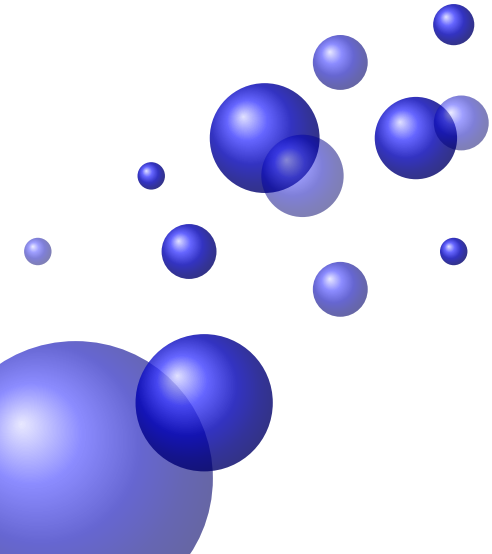
Rediger un gros document avec LaTeX

Nom de l'auteur

Sous la direction de Le Tuteur.



Le February 15, 2018

Rediger un gros document avec LaTeX

Nom de l'auteur

Sous la direction de Le Tuteur.

Le February 15, 2018

Abstract

My abstract: vous pouvez notez ici que l'espacement entre le mot abstract et les : n'est pas le même qu'en français, comme le veut la typographie anglaise.

Acknowlegdment

Je tiens a remercier toutes les personnes qui m'ont aide a rediger cet article, Namrod pour la partie bibliographie, Francis Walter, pour ses conseils ainsi que les personnes ayant participe a la correction de ce document.

A mon pere, ma mere, mes freres et soeurs

Sommary

Introduction

Context

In almost every scientific field, measurements are performed over time. These observations lead to a collection of organized data called time series. Today time series data are being generated at an unprecedented speed from almost every application domain, e.g.:

- In astronomy, telescopes scan the sky and capture light rays that are used in the study of the universe. In Large Synoptic Survey Telescope (LSST) project [lss], telescopes will capture the electromagnetic radiation of the sky during ten years to calculate the acceleration of the expansion of the universe. This will result in an astronomical catalogs of time series.
- In paleoecology, ...
- In medicine, the analysis is electrocardiogram is used to prevent heart attacks[]. Those electrocardiograms are long time series obtained by recording the electrical activity of the heart over a period.
- In biomechanics, the study of human locomotion is perform using sensors that record the efforts performed and the movements of the body during the locomotion.

As a consequence, in the last decade there has been a dramatically increasing amount of interest in querying and mining such data.

Issues

Time-series data mining unveils numerous facets of complexity. The most prominent problems arise from the uncertainty contained in time series data, the difficulty of defining a form of similarity measure based on human perception, and the high

dimensionality of time-series data. These constraints show us that three major issues are involved.

- Uncertainty. How to compare the shape of time series without knowing their exact value? How to measure the impact of uncertainty contained in time series or to reduce the adverse effects of uncertainty?
- Similarity measurement. How can any pair of time-series be distinguished or matched? How can an intuitive distance between two series be formalized? This measure should establish a notion of similarity based on perceptual criteria, thus allowing the recognition of perceptually similar objects even though they are not mathematically identical.
- Data representation. How can the fundamental shape characteristics of a time-series be represented? What invariance properties should the representation satisfy? A representation technique should derive the notion of shape by reducing the dimensionality of data while retaining its essential characteristics.

The aim of our work is to propose algorithm to deal with thoses caracteristics of time series.

Context of the thesis

This thesis apprehends these scientific questions from a data mining point of view, within the framework of the analysis of time series coming from Manual Wheelchair locomotion. Also, even if the issues addressed are not limited to the field of biomechanics time series and concern other areas of applications, this thesis will deal with the analysis of time series coming from Ergometer Wheelchair FRET-2.

For improving the mobility of persons confined to manual wheelchairs, it is necessary to be able to "assess" people in their daily environment, and a field ergometer wheelchair (FRET-1) has been designed and manufactured for this purpose []. ,]. This ergometer is equipped with a moment sensor that measures the forces applied to the handrails as well as the acceleration and movement of the FRET-1 [,]. It, therefore, makes it possible to measure and calculate a large number of the mechanical parameters of manual wheelchair locomotion.

However, the time series produced by this moment sensor have specific characteristics:

- they are long because of the acquisition frequency of the sensor (between 80 and 100 Hz),
- they are cyclic; these cycles come from the cyclical character of the locomotion in Manual Wheelchair which consists of a succession of period of pushing and freewheeling,

- they are uncertain, this uncertainty is observed during the calibration of the sensor.

Our work consists of proposing algorithms to extract relevant information from these time series while taking into account their characteristics. The methods developed in this work have the aim to assist practitioners for the analysis of Manual Wheelchair locomotion; then, special attention will be given to the readability and ease of interpretation of the results provided by them.

Plan

The thesis is organised as follow

- **Chapter 1** present the state of art
- **Chapter 2** Dynamic Time Warping (DTW) is a time series alignment algorithm that is often used because it considers that it exits small distortions between time series during their alignment. However, DTW sometimes produces pathological alignments that occur when, during the comparison of two time series X and Y, one data point of the time series X is compared to a large subsequence of data points of Y. In this paper, we demonstrate that to compress time series using Piecewise Aggregate Approximation (PAA) is a simple strategy that greatly increases the quality of the alignment with DTW this is particularly true for synthetic data sets.
- **Chapter 3** The abstract.
- **Chapter 4** The analysis of cyclic time series from biomechanics is based on the comparison of the properties of their cycles. As usual algorithms of time series classification ignore this particularity, we propose a symbolic representation of cyclic time series based on the properties of cycles, named SAX-P. The resulting character strings can be compared using the Dynamic Time Warping distance. The application of SAX-P to propulsive moments of three subjects (S1, S2, S3) moving in Manual Wheelchair highlight the asymmetry of their propulsion. The symbolic representation SAX-P facilitates the reading of the cyclic time series and the clinical interpretation of the classification results.

Part I

State of the Art

Chapter 1

Analysis of the wheelchair locomotion

1.1 Introduction

Wheelchair locomotion concerns many people, for different reasons: genetic(myopathy), accidental (spinal cord injury, lower extremity amputee), degenerative (multiple sclerosis, poliomyelitis) or just related to the natural aging of locomotor functions (muscle degeneration, arthritis of the lower limbs, etc.). Then, in the 34 developed countries, it is estimated that 1% or 10,000,000 people require a wheelchair. In the 156 developing countries, it is estimated that at least 2% or 121,800,000 people require a wheelchair. Overall, of the 7,091,500,000 people in the world, approximately 131,800,000 or 1.85% need a wheelchair [].However, the use of the manual wheelchair is not without risk.

1.2 The problem of locomotion manual wheelchair locomotion

Although the use of FRM improves the mobility of its users, doctors quickly realized that its use often led to sedentarization, leading to problems of obesity, diabetes, etc. Also, to promote daily physical activity, sport has been strongly encouraged [147, 203, 211, 259]. However, intensive and prolonged sports practice in FRM can lead to specific injuries and pains [79, 125, 274, 277], especially in the shoulder [11, 14, 25, 79, 192,], and at the elbow, wrist and hand [140, 141, 241, 232 279, 295, 352]. Thus, according to studies conducted between 1991 and 2000, it has been reported that 30 [11] to 73% [241] of paraplegic individuals suffered from shoulder pain. In addition, sitting and prolonged sitting of FRM users causes dermatological problems such as bedsores or pressure ulcers, due to immobility, loss of sensitivity and incontinence. In addition, these symptoms are recognized as a major cause of discontinuation of the use of FRM [314, 336], thus the sedentarization of users. Lundqvist et al. [210]

showed that upper limb pain was the only factor correlated with poor quality of life in FRM subjects. The difficulty for the therapist is to practice a daily physical activity, adapted to the individual, and limit orthopedic problems and thus promote the use of the FRM over time.

Given the problems faced by manual wheelchair users at the level of their autonomy and health, van der Woude et al. [304, 305] summarized the issues of manual wheelchair locomotion research into three main areas:

- Improving the interface between the subject and his manual wheelchair, that is to say, the ergonomy and the adequacy of the system {subject + manual wheelchair} with the external physical environment (ramps, lifts, corridor widths, etc.).
- The improvement of the manual wheelchair regarding the design and the mechanical principles of propulsions;
- **Improving the subject's physical abilities**, that is, improving propulsion techniques, as well as rehabilitation techniques and training programs.

Bio-mechanics work has been conducted in LIMOS to identify and quantify traumatic factors such as. This work led to the construction of a measuring tool: an Ergo-meter Field Chair.

1.3 Tools to evaluate manual wheelchair locomotion

1.4 Knowledge discovery on wheelchair time series

Locomotion in manual wheelchair causes significant mechanical stresses in the upper limbs. To remedy this problem, biomechanics have been conducted to identify and quantify traumatic factors such as:

- The doctoral thesis of Nicolas de Saint REMY (2005) [1] who proposed a mechanical model relating the forces applied to a Manual Wheelchair and its displacement as illustrated in Figure 1.1 and Equation 1.1. It made it possible to highlight the fact that the acceleration of the FRM is a function of the movements of the subject:
- The doctoral thesis of Christophe SAURET (2010) [2] who proposed a method of calculating the mechanical power developed by manual wheelchair users to move. This model analyzes the kinetics (trajectory and speed) of the subject segments and the Manual Wheelchair. A segment is the body part of a user or a manual wheelchair between two markers. Figure 1.2 shows the layout of the markers used for this analysis.

1.5 Conclusion

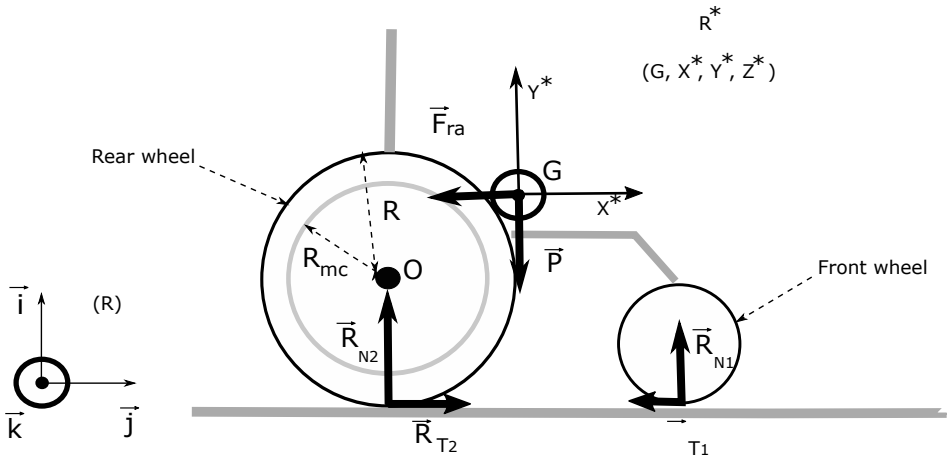


Figure 1.1: Balance of forces applied to a manual wheelchair during its use; the analysis of the movement of the subject-chair system has been reduced to that of its center of gravity

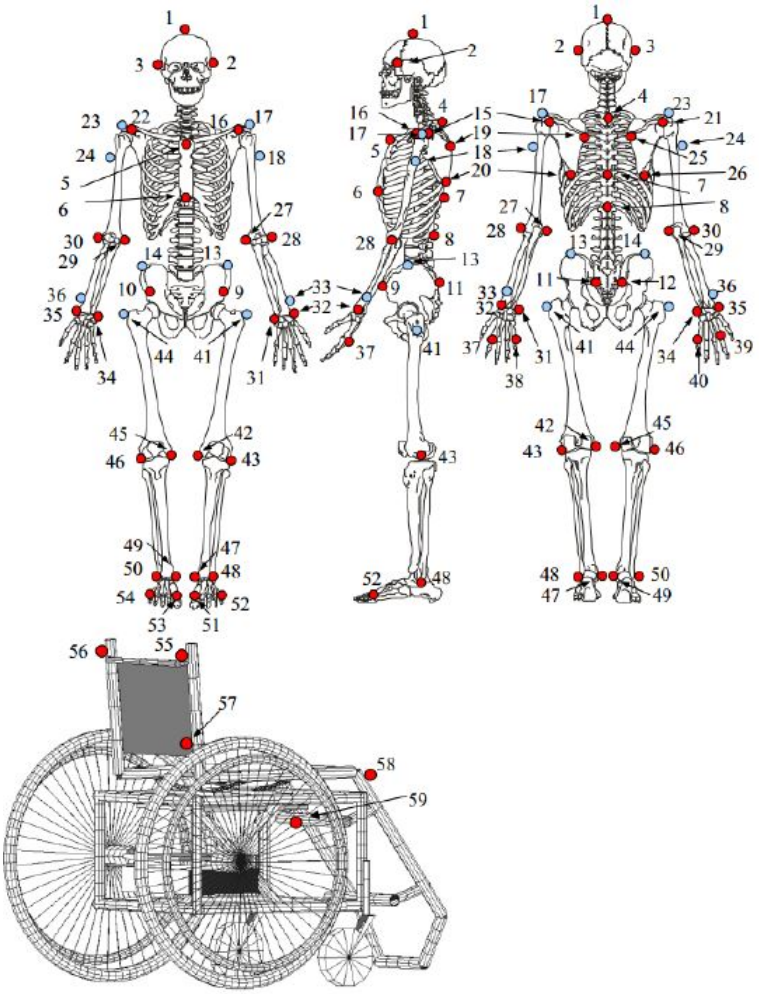


Figure 1.2: Balance of forces applied to a manual wheelchair during its use; the analysis of the movement of the subject-chair system has been reduced to that of its center of gravity

Chapter 2

Clustering of time series

2.1 Introduction

The field of computer science dealing with the extraction of time series knowledge is known as Temporal Knowledge Discovery which aims to extract from the knowledge of time series. But, what is a time series?

There are four categories of data concerning temporality [4]:

- Static data: these are data with no context temporal we have for example the radius of a wheel, the circumference of a circle, the gravity in a place.
- Sequences: This is an ordered sequence of events. This category includes an order but not time. As an example, we can cite a DNA sequence (GTTTTC-CCAGTCACGAC) [5].
- Time-indexed data: this is a temporal sequence of data; for example a set of measures taken at a more or less regular time interval.
- Full-time data: Each tuple has one or more time components. In this work, we refer to time series of indexed data by the time.

Time series analysis has two main and non-disjoint objectives which are:

- The prediction of missing values of a time series [6].
- The evaluation of dissimilarity or the classification of time series.

Althoug prediction is an important task of temporal data mining, the classification of time series remains its main challenge. In fact, in many cases, the prediction of missing values of time series is based on the study of time series that are similar to it, i.e., that belong to the same class as the latter [4]

Han and Kamber [1] classified clustering methods developed for handing various static data into five major categories: partitioning methods, hierarchical methods, density based methods, grid-based methods, and model-based methods. Three of the five major categories of clustering methods for static data, specifically partitioning methods, hierarchical methods, and model based methods, have been utilized directly or modified for time series clustering.

Two key aspects for achieving effectiveness and efficiency when managing time series data are representation methods and similarity measures. Time series are essentially high dimensional data and directly dealing with such data in its raw format is very expensive in terms of processing and storage cost. It is thus highly desirable to develop representation techniques that can reduce the dimensionality of time series, while still preserving the fundamental characteristics of a particular data set. Many techniques have been proposed in

the literature for representing time series with reduced dimensionality [2], such as Discrete Fourier Transformation, Single Value Decomposition, Discrete Cosine Transformation, Discrete Wavelet Transformation, Piecewise Aggregate Approximation, Adaptive Piecewise Constant Approximation, Chebyshev polynomials, Symbolic Aggregate approximation, Indexable Piecewise Linear Approximation etc.

In conjunction with this, there are over a dozen distance measures [2] for similarity of time series data in the literature, e.g., Euclidean distance (ED), Dynamic Time Warping (DTW), distance based on Longest Common Subsequence (LCSS) ,Edit Distance with Real Penalty (ERP) , Edit Distance on Real sequence (EDR) , DISSIM , Sequence Weighted Alignment model (Swale) , Spatial Assembling Distance (SpADe) and similarity search based on Threshold Queries (TQuEST) . Moreover, there must be some criteria on the basis of which the performance of time series clustering method must be evaluated. Two different categories of evaluation are given based on known ground truth and unknown ground truth [3].

The rest of this paper is organized as follows: Section 2 explains Major time series approaches in three subsections. Subsection 2.1 explains Taxonomy-based-approaches. In subsections 2.2 and 2.3 Representations-based and Model- based approaches are explained respectively. Section 3 concludes this paper.

2.2 MAJOR TIME SERIES CLUSTERING APPROACHES

Time series clustering methods have been divided into three major categories depending upon whether they work directly with raw data, indirectly with features extracted from the raw data, or indirectly with models built from the raw data.

2.2.1 Literature Survey of Temporal- Proximity-Based Clustering Approach

This approach usually works directly with raw time series data, thus called raw-data-based approach, and the major modification lies in replacing the distance/similarity measure for static data with an appropriate one for time series.

M. Kumar [25] proposed a distance function based on the assumed independent Gaussian models of data errors and used a hierarchical clustering method to group seasonality sequences into a desirable number of clusters. The experimental results based on simulated data and retail data showed that the new method outperformed both k-means and Ward's method that do not consider data errors in terms of (arithmetic) average estimation error. They assumed that data used have been preprocessed to remove the effects of non-seasonal factors and normalized to enable comparison of sales of different items on the same scale.

T.W. Liao [26] developed a two-step procedure for clustering multivariate time series of equal or unequal length. The first step applies the k-means or fuzzy c-means clustering algorithm to time stripped data in order to convert multivariate real-valued time series into univariate discrete- valued time series. The converted variable is interpreted as state variable process. The second step employs the k-means or FCM algorithm again to group the converted univariate time series, expressed as transition probability matrices, into a number of clusters. The traditional Euclidean distance is used in the first step, whereas various distance measures including the symmetric version of Kullback–Liebler distance are employed in the second step.

T.W. Liao [27] applied several clustering algorithms including K-means, fuzzy c-means, and genetic clustering to multivariate battle simulation time series data of unequal length with the objective to form a discrete number of battle states. The original time series data were not evenly sampled and made uniform by using the simple linear interpolation method.

C. S. Möller-Level [28] , in their study of DNA Microarray data, proposed short time series (ST S) distance to measure the similarity in shape formed by the relative change of amplitude and the corresponding temporal information of uneven sampling intervals. All series are considered sampled at the same time points. By incorporating the STS distance into the standard fuzzy c-means algorithm, they revised the equations for computing the membership matrix and the prototypes (or cluster centers), thus developed a fuzzy time series clustering algorithm.

Shumway [29] proposed the clustering of nonstationary time series by applying locally stationary versions of Kullback– Leibler discrimination information measures that give optimal time–frequency statistics for measuring the discrepancy between two non-stationary time series. To distinguish earthquakes from explosions, an agglomerative hierarchical cluster analysis was performed until a final set of two clusters was obtained.

Vit Niennattrakul and Chotirat Ann Ratanamahatana for the clustering of multi-

media time series [5] applied K-means and K- medoids algorithms with dynamic time warping and demonstrated that K-means is much more generic clustering method when Euclidean distance is used, but it failed to give correct results when dynamic time warping is used as distance measure in averaging the shape of the time series. As the results of their experiments, they have confirmed that dynamic time warping should not be used as subroutine with K-means algorithm and K-medoids with dynamic time warping gives satisfactory results.

For clustering time series gene expression data Pooya Sobhe Bidari [6] presented two phase functional clustering as a new approach in gene clustering. The proposed approach is based on finding functional patterns of time series using Fuzzy C- Means and K-means algorithms.

Pearson correlation similarity measure is used to extract the expression patterns of genes. In this approach, genes are clustered by K-means and FCM methods according to theirs time series expression, then patterns of gene behavior are extracted. Then, new features are defined for the genes and by calculating Pearson correlation between new feature vectors, genes with similar expression behavior are obtained which can lead to find interconnections between genes.

For detecting climate change in multivariate data Hardy Kremer [7] proposes novel clustering and clustering tracing techniques. In this novel clustering approach, time series is split into disjoint, equal length intervals and then density based subsequence clustering approach is applied, and dynamic time warping is used as a distance measure.

Jian Yin [9] proposed a clustering algorithm for time series data. He proposed a encoded-bitmap-approach-based swap is used to improve the classical hierarchical method. Traffic flow data is used as time series and grey relation is used a similarity measure. After getting K clusters, encoded-bitmap- approach based swap is used to refine the K clusters and get the new K clusters. Experiments show that the proposed method has a better performance on the change trend of time series than classic algorithm.

Ville Hautamaki and Pekka Nykanen [10] defined an optimal prototype as an optimization problem and proposed a local search solution to it. They applied two Euclidean space clustering methods to time-series clustering: random swap and hierarchical clustering followed by k-means fine-tuning and it provided 10-22

S. Chandrakala and C. Chandra Sekhar [11] proposed a density based method for clustering of multivariate time series of variable length in kernel feature space. Kernal DBSCAN algorithm with Euclidean distance measure is used. They presented heuristic methods to find the initial values of the parameters used in our proposed algorithm. The performance of the proposed method is compared with the spectral clustering and kernel k-means clustering methods. Besides handling outliers, the proposed method performs as well as the spectral clustering method and outperforms the kernel k- means clustering method.

Dacheng Nie [13] analyzed time series by using NLCS (Normalized Longest Com-

mon Subsequence). NLCS is a similarity measurement widely used in comparing character sequences. In this paper, he developed the NLCS and presented a novel algorithm to precisely calculate the similarity of time series. The algorithm used the sum of all common subsequence instead of longest common subsequence which can't represent the similarity of sequences accurately. The experiments based on synthetic and real-life datasets showed that the proposed algorithm performed better in comparing the similarity of time series. Comparing with Euclidean distance on four cluster validity indices, the results lead to a better performance by k-means or self-organize map.

Aurangzeb Khan [16] used hybrid clustering algorithm to mine the frequent pattern in the stock or inventory data. He proposed an algorithm for mining patterns of huge stock data to predict factors affecting the sale of products. In the first phase of his method, he applied k-means algorithm to divide the stock data into three different clusters i.e. Dead Stock (DS), Slow Moving (SM) and Fast Moving (FM) on the basis of product categories. In the second phase, he applied Most Frequent Pattern (MFP) algorithm to find frequencies of property values of the corresponding items. MFP provides frequent patterns of item attributes in each category of products and also gives sales trend in a compact form. The experimental result showed that the proposed hybrid k-mean plus MFP algorithm can generate more useful pattern from large stock.

Songpol Ongwattanakul [15] and Dararat Srisai introduced a new variation of Dynamic time warping distance measure for time series shape averaging classification. According to them resampled DTW and Hybrid DTW give better accuracy and

high performance than original DTW but to improve the accuracy further they introduced Contrast Enhanced Dynamic Time Warping (CEDTW) that reduces the effect from data points that have non-trivial contribution to the measured distance and improves the accuracy in similarity measure.

Xueyan WU [17] proposed a method of data stream clustering for stock data analysis. The method aimed to retain shape and tend features during the clustering process. He divided the process of the proposed method into two parts i.e. online clustering and offline macro clustering. Online clustering extracted data flow characteristics and maintains the clustering feature vectors and offline macro clustering is the process which responded to user requests and achieved clustering. Experiments showed that the method had good results.

Aurangzeb Khan [16] used hybrid clustering algorithm to mine the frequent pattern in the stock or inventory data. He proposed an algorithm for mining patterns of huge stock data to predict factors affecting the sale of products. In the first phase of his method, he applied k-means algorithm to divide the stock data into three different clusters i.e. Dead Stock (DS), Slow Moving (SM) and Fast Moving (FM) on the basis of product categories. In the second phase, he applied Most Frequent Pattern (MFP) algorithm to find frequencies of property values of the

corresponding items. MFP provides frequent patterns of item attributes in each category of products and also gives sales trend in a compact form. The experimental result showed that the proposed hybrid k-mean plus MFP algorithm can generate more useful pattern from large stock.

Xueyan WU [17] proposed a method of data stream clustering for stock data analysis. The method aimed to retain shape and tend features during the clustering process. He divided the process of the proposed method into two parts i.e. online clustering and offline macro clustering. Online clustering extracted data flow characteristics and maintains the clustering feature vectors and offline macro clustering is the process which responded to to user requests and achieved clustering. Experiments showed that the method had good results.

Mengfan Zhang [18] and Tao Yang applied the computational verb theory (CVT) to analyze the stock market data. His goal was to find the most representative trends in the intra-day stock market data. First round of computational verb clustering algorithm was used to categorize the stock market data and in the second round of computational verb k-means clustering algorithm is used to refine the representative trends in the stock market data. Experiments showed that the applied method yielded the most representative curves in the stock market data.

Jianfei Wu [19] introduced an algorithm that used stock sector information directly in conjunction with time series subsequences for mining core patterns within the sectors of stock market data. He used the stream sliding window concepts. Two adjacent sliding windows were used, namely training window and evaluation window. The algorithm detected significant sectors in the training window, and built core patterns for the significant ones. The algorithm identified whether a stock sector currently shows coherent behavior. When coherent behavior of a stock sector was detected, core patterns were extracted. The core patterns were more stable than clusters found by some clustering algorithm DBScan. Through comparing with DBScan, we show the effectiveness of the proposed algorithm.

Huawang Shi [20] proposed a novel unascertained C-means clustering algorithm. He used the theory and method of unascertained measure and established clustering weights and a novel unascertained C-means clustering algorithm. Experimental results showed that the proposed method performed more robust to noise than the fuzzy C-means (FCM) clustering algorithm do.

S.R.Nanda [23] applied clustering to stock market data for portfolio management. She used stock returns at different times along with their valuation ratios and results analysis showed that k-means cluster analysis builds the most compact cluster as compared to SOM and Fuzzy C-means for stock classification data. She then selected stocks from clusters to build portfolio, minimizing portfolio risk and compared the returns with that of benchmark index.

2.2.2 Literature Survey of Representation- Based Clustering Approach

It is not easy to work directly with the raw data that are highly noisy. Feature based approach first converts a raw time series data into a feature vector of lower dimension and then clustering algorithms are applied.

T.-C. Fu [30] described the use of self-organizing maps for grouping data sequences segmented from the numerical time series using a continuous sliding window with the aim to discover similar temporal patterns dispersed along the time series. They introduced the perceptually important point (PIP) identification algorithm to reduce the dimension of the input data sequence in accordance with the query sequence. The distance measure between the PIPs found was defined as the sum of the mean squared distance along the vertical scale (the magnitude) and that along the horizontal scale (time dimension). To process multi resolution patterns, training patterns from different resolutions are grouped into a set of training samples to which the SOM clustering process is applied only once. Two enhancements were made to the SOM: filtering out those nodes (patterns) in the output layer that did not participate in the recall process and consolidating the discovered patterns with a relatively more general pattern by a redundancy removal step.

Dong Jixue [8] for mining the financial time series uses the wave cluster, which is a kind of grid cluster and the density cluster unify. In this, basic details and methods of phase space reconstruction are analyzed in details. All of these provided the theoretical basis and technical feasibility to time series data mining based on phase space reconstruction. After contrasting the different means of time series pattern mining, the problem of Time Series Data Mining framework TSDM is pointed out, and the temporal patterns mining method based Wave cluster is systematically presented. By the multiresolution property of wavelet transformations and the grid-based partition method, it could detect arbitrary-shape clusters at different scales and levels of detail.

Huiting Liu [12] proposed a new similar pattern matching method. Firstly, trends of time series are extracted by empirical mode decomposition, and the trends are translated into vectors to realize dimension reduction. Secondly, the vectors are clustered by a forward propagation learning algorithm. Finally, all the series that is similar with the query are found by calculating Euclidean distance between the query and the series that belong to the same category with it. Experimental results showed that EMD outperforms the FFT (Fast Fourier Transform) when they are used to reduce dimension.

M. Vlachos [31] presented an approach to perform incremental clustering of time series at various resolutions using the Haar wavelet transform. First, the Haar wavelet decomposition is computed for all-time series. Then, the k- means clustering algorithm is applied, starting at the coarse level and gradually progressing to finer levels. The final centers at the end of each resolution are reused as the initial centers

for the next level of resolution. Since the length of the data reconstructed from the Haar decomposition doubles as we progress to the next level, each coordinate of the centers at the end of level i is doubled to match the dimensionality of the points on level i+1. The clustering error is computed at the end of each level as the sum of number of incorrectly clustered objects for each cluster divided by the cardinality of the dataset.

Nicole Powell [14] compared unsupervised classification techniques such as k-means clustering with supervised classification techniques such as support vector machines for stock prices forecasting .He used Principal component analysis to reduce the dimension of the data set to select the component which have the biggest effect and concluded that for this application both method give comparable results but unsupervised classification techniques are better for stock trend forecasting because unsupervised methods fine pattern in data that is usually seen as uncorrelated.

Anthony J. T .Lee [24] presented an effective approach (Hierarchical agglomerative and recursive k-means clustering) to stock market prediction. The proposed method converted each financial report to feature vector and used hierarchical agglomerative clustering to divide this feature vector into

clusters and then, for each sub-cluster so that most feature vectors in each sub-clusters belonged to the same class. Then, for each sub-cluster, a centroid was chosen as the representative feature vector and finally this feature vector was employed to predict the stock price movements.

Chonghui GUO [22] presented a novel feature based approach to time series clustering which first converted the raw time series into feature vector of lower dimension by using ICA (Independent Component Analysis) algorithm and then applied a modified k-means clustering algorithm to the extracted feature vector . Finally to validate the feasibility and effectiveness of the proposed approach he used it to analyze the real world stock time series and achieved the reasonable results.

Jian Xin Wu [32] proposed a combination of ICA (independent component analysis) with SVR (support vector regression) for predicting the financial time series. The proposed method used SRM (structure risk minimization) principle. It removed the problem of learning algorithms based on ERM (empirical risk minimization) principle, which always have good fit for the training samples, but bad prediction for future samples. He used non-linear SVR (Support vector regression) for the prediction, and before applying SVR, he used ICA for the feature extraction. ICA considers independence which is a more strict condition than PCA which takes into account uncorrelated between features.

Geert Verdoolaege [33] proposed a new method for the detection of activated voxels in event related BOLD FMRI data. Firstly, he derived wavelet histograms from each voxel time series and modeled the derived statistics through GGD (generalized Gaussian distribution). Finally, he performed the K-Means clustering of the GGD's characterizing the voxel data in a synthetic data set using the KLD (Kullback- Liebler divergence) as a similarity measure.

The main issue of similarity search in time series database is to improve the search performance since time series data is usually is of high dimension and for this is important to reduce the search space for the efficient processing of similarity search. In this paper, D.Muruga Radha Devi [34], proposed a combination of using Vari-DWT and Polar wavelet. Vari-DWT is fast to compute and requires little storage for each sequence; it preserves Euclidean distance and allows good approximation with a subset of coefficients. But its drawbacks are, it shows poor performance for locally distributed time series data since it uses averages to reduce the dimensionality of the data and another limitation is it works best when the length of the time series is 2n, and hence becomes the reason of using Polar wavelet, it uses polar co- ordinates which are not affected from averages and, it works with the time sequences of any length without distorting the original signal. She evaluated the effectiveness of this approach by using real weather data and synthetic datasets.

Liu Suyi [35] did feature recognition for underwater images. Firstly, the preprocessing of underwater image was done to improve image quality so that seam feature could be recognized easily. Weld featured image was effectively segmented by Mean Shift Algorithm, and finally Hough Transform was used to recognize the feature of underwater weld images.

2.2.3 Literature Survey of Model-Based

Clustering Approach This class of approaches considers that each time series is generated by some kind of model or probability distributions. Time series are considered similar when the models characterizing individual series or the remaining residuals after fitting the model are similar.

Baragona [36] evaluated three meta-heuristic methods for partitioning a set of time series into clusters in such a way that (i) the cross-correlation maximum absolute value between each pair of time series that belong to the same cluster is greater than some given threshold, and (ii) the k-min cluster criterion is minimized with a specified number of clusters. The cross-correlations are computed from the residuals of the models of the original time series. Among all methods evaluated, Tabu search was found to perform better than single linkage, pure random search, simulation annealing and genetic algorithms based on a simulation experiment on ten sets of artificial time series generated from low-order univariate and vector ARMA models.

K. Kalpakis [37] studied the clustering of ARIMA time series, by using the Euclidean distance between the Linear Predictive Coding cepstra of two time-series as their dissimilarity measure. The cepstral coefficients for an AR(p) time series are derived from the auto-regression coefficients. The partition around medoids method that is a k-medoids algorithm was chosen, with the clustering results evaluated with the cluster similarity measure and Silhouette width. Based on a test of four data sets, they showed that the LPC cestrum provides higher discriminatory power to tell one time series from another and superior clustering than other widely used

methods such as the Euclidean distance between (the

first 10 coefficients of) the DFT, DWT, PCA, and DFT of the auto-correlation function of two time series.

Xiong and Yeung [38] proposed a model-based method for clustering univariate ARIMA series. They assumed that the time series are generated by k different ARMA models, with each model corresponds to one cluster of interest. An expectation-maximization (EM) algorithm was used to learn the mixing coefficients as well as the parameters of the component models that maximize the expectation of the complete-data log-likelihood. In addition, the EM algorithm was improved so that the number of clusters could be determined automatically.

L. Wang [39] presented a framework for tool wear monitoring in a machining process using discrete hidden Markov models. The feature vectors are extracted from the vibration signals measured during turning operations by wavelet analysis. The extracted feature vectors are then converted into a symbol sequence by vector quantization, which in turn is used as input for training the hidden Markov model by the expectation maximization approach. Yun Yang and Ke Chen [4] presented an unsupervised ensemble learning approach to time series clustering by combining RPCL (rival-penalized competitive learning) with different representations. This approach first exploits its capability of RPCL rule in clustering analysis of automated model selection on individual representations and then applies ensemble learning for the synergy of reconciling diverse partitions resulted from the use of different representations and augmenting RPCL network in automatic model selection and overcoming its inherent limitation. They evaluated their approach on 16 benchmark time series data mining task. Simulation results demonstrated that their approach yielded favorite results in clustering analysis of automatic model selection.

Yu-Chia Hsu [4] proposed a approach using self organizing map (SOM) for time series data clustering and similar pattern recognition to improve the optimal hedge ratio (OHR) estimation. Taiwan stock market hedging is used. Five SOM based models and two traditional models were compared in this approach. Experiments demonstrated the SOM approach provides a useful alternative to the OHR estimation.

Xin Huang proposed [40] a research on predicting agriculture drought based on fuzzy set and R/S analysis model. He used fuzzy clustering iteration method to cluster the data of many years rainfall and then considered sensitiveness coefficient as the foundation of calculating weight, which affected the crop output by valid rainfall in each growth stage. The results of application showed that the model is convenient and feasible in the application of forecasting the years of occurrence in agriculture drought.

Yupei Lin [41] tried to improve the prediction accuracy with correcting two deficiencies, sub intervals failing to well represent the data distribution structures and a single antecedent factor in the fuzzy relationship in current fuzzy time series model. First, he partitioned the universe of discourse in subintervals with the midpoints of

two adjacent clusters centers, and the subintervals are employed to fuzzify the time series into fuzzy time series. Then, the fuzzy time series model with multi factors high order fuzzy relationship is built-up to forecast the stock market. The results showed that the model improved the prediction accuracy when compared with the benchmark model.

Shan Gao [42] analyzed ARCH (Autoregressive Conditional Heteroscedasticity) effects of wind data series with Eviews

software. Firstly, he built an ARMA (Autoregressive Moving Average) model of wind speed time series and, tested the ARCH effect of residual ARMA Model by Lagrange Multiplier. Lastly, he compared the forecasting performances of ARMA-ARCH model with ARMA model and proved that ARMA-ARCH model possesses higher accuracy.

2.3 CONCLUSIONS

In recent years, there has been variety of interests in mining time series. Particularly, the clustering of time series has attracted the interest of researchers. As Time series data are frequently large and may contain outliers, therefore careful examination of the proposed algorithms is necessary. In this paper we have studied most recent techniques on the subject of time series clustering. The uniqueness and limitation of previous research are discussed and several possible topics for future research are identified. It is hoped that this review will serve as the steppingstone for those interested in advancing this area of research.

Table 2.1: My caption

Paper	Distance Measure	Algorithm	A
M. Kumar	Based on the assumed independent Gaussian models of data errors	Agglomerative Hierarchical	Seasonalit
T.-W. Liao	Euclidean and symmetric version of Kullback–Liebler distance	K-Means and Fuzzy C-Means	Battle sim
T.-W. Liao	Dynamic Time Warping	K-Medoids Based Genetic Clustering	Battle sim
C.S. Möller-Levet	Short time series (STS) distance	Modified Fuzzy C-Means	DNA mic
Shumway	Kullback–Leibler discrimination information measure	Agglomerative Hierarchical	Earthqua and mining explosion
Vit Niennattrakul	Dynamic Time Warping	K-Means, K-Medoids	Multimed series
Pooya Sobhe Bidari	Pearson Correlation	K-Means, Fuzzy C-Means	Pattern e
Hardy Kremer	Dynamic Time Warping	Density Based Subsequence Clustering	Detecting change
Jian Yin	Grey Relation	Hierarchical Clustering	Change trend of traffic flo data
S. Chandrakala	Euclidean	Kernal DBScan	Multivari series clustering
Aurangzeb Khan	Euclidean	K-Mean+ MFP(Most Frequent Pattern)	Stock and
Mengfan Zhang	CVT(Computational Verb Theory)	K-Means	Stock ma
S.R.Nanda	Euclidean	K-Means	Portfolio
Jianfei Wu	N/A	K-Means	Stock dat

Table 2.2: My caption

Paper	**Features**	**Distance Measure**	**Clustering Algori**
T.-C. Fu	Perceptually important points	Sum of the mean squared distance along the vertical and horizontal scales	Modified SOM
M. Vlachos	Haar wavelet transform	Euclidean	Modified k-means
Huiting Liu	Empirical mode decomposition	Euclidean	Forward propagatio learning algorithm
Chonghui GUO	Independent component analysis	Euclidean	Modified k-means
Jian Xin Wu	Independent component analysis	N/A	support vector regression
Geert Verdoolaege	Wavelet transform	Kullback- Liebler divergence	k-means
Liu Suyi	Hough transform	N/A	Mean shift algorithr
Dong Jixue	Wavelet transform	N/A	Grid-based partitior method

Table 2.3: My caption

Paper	**Model**	**Distance measure**	**Clustering algorithm**	
Baragona	ARMA	Cross-correlation based	Tabu search, GA and	Non-
K. Kalpakis	AR	Euclidean	Partitioning around medoids	Publ
Xiong and Yeung	ARMA mixture	Log-liklihood	EM learning	Publ
L. Wang	Discrete HMM	Log-liklihood	EM learning	Tool cond mon
Xin Huang	Fuzzy set and R/S analysis model	N/A	Fuzzy clustering iteration method	Pred agric
Shan Gao	ARMA-ARCH	N/A	N/A	To a effec

Part II

Our contribution

Chapter 3

Frobenius correlation based u-shapelets discovery for time series clustering

Abstract : *An u-shapelet is a sub-sequence of a time series used for clustering a time series dataset. The purpose of this paper is to discover u-shapelets on uncertain time series. To achieve this goal, we propose a dissimilarity score called FOTS whose computation is based on the eigenvector decomposition and the comparison of the autocorrelation matrices of the time series. This score is robust to the presence of uncertainty; it is not very sensitive to transient changes; it allows capturing complex relationships between time series such as oscillations and trends, and it is also well adapted to the comparison of short time series. The FOTS score is used with the Scalable Unsupervised Shapelet Discovery algorithm for the clustering of 17 datasets, and it has shown a substantial improvement in the quality of the clustering with respect to the Rand Index. This work defines a novel framework for the clustering of uncertain time series.*

3.1 Introduction

Uncertainty in time series comes from several sources. For instance, to protect privacy, privacy-preserving transformation [Papadimitriou *et al.*, Aggarwal] deliberately introduce uncertainty to the confidential data before further processing. In a sensor network, sensor readings are imprecise because of the presence of noise generated either by the equipment itself or other external influences [Cheng *et al.*]. Ignoring the uncertainty of the data can lead to rough or inaccurate conclusions, hence the need to implement uncertain data management techniques.

Several recent studies have focused on the processing of uncertainty in data mining. Two main approaches allow to take uncertainty into account in data mining tasks: either it is taken into account during the comparison by using appropriate dis-

tance functions [Rizvandi *et al.*, Hwang *et al.*, Rehfeld and Kurths, Orang and Shiria, Wang *et al.*a, Orang and Shirib], or its impact is reduced by transformations performed on the data [Orang and Shiric].This latter strategy is used natively by the u-shapelet algorithm.

3.1.1 U-shapelets algorithm for clustering Uncertain Time Series

U-shapelets clustering is a framework introduced by[Zakaria *et al.*] who suggested clustering time series from the local properties of their sub-sequences rather than using their global features of the time series [**?**]. In that aim, u-shapelets clustering first seeks a set of sub-sequences characteristic of the different categories of time series and classifies a time series according to the presence or absence of these typical sub-sequences in it.

Clustering time series with u-shapelets has several advantages. Firstly, u-shapelets clustering is defined for datasets in which time series have different lengths, which is not the case for most techniques described in the literature. Indeed, in many cases, the equal length assumption is implied, and the trimming to equal length is done by exploiting expensive human skill [Ulanova *et al.*]. Secondly, u-shapelets clustering is much more expressive regarding representational power. Indeed, it allows clustering only time series that can be clustered and do not cluster those that do not belong to any cluster.

Furthermore, it is very appropriate to use u-shapelets clustering with uncertain time series because it can ignore irrelevant data and thus, reduce the adverse effects of the presence of uncertainties in the time series. Despite this advantage, it is highly desirable to take into account the adverse impact of uncertainty during u-shapelet discovery.

3.1.2 Uncertainty and u-shapelets discovery issue

Traditional measurement of similarity like Euclidean distance (ED) or Dynamic Time Warping (DTW) do not always work well for uncertain time series data. Indeed, they aggregate the uncertainty of each data point of the time series being compared and thus amplify the negative impact of uncertainty. However, ED plays a fundamental role in u-shapelet discovery because it is used to compute the gap, i.e. the distance between the two groups formed by a u-shapelet candidate. The discovery of u-shapelet on uncertain time series could thus lead to the selection of a wrong u-shapelet candidate or to assign a time series to the wrong cluster.

In this study, our goal is to cluster uncertain time series with u-shapelets algorithm. Our work leverages the observation that the use of a dissimilarity function robust to uncertainty could improve the quality of the u-shapelets discovered and thus improve the clustering quality of uncertain time series.

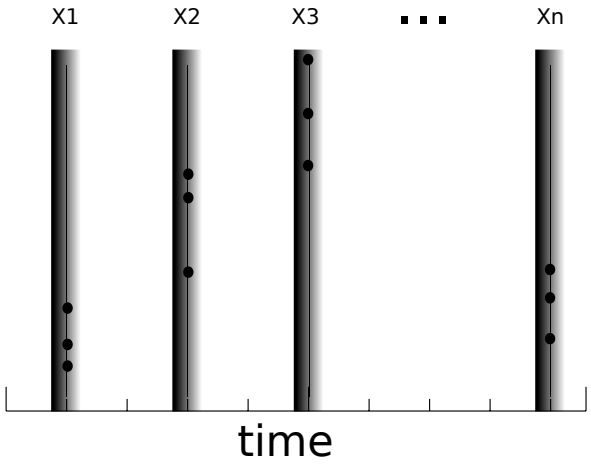


Figure 3.1: Multiset-based model of uncertain time series

3.1.3 Summary of contributions

- We review state of the art on similarity functions for uncertain time series and evaluate them for the comparison of small, uncertain time series.
- We introduce the Frobenius cOrrelation for uncertain Time series uShapelet discovery (FOTS), a new dissimilarity score based on local correlation, which has interesting properties useful for comparison of small, uncertain time series and that makes no assumption on the probability distribution of uncertainty in data.
- We put the source code at the disposal of the scientific community to allow extension of our work[**?**].

3.2 Definitions and Background

3.2.1 Related work

An Uncertain Time Series (UTS) $X =< X_1, \ldots, X_n >$ is a sequence of random variables where X_i is the random variable modeling the unknown real value number at timestamp i. There are two main ways to model uncertain time series: multiset-based model and PDF-based model.

In **Multiset-based model**, each element $X_i (1 \leq i \leq n)$ of an UTS $X =< X_1, \ldots, X_n >$ is represented as a set $\{X_{i,1}, \ldots, X_{i,N_i}\}$ of observed values (Fig. 3.1) and N_i denotes the number of observed values at timestamp i.

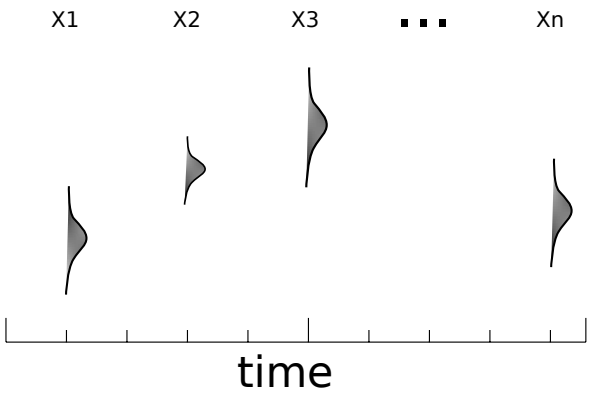


Figure 3.2: PDF-based model of uncertain time series

In **PDF-based model**, each element $X_i, (1 \leq i \leq n)$ of UTS $X =< X_1, \ldots, X_n >$ is represented as a random variable $X_i = x_i + X_{e_i}$, where x_i is the exact value that is unknown and X_{e_i} is a random variable representing the error (Fig. 3.2). It is this model that we consider this work.

Several similarity measures have been proposed for uncertain time series. They are grouped into two main categories: Traditional similarity measures and uncertain similarity measures.

- Traditional similarity measures such as Euclidean distance are those conventionally used with time series. They use a single uncertain value at each timestamp as an approximation of the unknown real value.
- Uncertain similarity measures use additional statistical information that quantifies the uncertainty associated with each approximation of the real value : this is the case of DUST, PROUD, MUNICH[Dallachiesa *et al.*]. [Orang and Shiric] demonstrates that the performances of Uncertain similarity measures associated with pre-processing of data are higher than those of traditional similarity measurements.

3.2.2 Review of u-shapelets

Definition 1. *Two datasets D_A and D_B are said to be **r-balanced** if only if $\frac{1}{r} < \frac{|D_A|}{|D_B|} < (1 - \frac{1}{r}), r > 1$*

Definition 2. *An **Unsupervised-Shapelet** is any sub-sequence that has a length shorter than or equal to the length of the shortest time series in the dataset, and that allows dividing the dataset into two **r-balanced** groups D_A and D_B; where D_A is the group of time series that contains a pattern **similar** to the shapelet and D_B is the group of time series that does not contain the shapelet.*

The similarity between a time series and a shapelet is evaluated using a distance function.

Definition 3. *The sub-sequence distance **sdist(S, T)** between a time series T and a sub-sequence S is the minimum of the distances between the sub-sequence S and all possible sub-sequences of T of length equal to the length of S.*

This definition opens the question of which distance measure to use for *sdist*. In general, the ubiquitous Euclidean distance (ED) is used, but it is not appropriate for uncertain time series [Orang and Shiria]. In the following section, we introduce a dissimilarity function that is more adapted to uncertainty.

Computing the *sdist* between a u-shapelet candidate and all time series in a dataset creates an orderline:

Definition 4. *An orderline is a vector of sub-sequence distances sdist(S, Ti) between a u-shapelet and all time series Ti in the dataset.*

The computation of the orderline is time-consuming. An orderline for a single u-shapelet candidate is $O(NMlog(M))$ where N is the number of time series in the dataset and M is the average length of the time series. The brute force algorithm for U-shapelets discovery requires K such computations, where K is the number of sub-sequences. The strategy used by [Ulanova *et al.*] in **Scalable Unsupervised Shapelet algorithm** consists in filtering the K candidate segments by considering only those allowing to build r-balanced groups. This selection is made efficiently thanks to a hash algorithm.

The assessment of a u-shapelet quality is based on its separation power which is calculated as follows :

$$gap = \mu_B - \sigma_B - (\mu_A - \sigma_A), \tag{3.1}$$

where μ_A (resp. μ_B) denotes mean(sdist(S, D_A)) (resp. mean(sdist(S, D_B))), and σ_A (resp. σ_B) represents standard deviation of $sdist(S, D_A)$ (resp. standard deviation of $sdist(S, D_B)$). If D_A or D_B consists of only one element (or of an insignificant number of elements that cannot represent a separate cluster), the gap score is assigned to zero. This ensures that a high gap scored for a u-shapelet candidate corresponds to a true separation power.

3.2.3 Review on uncertain similarity functions

Uncertain similarity measures can be grouped into two broad categories : deterministic similarity measurements and probabilistic similarity measurements.

Deterministic Similarity Measures

Like traditional similarity measures, deterministic similarity measures return a real number as the distance between two uncertain time series. **DUST** is an example of deterministic similarity measure.

DUST [Murthy and Sarangi] Given two uncertain time series $X =< X_1, \ldots, X_n >$ and $Y =< Y_1, \ldots, Y_n >$, the distance between two uncertain values X_i, Y_i is defined as the distance between their true (unknown) values r(X_i), r(Y_i): $dist(X_i, Y_i) = |r(X_i) - r(Y_i)|$. This distance is used to measures the similarity of two uncertain values.

$\varphi(|X_i - Y_i|)$ is the probability that the real values at timestamp i are equal, given the observed values at that instant :

$$\varphi(|X_i - Y_i|) = Pr(dist(0, |X_i - Y_i|) = 0). \tag{3.2}$$

This similarity function is then used inside the *dust* dissimilarity function:

$$dust(X_i, Y_i) = \sqrt{-log(\varphi(|X_i - Y_i|)) + log(\varphi(0))}. \tag{3.3}$$

The distance between uncertain time series $X =< X_1, \ldots, X_n >$ and $Y =< Y_1, \ldots, Y_n >$ in $DUST$ is then defined as follows:

$$DUST(X, Y) = \sqrt{\sum_{i=1}^{n} dust(X_i, Y_i)^2}. \tag{3.4}$$

The problem with the deterministic uncertain distances like $DUST$ is that their expression varies as a function of the probability distribution of uncertainty, and the probability distribution of the uncertainty is not always available in time series datasets.

Probabilistic Similarity Measures

Probabilistic similarities measures do not require knowledge of the uncertainty probability distribution. Furthermore, they provide the users with more information about the reliability of the result. There are several probabilistic similarity functions, as MUNICH, PROUD, PROUDS or Local Correlation.

MUNICH [Aßfalg *et al.*] This distance function is suitable for uncertain time series represented by the multiset based model. The probability that the distance between two uncertain time series X and Y is less than a threshold ε is equal to the number of distances between X and Y, which are less than ε, over the possible number of distances:

$$Pr(distance(X, Y)) \leq \varepsilon = \frac{|\{d \in dists(X, Y) | d \leq \varepsilon\}|}{|dists(X, Y)|} \tag{3.5}$$

The computation of this distance function is very time-consuming.

PROUD [Yeh *et al.*] Let$X =< X_1, ..., X_n >$ and $Y =< Y_1, ..., Y_n >$ be two UTS each modeled by a sequence of random variables, the PROUD distance between X and Y is $d(X,Y) = \sum_{i=1}^{n} (X_i - Y_i)^2$. According to the central limit theorem [**?**], the cumulative distribution of the distances approaches asymptotically a normal distribution:

$$d(X,Y) \propto N(\sum_i E[(X_i - Y_i)^2], \sum_i Var[(X_i - Y_i)^2]) \tag{3.6}$$

As a consequence of that feature of PROUD distance, the table of the normal centered reduced law can be used to compute the probability that the normalized distance is lower than a threshold:

$$Pr(d(X,Y)_{norm} \leq \epsilon). \tag{3.7}$$

A major disadvantage of PROUD is its inadequacy for comparing time series of small lengths like u-shapelets. Indeed, the calculation of the probability that the PROUD distance is less than a value is based on the assumption that it follows **asymptotically** a normal distribution. Thus, this probability will be all the more accurate as the compared time series are long (more than 30 data points).

PROUDS [Orang and Shiric] is an enhanced version of PROUD, which suppose that random variables coming from time series are independent and identically distributed.

Definition 5. *The normal form of a standard time series $X =< X_1, \ldots, X_n >$ is defined as $\hat{X} =< \hat{X}_1, \ldots, \hat{X}_n >$ in which for each timestamp i $(1 \leq i \leq n)$, we have:*

$$\hat{X}_i = \frac{X_i - \bar{X}}{S_X},\ \bar{X} = \sum_{i=1}^{n} \frac{X_i}{n},\ S_X = \sqrt{\sum_{i=1}^{n} \frac{(X_i - \bar{X})^2}{(n-1)}}. \tag{3.8}$$

PROUDS defines the distance between two normalized time series $\hat{X} =< \hat{X}_1 ... \hat{X}_n >$ and $\hat{Y} =< \hat{Y}_1 ... \hat{Y}_n >$ (Definition 5) as follows:

$$Eucl(\hat{X}, \hat{Y}) = 2(n-1) + 2\sum_{i=1}^{n} \hat{X}_i \hat{Y}_i \tag{3.9}$$

For the same reasons as PROUD, PROUDS is not suitable for short time series comparison. Another disadvantage of PROUDS is that it assumes that the

random variables are independent : this hypothesis is strong and particularly inappropriate for short time series like u-shapelets. A more realistic hypothesis with time series would be to consider that the random variables constituting the time series are M-dependent. Random variables of a time series are called M-dependent if $X_i, X_{i+1}, ..., X_{i+M}$ are dependent (correlated) and the variables X_i and X_{i+M+1} are independent. However, the M-dependent assumption could make PROUDS writing more complex and its use more difficult because of the choice of the parameter M.

Uncertain Correlation [Orang and Shirib] : Correlation analysis techniques are useful for feature selection in uncertain time series data. Indeed, correlation indicates the degree of dependency of a feature on other features. Using this information, redundant features can be identified. The same strategy can be useful for u-shapelet discovery. Uncertain correlation is defined as follows :

Definition 6. *(Uncertain time series correlation) Given UTS* $X =< X_1, \ldots, X_n >$ *and* $Y =< Y_1, \ldots, Y_n >$*, their correlation is defined as:*

$$Corr(X, Y) = \sum_{i=1}^{n} \hat{X}_i \hat{Y}_i / (n-1), \tag{3.10}$$

where $\hat{X}_i$ *and* $\hat{Y}_i$ *are normal forms of* X_i *and* Y_i *(Definition 5), respectively.* X_i *and* Y_i *are supposed to be independant continous random variables.*

If we know the probability distribution of random variables, it is possible to determine the probability density function associated with the correlation, which will subsequently be used to calculate the probability that the correlation between two time series is greater than a given threshold. Uncertain correlation has however some drawbacks :

- It is too sensitive to transient changes, often leading to widely fluctuating scores;
- It cannot capture complex relationship in timeseries;
- It requires to know the probability distribution function of the uncertainty or to make some assumption on the independence of the random variables contained in time series.

Because of all thoses drawbacks, uncertain correlation cannot be used as it is for u-shapelet discovery. The next paragraph presents a generalisation of correlation coefficient that is not an uncertain similarity function but is still interesting for u-shapelet discovery.

Local Correlation [Papadimitriou *et al.*] is a generalization of the correlation. It computes a time-evolving correlation scores that tracks a local similarity on multivariate time series based on local autocorrelation matrix. The autocorrelation matrix **allows capturing complex relationship** in time series like the key oscillatory (e.g., sinusoidal) as well as aperiodic trends (e.g., increasing or decreasing) that are present. The use of autocorrelation matrices which are computed based on overlapping windows allows **reducing the sensibility to transient changes** in time series.

Definition 7. *(Local autocovariance, sliding window). Given a time series X, a sample set of windows with length w, the local autocorrelation matrix estimator $\hat{\Gamma}_t$ using a sliding window is defined at time $t \in \mathbb{N}$ as (Eq.3.11) :*

$$\hat{\Gamma}_t(X, w, m) = \sum_{\tau=t-m+1}^{t} x_{\tau,w} \otimes x_{\tau,w}. \tag{3.11}$$

where $\boldsymbol{x}_{\tau,\omega}$ is a sub-sequence of the time series of length w and started at τ, $x \otimes y = xy^T$ is the outer product of x and y. The sample set of m windows is centered around time t. We typically fix the number of windows to $m = w$.

Given the estimates $\hat{\Gamma}_t(X)$ and $\hat{\Gamma}_t(Y)$ for the two time series, the next step is how to compare them and extract a correlation score. This goal is reached using the spectral decomposition; The eigenvectors of the autocorrelations matrices capture the key aperiodic and oscillatory trends, even **in short time series**. Thus, the subspaces spanned by the first few (k) eigenvectors are used to locally characterize the behavior of each series. Definition 8 formalizes this notion:

Definition 8. *(LoCo score). Given two series X and Y, their LoCo score is defined by*

$$\ell_t(X, Y) = \frac{1}{2}(\|\boldsymbol{U}_X^T \boldsymbol{u}_Y\| + \|\boldsymbol{U}_Y^T \boldsymbol{u}_X\|) \tag{3.12}$$

Where $\boldsymbol{U}_X$ and $\boldsymbol{U}_Y$ are the k first eigenvector matrices of the local autocorrelation $\hat{\Gamma}_t(X)$ and $\hat{\Gamma}_t(Y)$ respectively, and u_X and u_Y are the corresponding eigenvectors with the largest eigenvalue.

Intuitively, two time series X and Y will be considered as close when the angle α formed by the space carrying the information of the time series X and the vector carrying the information the time series Y is zero. In other words X and Y will be close when the value of the $cos(\alpha)$ will be 1. The only assumption made for the computation of LoCo similarity is that the mean of time series data point is zero. This could be easily achieve with z-normalization. LoCo similarity function has many interesting properties and does not require to:

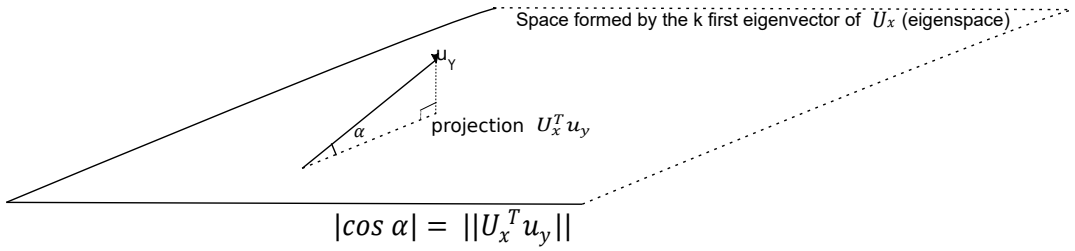


Figure 3.3: Geometric representation of loco similarity.

- Know the probability distribution of the uncertainty,
- Assume the independence of the random variables or the length of u-shapelets.

It is therefore interesting for feature selection, but we still need a dissimilarity function to be able to discover u-shapelet. In the next paragraph, we define a dissimilarity function that has the same properties as LoCo and that is robust to the presence of uncertainty.

3.3 Our Approach

3.3.1 Dissimilarity function

The LoCo similarity function defined on two multivariate time series X and Y approximately corresponds to the absolute value of the cosine of the angle formed by the eigenspaces of X and Y ($|cos(\alpha)|$). A straightforward idea would be to use the $sin(\alpha)$ or α-value as a dissimilarity function but this approach does not work so well; the sine and the angle are not discriminant enough for eigenvector comparison for clustering purpose. We thus propose the following dissimilarity measure (Definition. 9).

Definition 9. *(FOTS : Frobenius cOrrelation for uncertain Time series uShapelet discovery) Given two series X and Y, their FOTS score is defined by*

$$FOTS(X,Y) = \|U_X - U_Y\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{k}(U_X - U_Y)_{ij}^2} \tag{3.13}$$

where $\|\|_F$ is the Frobenius norm.

Because the FOTS computation is based on the comparison of the k-first eigenvectors of the autocorrelation matrices of the time series, it has the same desirable properties of the LoCo similarity function, that is:

- It **allows to capture complex relationship** in time series like the key oscillatory (e.g., sinusoidal) as well as aperiodic (e.g., increasing or decreasing) trends that are present;
- It allows to **reduce the sensibility to transient changes** in time series;
- It is appropriate for the **comparison of short timeseries**.

Moreover, the FOTS dissimilarity function is **robust to the presence of uncertainty** due to the spectral decomposition of the autocorrelation matrices of the time series. The robustness of FOTS to the uncertainty is confirmed by the following theorem:

Theorem 1. ***(HoffmanWielandt)*** *[Bhatia and Bhattacharyya] If X and $X + E$ are $n \times n$ symmetric matrices, then :*

$$\sum_{i=1}^{n} (\lambda_i(X + E) - \lambda_i(X))^2 \leq ||E||_F^2. \tag{3.14}$$

where $\lambda_i(X)$ is the ith largest eigenvalue of X, and $||E||_F^2$ is the squared of the Frobenius norm of E.

The next section explains how FOTS is integrated in the Scalable Unsupervised Shapelet discovery algorithm.

3.3.2 Scalable u-shapelets Algorithm with FOTS score

In this section we do not define a new SUShapelet algorithm, but we explain how we use SUShapelet algorithm with FOTS score (FOTS-SUSh) to deal with uncertainty.

Two main criteria make possible to evaluate the quality of a u-shapelet:

- It has to produce two r-balanced groups.
- It must build two well separated groups, i.e., groups whose gap is maximal.

The gap is, therefore, an essential criterion for the selection of u-shapelets candidate. It is subject to uncertainty because its calculation is based on the Euclidean distance. To remedy this, we propose to use the FOTS score instead of a simple Euclidean distance when calculating the gap in the Scalable u-shapelet algorithm. Algorithms 3 and 4 present a more formal definition:

Definition 10. *The sub-sequence FOTS dissimilarity $\boldsymbol{sd_f(S, T)}$ between a time series T and a sub-sequence S is the minimum of the FOTS score between the sub-sequence S and all possible sub-sequences of T of length equal to the length of S.*

Algorithm 1: ComputeOrderline

Input: u-shapeletCandidate : s,
time series dataset : D
Output: Distance between the u-shapelet Candidate and all the time series of the dataset

```
1 function ComputeOrderline(s, D)
2     dis ← {}
3     s ← zNorm(s)
4     forall i ∈ {1, 2, ..., |D|} do
5         ts ← D(i, :)
6         dis(i) ← sd_f(s, ts)
7     return dis|s|
```

3.4 Experimental Evaluation

3.4.1 Clustering with u-shapelets

The algorithm iteratively splits the data with each discovered u-shapelet: each u-shapelet splits the dataset into two groups D_A and D_B. The time series that belong to D_A are considered as members of the cluster form by the u-shapelet and are then removed from the dataset. A new u-shapelet search continues with the rest of the data until there is no more time series in the dataset or until the algorithm is no more able to find u-shapelet. As a stopping criterion for the number of u-shapelets extracted, the decline of the u-shapelet gap score is examined: the algorithm stops when the gap score of the newly-found u-shapelet becomes less than half of the gap score of the first discovered u- shapelet. This approach is a direct implementation of the u-shapelet definition

Choosing the length N of a uShapelet : The choice of the length of u-shapelet is directed by the knowledge of the domain to which the time series belongs. As part of these experiments, we tested all numbers between 4 and half the length of the time series. We consider as length of u-shapelet the one allowing to better cluster the time series.

Choosing the length w of the windows : The use of overlapping windows for calculating the autocorrelation matrix makes it possible to capture the oscillations present in the time series. During these experiments, we consider that the size of the window is equal to half the length of the u-shapelet.

Choosing the number k of eigenvectors: A practical choice is to fix k to a small value; we use $k = 4$ throughout all experiments. Indeed, key aperiodic trends

Algorithm 2: ComputeGap

```
Input: u-shapeletCandidate : s,
timeseries dataset : D,
lb, ub : lower/upper bound of reasonable number of time series in cluster
Output: gap : gap score
function ComputeGap(s, D, lb, ub)
    dis ← ComputeOrderline(s, D)
    gap ← 0
    for i ← lb to ub do
        D_A ← dis ≤ dis(i), D_B ← dis > dis(i)
        if lb ≤ |D_A| ≤ ub then
            m_A ← mean(D_A), m_B ← mean(D_B)
            s_A ← std(D_A), s_B ← std(D_B)
            currGap ← m_B − s_B − (m_A + s_A)
            if currGap > gap then
                gap ← currGap
    return gap
```

are captured by one eigenvector, whereas key oscillatory trends manifest themselves in a pair of eigenvectors.

3.4.2 Evaluation Metric

To appreciate the quality of the u-shapelets found, we use them for a clustering task. The quality of clustering is evaluated from the Rand Index [Rand] which is calculated as follows:

Let Lc be the cluster labels returned by a clustering algorithm and Lt be the set of ground truth class labels. Let A be the number of time series that are placed in the same cluster in Lc and Lt, B be the number of time series in different clusters in Lc and Lt, C be the number of time series in the same cluster in Lc but not in Lt and D be the number of time series in different clusters in Lc but in same cluster in Lt. The Rand Index is equals to :

$$Rand\,Index = (A + B)/(A + B + C + D) \tag{3.15}$$

3.4.3 Comparison with u-shapelet

Similarly to [Dallachiesa *et al.*], we tested our method on 17 datasets coming from UCR archive [Chen *et al.*2015] representing a wide range of application domains.

The training and testing sets have been joined to obtained bigger datasets. Table 3.1 present detailed information about tested datasets.

Data-set	**Size of dataset**	**Length**	**No. of Classes**	**Type**
50words	905	270	50	IMAGE
Adiac	781	176	37	IMAGE
Beef	60	470	5	SPECTRO
Car	120	577	4	SENSOR
CBF	930	128	3	SIMULATED
Coffee	56	286	2	SPECTRO
ECG200	200	96	2	ECG
FaceFour	112	350	4	IMAGE
FISH	350	463	7	IMAGE
Gun_Point	200	150	2	MOTION
Lighting2	121	637	2	SENSOR
Lighting7	143	319	7	SENSOR
OliveOil	60	570	4	SPECTRO
OSULeaf	442	427	6	IMAGE
SwedishLeaf	1125	128	15	IMAGE
synthetic_control	600	60	6	SIMULATED
FaceAll	2250	131	14	IMAGE

Table 3.1: Datasets

Table 3.2 presents the comparison of the two algorithms.

3.4.4 Discussion

The use of the FOTS score associated with the SUShapelet algorithm makes it possible to discover different u-shapelets than those found by the Euclidean distance. The FOTS-SUSh improves the results of time series clustering because the FOTS score takes into account the intrinsic properties of the time series when searching for u-shapelets and is robust to the presence of uncertainty. This improvement is particularly significant when the FOTS score is used for the clustering of time series containing several small oscillations. Indeed, these oscillations are not captured by the Euclidean distance but are by the FOTS score whose calculation is based on the autocorrelation matrix. This observation is illustrated by the result obtained on SwedishLeaf dataset.

Time complexity analysis

ED can be computed in $\mathcal{O}(n)$ and FOTS score is computed in $\mathcal{O}(n^{\omega})$, $2 \leq \omega \leq 3$ due to the time complexity of the eigenvector decompositions [**?**]. The computation

Datasets	RI_SUSh	RI_FOTS
50words	0.811	**0.877**
Adiac	0.796	**0.905**
Beef	0.897	**0.910**
Car	0.708	**0.723**
CBF	0.578	**0.909**
Coffee	0.782	**0.896**
ECG200	0.717	**0.866**
FaceFour	0.859	**0.910**
FISH	0.775	**0.899**
Gun_Point	0.710	**0.894**
Lighting2	0.794	**0.911**
Lighting7	0.757	**0.910**
OliveOil	0.714	**0.910**
OSULeaf	0.847	**0.905**
SwedishLeaf	0.305	**0.909**
synthetic_control	0.723	**0.899**
FaceAll	0.907	**0.908**

Table 3.2: Comparison of the Rand Index of SUSH (RI_SUSh) and FOTS-SUSh (RI_FOTS). The best Rand Index is in bold

of FOTS score is then more expensive than that of ED. However, its use remains relevant for u-shapelet research as they are often small.

3.5 Conclusion and Future Work

The purpose of this work was to discover u-shapelets on uncertain time series. To answer this question, we have proposed a dissimilarity score (FOTS) adapted to the comparison of short time series, whose computation is based on the comparison of the eigenvector of the autocorrelation matrices of the time series. This score is robust to the presence of uncertainty, it is not very sensitive to transient changes, and it allows capturing complex relationships between time series such as oscillations and trends. The FOTS score was used with the Scalable Unsupervised Shapelet Discovery algorithm for clustering 17 literature datasets and showed an improvement in the quality of clustering evaluated using the Rand Index. By combining the benefits of the u-shapelets algorithm, which reduces the adverse effects of uncertainty, and the benefits of the FOTS score, which is robust to the presence of uncertainty, this work is defining a framework for clustering uncertain time series. As a perspective to this work, we plan to use the FOTS score for fuzzy clustering of uncertain time series.

Chapter 4

Compression for a better classification with Dynamic Time Warping

Abstract : *Dynamic Time Warping (DTW) is a time series alignment algorithm that is often used because it considers that it exits small distortions between time series during their alignment. However, DTW sometimes produces pathological alignments that occur when, during the comparison of two time series X and Y, one data point of the time series X is compared to a large subsequence of data points of Y. In this paper, we demonstrate that to compress time series using Piecewise Aggregate Approximation (PAA) is a simple strategy that greatly increases the quality of the alignment with DTW this is particularly true for synthetic data sets.*

4.1 Introduction

Time series databases are often large and several transformations have been introduced in order to represent them in a more compact way. One of these transformations is Piecewise Aggregate Approximation (PAA) [**?**], which consists in dividing a time series into several segments of fixed length and replacing the data points of each segment with their averages. Due to its simplicity and low computational time, PAA has been widely used as a basic primitive by other temporal data mining algorithms such as [Lin *et al.*2003, Sun *et al.*2014, Lkhagva *et al.*], in order

- To construct symbolic representations of time series; [Camerra *et al.*, Ulanova *et al.*].
- To construct an index for time series; [Zhao and Itti, Keogh and Pazzanib, Kate]. Indeed, PAA allows queries which are shorter than length for which the index was built, this very desirable feature is impossible in Discrete Fourier Transform, Singular Value Decomposition and Discrete Wavelet Transform.
- To classify time series.

4.1.1 Why the use of PAA can improve alignment with Dynamic Time Warping

An important task is time series comparison that can be done in two main ways. Either the comparison method considers that there is no time distortion as in Euclidian distance (ED), or it considers that some small time distortions exist between time axis of time series as in Dynamic Time Warping alignment algorithm (DTW) [Zhang *et al.*b]. Since time distortion often exists between time series, DTW often has better results than the ED [Chen *et al.*2015]. An exhaustive comparison of time series algorithms [Bagnall *et al.*a] shows that DTW is among the efficient techniques to be used. However, DTW has two major drawbacks: the comparison of two time series under DTW is time-consuming [Rakthanmanon *et al.*2012] and DTW sometimes produces pathological alignments [Keogh and Pazzania]. A pathological alignment occurs when, during the comparison of two time series X and Y, one datapoint of the time series X is compared to a large subsequence of datapoints of Y. A pathological alignment causes a wrong comparison.

Three categories of methods are used to avoid pathological alignments with DTW:

- The first one adds constraints to DTW [Ratanamahatana and Keogh, Yu *et al.*, Candan *et al.*, Sakoe and Chiba1978, Jeong *et al.*]. The main idea here is to limit the length of the subsequence of a time series that can be compared to a single datapoint of another time series.

- The second one suggests skipping datapoints that produce pathological alignment during the comparison of two time series [Longin *et al.*, Itakura, Myers *et al.*].

- The third one proposes to replace the datapoints of time series with a high-level abstraction that captures the local behavior of those time series. A high-level abstraction can be a histogram of values that captures the repartition of time series datapoints in space [Zhang *et al.*b] or a feature that captures the local properties of time series, such as the trend with Derivative DTW (DDTW) [Keogh and Pazzania].

Another simple but yet interesting way to capture local properties of time series is to consider mean of segments of the time series as PAA does. Indeed, the use of the mean reduces the harmful effects of singularities contained in the data and thus allows to avoid the pathological alignments. However, one major challenge with PAA is the choice of the number of segments to consider especially with long time series.

4.1.2 The problem of choosing a suitable segment number for PAA

If the number of segments considered with PAA is too small, the resulted representation is compact, but it contains less information. On the other side, if the number of segments is too large, the obtained representation is less compact and more prone to the noise contained in the original time series (Fig. 4.1). Our idea is that a number of segments for PAA will be considered as good if it allows obtaining a compact representation of the time series, and also if it preserves the quality of the alignment of time series. So when considering classification task, one of the best classification algorithm to use for evaluating the quality of time series alignment is one nearest neighbor (1NN). Indeed, its classification error directly depends on time series alignment, since 1NN has no other parameters [Wang *et al.*b].

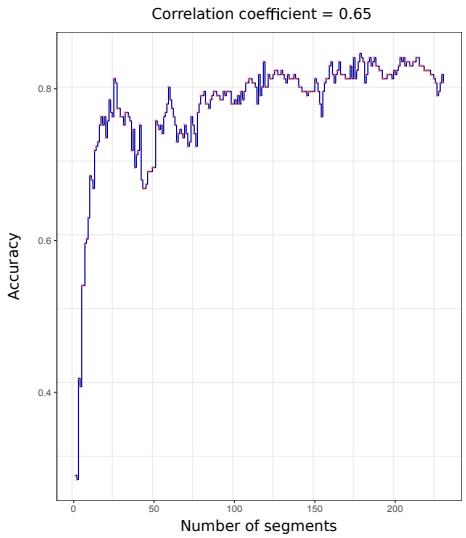


Figure 4.1: Relation between Accuracy and the number of segment on FISH dataset. The accuracy is computed from the algorithm one nearest neighbor associated with PDTW. When the number of segments considered is very small, there is a loss of information and the accuracy is reduced. However, considering all the points in the time series, we also do not obtain maximum accuracy due to the presence of noise or singularities [Keogh and Pazzania] in the data.

4.1.3 Summary of Contributions

In this paper,

- We define the problem of preprocessing time series with PAA for a better classification with DTW.

- We propose a parameter free heuristic for aligning piecewise aggregate time series with DTW, which approximates the optimal value of the number of segments to be considered with PAA.

- We make our source code and all our results available to allow the reproducibility of our experiments. work.

The rest of the paper is organized as follow: In Section 4.2 we recall the definitions and background. Section 4.3 explains our approach. Section 4.4 presents experimental results and comparisons to others methods. Section 4.5 offers conclusions and venues for future work.

4.2 Background and related work

Let's recall some definitions.

Definition 11. *: A **time series** $X = x_1, \cdots, x_n$ is a sequence of numerical values representing the evolution of a specific quantity over time. x_n is the most recent value.*

Definition 12. *: A segment X_i of length l of the time series X of length n ($l < n$) is a sequence constituted by l variables of X starting at the position i and ending at the position $i + l - 1$. We have: $X_i = x_i, x_{i+1}, ..., x_{i+l-1}$*

Definition 13. *: The arithmetic average of the data points of a segment X_i of length l is noted $\bar{X}_i$ and is defined by:*

$$\bar{X}_i = \frac{1}{l} \sum_{j=0}^{l-1} x_{i+j} \tag{4.1}$$

Definition 14. *: Let T be the set of time series. The Piecewise Aggregate Approximation (PAA) is defined as follows:*

$$\begin{aligned} & PAA : T \times \mathbb{N}^* \to T \\ & (X, N) \mapsto PAA(X, N) \quad = \quad \begin{cases} \bar{X}_1, \cdots, \bar{X}_N \; if \; N < |X| \\ X \; otherwise \end{cases} \end{aligned} \tag{4.2}$$

Definition 15. *: Let $d \subseteq T$ be a subset of time series, $N \in \mathbb{N}^*$, $PAAset(d, N) = \{PAA(X, N), \forall X \in d\}$*

4.2.1 Dynamic Time Warping algorithm.

DTW [**?**] is an algorithm of time series alignment algorithm that performs a non-linear alignment while minimizing the distance between two time series. To align two time series : $X = x_1, x_2, \cdots, x_n$; $Y = y_1, y_2, \cdots, y_m$, the algorithm constructs an $n \times m$ matrix where the cell (i, j) of the matrix corresponds to the squared distance $(x_i - y_j)^2$ between x_i and y_j. To find the best alignment between two time series, DTW constructs the path that minimizes the sum of squared distances. This path, noted $W = w_1, w_2, \ldots, w_k, \ldots, w_K$, must respect the following constraints:

- Boundary constraint: $w_1 = (1, 1)$ and $w_K = (n, m)$
- Monotonicity constraint: given $w_k = (i, j)$ and : $w_{k+1} = (i', j')$ then : $i \leq i'$ and $j \leq j'$
- Continuity constraint: given $w_k = (i, j)$ and : $w_{k+1} = (i', j')$ then : $i' \leq i + 1$ and : $j' \leq j + 1$

The warping path is computed by an algorithm based on the dynamic programming paradigm that solves the following recurrence:

$$\gamma(i, j) = d(x_i, y_j) + min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\}, \tag{4.3}$$

where $d(x_i, y_j)$ is the squared distance contained in the cell (i, j) and $\gamma(i, j)$ is the cumulative distance at the position (i, j) that is computed by the sum of the squared distance at the position (i, j) and the minimal cumulative distance of its three adjacent cells.

4.2.2 Piecewise Dynamic Time Warping

Piecewise Dynamic Time Warping Algorithm (PDTW) [Keogh and Pazzanib] is the DTW algorithm applied on Piecewise Aggregate time series [**?**]. Let $N \in \mathbb{N}^*$, X and Y be two time series.

$$PDTW(X, Y, N) = DTW(PAA(X, N), PAA(Y, N)). \tag{4.4}$$

4.3 Heuristic search of the number of segments

4.3.1 Problem definition.

Definition 16. *Let $D = \{d_i\}$ be a set of datasets composed of time series and $X \in d_i$ be a time series of the dataset d_i; we note $|X| = n$ the length of the time series X. Let $N \in \mathbb{N}^* and\, N \leq n$, $1NNPDTW(d_i, N)$ is the classification error of 1-NN with PDTW using N segments on d_i.*

Our goal is to find a number of segments $N \in \{1 \dots n\}$ such that

$$1NNPDTW(d_i, N) = \min_{1 \leq \alpha \leq n} \{1NNPDTW(d_i, \alpha)\}.$$

The simplest way to find the value for the number of segments that minimized the classification error is to test all the possible values. Obviously, this method is time-consuming as we have to test n values to find the one that has the minimal classification error. The time complexity of this process is :

$$O((\frac{|trainingset|}{2})^2 \times \sum_{N \in C} N^2),\ |C| = n,$$

where C is the set of values for the number of segments.

To reduce the time of the search, the FDTW proposes to look for the number of segments with the minimal classification error without testing all the possible values.

4.3.2 Greedy Randomized Adaptive Search Procedures

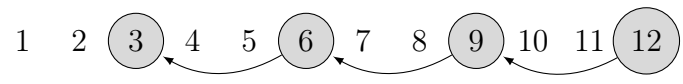
The Greedy Randomized Adaptive Search Procedures (GRASP) is a multi-start, or iterative metaheuristic proposed by Feo and Resende (1995) [Feo and Resende1995], in which each iteration consists of two phases: firstly a new solution is constructed by a greedy randomized procedure and then is improved using a local search procedure.

The greediness criterion establishes that elements with the best quality are added to a restricted candidate list and chosen at random when building up the solution. The candidates obtained by greedy algorithms are not necessarily optimal. So, those candidates are used as initial solutions to be explored by local search. The heuristic that we proposed is build upon GRASP and strengthened with an inclusion of specific global search component.

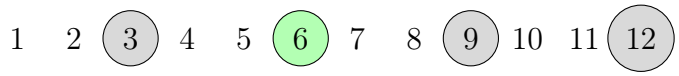
4.3.3 Parameter free heuristic

The idea of our heuristic is the following:

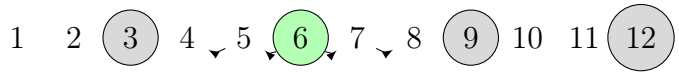
1. We choose N_c candidates distributed in the space of possible values to ensure that we are going to have small, medium and large values as candidates. The candidates values are: n, $n - \left\lfloor \frac{n}{N_c} \right\rfloor$, $n - 2 \times \left\lfloor \frac{n}{N_c} \right\rfloor$, ..., $n - N_c \times \left\lfloor \frac{n}{N_c} \right\rfloor$. For instance, if the length of time series is $n = 12$ and the number of candidates is $N_c = 4$, we are going to select the candidates 12, 9, 6, 3.



2. We evaluate the classification error with $1NNPDTW$ for each chosen candidate, and we select the candidate that has the minimal classification error: it is the best candidate. In our example, we may suppose that we get the minimal value with the candidate 6 : it is thus the best candidate at this step.



3. We respectively look between the predecessor (i.e., 3 here) and successor (i.e., 9 here) of the best candidate for a number of segments with a lower classification error : this number of segments corresponds to a local minimum. In our example, we are going to test values 4, 5, 7 and 8 to see if there is a local minimum.



4. We restart at step one while choosing different candidates during each iteration to ensure that we return a good local minimum. We fix the number of iterations to $k \leq \lfloor log(n) \rfloor$. At each iteration, the first candidate is $n-(number\_of\_iteration-1)$.

In short, in the worst case, we test the first N_c candidates to find the best one. Then, we test $\frac{2n}{N_c}$ other candidates to find the local minimum. We finally perform $nb(N_c) = N_c + \frac{2n}{N_c}$ tests. The number of tests to be performed is a function of the number of candidates. Hence, how many candidates should we consider to reduce the number of tests? The first derivative of nb function vanishes when $N_c = \sqrt{2n}$ and its second derivative is positive; so the minimal number of tests is obtained when the number of candidates is : $N_c = \sqrt{2n}$.

Lemma 1. : For a given a dataset d_i, $FDTW(d_i) \leq 1NNDTW(d_i)$. The quality of the alignment of our heuristic is better than that of DTW.

Proof : $1NNDTW(d_i) = 1NNPDTW(d_i, n)$. Then, $1NNDTW(d_i)$ is one of the candidates considered by the heurisitic $FDTW$. Since $FDTW$ returns the minimal classification error from all candidates, the classification error of $1NNDTW$ is always greater than or equal to $FDTW$.

A heuristic does not always give the optimal value. To ensure that it gives a result not far from the optimal value, one approach is to guarantee that the result of the heuristic always lies in an interval with respect to the optimal value [Ibarra and Kim].

In our case, we are looking for the number of segments that allows a good alignment of time series. The alignment is good when the classification error with 1NN is minimal or when the accuracy is maximal.

Let d_i be a dataset:

$acc_{max(d_i)} = 1 - \min_{1 \leq \alpha \leq n} \{1NNPDTW(di, \alpha)\}$ is the maximal accuracy for the dataset d_i,

$acc_{DTW} = 1 - 1NNDTW(d_i)$ is the accuracy with d_i and 1NNDTW and

$acc_{FDTW} = 1 - FDTW(d_i)$ is the accuracy of our heuristic.

To ensure the quality of our heuristic FDTW, wee hypothesized tat $1NNDTW$ is better than Zero Rule classifier. Zero Rule classifier is a simple classifier that predicts the majority class of test data (if nominal) or average value (if numeric). Zero Rule is often used as baseline classifier [Cuřín *et al.*]. The minimal value of the accuracy of Zero Rule is $\frac{1}{c}$ where c is the number of classes of the dataset.

Proposition 1. : For a given dataset d_i that has c_i classes, $c_i \in \mathbb{N}^*$,

$if\ acc_{DTW} \geq \frac{1}{c_i}\ then\ \frac{1}{c_i} \times acc_{max} \leq acc_{FDTW} \leq acc_{max}$

Proposition 1 shows that when 1NN associated with DTW has a better accuracy than the baseline classifier Zero Rule, the FDTW heuristic is an approximation.

: By definition, $acc_{FDTW} \leq acc_{max}$ We look for $\beta \in \mathbb{N}$ such that

$$\frac{1}{\beta} \times acc_{max} \leq acc_{FDTW} \tag{4.5}$$

$$\frac{1}{\beta} \times acc_{max} \leq acc_{FDTW} \Leftrightarrow \frac{acc_{max}}{acc_{FDTW}} \leq \beta \tag{4.6}$$

$$However, \quad \frac{acc_{max}}{acc_{FDTW}} \leq \frac{1}{acc_{FDTW}} \tag{4.7}$$

$$because \quad acc_{max} \leq 1 \tag{4.8}$$

$$And, \quad \frac{1}{acc_{FDTW}} \leq \frac{1}{acc_{DTW}} \tag{4.9}$$

$$because \quad acc_{DTW} \leq acc_{FDTW} \tag{4.10}$$

$$So, \quad \frac{1}{acc_{DTW}} \leq c_i \tag{4.11}$$

$$because \quad \frac{1}{c_i} \leq acc_{DTW} \quad by \quad hypothesis \tag{4.12}$$

we take $\beta = c_i$.

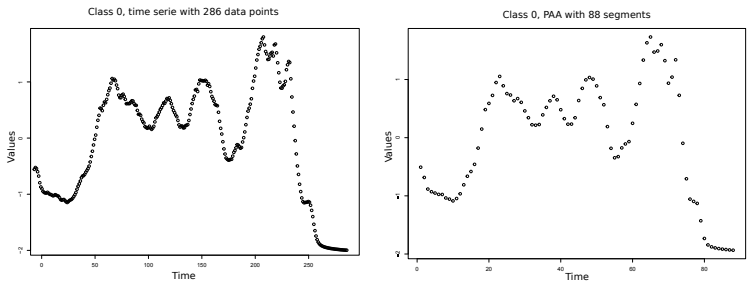


Figure 4.2: Coffee dataset time series compression with PAA: original time series (left) versus PAA represetion using 88 segments (right). The number of segments is found by FDTW and allow to reduce the length of the time series while retaining the information that it contains.

4.4 Experiments and discussion

4.4.1 Datasets

The performance of FDTW has been evaluated on 84 datasets of the UCR time series datamining archive [Chen *et al.*2015], which provides a large collection of datasets that cover various domains. Each dataset is divided into a training set and a testing set.

4.4.2 Compression

When it is used with a suitable segments number determined with FDTW, PAA allows compression of the time series of the **Coffee** without loss of information. Although they are more compact, the obtained time series capture the main variations of the original time series (Fig. 4.2).

4.4.3 Classification

To evaluate the quality of FDTW, we compared its classification errors with that of 35 other classification algorithms [Bagnall *et al.*b] of the literature on 84 datasets of UCR archive. The classification error was calculated based on the holdout model evaluation. FDTW used the training set to find the number of segments N using 3-fold cross-validation. If two numbers of segments N_1 and N_2 are associated with the same classification error, we retain the largest. The performances of the algorithms are compared using the Nemenyi test that compares all the algorithms pairwise and provides an intuitive way to visualize the results (Fig. 4.3). The Nemenyi test allows ranking the classification algorithms according to their average accuracy on 84 datasets.

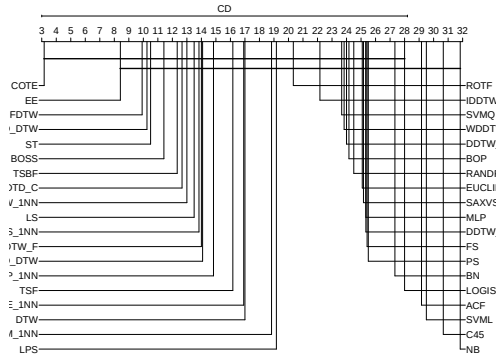


Figure 4.3: Critical difference diagram for FDTW and 36 other classification algorithms on 6 simulated datasets.

The value of the segment number N found on the training set may in some cases not be appropriate for the testing set. We speak of an error of generalization which is due to the representativeness of the training set. Thus, FDTW obtains good results on the simulated data sets 3rd / 37 algorithms in terms of average accuracy (Fig. 4.3) because the data of the training set and the testing set are generated by the same models.

However, to evaluate the significance of the difference between the classification algorithms on 84 datasets, we use the Wilcoxon signed rank test with continuity correction which has more statistical power.The results of these experiments are summarized below.

The experiments show that despite data compression :

- FDTW have better performance than Naive Bayes (NB), C45, logistic regression (Logistic), BN;
- FDTW has similar performance to that of 26 other algorithms in the literature, namely : SVMQ, RANDF, ROTF, MLP, EUCLIDEAN_1_NN, DDTW_R1_1NN, DDTW_RN_1NN, ERP_1NN, LCSS_1NN, MSM_1NN, TWE_1NN, WDDTW_1NN, WDTW_1NN, DD_DTW, DTD_C, LS, BOP, SAXVSM, TSF, TSBF, LPS, PS, CID_DTW, SVML, FS, ACF;
- DTW_F, Shapelet Transform (ST), BOSS, Elastic Ensemble (EE) and COTE perform better overall than FDTW.

This demonstrates its competitiveness. Moreover, FDTW outperforms the best result reported in the literature on UWaveGestureLibraryAll dataset (Fig. 4.4). The challenge with the UWaveGestureLibraryAll dataset is to recognize the gesture made by a user from measurements made by accelerometers. As reported here [Bagnall *et al.*a] the best accuracy obtained on this dataset is 83.44% with TSBF algorithm. FDTW outperforms this result and allows to obtain **91.87%** of accuracy.

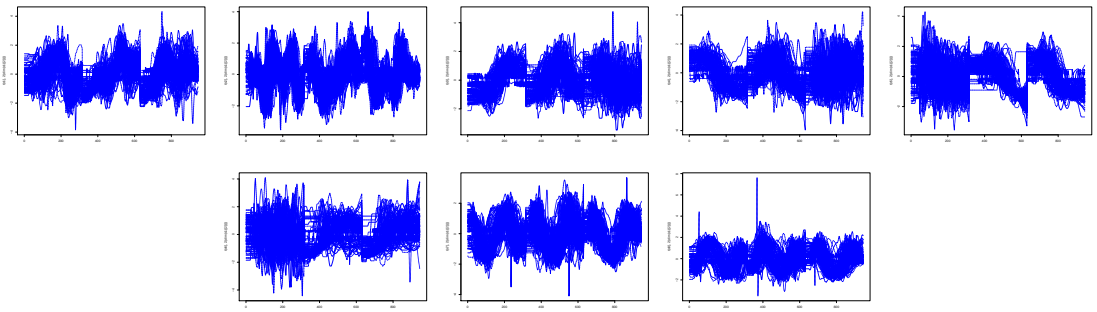


Figure 4.4: Eight types of time series corresponding to the vocabulary of 8 gestures.

4.5 Conclusion

Our problem in this paper was to choose an appropriate number of segments to compress time series with PAA in order to improve the alignment with DTW. To achieve this goal we proposed a parameter Free heuristic named FDTW with approximate the optimal number of segment to use. The experiments show that our heuristic increased the quality of alignment of time series with especially on synthetic datasets where DTW associated with PAA perform better than any other variant of DTW on a classification task and was rank 3/37 behind two ensemble classification algorithms COTE and EE. This work allows reducing the storage space and the processing time of time series while increasing the quality of the alignment of DTW. The same strategy presented in FDTW can be used to find the number of segments to be considered for the indexation and for symbolic representations of time series like SAX [Lin *et al.*2003], ESAX [Lkhagva *et al.*], SAX-TD [Sun *et al.*2014].

Chapter 5

Symbolic representation of cyclic time series based on properties of cycles

Abstract : *The analysis of cyclic time series from bio-mechanics is based on the comparison of the properties of their cycles. As usual algorithms of time series classification ignore this particularity, we propose a symbolic representation of cyclic time series based on the properties of cycles, named SAX-P. The resulting character strings can be compared using the Dynamic Time Warping distance. The application of SAX-P to propulsive moments of three subjects (S1, S2, S3) moving in Manual Wheelchair highlight the asymmetry of their propulsion. The symbolic representation SAX-P facilitates the reading of the cyclic time series and the clinical interpretation of the classification results.*

5.1 Introduction

Generally, during his locomotion, the human being performs cyclic movements (eg walking, running, swimming, cycling). The bio-mechanical analysis of these movements is performed with various measuring instruments (eg force and acceleration sensors, kinematic analysis systems) that enable continuous recording over long periods of many kinematic and dynamic parameters. These recordings produce long time series composed of many cycles or patterns, representative of the movements made and effort produced by the subject during his displacement (Fig. 5.1).

These cycles are the time series analysis units and have several characteristic properties such as the minimum value, the area under the cycle [Vegter *et al.*2014] (Fig. 5.2).

For comparing time series, several previous studies suggested to break them into small segments and then to compare the properties of their segments. A segment of a time series is a sequence of consecutive values belonging to it [Abonyi *et al.*2003].

[**?**] proposed replacing each segment of a time series $X = x_1, x_2, \cdots, x_n$ by its

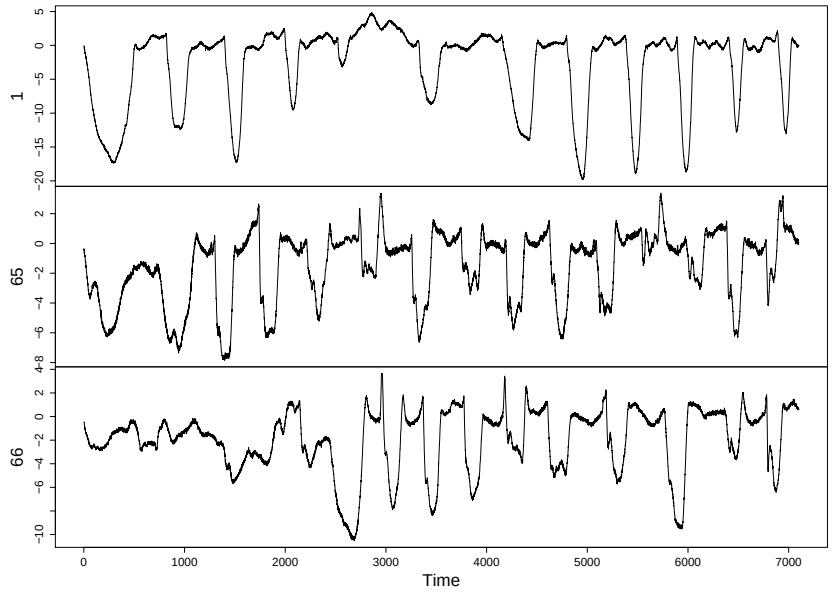


Figure 5.1: Cyclic time series form manual wheelchair locomotion

mean values; $\bar{x}_i = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{(\frac{n}{N})i} x_j$ transforming the time series, which is a sequence of values, in the suite of the means of its N segments $\bar{X} = \bar{x}_1, \bar{x}_2, \cdots, \bar{x}_N$ This method is known as Piecewise Aggregate Approximation (PAA) (Fig. 5.3). The time series C and Q are then compared by calculating the distance DR between the suite $\bar{C}$ and $\bar{Q}$ of the means of their segments :

$$DR(\bar{C}, \bar{Q}) = \sqrt{\frac{n}{N} \sum_{i=1}^{N} (\bar{c}_i - \bar{q}_i)^2} \tag{5.1}$$

The main objective of PAA was to reduce the length of the time series. However, as it computes the segments means, it also allows us to compare two time series C and Q from the properties of their segments (Equation 5.1).

[Lin *et al.*2003] were based on the PAA method to provide a symbolic representation of time series called Symbolic Aggregate Approximation (SAX). The objective of SAX is to assign a letter to each segment. To do this, the domain of the values of the time series is divided into intervals so that every point of the temporal series has approximately the same probability to belong to an interval and a letter is associated with each of these intervals. Then each segment of the time series is associated with the letter of the interval to which belongs its average (Fig. 5.4).

With SAX, the distance $MINDIST$ between two strings $\hat{Q}$ and $\hat{C}$ of length N is calculated from the distance between the borders of the intervals represented by each character in the string (Equation 5.2).

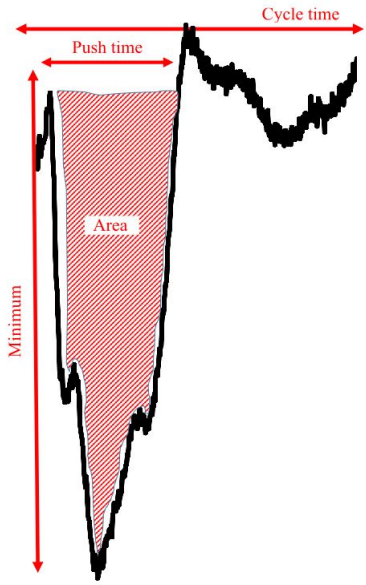


Figure 5.2: Properties of a cycle

$$MINDIST(\hat{Q}, \hat{C}) = \sqrt{\frac{n}{N}\sum_{i=1}^{N}(dist(\hat{q}_i, \hat{c}_i))^2} \tag{5.2}$$

$\hat{q}_i$ *et* $\hat{c}_i$ are characters and $dist()$ is the distance between the borders of the intervals which represent these characters [Lin *et al.*2003]. However, two segments with very different shapes can have the same average and be represented by the same letter: the mean is not enough to define a segment. In order to solve this problem, [Lkhagva and Kawagoe2006] proposed the ESAX model that considers three properties for each segment: its mean, its minimum and maximum (Fig. 5.5).

Thereafter, [Sun *et al.*2014] proposed the SAX-TD model that takes into account two properties for each segment: its mean and trend. They then adjust the distance used by the SAX method for it to take into account the trend (Fig. 5.6).

Both methods provide better results than the SAX method [Sun *et al.*2014]. However, they have the disadvantage of increasing the number of symbols required to represent the time series. Indeed, the method ESAX triple the size of the representation of a time series provided by the SAX method, while the SAX-TD method the double. In addition, the previous four methods have two major drawbacks: they consider fixed-size segments, while the cycles are variable-sized segments, and they do not take into account the characteristic properties of cycles such as the duration and the surface under a cycle. Our goal is to provide a symbolic representation that takes into account several properties for each cycle, but without increasing the number of symbols used for the representation.

The symbolic representations obtained have another advantage; they allow to

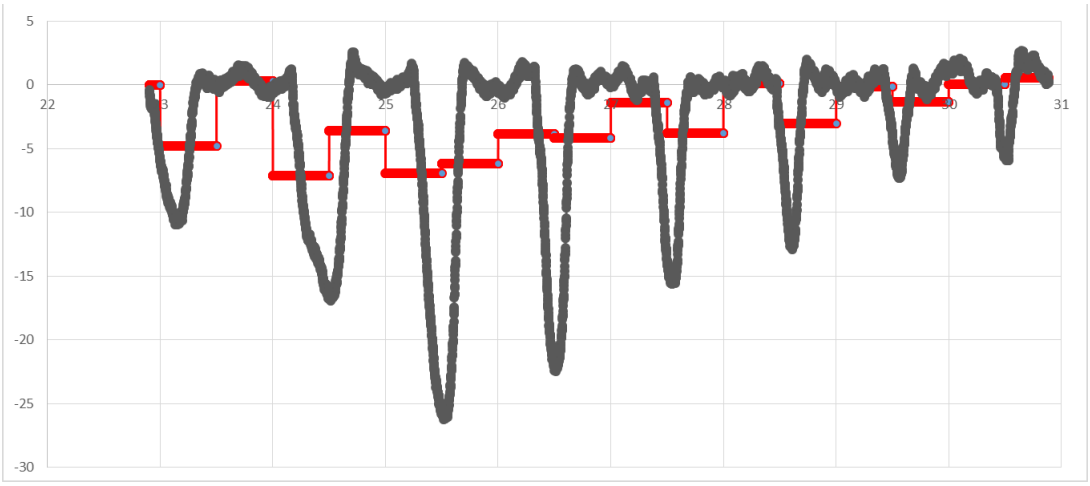


Figure 5.3: Piecewise aggregate approximation of a cyclic time series

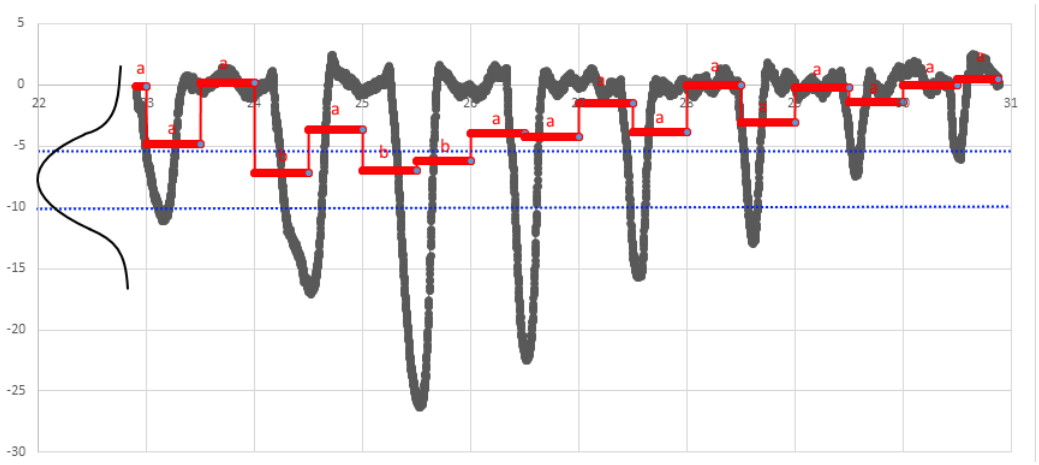


Figure 5.4: Symbolic Aggregate approXimation of a cyclic time series

use a large number algorithms available for sequence analysis like novelty detection (finding unusual shapes or sub-sequences), motif discovery (finding repeated shapes or sub-sequences) [Begum and Keogh2014], clustering, classification, indexing and also some interesting algorithms for text processing or the bio-informatics community [Aach and Church2001, Papapetrou *et al.*2011, Dietterich2002].

5.2 SAX-P

A prerequisite to be able to build a symbolic representation based on the cycles of the cyclic time series is to be able to segment the cyclic time series into consecutive cycles.

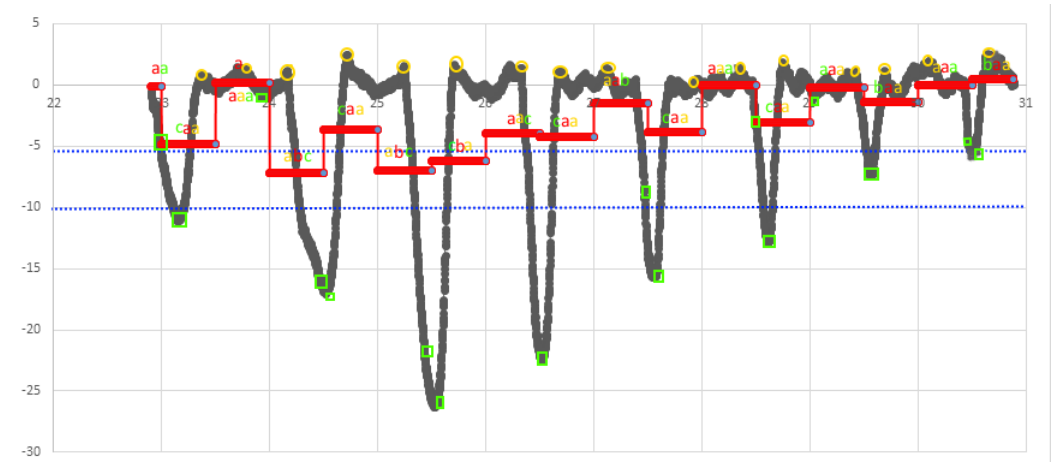


Figure 5.5: Extended Symbolic Aggregate approXimation of a cyclic time series

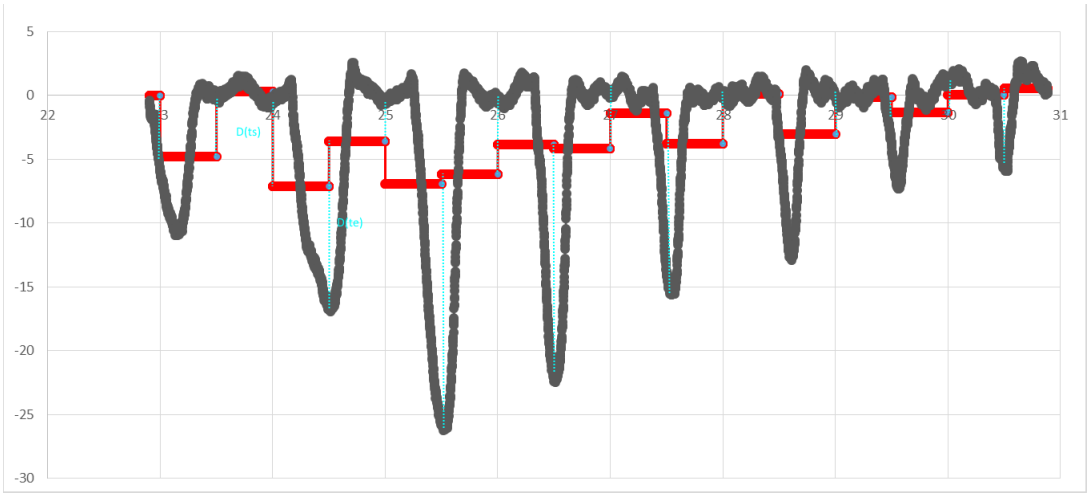


Figure 5.6: Trend Symbolic Aggregate approXimation of a cyclic time series

5.2.1 Segmentation of cyclic time series

The principle used to segment cyclic time series is as follows: A cycle contains all the data points between the beginning of two consecutive peaks. To locate the peaks, we set a threshold (Fig. 5.8). The threshold considered can be the first or the second quartile of the time series data point.

If the current value of the time series is below this threshold, then it is a peak. It is then necessary to turn back to find the moment of the beginning of the peak. The figure (Fig. 5.9) presents the results obtained after segmentation of a cyclic time series.

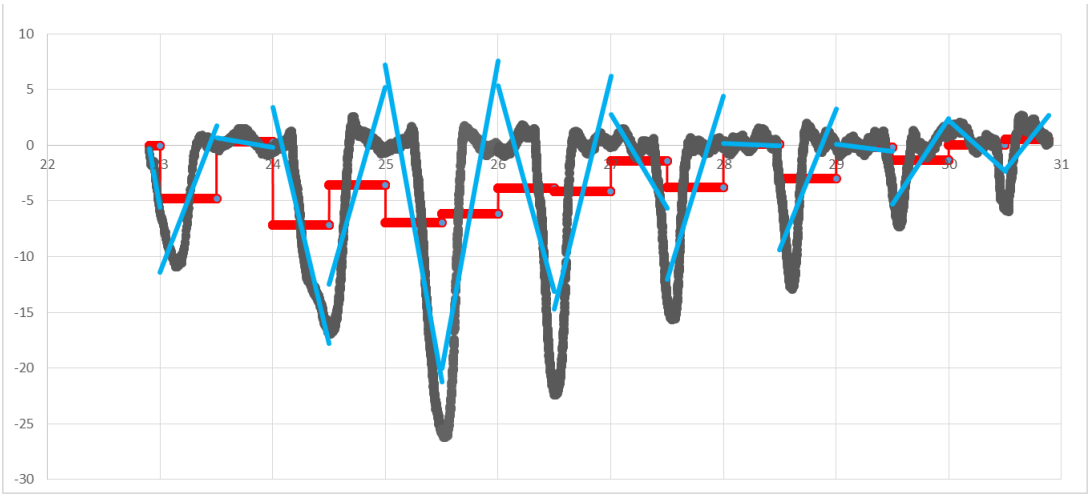


Figure 5.7: Properties of a cycle

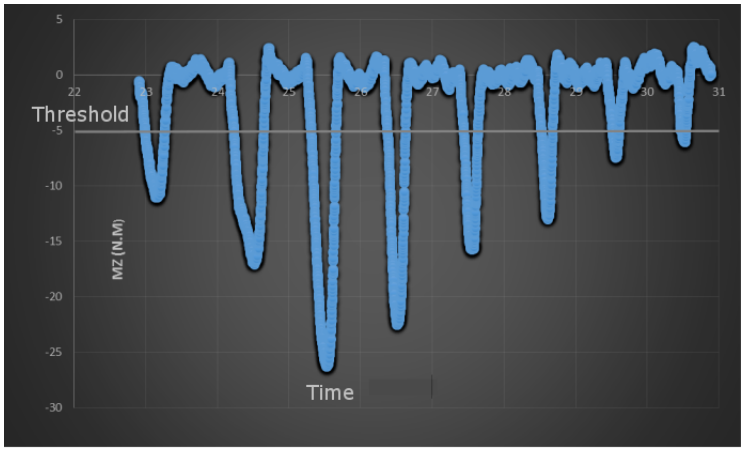


Figure 5.8: Threshold for the segmentation of cyclic time series

5.2.2 From cycles to letters

The method SAX-P is based on SAX and works as follows:

1. A cyclic time series is split in successive segments using a threshold for identifying the beginning and the end of cycles, which have variable durations;

2. Several parameters (properties) are computed on each segment: cycle time, push time, mean, median, standard deviation, minimum and maximum values, and the area under the time series curve. As all these parameters have different units, they must be normalized (i.e. centered and reduced) (Fig. 5.10);

3. Segments are then gathered in clusters using a classification algorithm [Esling and Agon2012] and each cluster is named by a capital letter (Fig. 5.11);

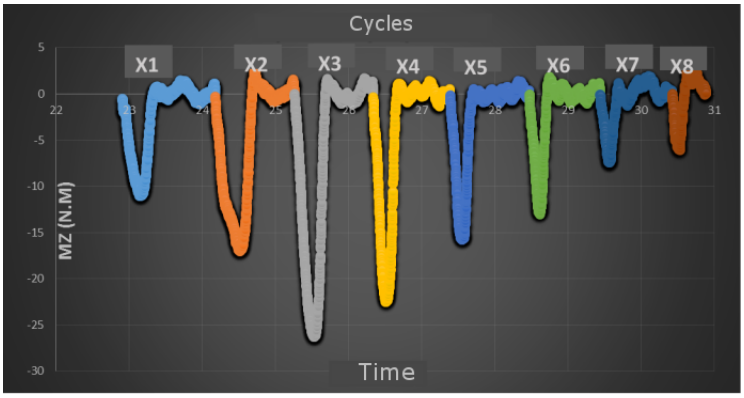


Figure 5.9: Segmentation

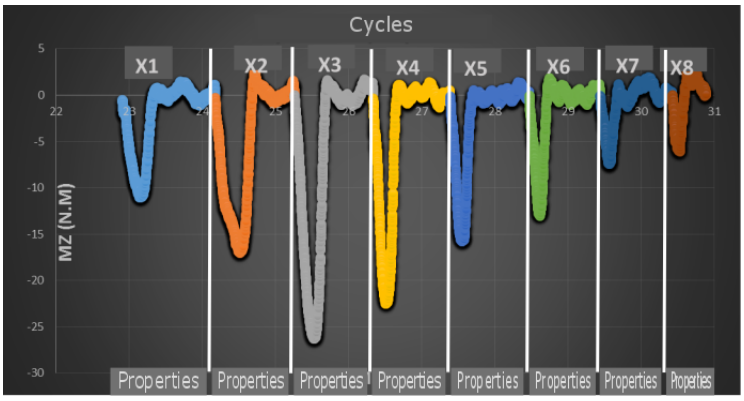


Figure 5.10: Some properties are computed on each cycle

4. Each segment is replaced by the letter of the cluster to which it belongs, so that the initial cyclic time series is then represented by a string of characters (Fig. 5.12);

The distance between two strings, which may have different numbers of characters, is computed using Dynamic Time Warping [Petitjean *et al.*2014] which is known as the best distance measure for several domains [Ding *et al.*2008]. The distance between two characters is the euclidean distance between the centers of the classes represented by those characters.

Unlike SAX, ESAX and SAX-TD methods that require to fix the length of segments to consider when building the symbolic representation of a time series, SAX-P considers the cycles which constitute basic unit of analysis of time series recorded during cyclic movements and also allows taking into account several characteristic features for each cycle. Figure 5.12 presents the symbolic representations obtained with the SAX method (in small letters) and SAX-P (in capital letters). It illustrates that SAX-P unlike SAX considers cycles of the time series during the construction of the symbolic representation.

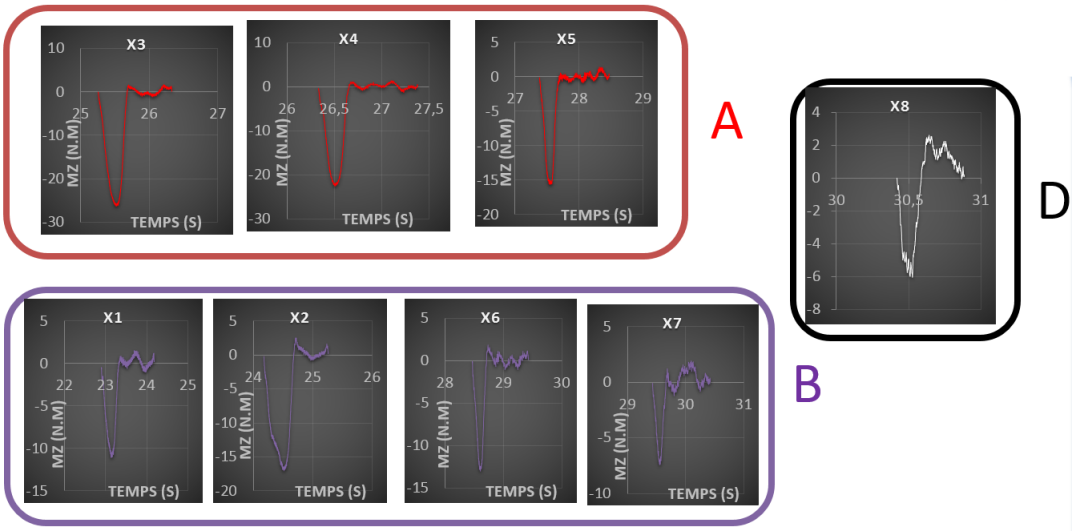


Figure 5.11: Classification of cycles based on properties

5.3 Application to manual wheelchair locomotion

This method has been applied to the axial moment (Mz) measured by both right and left rear wheels of an instrumented Manual WheelChair (MWC) during five there and back 10-m linear displacements between two cones performed by three handicapped subjects. We group propulsion cycles into 5 clusters (Table 5.1) and we obtained a symbolic representation for Mz (Table 5.2).

An important task of analyzing manual wheelchair locomotion is the comparison of its rolling movement. Experts from the fields seek to compare two movements simultaneously taking into account several criteria or properties. Applying SAX-P method to the time series of axial moments exerted by a wheelchair user on the right and left wheels greatly facilitates the comparison, the analysis and the interpretation of these time series:

- At first sight (Table 2), it immediately appears that during their second 10-m run the three subjects analyzed here did not exert the same number of pushes for moving a MWC on the same distance (S1: 7-8; S2: 12-13; S3: 5-7);
- It is also obvious that each subject did not exert the same number of pushes on both rear wheels. Moreover, although right and left pushes exerted by one subject globally belong to the same clusters, the total distance between all these pushes can be more (S2: 354) or less (S3: 44) high. Both these

Cluster	A	B	C	D	E
Nb of cycles	18	36	59	18	104
Cycle time (s)	1.2	1.0	1.0	1.7	0.8
Push time (s)	0.6	0.3	0.4	1.0	0.3
Mz Min (Nm)	-22.3	-17.4	-11.4	-8.7	-6.4
Mz Max (Nm)	0.1	0.1	0.1	0.7	0.1
Mean (Nm)	13.6	-8.1	-6.2	-3.0	-3.3
Median (Nm)	-16.1	-10.8	-7.4	-4.2	-3.9
IRQ (Nm)	12.4	10.7	6.0	4.3	3.1
SD (Nm)	7.1	5.6	3.4	2.6	1.8
Area (Nm.s)	-7.1	-2.3	-2.2	-1.8	-1.0

Table 5.1: Average vectors of the properties of classes (A, B, C, D, E) used for the symbolic representation of the axial moment (Mz) SAX-P takes into account the surface under the push, the time-push and the time-cycle.

Subject	S1		S2		S3	
Push	Right	Left	Right	Left	Right	Left
1	C	A	C	D	E	D
2	B	B	E	E	D	E
3	B	B	C	E	C	E
4	B	B	C	E	E	E
5	B	B	C	D	E	E
6	C	B	E	C		D
7	B	C	E	E		E
8	E		E	E		
9			C	C		
10			E	E		
11			C	E		
12			C	E		
13				E		
DTW	268		354		44	

Table 5.2: Strings of characters obtained with SAX-P method on times series of axial moments applied by the three subjects on right and left rear wheels of an instrumented MWC during their second 10-m run.

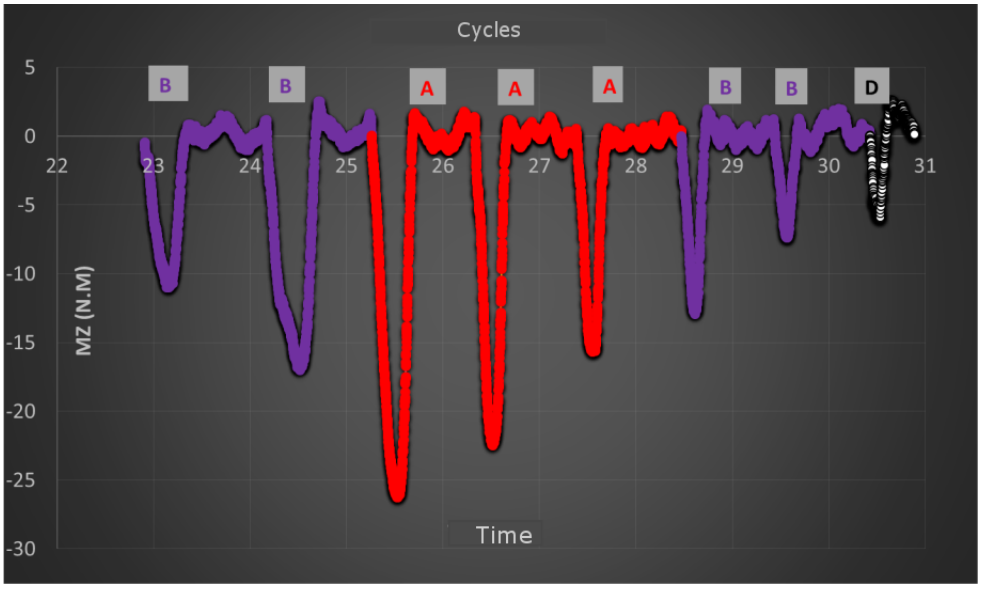


Figure 5.12: Symbolic representation of cyclic time series

observations clearly demonstrate that the three subjects did not propel their MWC symmetrically during this particular exercise. The first results of the evaluation of SAX-P on a classification task are presented on the web page [**?**]

5.4 Conclusion

In this ongoing work, we proposed a method of symbolic representation of cyclic time series called SAX-P. This method is used to represent a cyclic time series as a string, each character representing a class of the cycles of the considered time series. The character strings obtained were then compared using the Dynamic Time Warping distance. The SAX-P model has been applied to propulsive moments measured during the movements in a straight line by three subjects in MWC. The preliminary results obtained have particularly showed that these subjects had different modes of propulsion and propulsion cycles of the same subject were not symmetrical. Ongoing research is devoted on applying this new symbolic representation to a supervised classification of cyclic time-series in bio-mechanics.

Chapter 6

Application to manual wheelchair locomotion

6.1 Introduction

To improve the efficiency of wheelchair propulsion, a Wheelchair Ergometer (FRET-2) equipped with sensors has been manufactured. The sensors installed on the wheelchair measure the physical stresses applied to the Manual Wheelchair (FRM) during actual use and record them. The following paragraphs present.

- A description of the measurements recorded by the sensors during actual use;
- An analysis of the data obtained by applying the algorithms as mentioned above (in our contribution).

6.2 Description of the dataset

The sensors are located on the right and left wheels of the manual wheelchair, on the footrest, on the seat and the backrest (see Figure 6.1). These sensors measure the forces and moments of these forces applied to each of the systems mentioned above. The moment of a force concerning a given point is a vectorial physical quantity which translates the ability of a force to turn a mechanical system around that point, often called a pivot [20]. The sensors installed on the FRM were used to measure the kinematic parameters (speed, acceleration) of the movement of the Manual Wheelchair (FRM), as well as its position relative to the Earth's magnetic north.

The measurements recorded by the sensors and subjected to our analysis consist of 44 attributes; 30 of the 44 attributes relate to the measurement of the torque

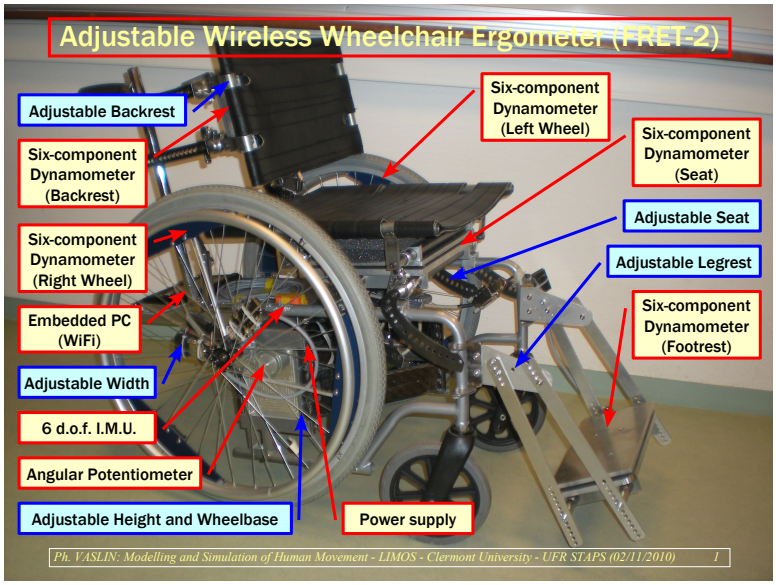


Figure 6.1: title

constituted by force applied to the systems mentioned above and the moment of this force to an axis of rotation. For each of the five systems, we have three components of the force (Fx, Fy, Fz) and the momentum (Mx, My, Mz) that apply to it. The 14 other attributes tell us about the kinematics of the manual wheelchair and its position relative to the Earth's magnetic north. The detailed description of these data is presented in Table 4.1.

To analyze the propulsion in FRM, we are interested in the moment along the axis of Z (Mz) applied to the right and left wheels of the FRM. This will allow us to identify the propulsion cycles during the move. A propulsion cycle consists of a push time interval and a consecutive freewheeling time interval that materializes on the Z-moment by a peak and drag as shown in Figure 4.2. The description of the data recorded by the sensors is presented in Appendix B.

6.2.1 Torque sensor

6.2.2 Characteristic properties of the data

The length

The cycles

The uncertainty

6.3 Analysis based on propulsion technique

6.4 Analysis based on propulsion capabilities

6.5 Propulsion technique versus propulsion capabilities

6.6 Conclusion

Appendix A

Hellinger Based Distance for Uncertain time series

Deterministic Measures like traditional similarity measures, return a real number as the distance between two uncertain time series.

A.1 DUST

[Murthy and Sarangi] is the only deterministic similarity measure defined for uncertain time series. Given two uncertain time series $X =< X_1, \ldots, X_n >$ and $Y =< Y_1, \ldots, Y_n >$, the distance between two uncertain values X_i, Y_i is defined as the distance between their true (unknown) values r(X_i), r(Y_i): $dist(X_i, Y_i) = |r(X_i) - r(Y_i)|$. This distance is used to measures the similarity of two uncertain values. $\varphi(|X_i - Y_i|)$ is the probability that the reals values at timestamp i are equal, given the observed values at that timestamp i.e.

$$\varphi(|X_i - Y_i|) = Pr(dist(0|r(X_i) - r(Y_i)| = 0). \tag{A.1}$$

This similarity function is then used inside the dust dissimilarity function:

$$dust(X_i, Y_i) = \sqrt{-log(\varphi(|X_i - Y_i|)) + log(\varphi(0))}. \tag{A.2}$$

The distance between uncertain time series $X =< X_1, \ldots, X_n >$ and $Y =< Y_1, \ldots, Y_n >$ in DUST is then defined as follows:

$$DUST(X, Y) = \sqrt{\sum_{i=1}^{n} dust(X_i, Y_i)^2}. \tag{A.3}$$

The disadvantage of DUST is that it breaks the triangle inequality for small distances. Triangular inequality is a desirable property of dissimilarity functions because it makes it possible to speed-up the comparison of time series. For example,

Figure A.1: Bhattacharyya

for density based clustering two time series A and B are considered similar if the distance between them is less than ϵ. Thus, if the sum of the distances d (A, B) and d (B, C) is less than ϵ, we deduce that the distance d (A, C) is also without calculating it. The triangular inequality is also used for the exact indexing of time series [Keogh *et al.*].

To remedy this, we introduce a new deterministic distance function based on the Hellinger distance that evaluate the dissimilarity between uncertain time series and respects triangular inequality.

A.2 Hellinger Based Distance

To evaluate the similarity between two probability distributions, we can measure the area of intersection between these two probability distributions (Figure A.1) . If the area of this intersection is zero, then the probability distributions are disjoint, if it is 1 then the probability distributions are identical. The area of this intersection can be calculated using the Bhattacharyya coefficient (B) [Patra *et al.*].

The **Hellinger** distance, based on the use of the Bhattacharyya coefficient, allows to measure the dissimilarity between two probability distributions. It is defined as follows:

Definition 17. *The Hellinger distance between two probability measures P and Q that are absolutely continuous relative to some σ-finite measure μ on a measurable space (x, β) is defined by the formula:*

$$H(P,Q) = \{2[1 - B(P,Q)]\}^{\frac{1}{2}}, \tag{A.4}$$

where

$$B(P,Q) = \int \sqrt{\frac{dP}{d\mu}} \sqrt{\frac{dQ}{d\mu}} d\mu. \tag{A.5}$$

Theorem 2. *The Hellinger distance satisfy the triangle inequality [Ibragimov and Has' minskii].*

Based on Hellinger distance we define the HBD distance (Hellinger Based Distance) which measures the dissimilarity between two uncertain time series:

Definition 18. *The distance between uncertain time series* $X =< X_1, \ldots, X_n >$ *and* $Y =< Y_1, \ldots, Y_n >$ *under Hellinger Based Distance is then defined as follows:*

$$HBD(X,Y) = \sqrt{\sum_{i=1}^{n} H(X_i, Y_i)^2}. \tag{A.6}$$

Theorem 3. *HBD distance satisfy the triangle inequality.*

Proof. Let $X =< X_1, \ldots, X_n >$, $Y =< Y_1, \ldots, Y_n >$, $Z =< Z_1, \ldots, Z_n >$ be three uncertain time series, we want to proof that

$$\sqrt{\sum_{i=1}^{n} H(X_i, Y_i)^2} + \sqrt{\sum_{i=1}^{n} H(Y_i, Z_i)^2} \geq \sqrt{\sum_{i=1}^{n} H(X_i, Z_i)^2}. \tag{A.7}$$

First, let us show that:

$$\left(\sqrt{\sum_{i=1}^{n} H(X_i, Y_i)^2}\right) \times \left(\sqrt{\sum_{i=1}^{n} H(Y_i, Z_i)^2}\right) \geq \sum_{i=1}^{n} H(X_i, Y_i) H(Y_i, Z_i) \tag{A.8}$$

By squaring the two members of the inequality (A.8) we obtain

$$\left(\sum_{i=1}^{n} H(X_i, Y_i)^2\right) \times \left(\sum_{i=1}^{n} H(Y_i, Z_i)^2\right) \geq \left(\sum_{i=1}^{n} H(X_i, Y_i) H(Y_i, Z_i)\right)^2 \tag{A.9}$$

$$i.e. \left(\sum_{i=1}^{n} H(X_i, Y_i)^2\right) \times \left(\sum_{i=1}^{n} H(Y_i, Z_i)^2\right) - \left(\sum_{i=1}^{n} H(X_i, Y_i) H(Y_i, Z_i)\right)^2 \geq 0 \tag{A.10}$$

By developing and reducing the expression(A.10), we obtain

$$i.e. \sum_{i,j \epsilon \{1,\ldots,k\} \, and \, i \neq j} \left(H(X_i, Y_i) - H(Y_j, Z_j)\right)^2 \geq 0 \tag{A.11}$$

This shows that the inequality (A.8) is true. Let us now show that HBD satisfies the triangular inequality : according to Theorem 2,

$$H(X_i, Y_i) + H(Y_i, Z_i) \geq H(X_i, Z_i) \tag{A.12}$$

By squaring the two members of the inequality, we obtain

$$H(X_i, Y_i)^2 + H(Y_i, Z_i)^2 + 2H(X_i, Y_i)H(Y_i, Z_i) \geq H(X_i, Z_i)^2. \tag{A.13}$$

$$i.e. \sum_{i=1}^{n} H(X_i, Y_i)^2 + \sum_{i=1}^{n} H(Y_i, Z_i)^2 + 2\sum_{i=1}^{n} H(X_i, Y_i)H(Y_i, Z_i) \geq \sum_{i=1}^{n} H(X_i, Z_i)^2 \tag{A.14}$$

according to inequality A.8, we obtain

$$\sum_{i=1}^{n} H(X_i, Y_i)^2 + \sum_{i=1}^{n} H(Y_i, Z_i)^2 + 2\left(\sqrt{\sum_{i=1}^{n} H(X_i, Y_i)^2}\right) \times \left(\sqrt{\sum_{i=1}^{n} H(Y_i, Z_i)^2}\right) \geq \sum_{i=1}^{n} H(X_i, Z_i)^2 \tag{A.15}$$

$$i.e. \left(\sqrt{\sum_{i=1}^{n} H(X_i, Y_i)^2} + \sqrt{\sum_{i=1}^{n} H(Y_i, Z_i)^2}\right)^2 \geq \sum_{i=1}^{n} H(X_i, Z_i)^2. \tag{A.16}$$

$$i.e. \sqrt{\sum_{i=1}^{n} H(X_i, Y_i)^2} + \sqrt{\sum_{i=1}^{n} H(Y_i, Z_i)^2} \geq \sqrt{\sum_{i=1}^{n} H(X_i, Z_i)^2}. \tag{A.17}$$

This is what had to be demonstrated □

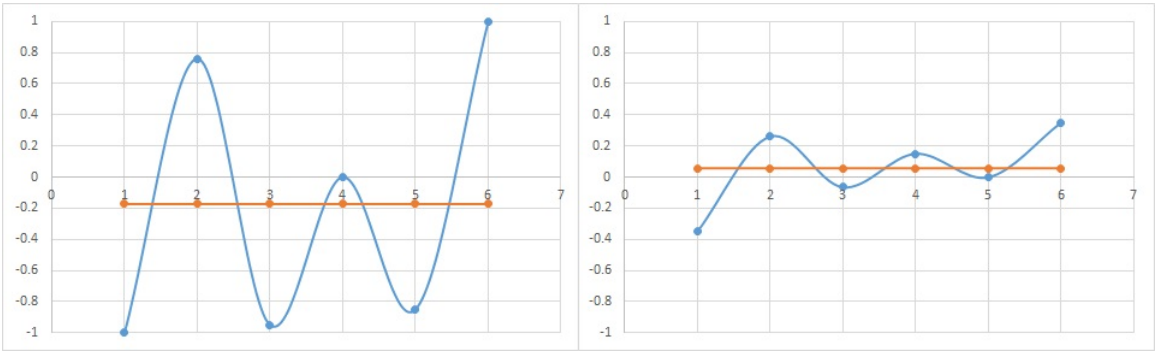
The problem with the deterministic uncertain distance distances DUST and HBD is that their expression varies as a function of the probability law that uncertainty follows. Their use therefore requires knowledge of the law of probability of the uncertainty contained in the data, which is not always possible in practice.

Appendix B

An optimal approach to time series segmentation: Application to the supervised classification

B.1 Introduction

Time series databases are often extremely large. This is particularly the case of the Large Synoptic Survey Telescope (LSST) database which records data from of telescopes [lss]. She has billions of time series (10 Petabytes). The time series recorded in these databases are sometimes very long. Another example is the SACR-FRM project that uses sensors to measure the efforts of a manual wheelchair user at a frequency of 1000 Hz [SAC]. Ten minutes recording time series of 600 000 measurements. Faced with this, several scientific works were carried out with the aim of reducing the storage space of time series and accelerating their treatment. A widely used approach is to change the representation of time series to reduce their length. This technique was introduced by Agrawal et al. [Agrawal *et al.*1993]; he uses the discrete Fourier transform to obtain a compact representation of the time series. Other methods have also been used: the decomposition in eigenvalue [Wu *et al.*1996], the discrete wavelet transform [Chan and Fu1999] and approximate aggregation by segments (PAA) [Keogh *et al.*]. This last method has shown its effectiveness compared to previous three because it is easier to understand, to implement, but also faster and allows to build indexes in linear time. PAA suggests splitting the time series into segments of the same size, then replace each segment by the average of its points. This method generates a representation compact, able to have a few segments as possible to reduce space storage and time comparison time series. However, too compact a representation distorts the time series and causes a loss of information. How then to choose the right number of segments to consider? Our work is based on a simple observation: the use of the average arithmetic is relevant



(a) (b)

Figure B.1: These figures show the average of two segments. In the first case (a) the data points of the segment are far from the average, in the second case (b) they are close to the average. Replace data points of a segment by their average introduces an error that the can be measured from the gap between the points and the average.

when the variance of the population is small as illustrated by the figure B.1.

We define here a minimal bound for the number of segments to be considered, and we propose an algorithm which allows choosing the number of segments which minimizes their mean squared error, this to reduce the length of the time series without altering the information they contain.

The rest of this chapter is organized as follows: the B.2 section presents a formal definition of our problem and an algorithm used to solve it; the section B.3 presents and comments on the results of the experiments and the section B.4 concludes the paper and presents perspectives for this work.

B.2 Granularity of time series segments

B.2.1 Notations and definitions

Definition 1: A **time series** or **time series** $X = x_0, x_1, \cdots, x_m$ is a sequence of numerical values representing the evolution of a specific quantity over time. x_m is the most recent value.

Definition 2: A segment X_i of length l of the time series X of size m ($l < m$) is a sequence consisting of l consecutive variables X beginning at the position i and ending at the position $i + l - 1$. We have: $X_i = x_i, x_{i+1}, ..., x_{i+l-1}$

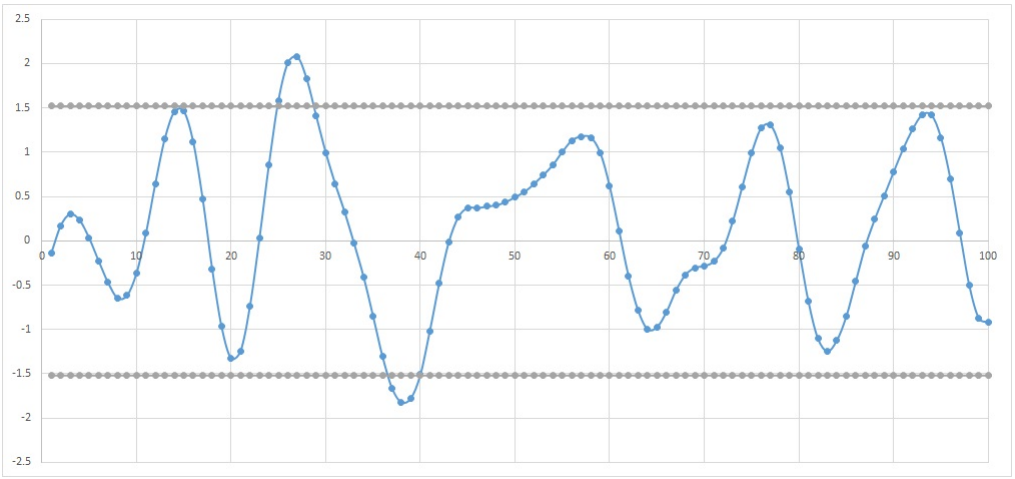


Figure B.2: This figure shows the first 100 points of the first time series of the fordA dataset available in the UCR [Chen *et al.*2015] database. Time series are normalized. The two horizontal lines delimit the interval corresponding to twice the standard deviation and minus two times the standard deviation of the points of the time series. We can observe that the points outside this range are at the ends of the time series.

Definition 3: The arithmetic mean of the data points of a segment X_i of size l is denoted $\bar{X}_i$ and is defined by

$$\bar{X}_i = \frac{1}{l} \sum_{j=0}^{l-1} x_{i+j}$$

B.2.2 information theory and minimum number of segments

To mitigate the effects of noise during time series processing, Keogh and Kasetty [Keogh and Kasetty2003] recommend that they are normalized. Normalizing the time series makes them compatible with a normal distribution [Lin *et al.*2007]. In this case, 95 % of the points in the time series are between minus two times the standard deviation (σ) and twice the standard deviation of the points, and thus 5 % of the points of the series are outside this range. These points correspond to the ends of the series as shown in the figure B.2.

B.2.3 Notations and definitions

Definition 1: A **time series** or **time series** $X = x_0, x_1, \cdots, x_m$ is a sequence of numerical values representing the evolution of a specific quantity over time. x_m is the most recent value.

Definition 2: A segment X_i of length l of the time series X of size m ($l < m$) is a sequence consisting of l consecutive variables X beginning at the position i and ending at the position $i + l - 1$. We have: $X_i = x_i, x_{i+1}, ..., x_{i+l-1}$

Definition 3: The arithmetic mean of the data points of a segment X_i of size l is denoted $\bar{X}_i$ and is defined by

$$\bar{X}_i = \frac{1}{l} \sum_{j=0}^{l-1} x_{i+j}$$

Also, information theory tells us that the amount of information relating to an event is equal to $-log_2(p)$ where p is the probability of the event [Shannon2001] . In other words, a very likely event ($p \longrightarrow 1$) brings less information than an unlikely event ($p \longrightarrow 0$). Therefore, a point outside the interval $[-2\sigma, 2\sigma]$ provides more information than a point in that range. Indeed, the probability that one point is in the range is 0.95 while the probability that one point is out of range is $0.05 = \frac{1}{20}$.

If we choose a minimum number of (α) segments less than 5 % of the length of the time series, we run the risk of aggregating the points within the interval $[-2\sigma, 2\sigma]$ and those outside this range. This will have two consequences: on the one hand, to alter the information carried by these points. On the other hand, increase the variance of the segments obtained. Because we will aggregate the points at the ends and those near the average. So we chose to consider 5% of the number of points in the time series as the minimum number of segments. This allows us to define the following functions of $\mathbb{N} \to \mathbb{N}$:

$$\alpha : n \mapsto \alpha(n) = \begin{cases} \lfloor \frac{n}{20} \rfloor \ if \ \lfloor \frac{n}{20} \rfloor \geq 2 \\ 2 \ otherwise \end{cases}$$

$$\beta : n \mapsto \beta(n) = \left\lfloor \frac{n}{2} \right\rfloor$$

β gives the maximum number of segments. Indeed, a segment is made up of at least 2 points, so there is at most $\lfloor \frac{n}{2} \rfloor$ segments. The number of segments W that we will consider is greater than or equal to $\alpha(|X|)$ and less than or equal to $\beta(|X|)$. The next subsection explains how we choose the value of W.

B.2.4 Minimize the squared error to choose the number of segments

After dividing into segments, we replace each segment by the average of the points that constitute it. The variance between the points of each segment can be measured from the mean squared error. Our problem is therefore the following:

Let $X = x_0, x_1, \cdots, x_n$ a time series of size n, look for $W \in \mathbb{N}$ such that $\alpha(n) \leq W \leq \beta(n)$ and $\frac{1}{n}\sum_{i=1}^{W}\sum_{j=(i-1)k}^{ik}(\bar{X}_i - X_j)^2$ is minimal. Where $\bar{X}_i$ is the arithmetic mean of a segment of length k.

To solve this problem, we propose an algorithm that proceeds as follows:

1. For each value of W, with $\alpha(n) \leq W \leq \beta(n)$
   (a) Calculate the squared error of each segment $X_i = x_i, x_{i+1}, ..., x_{i+l-1}$;
   (b) Calculate the mean of the quadratic errors;
2. The value of W returned is the one that produces an average squared error minimum;

Algorithm 3 details the previous principle.

Algorithm 3: optimalNumberOfSegment

Input: length_min, length_max : repectively the minimal and the maximal length of a segment.
v : a time series
Output: The optimal number of segment to be use with Piecewise Aggregate approXimation

```
1 function optimalNumberOfSegment(length_min, length_max, v)
2     len_v ← length(v)
3     n ← length_max − length_min + 1
4     forall i ∈ {length_min, ..., length_max} do
5         x[j, 1] ← i
6         z[j, 1] ← (1/len_v) ∗ sum_SSE(v, i)
7         computation of the error j ← j + 1
8     ind_min ← indice_minimun(z)
9     return floor(len_v/x[ind_min, 1])
```

Complexity of the algorithm The calculation of the squared error of a segment is done in $O(\lfloor \frac{n}{W} \rfloor)$.

The time complexity of calculating the mean squared error for segment splitting is $O(n)$.

The number of segments varies from $\lfloor \frac{n}{20} \rfloor$, $\lfloor \frac{n}{19} \rfloor$ $\ldots$ $\lfloor \frac{n}{2} \rfloor$. There are 19 possible divisions in segments. The time complexity of calculating the value of W which minimizes the error mean squared is $19 \times O(n) = O(n)$.

To exploit the compact representations of the time series, we must be able to compare them. The next subsection presents how to compare compact time series that we get.

Algorithm 4: sum_SSE

Input: v : a time series.
nbPoints : the length of a segment
Output: The sum of squares error associated with a segment length

```
1  function sum_SSE(v, nbPoints)
2      n ← length(v)
3      ind_debut ← 1
4      aux_se ← c()
5      tab_indices_debut ← c()
6      i ← 0
7      while (ind_debut + nbPoints) ≤ n do
8          tab_indices_debut[i] ← ind_debut
9          ind_debut ← ind_debut + nbPoints
10         i ← i + 1
11     m ← length(tab_indices_debut)
12     forall i ∈ {1, ..., m} do
13         aux_se[i] ← SSE_segment(v, nbPoints, tab_indices_debut[i])
14     return sum(aux_se)
```

B.2.5 Dynamic Time Warping Algorithm and Comparison of Representations compact

The dynamic time warping algorithm [Keogh and Ratanamahatana] allows to carry out a non-linear correspondence between two time series by minimizing the distance between the two. It proceeds as follows: Be two time series

$$X = x_1, x_2, \cdots, x_n;$$

$$Y = y_1, y_2, \cdots, y_m.$$

To align them, the algorithm constructs a matrix $n \times m$ where the element (i, j) of the matrix corresponds to the square distance $(x_i - y_j)^2$ which is the alignment between x_i and y_j. To find the best alignment between the two time series, we build the path in the matrix that minimizes the sum of the square distances. This path is calculated by dynamic programming from the following recurrence:

$$\gamma(i, j) = d(x_i, y_j) + min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\}$$

where $d(x_i, y_j)$ is the square of the distance contained in the cell (i, j) and $\gamma(i, j)$ is the cumulative distance at the position (i, j) which is calculated by the sum of the square of the distance to the position (i, j) and the minimum cumulative distance of its three adjacent cells.

Approximate aggregation by segment provides distance-based distance measurement Euclidean to compare compact representations. However, we chose to use the dynamic time warping algorithm. For the following reasons:

1. The dynamic time warping algorithm is known to have the best performance for sequence alignment in several areas: in robotics, biometrics, music, climatology, aviation, in gesture recognition, cryptanalysis, astronomy, exploration space [Rakthanmanon *et al.*2012].

2. piecewise aggregate approximation of the time series leads to temporal deformation. Indeed, with two time series of size n, we can apply our algorithm to the first time series, reduce it to N_1 segments and reduce the second to N_2 segments with $N_1 < N_2$.

B.3 Results and Discussion

First, we present the datasets used during the experiment. Then we evaluate the performance of the proposed method from the reduction of the length of time series and classification errors.

B.3.1 Datasets

We performed tests on 85 datasets that come from the UCR database [Chen *et al.*2015] . Detailed information on the datasets is presented in the table B.1. In the UCR database, each data set is divided into a learning set and a test set. Datasets contain between 2 and 60 classes and the time series of these datasets have lengths that range from 24 to 2709 points. The table B.1 presents a detailed description of the datasets. The following paragraph presents the assessment of the performance of our algorithm on these datasets.

N	**Name**	**Nb. of classes**	**Size of training set**	**Size of testing set**
1	50Words	50	450	455
2	Adiac	37	390	391
3	ArrowHead	3	36	175
4	Beef	5	30	30
5	BeetleFly	2	20	20
6	BirdChicken	2	20	20
7	Car	4	60	60
8	CBF	3	30	900
9	ChlorineConcentration	3	467	3840
10	CinC_ECG_torso	4	40	1380
Continue to the next page				

N	Name	Nb. of classes	Size of training set	Size of testing set
Following ...				
11	Coffee	2	28	28
12	Computers	2	250	250
13	Cricket_X	12	390	390
14	Cricket_Y	12	390	390
15	Cricket_Z	12	390	390
16	DiatomSizeReduction	4	16	306
17	DistalPhalanxOutlineAgeGroup	3	139	400
18	DistalPhalanxOutlineCorrect	2	276	600
19	DistalPhalanxTW	6	139	400
20	Earthquakes	2	139	322
21	ECG	2	100	100
22	ECG5000	5	500	4500
23	ECGFiveDays	2	23	861
24	ElectricDevices	7	8926	7711
25	Face (all)	14	560 1	690
26	Face (four)	4	24	88
27	FacesUCR	14	200	2050
28	Fish	7	175	175
29	FordA	2	1320	3601
30	FordB	2	810	3636
31	Gun-Point	2	50	150
32	Ham	2	109	105
33	HandOutlines	2	370	1000
34	Haptics	5	155	308
35	Herring	2	64	64
36	InlineSkate	7	100	550
37	InsectWingbeatSound	11	220	1980
38	ItalyPowerDemand	2	67	1029
39	LargeKitchenAppliances	3	375	375
40	Lightning-2	2	60	61
41	Lightning-7	7	70	73
42	MALLAT	8	55	2345
43	Meat	3	60	60
44	MedicalImages	10	381	760
45	MiddlePhalanxOutlineAgeGroup	3	154	400
...	...	...	...	...

Table B.1: 85 UCR datasets used for experimental validation. The full list is available here [Chen *et al.*2015]

End

B.3.2 Comparison of algorithm performance

The tables **??** and B.2 present the comparison of the classification error of 1-Nearest Neighbor (1-NN) algorithms associated with Euclidean distance (4), 1-NN, associated with the dynamic time warping algorithm using a constraint (a deformation window) (5), 1-NN associated with the algorithm of unconstrained dynamic time warping (DTW) applied to the raw data (6) and the 1-NN algorithm associated with DTW applied to the data pre-processed by our algorithm (7). The (4) algorithm gives the best results that are to say ($(4) \leq (5)$ *and* $(4) \leq (6)$ *and* $(4) \leq (7)$) on 20 datasets, the algorithm (5) is the best on 47 datasets, the (6) algorithm is the best on 21 datasets, the (7) algorithm is the best on 21 datasets. Although no of these algorithms have the best performance on all datasets, the algorithm (5) averaged the smallest misclassification **0.237** and the most expensive (4) algorithm large average error **0,288**. The (6) and (7) algorithms have relatively close average error rates equal to **0,256** and **0,258** respectively.

To evaluate the effects of the **change of representation** of the time series on their **classification**, we compare the length of the time series and the errors of classification presented by the columns (6) and (7) of the tables **??** and B.2. Indeed, these two columns use the same 1-NN classification algorithm and the same function distance DTW. The only difference between these columns is the nature of the data; the (6) column uses the raw data and the column (7) the compacted data with the method described above.

- Regarding the length of the time series; the (6) algorithm uses all the points of the time series. On the other hand, the (7) algorithm uses compact representations whose length varies between **15** % and **34** % of the initial length of the time series. On average, the compact representations have a length equal to **20** % of the initial time series

- For classification errors, the error (7) > (6) on 40 datasets, the error of (7) = (6) on 3 datasets and the error of (7) < (6) on 42 datasets.

These results are encouraging because despite the reduction in the length of the time series errors, the classification error with the compact representation is less than or equal to that of the raw data classification for 45 datasets out of the 85 available in the UCR base. These results are summarized in Figure B.3. One of the reasons for this observed improvement over 42 datasets is as follows: the dynamic time warping algorithm is sensitive to noise, therefore by aggregating the points of the segments, we reduce the effects of noise.

(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**	**(7)**
1	54	0,20	0,369	0,242 (6)	0,31	***0,279***
Continue to the next page						

(1)	(2)	(3)	(4)	(5)	(6)	(7)
Following ...						
2	35	0,20	*0,389*	0,391 (3)	**0,396**	0,425
3	50	0,20	*0,2*	*0,200 (0)*	0,297	**0,246**
4	78	0,17	*0,333*	*0,333 (0)*	**0,367**	0,433
5	85	0,17	*0,25*	0,300 (7)	0,3	**0,300**
6	85	0,17	0,45	0,300(6)	*0,25*	***0,250***
7	96	0,17	0,267	0,233 (1)	0,267	***0,217***
8	32	0,25	0,148	0,004 (11)	0,003	***0,002***
9	33	0,20	*0,35*	*0,35 (0)*	**0,352**	0,414
10	234	0,14	0,103	*0,07 (1)*	0,349	**0,285**
11	57	0,20	*0*	*0,000 (0)*	***0***	0,036
12	120	0,17	0,424	0,380 (13)	***0,3***	0,416
13	60	0,20	0,423	0,228 (10)	0,246	***0,241***
14	60	0,20	0,433	*0,238 (17)*	**0,256**	0,277
15	60	0,20	0,413	0,254 (5)	0,246	***0,244***
16	69	0,20	0,065	0,065 (0)	***0,033***	0,072
17	20	0,25	0,218	0,228 (1)	0,208	***0,198***
18	20	0,25	0,248	*0,232 (2)*	***0,232***	0,255
19	20	0,25	0,273	*0,272 (0)*	**0,29**	0,310
20	85	0,17	0,326	*0,258 (22)*	***0,258***	0,276
21	24	0,25	*0,12*	*0,120 (0)*	0,23	**0,180**
Continue to the next page						

(1)	(2)	(3)	(4)	(5)	(6)	(7)
Following ...						
22	35	0,25	0,075	0,075 (1)	0,076	***0,072***
23	34	0,25	*0,203*	*0,203 (0)*	**0,232**	0,259
24	24	0,25	0,45	*0,376 (14)*	**0,399**	0,438
25	32	0,24	0,286	*0,192 (3)*	***0,192***	0,253
26	70	0,20	0,216	*0,114 (2)*	0,17	**0,170**
27	32	0,24	0,231	*0,088 (12)*	**0,095**	0,177
28	77	0,17	0,217	*0,154(4)*	**0,177**	0,263
29	83	0,17	*0,341*	*0,341 (0)*	0,438	**0,359**
30	83	0,17	0,442	0,414 (1)	0,406	***0,360***
31	30	0,20	0,087	0,087 (0)	0,093	***0,047***
32	71	0,16	*0,4*	*0,400 (0)*	0,533	**0,419**
33	387	0,14	0,199	*0,197 (1)*	**0,202**	0,206
34	182	0,17	0,63	*0,588 (2)*	0,623	**0,623**
35	85	0,17	0,484	*0,469 (5)*	***0,469***	0,500
36	268	0,14	0,658	*0,613 (14)*	0,616	**0,615**
37	51	0,20	0,438	*0,422 (2)*	0,645	**0,611**
38	8	0,33	*0,045*	*0,045 (0)*	0,05	**0,048**
39	120	0,17	0,507	0,205 (94)	0,205	***0,203***
40	106	0,17	0,246	*0,131 (6)*	***0,131***	0,164
Continue to the next page						

(1)	(2)	(3)	(4)	(5)	(6)	(7)
Following ...						
41	63	0,20	0,425	0,288 (5)	0,274	***0,219***
42	170	0,17	0,086	0,086 (0)	***0,066***	0,097
43	74	0,17	*0,067*	*0,067*	*(0) 0,067*	***0,067***
44	24	0,24	0,316	*0,253 (20)*	**0,263**	0,288
45	20	0,25	0,26	0,253 (5)	***0,25***	0,268
46	20	0,25	*0,247*	0,318 (1)	0,352	**0,268**
47	20	0,25	0,439	0,419 (2)	***0,416***	0,419
48	21	0,25	*0,121*	0,134 (1)	0,165	**0,133**
49	125	0,17	*0,171*	0,185 (1)	**0,209**	0,222
50	125	0,17	*0,12*	0,129 (1)	**0,135**	0,146
51	95	0,17	*0,133*	*0,133 (0)*	0,167	**0,167**
52	71	0,17	0,479	0,388 (7)	0,409	***0,355***
53	20	0,25	*0,239*	*0,239 (0)*	**0,272**	0,273
54	170	0,17	0,891	0,773 (14)	***0,772***	0,809
55	36	0,25	0,038	*0,000 (6)*	*0*	***0,000***
56	20	0,25	0,215	0,215 (0)	***0,195***	0,249
57	20	0,25	*0,192*	0,210 (1)	**0,216**	0,251
58	20	0,25	0,292	*0,263 (6)*	***0,263***	0,280
59	120	0,17	0,605	0,560 (8)	0,536	***0,501***
Continue to the next page						

(1)	(2)	(3)	(4)	(5)	(6)	(7)
Following ...						
60	120	0,17	0,64	*0,589 (17)*	**0,603**	0,645
61	83	0,17	0,461	*0,300 (3)*	0,35	**0,339**
62	85	0,17	0,248	0,198 (4)	0,232	***0,210***
63	120	0,17	0,659	*0,328 (15)*	0,357	**0,349**
64	17	0,24	0,305	0,305 (0)	0,275	***0,250***
65	16	0,25	*0,141*	*0,141 (0)*	**0,169**	0,189
66	170	0,17	0,151	0,095 (16)	***0,093***	0,124
67	47	0,20	0,062	0,062 (0)	0,06	***0,055***
68	32	0,25	0,211	*0,154 (2)*	0,208	**0,184**
69	79	0,20	0,1	0,062 (8)	0,05	***0,048***
70	15	0,25	0,12	0,017 (6)	***0,007***	0,017
71	55	0,20	0,32	0,250 (8)	0,228	***0,193***
72	68	0,20	0,192	*0,092 (5)*	0,162	**0,154**
73	55	0,20	0,24	0,010 (3)	***0***	0,070
74	32	0,25	0,09	0,002 (4)	*0*	***0,000***
75	20	0,24	0,253	0,132 (5)	***0,096***	0,283
76	63	0,20	0,261	0,227 (4)	0,273	***0,252***
77	63	0,20	0,338	*0,301 (4)*	0,366	**0,346**
78	63	0,20	0,35	*0,322 (6)*	0,342	**0,334**
Continue to the next page						

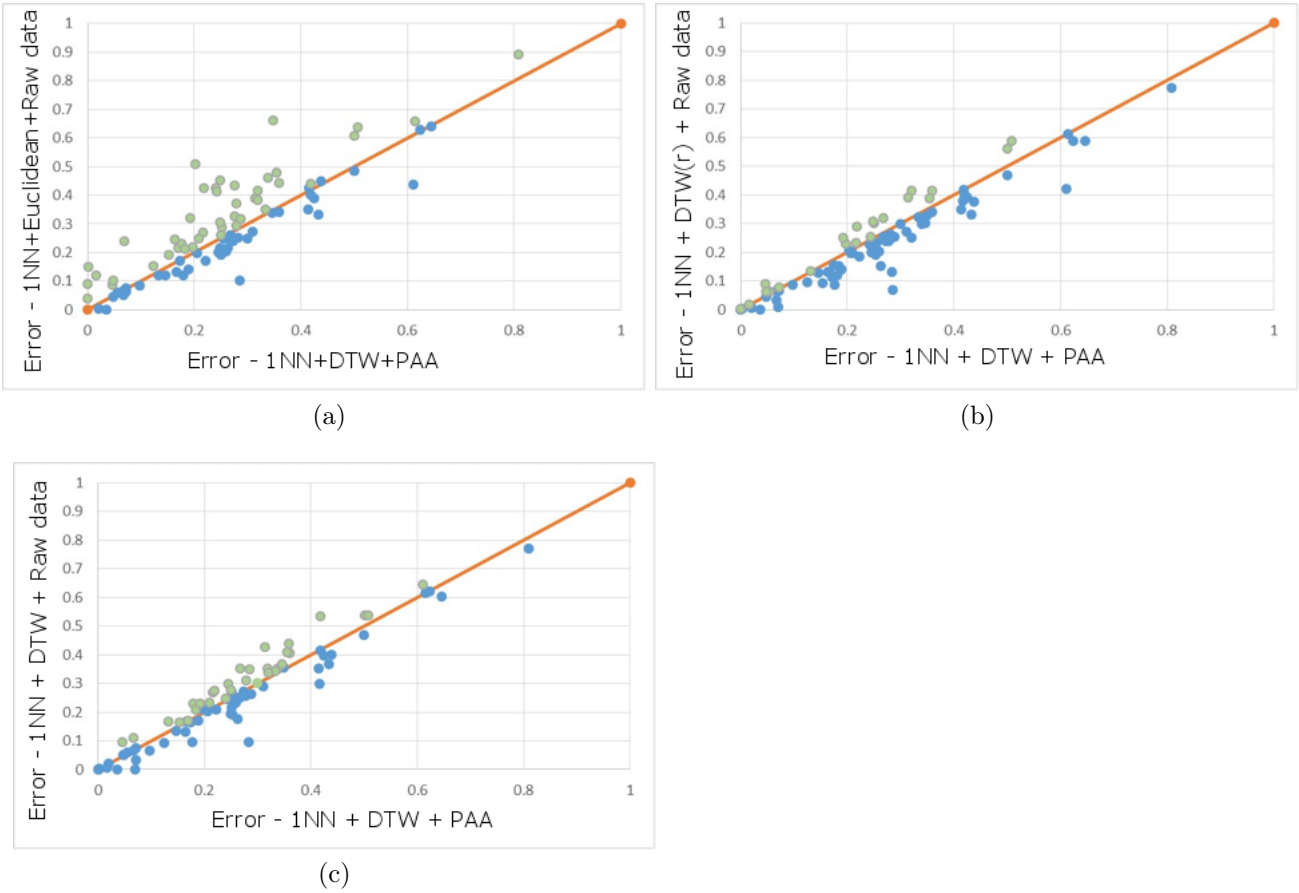
(1)	(2)	(3)	(4)	(5)	(6)	(7)
Following ...						
79	157	0,17	0,052	*0,034 (4)*	0,108	**0,067**
80	30	0,20	0,005	0,005 (1)	***0,02***	0,021
81	46	0,20	0,389	0,389 (0)	0,426	***0,315***
82	54	0,20	0,382	*0,252 (8)*	0,351	**0,320**
83	150	0,17	0,635	0,586 (3)	0,536	***0,508***
84	150	0,17	0,414	0,414 (9)	0,337	***0,320***
85	71	0,17	0,17	*0,155 (2)*	**0,164**	0,174
$\bar{X}$			0,288	0,237	0,256	0,258
σ			0,175	0,161	0,166	0,160

Table B.2: The **(1)** column presents **numbers** of the datasets. The column **(2)** the **reduced length** of the time series. The column **(3)** is the **ratio** of the length of the reduced time series over the length of the initial time series. The **(4)** column designates the **1-Nearest Neighbor** algorithm, associated to the **Euclidean distance**. The **(5)** column designates the algorithm of **1- Nearer Neighbor**, associated with the algorithm of **dynamic dynamic temporal deformation** using a **constraint** called deformation window that allows to stop the comparison of time series when one perceives that they are very different. The **(6)** column represents **1-Nearest Neighbor** algorithm associated to the **unconstrained dynamic time warping** applied to the **raw data**. The **(7)** column represents the **algorithm. 1-Nearest Neighbor** associated with the **dynamic time warping algorithm without constraints**, applied on the **compact representations** produced by our algorithm. We compare firstly, the classification error of the algorithms (6) and (7) the smallest error is in **bold**. We then compare the classification errors of algorithms (4), (5), (6) and (7) the smallest error is put **italicized**.

End

B.4 Conclusion

The purpose of this article was to propose an algorithm for choosing the number of segments to consider for the compact representation of a time series. For this, we have defined a minimum bound for the number of segments to be considered which is equal to 5 % of the length of the time series. We have proposed an algorithm that



(a) (b)

(c)

Figure B.3: Two-to-two comparison of the classification errors of the algorithm 1-Nearest Neighbor (1-NN) using Euclidean distance with 1-NN using two variations of the temporal warping algorithm on data raw and compact data

chooses the number W of segments minimizing the mean squared error. Results of experiments conducted on 42 datasets have shown that the number of segments chosen allows two improvements

- significantly reduce the length of the series temporal; time series of reduced size has a length which varies between 15% and 34% of the initial time series length
- improves supervised classification results on a set of 85 datasets used in the literature.

As a perspective for this work, we plan to vary the number of segments W from 2 to $\frac{n}{2}$ to see if our value of W is optimum for a task classification. We also plan to compare the results of this compact representation to those of other representations of literature. We also plan to parallelize our algorithm to calculate the right number of segments in linear time (almost trivial). This work allows reducing the storage space and the processing time of the time series. It also allows choosing the number of segments to consider when designing representations symbolic of time series. Indeed, several symbolic representations of series of the literature (SAX [Lin *et al.*2003], ESAX [Lkhagva *et al.*], 1d-SAX [Malinowski *et al.*2013], SAX-TD [Sun *et al.*2014], SAX-P [Siyou Fotso *et al.*2015]) use the division into segments recommended by PAA.

Bibliography

[Aach and Church2001] J. Aach and G. M. Church. Aligning gene expression time series with time warping algorithms. *Bioinformatics*, 17(6):495–508, June 2001.

[Abonyi *et al.*2003] Janos Abonyi, Balazs Feil, Sandor Nemeth, and Peter Arva. Fuzzy clustering based segmentation of time-series. *Advances in Intelligent Data Analysis V*, pages 275–285, 2003.

[Aggarwal] Charu C Aggarwal. On unifying privacy and uncertain data models. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 386–395. IEEE.

[Agrawal *et al.*1993] Rakesh Agrawal, Christos Faloutsos, and Arun N. Swami. Efficient Similarity Search In Sequence Databases. In *FODO '93 Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*, pages 69–84. Springer-Verlag, oct 1993.

[Aßfalg *et al.*] Johannes Aßfalg, Hans-Peter Kriegel, Peer Kröger, and Matthias Renz. Probabilistic similarity search for uncertain time series. In *SSDBM*, pages 435–443. Springer.

[Bagnall *et al.*a] Anthony Bagnall, Eamonn Keogh, Jason Lines, Aaron Bostrom, and James Large. Time Series Classification Website. `http://timeseriesclassification.com`.

[Bagnall *et al.*b] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. pages 1–55.

[Batista *et al.*] Gustavo EAPA Batista, Eamonn J Keogh, Oben Moses Tataw, and Vinicius MA De Souza. Cid: an efficient complexity-invariant distance for time series. 28(3):634–669.

[Begum and Keogh2014] Nurjahan Begum and Eamonn Keogh. Rare time series motif discovery from unbounded streams. *Proceedings of the VLDB Endowment*, 8(2):149–160, October 2014.

[Bhatia and Bhattacharyya] Rajendra Bhatia and Tirthankar Bhattacharyya. A generalization of the Hoffman-Wielandt theorem. 179:11–17.

[Camerra *et al.*] Alessandro Camerra, Themis Palpanas, Jin Shieh, and Eamonn Keogh. isax 2.0: Indexing and mining one billion time series. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 58–67. IEEE.

[Candan *et al.*] K Selçuk Candan, Rosaria Rossini, Xiaolan Wang, and Maria Luisa Sapino. sdtw: computing dtw distances using locally relevant constraints based on salient feature alignments. 5(11):1519–1530.

[Chan and Fu1999] Kin-Pong Chan and Ada Wai-Chee Fu. Efficient time series matching by wavelets. In *Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337)*, pages 126–133. IEEE, 1999.

[Chen *et al.*2015] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. The ucr time series classification archive. `http://www.cs.ucr.edu/~eamonn/time_series_data/`, July 2015. `www.cs.ucr.edu/~eamonn/time_series_data/`.

[Cheng *et al.*] Reynold Cheng, Dmitri V Kalashnikov, and Sunil Prabhakar. Evaluating probabilistic queries over imprecise data. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 551–562. ACM.

[Chiu *et al.*] Bill Chiu, Eamonn Keogh, and Stefano Lonardi. Probabilistic discovery of time series motifs. In *ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 493–498. ACM.

[Chu *et al.*] Selina Chu, Eamonn J Keogh, David M Hart, Michael J Pazzani, et al. Iterative deepening dynamic time warping for time series. In *SDM*, pages 195–212. SIAM.

[Cuřín *et al.*] Jan Cuřín, Pascal Fleury, Jan Kleindienst, and Robert Kessl. Meeting state recognition from visual and aural labels. In *International Workshop on Machine Learning for Multimodal Interaction*, pages 24–25. Springer.

[Dallachiesa *et al.*] Michele Dallachiesa, Besmira Nushi, Katsiaryna Mirylenka, and Themis Palpanas. Uncertain time-series similarity: return to the basics. 5(11):1662–1673.

[Dean and Ghemawat2010] Jeffrey Dean and Sanjay Ghemawat. MapReduce. *Communications of the ACM*, 53(1):72, jan 2010.

[Dietterich2002] Thomas G. Dietterich. Machine Learning for Sequential Data: A Review. *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 16–30, August 2002.

[Ding *et al.*2008] Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. Querying and mining of time series data. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, August 2008.

[Esling and Agon2012] Philippe Esling and Carlos Agon. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):1–34, 2012.

[Feo and Resende1995] Thomas A Feo and Mauricio GC Resende. Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2):109–133, 1995.

[Gao *et al.*] Yifeng Gao, Jessica Lin, and Huzefa Rangwala. Iterative grammar-based framework for discovering variable-length time series motifs. In *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on*, pages 7–12. IEEE.

[Ghalwash *et al.*] Mohamed F Ghalwash, Vladan Radosavljevic, and Zoran Obradovic. Utilizing temporal patterns for estimating uncertainty in interpretable early decision making. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 402–411. ACM.

[Godsill] Simon J Godsill. On the relationship between Markov chain Monte Carlo methods for model uncertainty. 10(2):230–248.

[Hamilton] Jd Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. 57(2):357–384.

[Huang and Kinsner] B Huang and W Kinsner. ECG frame classification using dynamic time warping. 2:1105–1110.

[Hwang *et al.*] Jun Hwang, Yusuke Kozawa, Toshiyuki Amagasa, and Hiroyuki Kitagawa. GPU Acceleration of Similarity Search for Uncertain Time Series. In *2014 17th International Conference on Network-Based Information Systems*, pages 627–632. IEEE.

[Ibarra and Kim] Oscar H Ibarra and Chul E Kim. Fast approximation algorithms for the knapsack and sum of subset problems. 22(4):463–468.

[Ibragimov and Has' minskii] Ildar Abdulovič Ibragimov and Rafail Z Has' minskii. *Statistical estimation: asymptotic theory*, volume 16. Springer Science & Business Media.

[Itakura] Fumitada Itakura. Minimum prediction residual principle applied to speech recognition. 23(1):67–72.

[Jeong *et al.*] Young-Seon Jeong, Myong K Jeong, and Olufemi A Omitaomu. Weighted dynamic time warping for time series classification. 44(9):2231–2240.

[Kate] Rohit J. Kate. Using dynamic time warping distances as features for improved time series classification. 30(2):283–312.

[Keogh and Kasetty2003] Eamonn Keogh and Shruti Kasetty. On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. *Data Mining and Knowledge Discovery*, 7(4):349–371, 2003.

[Keogh and Pazzania] E J Keogh and M J Pazzani. Derivative dynamic time warping. pages 1–11.

[Keogh and Pazzanib] Eamonn J Keogh and Michael J Pazzani. Scaling up dynamic time warping for datamining applications. In *sixth ACM SIGKDD*, pages 285–289. ACM.

[Keogh and Ratanamahatana] Eamonn Keogh and Chotirat Ann Ratanamahatana. Exact indexing of dynamic time warping. 7(3):358–386.

[Keogh and Ratanamahatana2004] E Keogh and A Ratanamahatana. Everything you know about dynamic time warping is wrong. In *3rd Workshop on Mining Temporal and Sequential Data, in conjunction with 10th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD-2004), Seattle, WA*, 2004.

[Keogh *et al.*] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. Locally adaptive dimensionality reduction for indexing large time series databases. 30(2):151–162.

[Keogh *et al.*2001] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. *Knowledge and Information Systems*, 3(January):263–286, 2001.

[Kuczera and Parent] George Kuczera and Eric Parent. Monte Carlo assessment of parameter uncertainty in conceptual catchment models: the metropolis algorithm. 211(1):69–85.

[Lin *et al.*2003] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 2–11, New York, New York, USA, 2003. ACM, ACM Press.

[Lin *et al.*2007] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144, apr 2007.

[Lkhagva and Kawagoe2006] B. Lkhagva and K. Kawagoe. New Time Series Data Representation ESAX for Financial Applications. In *22nd International Conference on Data Engineering Workshops (ICDEW'06)*, pages x115–x115. IEEE, April 2006.

[Lkhagva *et al.*] Battuguldur Lkhagva, Yu Suzuki, and Kyoji Kawagoe. Extended sax: Extension of symbolic aggregate approximation for financial time series data representation. 7.

[Longin *et al.*] JL Longin, M Vasilis, W Qiang, L Rolf, AR Chotirat, and EJ Keogh. Elastic partial matching of time series. In *9th European Conference On Principles And Practice Of Knowledge Discovery In Databases, Porto, Portugal.*

[lss] LSST time series catalog - XLDB Use Cases.

[Malinowski *et al.*2013] Simon Malinowski, Thomas Guyet, René Quiniou, and Romain Tavenard. *1d-SAX: A Novel Symbolic Representation for Time Series*, volume 8207 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[Marszalek and Burczynski] A. Marszalek and T. Burczynski. Modeling and forecasting financial time series with ordered fuzzy candlesticks. 273:144–155.

[Murthy and Sarangi] Karin Murthy and Smruti Ranjan Sarangi. Generalized notion of similarities between uncertain time series. US Patent 8,407,221.

[Myers *et al.*] Cory Myers, Lawrence R. Rabiner, and Aaron E. Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. 28(6):623–635.

[Orang and Shiria] Mahsa Orang and Nematollaah Shiri. An experimental evaluation of similarity measures for uncertain time series. In *Proceedings of the 18th International Database Engineering & Applications Symposium on - IDEAS '14*, pages 261–264. ACM Press.

[Orang and Shirib] Mahsa Orang and Nematollaah Shiri. Correlation analysis techniques for uncertain time series. 50(1):79–116.

[Orang and Shiric] Mahsa Orang and Nematollaah Shiri. Improving performance of similarity measures for uncertain time series using preprocessing techniques. In *Proceedings of the 27th International Conference on Scientific and Statistical Database Management - SSDBM '15*, pages 1–12. ACM Press.

[Papadimitriou *et al.*] Spiros Papadimitriou, Feifei Li, George Kollios, and Philip S Yu. Time series compressibility and privacy. In *Proceedings of the 33rd international conference on Very large data bases*, pages 459–470. VLDB Endowment.

[Papapetrou *et al.*2011] Panagiotis Papapetrou, Vassilis Athitsos, Michalis Potamias, George Kollios, and Dimitrios Gunopulos. Embedding-based subsequence matching in time-series databases. *ACM Transactions on Database Systems*, 36(3):1–39, August 2011.

[Patra *et al.*] Bidyut Kr Patra, Raimo Launonen, Ville Ollikainen, and Sukumar Nandi. A new similarity measure using bhattacharyya coefficient for collaborative filtering in sparse data. 82:163–177.

[Petitjean *et al.*2011] François Petitjean, Alain Ketterlin, and Pierre Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693, March 2011.

[Petitjean *et al.*2014] François Francois Petitjean, Germain Forestier, Geoffrey I. Webb, Ann E. Nicholson, Yanping Chen, and Eamonn Keogh. Dynamic Time Warping Averaging of Time Series Allows Faster and More Accurate Classification. In *2014 IEEE International Conference on Data Mining*, pages 470–479. IEEE, December 2014.

[Rakthanmanon *et al.*2012] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 262–270, 2012.

[Rand] William M Rand. Objective criteria for the evaluation of clustering methods. 66(336):846–850.

[Ratanamahatana and Keogh] Chotirat Ann Ratanamahatana and Eamonn Keogh. *Making Time-series Classification More Accurate Using Learned Constraints*, pages 11–22.

[Rehfeld and Kurths] K. Rehfeld and J. Kurths. Similarity estimators for irregular and age-uncertain time series. 10(1):107–122.

[Rizvandi *et al.*] Nikzad Babaii Rizvandi, Javid Taheri, Reza Moraveji, and Albert Y. Zomaya. A study on using uncertain time series matching algorithms for MapReduce applications. 25(12):1699–1718.

[SAC] Projet SACR-FRM (Approches de la Sociologie, de la Biomécanique et de l'Intelligence Artificielle...) | ANR - Agence Nationale de la Recherche.

[Sakoe and Chiba1978] H Sakoe and S Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, February 1978.

[Senin *et al.*] Pavel Senin, Jessica Lin, Xing Wang, Tim Oates, Sunil Gandhi, Arnold P Boedihardjo, Crystal Chen, Susan Frankenstein, and Manfred Lerner. Grammarviz 2.0: a tool for grammar-based pattern discovery in time series. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 468–472. Springer.

[Shannon2001] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3, jan 2001.

[Siyou Fotso *et al.*] Vanel Steve Siyou Fotso, Engelbert Mephu Nguifo, and Philippe Vaslin. Comparison of classification algorithms to fdtw. `http://fc.isima.fr/~siyou/fdtw`.

[Siyou Fotso *et al.*2015] Vanel Steve Siyou Fotso, Engelbert Mephu Nguifo, and Philippe Vaslin. Symbolic representation of cyclic time series: application to biomechanics. In *ICML'15 Workshop on Constructive Machine Learning*, July 2015.

[Sun *et al.*2014] Youqiang Sun, Jiuyong Li, Jixue Liu, Bingyu Sun, and Christopher Chow. An improvement of symbolic aggregate approximation distance measure for time series. *Neurocomputing*, 138:189–198, August 2014.

[Ulanova *et al.*] Liudmila Ulanova, Nurjahan Begum, and Eamonn Keogh. Scalable clustering of time series with u-shapelets. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 900–908. SIAM.

[Vegter *et al.*2014] Riemer J K Vegter, Claudine J Lamoth, Sonja de Groot, Dirkjan H E J Veeger, and Lucas H V van der Woude. Inter-individual differences in the initial 80 minutes of motor learning of handrim wheelchair propulsion. *PloS one*, 9(2):e89729, January 2014.

[Wang *et al.*a] Wei Wang, Guohua Liu, and Dingjia Liu. Chebyshev Similarity Match between Uncertain Time Series. 2015:1–13.

[Wang *et al.*b] Xiaoyue Wang, Abdullah Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn Keogh. Experimental comparison of representation methods and distance measures for time series data. 26(2):275–309.

[Wu *et al.*1996] Daniel Wu, Ambuj Singh, Divyakant Agrawal, Amr El Abbadi, and Terence R. Smith. Efficient retrieval for browsing large image databases. In *Proceedings of the fifth international conference on Information and knowledge*

management - CIKM '96, pages 11–18, New York, New York, USA, nov 1996. ACM Press.

[Yeh *et al.*] Mi-Yen Yeh, Kun-Lung Wu, Philip S Yu, and Ming-Syan Chen. Proud: a probabilistic approach to processing similarity queries over uncertain data streams. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, pages 684–695. ACM.

[Yu *et al.*] Daren Yu, Xiao Yu, Qinghua Hu, Jinfu Liu, and Anqi Wu. Dynamic time warping constraint learning for large margin nearest neighbor classification. 181(13):2787–2796.

[Zakaria *et al.*] Jesin Zakaria, Abdullah Mueen, and Eamonn Keogh. Clustering time series using unsupervised-shapelets. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 785–794. IEEE.

[Zhang *et al.*a] XX Zhang, Q Liu, DM Liu, and W Xie. A survey on anonymity for privacy preserving data mining. In *Information Science and Electronic Engineering: Proceedings of the 3rd International Conference of Electronic Engineering and Information Science (ICEEIS 2016), 4-5 January, 2016, Harbin, China*, pages 343–346. CRC Press.

[Zhang *et al.*b] Zheng Zhang, Ping Tang, and Rubing Duan. Dynamic time warping under pointwise shape context. 315:88–101.

[Zhao and Itti] Jiaping Zhao and Laurent Itti. shapedtw: shape dynamic time warping.

List of Figures

List of Tables

Contents