Time-series data mining

Philippe Esling, Carlos Agon

Time series data mining

PHILIPPE ESLING and CARLOS AGON, Institut de Recherche et Coordination Acoustique / Musique (IRCAM), Paris

In almost every scientific field, measurements are performed over time. These observations lead to a collection of organized data called *time series*. The purpose of time series data mining is to try to extract all meaningful knowledge from the *shape* of data. Even if humans have a natural capacity to perform these tasks, it remains a complex problem for computers. In this paper we intend to provide a survey of the techniques applied for time series data mining. The first part is devoted to an overview of the tasks that have captured most of the interest of researchers. Considering that in most cases, time series task relies on the same components for implementation, we divide the literature depending on these common aspects, namely *representation* techniques, *distance* measures and *indexing* methods. The study of the relevant literature has been categorized for each individual aspects. Four types of robustness could then be formalized and any kind of distance could then be classified. Finally, the study submit various research trends and avenues that can be explored in the near future. We hope that this paper can provide a broad and deep understanding of the time series data mining research field.

Categories and Subject Descriptors: G.3 [**Probability and Statistics**]: Time Series Analysis; H.2.8 [**Database Management**]: Database Applications; H.3.1 [**Information storage and retrieval**]: Content Analysis and Indexing; H.3.3 [**Information storage and retrieval**]: Information Search and Retrieval

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Distance measures, data indexing, data mining, query by content, sequence matching, similarity measures, stream analysis, temporal analysis, time series

1. INTRODUCTION

A time series represents a collection of values obtained from sequential measurements over time. Time series data mining stems from the desire to reify our natural ability to visualize the *shape* of data. Humans rely on complex schemes in order to perform such tasks. We can actually avoid focusing on small fluctuations in order to derive a notion of *shape* and identify almost instantly similarities between patterns on various time scales. Major time series related tasks include query by content [Faloutsos et al. 1994], anomaly detection [Weiss 2004], motif discovery [Lin et al. 2004], prediction [Weigend and Gershenfeld 1994], clustering [Lin and Keogh 2005], classification [Bakshi and Stephanopoulos 1994] and segmentation [Keogh et al. 2003]. Despite the vast body of work devoted to this topic in the early years, [Antunes and Oliveira 2001] noted that *"the research has not been driven so much by actual problems but by an interest in proposing new approaches"*. However, with the ever-growing maturity of time series data mining techniques, this statement seems to have become obsolete. Nowadays, time series analysis covers a wide range of real-life problems in various fields of research. Some examples include economic forecasting [Song and Li 2008], intrusion detection [Zhong et al. 2007], gene expression analysis [Lin et al. 2008], medical surveillance [Burkom et al. 2007] and hydrology [Ouyang et al. 2010].

Time series data mining unveils numerous facets of complexity. The most prominent problems arise from the high dimensionality of time series data and the difficulty of defining a form of simi-

Author's addresses: P. Esling and C. Agon, 1, place Igor Stravinsky, F-75004, Paris, France. esling@ircam.fr

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

larity measure based on human perception. With the rapid growth of digital sources of information, time series mining algorithms will have to match increasingly massive datasets. These constraints show us that three major issues are involved:

— *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.
— *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.
— *Indexing method:* How should a massive set of time series be organized to enable fast querying? In other words, what *indexing mechanism* should be applied? The indexing technique should provide minimal space consumption and computational complexity.

These implementation components represent the core aspects of time series data mining systems. However these are not exhaustive as many tasks will require the use of more specific modules. Moreover, some of these are useless for some specific tasks. Forecasting (cf. section 3.5) is the most blatant example of a topic that requires more advanced analysis processes as it is more closely related to statistical analysis. It may require the use of a time series representation and a notion of similarity (mostly used to measure prediction accuracy) whereas model selection and statistical learning are also at the core of forecasting systems. The components that are *common* to most time series mining tasks have therefore been analyzed and other components found in related tasks have been briefly discussed.

The following part of this paper has been organized as follows: first introducing the fundamental concepts of time series data mining (section 2); then presenting an overview of the tasks to which most of the research in this field has been devoted (section 3); then reviewing the literature based on the three core components for implementation (section 4) and finally reviewing the research trends for future work in this field (section 5).

2. DEFINITIONS

The purpose of this section is to provide a definition for the terms used throughout this paper.

Definition 2.1. A *time series* T is an ordered sequence of n real-valued variables

$$T = (t_1, \ldots, t_n), t_i \in \mathbb{R}$$

A time series is often the result of the observation of an underlying process in the course of which values are collected from measurements made at uniformly spaced *time instants* and according to a given *sampling rate*. A time series can thus be defined as a set of contiguous time instants. The series can be *univariate* as in definition 2.1 or *multivariate* when several series simultaneously span multiple dimensions within the same time range.

Time series can cover the full set of data provided by the observation of a process and may be of considerable length. In the case of streaming, they are semi-infinite as time instants continuously feed the series. It thus becomes interesting to consider only the *subsequences* of a series.

Definition 2.2. Given a time series $T = (t_1, \ldots, t_n)$ of length n, a *subsequence* S of T is a series of length $m \leq n$ consisting of contiguous time instants from T

$$S = (t_k, t_{k+1}, \ldots, t_{k+m-1})$$

with $1 \leq k \leq n-m+1$. We denote the set of all subsequences of length m from T as $\mathbf{S}_T^m$.

For easier storage, massive time series sets are usually organized in a database.

Definition 2.3. A *time series database DB* is an unordered set of time series.

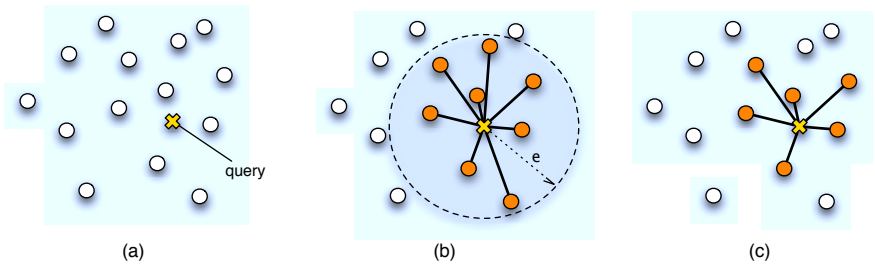

Fig. 1. Diagram of a typical query by content task represented in a 2-dimensional search space. Each point in this space represents a series whose coordinates are associated with its features. (a) When a query is entered into the system, it is first transformed into the same representation as that used for other datapoints. Two types of query can then be computed. (b) A $\varepsilon-$*range query* will return the set of series that are within distance ε of the query. (c) A $K-$*Nearest Neighbors query* will return the K points closest to the query.

As one of the major issues with time series data mining is the *high dimensionality* of data, the database usually contains only simplified representations of the series.

Definition 2.4. Given a time series $T = (t_1, ..., t_n)$ of length n, a *representation* of T is a model $\bar{T}$ of reduced dimensionality $\bar{d}$ ($\bar{d} \ll n$) such that $\bar{T}$ closely approximates T.

Nearly every task of time series data mining relies on a notion of similarity between series. After defining the general principle of similarity measures between time series, we will see (section 4.3) how these can be specified.

Definition 2.5. The *similarity measure* $\mathscr{D}(T,U)$ between time series T and U is a function taking two time series as inputs and returning the *distance* d between these series.

This distance has to be *non-negative*, i.e. $\mathscr{D}(T,U) \geq 0$. If this measure satisfies the additional *symmetry* property $\mathscr{D}(T,U) = \mathscr{D}(U,T)$ and *subadditivity* $\mathscr{D}(T,V) \leq \mathscr{D}(T,U) + \mathscr{D}(U,V)$ (also known as the *triangle inequality*), the distance is said to be a *metric*. As will be seen below (section 4.4), on the basis of the triangle inequality, metrics are very efficient measures for indexing. We may also extend this notion of distance to the subsequences.

Definition 2.6. The *subsequence similarity measure* $\mathscr{D}_{subseq}(T,S)$ is defined as

$$\mathscr{D}_{subseq}(T,S) = min\left(\mathscr{D}\left(T,S'\right)\right)$$

for $S' \in \mathbf{S}_S^{|T|}$. It represents the distance between T and its best matching location in S.

3. TASKS IN TIME SERIES DATA MINING

This section provides an overview of the tasks that have attracted wide research interest in time series data mining. These tasks are usually just defined as theoretical objectives though concrete applications may call for simultaneous use of multiple tasks.

3.1. Query by content

Query by content is the most active area of research in time series analysis. It is based on retrieving a set of solutions that are most similar to a query provided by the user. Figure 1 depicts a typical query by content task, represented on a 2-dimensional search space. We can define it formally as

Definition 3.1 (*Query by content*). Given a query time series $Q = (q_1, ..., q_n)$ and a similarity measure $\mathscr{D}(Q,T)$, find the ordered list $\mathscr{L} = \{T_1, \ldots, T_n\}$ of time series in the database *DB*, such that $\forall T_k, T_j \in \mathscr{L}, k > j \Leftrightarrow \mathscr{D}(Q,T_k) > \mathscr{D}(Q,T_j)$.

The content of the result set depends on the *type* of query performed over the database. The previous definition is in fact a generalized formalization of a query by content. It is possible to

specify a threshold ε and retrieve all series whose similarity with the query $\mathscr{D}(Q,T)$ is less than ε. This type of query is called an ε-*range query*.

Definition 3.2 (ε-*range query*). Given a query time series $Q=(q_1,...,q_n)$, a time series database DB, a similarity measure $\mathscr{D}(Q,T)$ and a threshold ε, find the set of series $\mathscr{S}=\{T_i \mid T_i \in DB\}$ that are within distance ε from Q. More precisely, find $\mathscr{S}=\{T_i \in DB \mid \mathscr{D}(Q,T_i) \leq \varepsilon\}$

Selecting this threshold is obviously highly data-dependent. Users usually want to retrieve a set of solutions by constraining the number of series it should contain, without knowing how far they will be from the query. It is thus possible to query the K most similar series in the database ($K-$Nearest Neighbors query).

Definition 3.3 (*K-Nearest Neighbors*). Given a query time series $Q=(q_1,...,q_n)$, a time series database DB, a similarity measure $\mathscr{D}(Q,T)$ and an integer K, find the set of K series that are the most similar to Q. More precisely, find $\mathscr{S}=\{T_i \mid T_i \in DB\}$ such that $|\mathscr{S}|=K$ and $\forall T_j \notin \mathscr{S}$, $\mathscr{D}(Q,T_i) \leq \mathscr{D}(Q,T_j)$

Such queries can be called on complete time series; however, the user may also be interested in finding every subsequence of the series matching the query, thus making a distinction between *whole series matching* and *subsequence matching*. This distinction between these types of queries is thus expressed in terms of $\varepsilon-$range query

Definition 3.4 (*Whole series matching*). Given a query Q, a similarity measure $\mathscr{D}(Q,T)$ and a time series database DB, find all series $T_i \in DB$ such that $\mathscr{D}(Q,T_i) \leq \varepsilon$

Definition 3.5 (*Subsequence matching*). Given a query Q, a similarity measure $\mathscr{D}(Q,T)$ and a database DB, find all subsequences $T_i^{'}$ of series $T_i \in DB$ such that $\mathscr{D}_{subseq}(Q,T_i^{'}) \leq \varepsilon$

In former times, time series mining was almost exclusively devoted to this task (cf. seminal work by [Agrawal et al. 1993]). In this paper, the representation was based on a set of coefficients obtained from a Discrete Fourier Transform (DFT) to reduce the dimensionality of data. These coefficients were then indexed with a R*-tree [Beckmann et al. 1990]. False hits were removed in a post-processing step, applying the Euclidean distance to complete time series. This paper laid the foundations of a reference framework that many subsequent works just enlarged by using properties of the DFT [Rafiei and Mendelzon 1998] or similar decompositions such as Discrete Wavelet Transform (DWT) [Chan and Fu 1999], that has been shown to have similar efficiency depending on the dataset at hand [Popivanov and Miller 2002]. The Discrete Cosine Transform (DCT) has also been suggested [Korn et al. 1997] but it appeared later that it did not have any advantage over other decompositions [Keogh et al. 2004]. Several numeric transformations – such as random projections [Indyk et al. 2000], Piecewise Linear Approximation (PLA) [Shatkay and Zdonik 1996], Piecewise Approximate Aggregation (PAA) [Keogh et al. 2001; Yi and Faloutsos 2000] and Adaptive Piecewise Constant Approximation (APCA) [Keogh et al. 2001] – have been used as representations. Symbolic representations have also been widely used. A shape alphabet with fixed resolution was originally proposed in [Agrawal et al. 1995]. Other symbolic representations have been proposed, such as the bit level approximation [Ratanamahatana et al. 2005] or the Symbolic Aggregate approXimation (SAX) [Lin et al. 2003]; the latter one has been shown to outperform most of the other representations [Stiefmeier et al. 2007]. We will find below a detailed overview of representations (section 4.2), distance measures (section 4.3) and indexing techniques (section 4.4).

Other important extensions to query by content include the handling of scaling and gaps [Vlachos et al. 2002], noise [Vlachos et al. 2004], query constraints [Goldin and Kanellakis 1995] and time warping, either by allowing false dismissals [Yi et al. 1998] or working without constraints [Sakurai et al. 2005]. Lower bounding distances without false dismissals for DTW were proposed in [Kim et al. 2001] and [Keogh and Ratanamahatana 2005] which allows exact indexing. The recent trend of query by content systems seems to be focused on streams. Given the continuously growing bandwidth, most of next generation analysis will most likely have to be performed over stream data. The

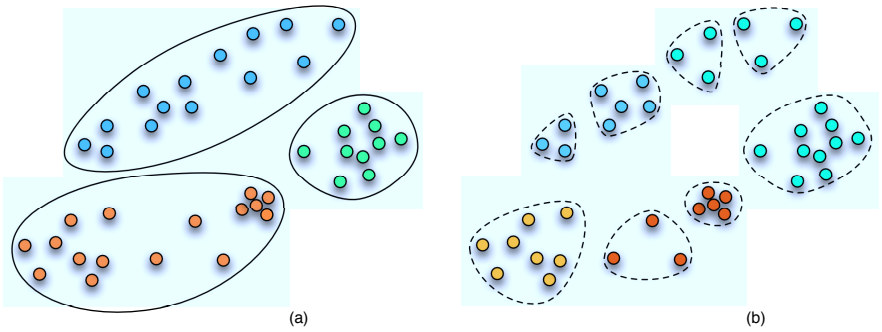

Fig. 2. Two possible outputs from the same clustering system obtained by changing the required number of clusters with (a) $N = 3$ and (b) $N = 8$. As we can see, the clustering task is a non trivial problem that highly depends on the way parameters are initialized and the level of detail targeted. This parameter selection issue is common to every clustering task, even out of the scope of time series mining.

dynamic nature of streaming time series precludes using the methods proposed for the static case. In a recent study, [Kontaki et al. 2009] introduced the most important issues concerning similarity search in static and streaming time series databases. In [Kontaki et al. 2007], the use of an incremental computation of DFT allows to adapt to the stream update frequency. However, maintaining the indexing tree for the whole streaming series seems to be uselessly costly. [Assent et al. 2009] proposed a filter-and-refine DTW algorithm called Anticipatory DTW, which allows faster rejection of false candidates. [Lian et al. 2010] proposed a weighted locality-sensitive hashing (WLSH) technique applying to approximate queries and working by incremental updating adaptive to the characteristics of stream data. [Lian and Chen 2007] proposed three approaches, polynomial, DFT and probabilistic, to predict future unknown values and answer queries based on the predicated data. This approach is a combination of prediction (cf. section 3.5) and streaming query by content; it is representative of an effort to obtain a convergence of approaches that seem to be heterogeneous.

3.2. Clustering

Clustering is the process of finding natural groups, called *clusters*, in a dataset. The objective is to find the most homogeneous clusters that are as distinct as possible from other clusters. More formally, the grouping should maximize inter-cluster variance while minimizing intra-cluster variance. The algorithm should thus automatically locate which groups are intrinsically present in the data. Figure **2** depicts some possible outputs of a clustering algorithm. It can be seen in this figure that the main difficulty concerning any clustering problem (even out of the scope of time series mining) usually lies in defining the correct number of clusters. The time series clustering task can be divided into two sub-tasks.

3.2.1. Whole series clustering. Clustering can be applied to each complete time series in a set. The goal is thus to regroup entire time series into clusters so that the time series are as similar to each other as possible within each cluster.

Definition 3.6. Given a time series database *DB* and a similarity measure $\mathscr{D}(Q,T)$, find the set of clusters $\mathscr{C} = \{c_i\}$ where $c_i = \{T_k \mid T_k \in DB\}$ that maximizes inter-cluster distance and minimizes intra-cluster variance. More formally $\forall i_1, i_2, j$ such that $T_{i_1}, T_{i_2} \in c_i$ and $T_j \in c_j$ $\mathscr{D}(T_{i_1}, T_j) \gg \mathscr{D}(T_{i_1}, T_{i_2})$

There have been numerous approaches for whole series clustering. Typically, after defining an adequate distance function, it is possible to adapt any algorithm provided by the generic clustering topic. Clustering is traditionnally performed by using Self Organizing Maps (SOM) [Chappelier and Grumbach 1996], Hidden Markov Models (HMM) [Smyth 1997] or Support Vector Machines (SVM) [Yoon et al. 2005]. [Gaffney and Smyth 1999] proposed a variation of the Expectation Ma-

ximization (EM) algorithm. However, this model-based approach has usually some scalability problems and implicitly presupposes the existence of an underlying model which is not straightforward for every dataset. Using Markov chain Monte Carlo (MCMC) methods, [Fröhwirth-Schnatter and Kaufmann 2008] makes an estimation about the appropriate grouping of time series simultaneously along with the group-specific model parameters. A good survey of generic clustering algorithms from a data mining perspective is given in [Berkhin 2006]. This review focuses on methods based on classical techniques that can further be applied to time series. A classification of clustering methods for various static data is proposed in [Han and Kamber 2006] following five categories: *partitioning*, *hierarchical*, *density-based*, *grid-based* and *model-based*. For the specificities of time series data, three of these five categories (partitioning, hierarchical and model-based) have been applied [Liao 2005]. Clustering of time series is especially useful for data streams; it has been implemented by using clipped data representations [Bagnall and Janacek 2005], Auto-Regressive (AR) models [Corduas and Piccolo 2008], $k-$Means [Vlachos et al. 2003] and – with greater efficiency – k-center clustering [Cormode et al. 2007]. Interested readers may refer to [Liao 2005] who provides a thorough survey of time series clustering issues by discussing the advantages and limitations of existing works as well as avenues for research and applications.

3.2.2. Subsequence clustering. In this approach, the clusters are created by extracting subsequences from a single or multiple longer time series.

Definition 3.7. Given a time series $T = (t_1, ..., t_n)$ and a similarity measure $\mathscr{D}(Q,C)$, find the set of clusters $\mathscr{C} = \{c_i\}$ where $c_i = \left\{ T_j^{'} \mid T_j^{'} \in \mathbf{S}_T^n \right\}$ is a set of subsequences that maximizes inter-cluster distance and intra-cluster cohesion.

In [Hebrail and Hugueney 2000], the series are sliced into non-overlapping windows. Their width is chosen by investigating the periodical structure of the time series by means of a DFT analysis. This approach is limited by the fact that, when no strong periodical structure is present in the series, non-overlapping slicing may miss important structures. A straightforward way to extend this approach can therefore be to extract shorter overlapping subsequences and then cluster the resulting set. However, this overlapping approach has been shown to produce meaningless results [Keogh et al. 2003]. Despite these deceptive results, the authors pointed out that a meaningful subsequence clustering system could be constructed on top of a motif mining [Patel et al. 2002] algorithm (cf. section 3.7). [Denton 2005] was first to suggest an approach to overcome this inconsistency by not forcing the algorithm to use all subsequences in the clustering process. In the context of intrusion detection, [Zhong et al. 2007] studied multiple centroid-based unsupervised clustering algorithms, and proposed a self-labeling heuristic to detect any attack within network traffic data. Clustering is also one of the major challenges in bioinformatics, especially in DNA analysis. [Kerr et al. 2008] surveyed state-of-the-art applications of gene expression clustering and provided a framework for the evaluation of results.

3.3. Classification

The classification task seeks to assign labels to each series of a set. The main difference when compared to the clustering task is that classes are known in advance and the algorithm is trained on an example dataset. The goal is first to learn what the distinctive *features* distinguishing classes from each others are. Then, when an unlabeled dataset is entered into the system, it can automatically determine which class each series belongs to. Figure 3 depicts the main steps of a classification task.

Definition 3.8. Given an unlabeled time series T, assign it to one class c_i from a set $\mathscr{C} = \{c_i\}$ of predefined classes.

There are two types of classification. The first one is the *time series classification* similar to whole series clustering. Given sets of time series with a label for each set, the task consists in training a

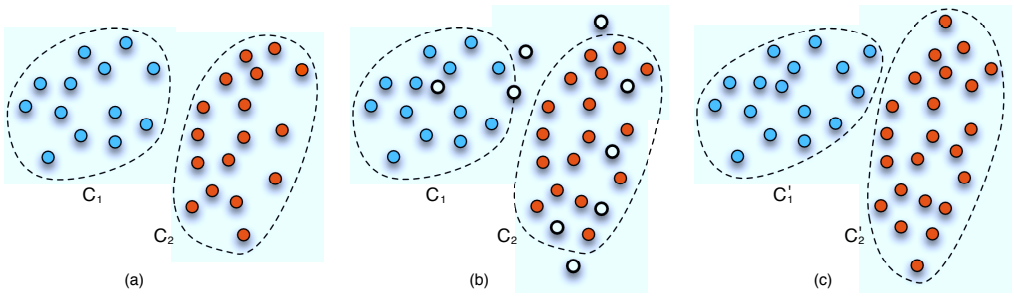

Fig. 3. The three main steps of a classification task. (a) A training set consisting of two pre-labeled classes C_1 and C_2 is entered into the system. The algorithm will first try to learn what the characteristic features distinguishing one class from another are; they are represented here by the class boundaries. (b) An unlabeled dataset is entered into the system that will then try to automatically deduce which class each datapoint belongs to. (c) Each point in the set entered has been assigned to a class. The system can then optionally adapt the classes boundaries.

classifier and labeling new time series. An early approach to time series classification was presented in [Bakshi and Stephanopoulos 1994]. However, it is based on simple trends whose results are therefore hard to interpret. A piecewise representation was later proposed in [Keogh and Pazzani 1998], it is robust to noise and weighting can be applied in a relevance feedback framework. The same representation was used in [Geurts 2001]; it is apparently not too robust to outliers. To overcome the obstacle of high dimensionality, [Jeng and Huang 2008] used Singular Value Decomposition to select essential frequencies. However, it implies higher computational costs. In a recent study, [Rodriguez and Kuncheva 2007] compared three types of classifiers: nearest neighbor, support vector machines and decision forests. All three methods seems to be valid, though highly depending on the dataset at hand. 1-NN classification algorithm with DTW seems to be the most widely used classifier; it was shown to be highly accurate [Xi et al. 2006], though computing speed is significantly affected by repeated DTW computations. To overcome this limitation [Srisai and Ratanamahatana 2009] proposed a template construction algorithm based on the Accurate Shape Averaging (ASA) technique. Each training class is represented by only one sequence so that any incoming series is compared only with one averaged template per class. Several other techniques have been introduced, such as ARMA models [Deng et al. 1997] or HMM [Zhong and Ghosh 2002]. In the context of clinical studies, [Lin et al. 2008] enhanced HMM approaches by using discriminative HMMs in order to maximize inter-classes differences. Using the probabilistic transitions between fewer states results in the patients being aligned to the model and can account for varying rates of progress. This approach has been applied in [Lowitz et al. 2009], in order to detect post-myocardial infarct patients. Several machine learning techniques have also been introduced such as neural networks [Nanopoulos et al. 2001] or Bayesian classification [Povinelli et al. 2004]. However, many of these proposals have been shown to be overpowered by a simple 1NN-DTW classifier [Xi et al. 2006]. A double-loop EM algorithm with a Mixture of Experts network structure has been introduced in [Subasi 2007] for the detection of epileptic seizure based on the EEG signals displayed by normal and epileptic patients. A well-known problem in classification tasks is the *overtraining*, i.e. when too many training data lead to an over-specified and inefficient model. [Ratanamahatana and Wanichsan 2008] suggested a stopping criterion to improve the data selection during a self training phase. [Zhang et al. 2009] proposed a time series reduction, which extracts patterns that can be used as inputs to classical machine-learning algorithms. Many interesting applications to this problem have been investigated such as brain–computer interface based on EEG signals; they have been reviewed in [Lotte et al. 2007].

3.4. Segmentation

The segmentation (or *summarization*) task aims at creating an accurate approximation of time series, by reducing its dimensionality while retaining its essential features. Figure 4 shows the output of a

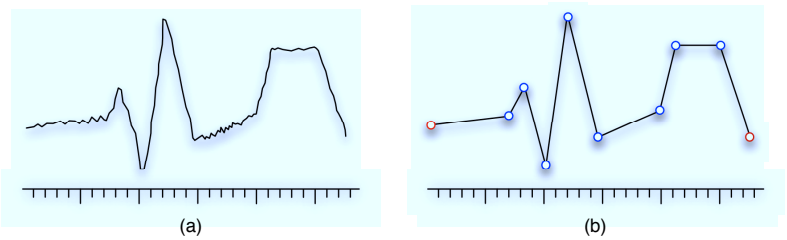

Fig. 4. Example of application of a segmentation system. From (a) usually noisy time series containing a very large number of datapoints, the goal is to find (b) the closest approximation of the input time series with the maximal dimensionality reduction factor without loosing any of its essential features.

segmentation system. Section 4.2 will show that most time series representations try to solve this problem implicitly.

Definition 3.9. Given a time series $T = (t_1, ..., t_n)$, construct a model $\bar{T}$ of reduced dimensionality $\bar{d}$ ($\bar{d} \ll n$) such that $\bar{T}$ closely approximates T. More formally $|R(\bar{T}) - T| < \varepsilon_r$, $R(\bar{T})$ being the reconstruction function and ε_r an error threshold.

The objective of this task is thus to minimize the reconstruction error between a reduced representation and the original time series. The main approach that have been undertaken over the years seems to be Piecewise Linear Approximation (PLA) [Shatkay and Zdonik 1996]. The main idea behind PLA is to split the series into most representative segments, and then fit a polynomial model for each segment. A good review on the most common segmentation methods in the context of PLA representation can be found in [Keogh et al. 2003]. Three basic approaches are distinguished. In *sliding windows*, a segment is grown until it exceeds some error threshold [Shatkay and Zdonik 1996]. This approach has shown poor performance with many real life datasets [Keogh et al. 2003]. The *top-down* approach consists in recursively partitioning a time series until some stopping criterion is met [Li et al. 1998]. This approach has time complexity $O\left(n^2\right)$ [Park et al. 1999] and is qualitatively outperformed by *bottom-up*. In this approach, starting from the finest approximation, segments are iteratively merged [Keogh and Pazzani 1998]. [Himberg et al. 2001] present fast greedy algorithms to improve previous approaches and a statistical method for choosing the number of segments is described in [Vasko and Toivonen 2002].

Several other methods have been introduced to handle this task. [Palpanas et al. 2008] introduced a representation of time series that implicitly handles the segmentation of time series. They proposed user-specified amnesic functions reducing the confidence to older data in order to make room for newer data. In the context of segmenting hydrological time series, [Kehagias 2004] proposed a maximum likelihood method using an HMM algorithm. However, this method offers no guarantee to yield the globally optimal segmentation without long execution times. For dynamic summary generation, [Ogras and Ferhatosmanoglu 2006] proposed an online transform-based summarization techniques over data streams that can be updated continuously. The segmentation of time-series can also be seen as a constrained clustering problem. [Abonyi et al. 2003] proposed to group time points by their similarity, provided that all points in a cluster come from contiguous time instants. Therefore, each cluster represents the segments in time whose homogeneity is evaluated with a local PCA model.

3.5. Prediction

Time series are usually very long and considered *smooth*, i.e. subsequent values are within predictable ranges of one another [Shasha and Zhu 2004]. The task of prediction is aimed at explicitly modeling such variable dependencies to forecast the next few values of a series. Figure 5 depicts various forecasting scenarios.

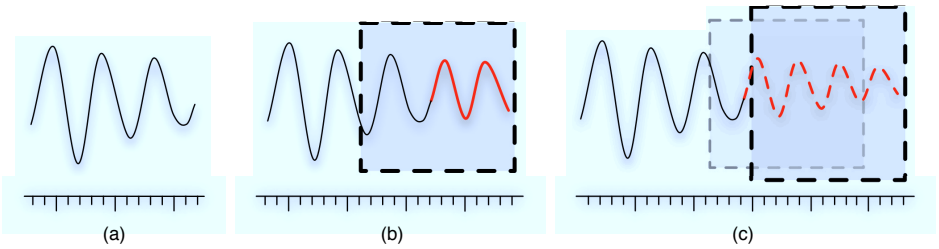

Fig. 5. A typical example of the time series prediction task. (a) The input time series may exhibit a periodical and thus predictable structure. (b) The goal is to forecast a maximum number of upcoming datapoints within a prediction window. (c) The task becomes really hard when it comes to having *recursive prediction*, i.e. the long term prediction of a time series implies reusing the earlier forecast values as inputs in order to go on predicting.

Definition 3.10. Given a time series $T = (t_1, ...t_n)$, predict the k next values $(t_{n+1}, ..., t_{n+k})$ that are most likely to occur.

Prediction is a major area in several fields of research. Concerning time series, it is one of the most extensively applied tasks. Literature about this is so abundant that dozens of reviews can focus on only a specific field of application or family of learning methods. Even if it can use time series representations and a notion of similarity to evaluate accuracy, It also relies on several statistical components that are out of the scope of this article, e.g. model selection and statistical learning. This task will be mentioned because of its importance but the interested reader willing to have further information may consult several references on forecasting [Brockwell and Davis 2002; Harris and Sollis 2003; Tsay 2005; Brockwell and Davis 2009] Several methods have been applied to this task. A natural option could be AR models [Box et al. 1976]. These models have been applied for a long time to prediction tasks involving signal de-noising or dynamic systems modeling. It is however possible to use more complex approaches such as neural networks [Koskela 2003] or clusters function approximation [Sfetsos and Siriopoulos 2004] to solve this problem. A polynomial architecture has been developed to improve a multilayer neural network in [Yadav et al. 2007] by reducing higher-order terms to a simple product of linear functions. Other learning algorithms, such as SOM, provided efficient supervised architectures. A survey of applications of SOM to time series prediction is given in [Barreto 2007]. Recent improvements for time series forecasting have been proposed; [Pesaran et al. 2006] proposed a Bayesian prediction for time series subject to discrete breaks, handling the size and duration of possible breaks by means of a hierarchical HMM. A dynamic genetic programming (GP) model tailored for forecasting streams was proposed in [Wagner et al. 2007] by adapting incrementally based on retained knowledge. The prediction task seems one of the most commonly applied in real-life applications, considering that market behavior forecasting relies on a wealth of financial data. [Bai and Ng 2008] proposed to refine the method of factor forecasting by introducing 'targeted predictors' selected by using a hysteresis (hard and soft thresholding) mechanism. The prediction task has also a wide scope of applications ranging from tourism demand forecasting [Song and Li 2008] to medical surveillance [Burkom et al. 2007]. In this paper, the authors compared the predictive accuracy of three methods, namely, non-adaptive regression, adaptive regression, and the Holt-Winters method; the latter appeared to be the best method. In a recent study, [Ahmed et al. 2009] carried out a large scale comparison for the major machine-learning models applied to time series forecasting, following which the best two methods turned out to be multilayer perceptron and Gaussian process regression. However, learning a model for long-term prediction seems to be more complicated, as it can use its own outputs as future inputs (*recursive prediction*). [Herrera et al. 2007] proposed the use of least-squares SVM, to solve this problem. [Cao and Tay 2009] further applied saliency analysis to SVM in order to remove irrelevant features based on the sensitivity of the network output to the derivative of the feature input. [Sorjamaa et al. 2007] proposed to combine direct prediction and an input selection in order to cope with long-term prediction of time series.

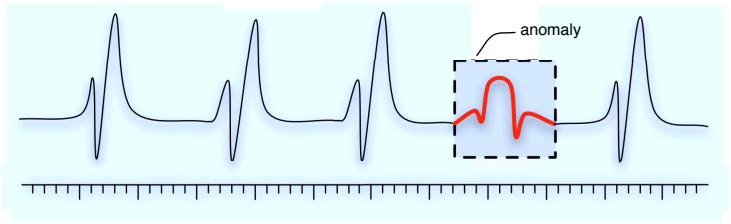

Fig. 6. An idealized example of the anomaly detection task. A long time series which exhibits some kind of periodical structure can be modeled thanks to a reduced pattern of "standard" behavior. The goal is thus to find subsequences which does not follow the model and may therefore be considered as anomalies.

3.6. Anomaly detection

The detection of anomalies seeks to find abnormal subsequences in a series. Figure 6 depicts an example of anomaly detection. It has numerous applications ranging from biosurveillance [Chuah and Fu 2007] to intrusion detection [Zhong et al. 2007].

Definition 3.11. Given a time series $T = (t_1, ..., t_n)$ and a model of its normal behavior, find all subsequences $T' \in \mathbf{S}_T^n$ which contain anomalies, i.e. do not fit the model.

A good discussion on the difficulties of mining rare events is given in [Weiss 2004]. The usual approach to detect anomalies is to first create a model of a series' normal behavior and characterize subsequences that stray too far from the model as anomalies. This approach can be linked to the prediction task. Indeed, if we can forecast the next values of a time series with a large accuracy, outliers can be detected in a straightforward manner and flagged as anomalies. This approach was undertaken first in [Ypma and Duin 1997] using SOM model to represent the expected behavior. A framework for novelty detection is defined in [Ma and Perkins 2003] and implemented based on Support Vector Regression (SVR). Machine learning techniques were also introduced to dynamically adapt their modelisation of normal behavior. [Ahmed et al. 2007] investigated the use of block-based One-Class Neighbor Machine and recursive Kernel-based algorithms and showed their applicability to anomaly detection. [Chen and Zhan 2008] proposed two algorithms to find anomalies in the Haar wavelet coefficients of the time series. A state-based approach is taken in [Salvador et al. 2004] using time point clustering so that clusters represents the normal behavior of a series. Another definition of anomalies, the time series *discords*, are defined as subsequences that are maximally different from all the remaining subsequences [Keogh et al. 2007]. This definition is able to capture the idea of most unusual subsequence within a time series and its unique parameter is the required length of the subsequences. Thanks to this definition [Yankov et al. 2008] proposed an exact algorithm that requires only two linear scans, thus allowing for the use of massive datasets. However, as several proposals, the number of anomalous subsequences must be specified prior to the search. Several real-life applications have also been outlined in recent research. Anomaly detection is applied in [Gupta et al. 2007] to detect fatigue damage in polycrystalline alloys, thus preventing problems in mechanical structures. An anomaly detection scheme for time series is used in [Chuah and Fu 2007] to determine whether streams coming from sensors contain any abnormal heartbeats. A recent overview and classification of the research on anomaly detection is presented in [Chandola et al. 2009], which provides a discussion on the computational complexity of each technique.

3.7. Motif discovery

Motif discovery consists in finding every subsequences (named *motif*) that appears recurrently in a longer time series. This idea was transferred from gene analysis in bioinformatics. Figure 7 depicts a typical example of motif discovery. Motifs were defined originally in [Patel et al. 2002] as *typical* non-overlapping subsequences. More formally

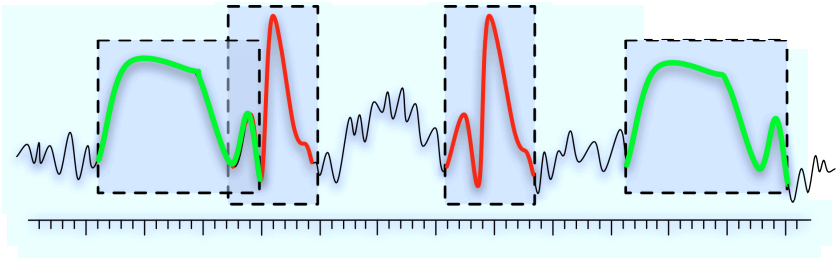

Fig. 7. The task of motif discovery consists in finding every subsequence that appears recurrently in a longer time series. These subsequences are named motifs. This task exhibits a high combinatorial complexity as several motifs can exist within a single series, motifs can be of various lengths and even overlap.

Definition 3.12. Given a time series $T = (t_1, \ldots, t_n)$, find all subsequences $T' \in \mathbf{S}_T^n$ that occurs repeatedly in the original time series.

A great interest for this research topic has been triggered by the observation that subsequence clustering produces meaningless results [Keogh et al. 2003]. The authors pointed out that motif discovery could be used as a subroutine to find meaningful clusters. In order to find motifs more efficiently, [Chiu et al. 2003] proposed to use the random projection algorithm [Buhler and Tompa 2002] which was successfully used for DNA sequences. However, because of its probabilistic nature, it is not guaranteed to find the exact set of motifs. [Ferreira et al. 2006] proposed an algorithm that can extract approximate motifs in order to mine time series data from protein folding/unfolding simulations. In [Liu et al. 2005], motif discovery is formalized as a continuous top-k motif balls problem in an m-dimensional space. However, the efficiency of this algorithm critically depends on setting the desired length of the pattern. [Tang and Liao 2008] introduced a k-motif-based algorithm that provides an interesting mechanism to generate summaries of motifs. [Yankov et al. 2007] showed that motif discovery can be severely altered by any slight change of *uniform scaling* (linear stretching of the pattern length) and introduced a scaling-invariant algorithm to determine the motifs. An algorithm for exact discovery of time series motifs has been recently proposed [Mueen et al. 2009], which is able to process very large datasets by using early abandoning on a linear re-ordering of data. [Mohammad and Nishida 2009] studied the constrained motif discovery problem which provides a way to incorporate prior knowledge into the motif discovery process. They showed that most unconstrained motif discovery problems can be transformed into constrained ones and provided two algorithms to solve such problem. The notion of motifs can be applied to many different tasks. The modeling of normal behavior for anomaly detection (cf. section 3.6) implies finding the recurrent motif of a series. For time series classification, significant speed-ups can be achieved by constructing motifs for each class [Zhang et al. 2009].

4. IMPLEMENTATION COMPONENTS

In this section, we review the implementation components common to most of time series mining tasks. As said earlier, the three key aspects when managing time series data are *representation* methods, *similarity* measures and *indexing* techniques. Because of the *high dimensionality* of time series, it is crucial to design low-dimensional representations that preserve the fundamental characteristics of a series. Given this representation scheme, the *distance* between time series needs to be carefully defined in order to exhibit perceptually relevant aspects of the underlying similarity. Finally the indexing scheme must allow to efficiently manage and query evergrowing massive datasets.

4.1. Preprocessing

In real-life scenarios, time series usually come from live observations [Reeves et al. 2009] or sensors [Stiefmeier et al. 2007] which are particularly subject to noise and outliers. These problems are

usually handled by preprocessing the data. Noise filtering can be handled by using traditional signal processing techniques like digital filters or wavelet thresholding. In [Himberg et al. 2001], Independent Component Analysis (ICA) is used to extract the main mode of the series. As will be explained in section 4.2, several representations implicitly handle noise as part of the transformation.

The second issue concerns the scaling differences between time series. This problem can be overcome by a linear transformation of the amplitudes [Goldin and Kanellakis 1995]. Normalizing to a fixed range [Agrawal et al. 1995] or first subtracting the mean (known as *zero mean / unit variance* [Keogh et al. 2001]) may be applied to both time series, however it does not give the optimal match of two series under linear transformations [Argyros and Ermopoulos 2003]. In [Goldin et al. 2004] the transformation is sought with optional bounds on the amount of scaling and shifting. However, normalization should be handled with care. As noted by [Vlachos et al. 2002], normalizing an essentially flat but noisy series to unit variance will completely modify its nature and normalizing small enough subsequences can provoke all series to look the same [Lin and Keogh 2005].

Finally, resampling (or *uniform time warping* [Palpanas et al. 2004]) can be performed in order to obtain series of the same length [Keogh and Kasetty 2003]. Down-sampling the longer series has been shown to be fast and robust [Argyros and Ermopoulos 2003].

4.2. Representation

As mentioned earlier, time series are essentially high dimensional data. Defining algorithms that work directly on the raw time series would therefore be computationally too expensive. The main motivation of representations is thus to emphasize the essential characteristics of the data in a concise way. Additional benefits gained are efficient storage, speedup of processing as well as implicit noise removal. These basic properties lead to the following requirements for any representation:

— Significant reduction of the data dimensionality
— Emphasis on fundamental shape characteristics on both *local* and *global* scales
— Low computational cost for computing the representation
— Good reconstruction quality from the reduced representation
— Insensitivity to noise or implicit noise handling

Many representation techniques have been investigated, each of them offering different trade-offs between the properties listed above. It is however possible to classify these approaches according to the kind of transformations applied. In order to perform such classification, we follow the taxonomy of [Keogh et al. 2004] by dividing representations into three categories, namely *non data-adaptive*, *data-adaptive* and *model-based*.

4.2.1. Non Data-Adaptive. In non data-adaptive representations, the parameters of the transformation remain the same for every time series regardless of its nature.

The first non data-adaptive representations were drawn from spectral decompositions. The DFT was used in the seminal work of [Agrawal et al. 1993]. It projects the time series on a sine and cosine functions basis [Faloutsos et al. 1994] in the real domain. The resulting representation is a set of sinusoidal coefficients. Instead of using a fixed set of basis functions, the DWT uses scaled and shifted versions of a mother *wavelet* function [Chan and Fu 1999]. This gives a multi-resolution decomposition where low frequencies are measured over larger intervals thus providing better accuracy [Popivanov and Miller 2002]. A large number of wavelet functions have been used in the literature like Haar [Chan et al. 2003], Daubechies [Popivanov and Miller 2002] or Coiflets [Shasha and Zhu 2004]. The Discrete Cosine Transform (DCT) uses only a cosine basis; it has also been applied to time series mining [Korn et al. 1997]. However, it has been shown that it does not offer any advantage over previously cited decompositions [Keogh et al. 2004]. Finally, an approximation by Chebychev polynomials [Cai and Ng 2004] has also been proposed but the results obtained have later been withdrawn due to an error in implementation.

Other approaches – more specific to time series – have been proposed. The Piecewise Aggregate Approximation (PAA) introduced by [Keogh et al. 2001] (submitted independently as Segmen-

ted Means in [Yi and Faloutsos 2000]) represents a series through the mean values of consecutive fixed-length segments. An extension of PAA including a multi-resolution property (MPAA) has been proposed in [Lin and Keogh 2005]. [Aßfalg et al. 2008] suggested to extract a sequence of amplitude-levelwise local features (ALF) to represent the characteristics of local structures. It was shown that this proposal provided weak results in [Ding et al. 2008]. Random projections have been used for representation in [Indyk et al. 2000]; in this case, each time series enters a convolution product with k random vectors drawn from a multivariate standard. This approach has recently been combined with spectral decompositions by [Reeves et al. 2009] with the purpose of answering statistical queries over streams.

4.2.2. Data-Adaptive. This approach implies that the parameters of a transformation are modified depending on the data available. By adding a data-sensitive selection step, almost all non data-adaptive methods can become data-adaptive. For spectral decompositions, it usually consists in selecting a subset of the coefficients. This approach has been applied to DFT [Vlachos et al. 2004] and DWT [Struzik et al. 1999]. A data-adaptive version of PAA has been proposed in [Megalooikonomou et al. 2004], with vector quantization being used to create a codebook of recurrent subsequences. This idea has been adapted to allow for multiple resolution levels [Megalooikonomou et al. 2005]. However, this approach has only been tested on smaller datasets. A similar approach has been undertaken in [Stiefmeier et al. 2007] with a codebook based on motion vectors being created to spot gestures. However, it has been shown to be computationally less efficient than SAX.

Several inherently data-adaptive representations have also been used. SVD has been proposed [Korn et al. 1997] and later been enhanced for streams [Ravi Kanth et al. 1998]. However, SVD requires computation of eigenvalues for large matrices and is therefore far more expensive than other mentioned schemes. It has recently been adapted to find multi-scale patterns in time series streams [Papadimitriou and Yu 2006]. PLA [Shatkay and Zdonik 1996] is a widely used approach for the segmentation task (cf. section **3.4**) The set of polynomial coefficients can be obtained either by interpolation [Keogh and Pazzani 1998] or regression [Huang and Yu 1999]. Many derivatives of this technique have been introduced. The Landmarks system [Perng et al. 2000] extends this notion to include a multi-resolution property. However, the extraction of features relies on several parameters which are highly data-dependent. APCA [Keogh et al. 2001] uses constant approximations per segment instead of polynomial fitting. Indexable PLA has been proposed by [Chen et al. 2007] to speed up the indexing process. [Palpanas et al. 2004] put forward an approach based on PLA, to answer queries about the recent past with greater precision than older data and called such representations amnesic. The method consisting in using a segmentation algorithm as a representational tool has been extensively investigated. The underlying idea is that segmenting a time series can be equated with the process of representing the most salient features of a series while considerably reducing its dimensionality. [Xie and Yan 2007] proposed a pattern-based representation of time series. The input series is approximated by a set of concave and convex patterns to improve the subsequence matching process. [Zhan et al. 2007] proposed a pattern representation of time series to extract outlier values and noise. The Derivative Segment Approximation (DSA) model [Gullo et al. 2009] is a representation based on time series segmentation through an estimation of derivatives to which DTW can be applied. The polynomial shape space representation [Fuchs et al. 2010] is a subspace representation consisting of trend aspects estimators of a time series. [Bandera et al. 2009] put forward a two-level approach to recognize gestures by describing individual trajectories with key-points, then characterizing gestures through the global properties of the trajectories.

Instead of producing a numeric output, it is also possible to discretize the data into symbols. This conversion into a symbolical representation also offers the advantage of implicitly performing noise removal by complexity reduction. A relational tree representation is used in [Bakshi and Stephanopoulos 1995]. Non-terminal nodes of the tree correspond to valleys and terminal nodes to peaks in the time series. The Symbolic Aggregate approXimation (SAX) [Lin et al. 2003], based on the same underlying idea as PAA, calls on equal frequency histograms on sliding windows to create a sequence of short words. An extension of this approach, called indexable Symbolic Aggregate

approXimation (iSAX) [Shieh and Keogh 2008], has been proposed to make fast indexing possible by providing zero overlap at leaf nodes. The grid-based representation [An et al. 2003] places a two dimensional grid over the time series. The final representation is a bit string describing which values were kept and which bins they were in. Another possibility is to discretize the series to a binary string (a technique called *clipping*) [Ratanamahatana et al. 2005]. Each bit indicates whether the series is above or below the average. That way, the series can be very efficiently manipulated. In [Bagnall et al. 2003] this is done using the median as the clipping threshold. Clipped series offer the advantage of allowing direct comparison with raw series, thus providing a tighter lower bounding metric. Thanks to a variable run-length encoding, [Bagnall et al. 2006] show that it is also possible to define an approximation of the Kolmogorov complexity. Recently, a very interesting approach has been proposed in [Ye and Keogh 2009]; it is based on primitives called *shapelets*, i.e. subsequences which are maximally representative of a class and thus fully discriminate classes through the use of a dictionary. This approach can be considered as a step forward towards bridging the gap between time series and shape analysis.

4.2.3. Model-based. The model-based approach is based on the assumption that the time series observed has been produced by an underlying model. The goal is thus to find parameters of such a model as a representation. Two time series are therefore considered similar if they have been produced by the same set of parameters driving the underlying model. Several parametric temporal models may be considered, including statistical modeling by feature extraction [Nanopoulos et al. 2001], ARMA models [Kalpakis et al. 2001] Markov Chains (MCs) [Sebastiani et al. 1999] or HMM [Panuccio et al. 2002]. MCs are obviously simpler than HMM so they fit well shorter series but their expressive power is far more limited. The Time Series bitmaps introduced in [Kumar et al. 2005] can also be considered as a model-based representation for time series, even if it mainly aims at providing a visualization of time series.

4.3. Similarity measure

Almost every time series mining task requires a subtle notion of similarity between series, based on the more intuitive notion of *shape*. When observing simultaneously multiple characteristics of a series, humans can abstract from such problems as amplitude, scaling, temporal warping, noise and outliers. The Euclidean distance is obviously unable to reach such a level of abstraction. Numerous authors have pointed out several pitfalls when using L_p norms [Ding et al. 2008; Keogh and Kasetty 2003; Yi and Faloutsos 2000]. However, it should be noted that, in the case of very large datasets, Euclidean distance has been shown [Shieh and Keogh 2008] to be sufficient as there is a larger probability that an almost exact match exists in the database. Otherwise, a similarity measure should be consistent with our intuition and provide the following properties:

(1) It should provide a recognition of perceptually similar objects, even though they are not mathematically identical;
(2) It should be consistent with human intuition;
(3) It should emphasize the most salient features on both *local* and *global* scales;
(4) A similarity measure should be universal in the sense that it allows to identify or distinguish arbitrary objects, i.e. no restrictions on time series are assumed;
(5) It should abstract from distortions and be invariant to a set of transformations.

Many authors have reported about various transformation invariances required for similarity. Given a time series $T = \{t_1, \ldots, t_n\}$ of n datapoints, we consider the following transformations:

— *Amplitude shifting*: The series $G = \{g_1, \ldots, g_n\}$ obtained by a linear amplitude shift of the original series $g_i = t_i + k$ with $k \in \mathbb{R}$ a constant.
— *Uniform amplification*: The series G obtained by multiplying the amplitude of the original series $g_i = k.t_i$ with $k \in \mathbb{R}$ a constant.
— *Uniform time scaling*: The series $G = \{g_1, \ldots, g_m\}$ produced by a uniform change of the time scale of the original series $g_i = t_{\lceil k.i \rceil}$ with $k \in \mathbb{R}$ a constant.

— *Dynamic amplification*: The series G obtained by multiplying the original series by a dynamic amplification function $g_i = h(i).t_i$ with $h(i)$ a function such that $\forall t \in [1 \dots n]$, $h'(t) = 0$ if and only if $t'_i = 0$.
— *Dynamic time scaling*: The series G obtained by a dynamic change of the time scale $g_i = t_{h(i)}$ with $h(i)$ a positive, strictly increasing function such that $h : \mathbb{N} \to [1 \dots n]$
— *Additive Noise*: The series G obtained by adding a noisy component to the original series $g_i = t_i + \varepsilon_i$ with ε_i an independent identically distributed white noise.
— *Outliers*: The series G obtained by adding outliers at random positions. Formally, for a given set of random time positions $\mathscr{P} = \{k \mid k \in [1 \dots n]\}$, $g_k = \varepsilon_k$ with ε_k an independent identically distributed white noise.

The similarity measure $\mathscr{D}(T,G)$ should be robust to any combinations of these transformations. This property lead to our formalization of four general types of robustness. We introduce properties expressing robustness for *scaling* (amplitude modifications), *warping* (temporal modifications), *noise* and *outliers*. Let $\mathscr{S}$ be a collection of time series, and let $\mathscr{H}$ be the maximal group of homeomorphisms under which $\mathscr{S}$ is closed. A similarity measure $\mathscr{D}$ on $\mathscr{S}$ is called *scale robust* if it satisfies

> *Property.* For each $T \in \mathscr{S}$ and $\alpha > 0$ there is a $\delta > 0$ such that $\|t_i - h(t_i)\| < \delta$ for all $t_i \in T$ implies $\mathscr{D}(T, h(T)) < \alpha$ for all $h \in \mathscr{H}$.

We call a similarity measure *warp robust* if the following holds

> *Property.* For each $T = \{t_i\} \in \mathscr{S}, T' = \{t_{h(i)}\}$ and $\alpha > 0$ there is a $\delta > 0$ such that $\|i - h(i)\| < \delta$ for all $t_i \in T$ implies that $\mathscr{D}(T,T') < \alpha$ for all $h \in \mathscr{H}$.

We call a similarity measure *noise robust* if it satisfies the following property

> *Property.* For each $T \in \mathscr{S}$ and $\alpha > 0$, there is a $\delta > 0$ such that $U = T + \varepsilon$ with $p(\varepsilon) = \mathscr{N}(0, \delta)$ implies $\mathscr{D}(T,U) < \alpha$ for all $U \in \mathscr{S}$

We call a measure *outlier robust* if the following holds

> *Property.* For each $T \in \mathscr{S}$, $\mathscr{K} = \{rand[1...n]\}$ and $\alpha > 0$, there is a $\delta > 0$ such that if $|\mathscr{K}| < \delta$ and $U_{k \in \mathscr{K}} = \varepsilon_k$ and $U_{k \notin \mathscr{K}} = T_k$ implies $\mathscr{D}(T,U) < \alpha$ for all $U \in \mathscr{S}$

Similarity measures can be classified in four categories. *Shape-based* distances compare the overall shape of the series. *Edit-based* distances compare two time series on the basis of the minimum number of operations needed to transform one series into another one. *Feature-based* distances extract features describing aspects of the series that are then compared with any kind of distance function. *Structure-based* similarity aims at finding higher-level structures in the series to compare them on a more global scale. We further subdivide this category into two specific subcategories. *Model-based* distances work by fitting a model to the various series and then comparing the parameters of the underlying models. *Compression-based* distances analyze how well two series can be compressed together. Similarity is reflected by higher compression ratios. As defined by [Keogh and Kasetty 2003], we refer to distance measures that compare the i–th point of a series to the i–th point of another as *lock-step* and measures that allow flexible (one-to-many / one-to-none) comparison as *elastic*.

4.3.1. Shape-based. The Euclidean distance and other L_p norms [Yi and Faloutsos 2000] have been the most widely used distance measures for time series [Keogh and Kasetty 2003]. However, these have been shown to be poor similarity measurements [Antunes and Oliveira 2001; Ding et al. 2008]. As a matter of fact, these measures does not match any of the types of robustness. Even if the problems of scaling and noise can be handled in a preprocessing step [Goldin and Kanellakis 1995], the warping and outliers issues need to be addressed with more sophisticated techniques. This is where the use of elastic measures can provide an elegant solution to both problems.

Handling the local distortions of the time axis is usually addressed using *non-uniform time warping* [Keogh and Pazzani 1998], more specifically with Dynamic Time Warping (DTW) [Berndt and Clifford 1994]. This measure is able to match various sections of a time series by allowing warping of the time axis. The optimal alignment is defined by the shortest warping path in a distance matrix. A warping path W is a set of contiguous matrix indices defining a mapping between two time series. Even if there is an exponential number of possible warping paths, the optimal path is the one that minimizes the global warping cost. DTW can be computed using dynamic programming with time complexity $O(n^2)$ [Ratanamahatana and Keogh 2004a]. However, several lower bounding measures have been introduced to speed up the computation. [Keogh and Ratanamahatana 2005] introduced the notion of upper and lower envelope that represents the maximum allowed warping. Using this technique, the complexity becomes $O(n)$. It is also possible to impose a *temporal constraint* on the size of the DTW warping window. It has been shown that these improve not only the speed but also the level of accuracy as it avoids the pathological matching introduced by extended warping [Ratanamahatana and Keogh 2004b]. The two most frequently used global constraints are the Sakoe-Chiba Band and the Itakura Parallelogram. [Salvador and Chan 2007] introduced the FastDTW algorithm which makes a linear time computation of DTW possible by recursively projecting a warp path to a higher resolution and then refining it. A drawback of this algorithm is that it is approximate and therefore offer no guarantee to finding the optimal solution. In addition to dynamic warping, it may sometimes be useful to allow a global scaling of time series to achieve meaningful results, a technique known as *uniform scaling* (US). [Fu et al. 2008] proposed the scaled and warped matching (SWM) similarity measure that makes it possible to combine the benefits of DTW with those of US.

Other shape-based measures have been introduced such as the Spatial Assembling Distance (SpADe) [Chen et al. 2007]; it is a pattern-based similarity measure. This algorithm identifies matching *patterns* by allowing shifting and scaling on both temporal and amplitude axes, thus being scale robust. The DISSIM [Frentzos et al. 2007] distance has been introduced to handle similarity at various sampling rates. It is defined as an approximation of the integral of the Euclidean distance. One of the most interesting recent proposals is based on the concept of elastic matching of time series [Latecki et al. 2005]. [Latecki et al. 2007] presented an optimal subsequence matching (OSB) technique that is able to automatically determine the best subsequence and warping factor for distance computation; it includes a penalty when skipping elements. Optimality is achieved through a high computational cost; however, it can be reduced by limiting the skipping range.

4.3.2. Edit-based. Edit-based methods (also known as *Levenshtein distance*) has originally been applied to characterize the difference between two strings. The underlying idea is that the distance between strings may be represented by the minimum number of operations needed to transform one string into another, with insertion, deletion and substitution. The presence of outliers or noisy regions can thus be compensated by allowing gaps in matching two time series. [Das et al. 1997] use the Longest Common Subsequence (LCSS) algorithm to tackle this problem. The LCSS distance uses a *threshold parameter* ε for point matching and a *warping threshold* δ. A fast approximate algorithm to compute LCSS has been described in [Bollobas et al. 1997]. [Vlachos et al. 2002] normalized the LCSS similarity by the length of the time series and allowed linear transformations. [Vlachos et al. 2006] introduced lower-bounding measure and indexing techniques for LCSS. DTW requires the matched time series to be well aligned and its efficiency deteriorates with noisy data as, when matching all the points, it also matches the outliers distorting the true distance between sequences. LCSS has been shown to be more robust than DTW under noisy conditions [Vlachos et al. 2002]; this heavily depends on the threshold setting. [Morse and Patel 2007] proposed the Fast Time Series Evaluation (FTSE) method for computing LCSS. On the basis of this algorithm, they proposed the Sequence Weighted Alignment model (Swale) that extends the ε threshold-based scoring techniques to include arbitrary match rewards and gap penalties. The Edit Distance on Real sequence (EDR) [Chen et al. 2005] is an adaptation of the edit distance to real-valued series. Contrary to LCSS, EDR assign penalties depending on the length of the gaps between the series. The Edit Distance with Real Penalty (ERP) [Chen and Ng 2004] attempts to combine the merits of DTW and edit distance

by using a *constant reference point.* For the same purpose, [Marteau 2008] submitted an interesting dynamic programming algorithm called Time Warp Edit Distance (TWED). TWED is slightly different from DTW, LCSS, or ERP algorithms. In particular, it highlights a parameter that controls a kind of stiffness of the elastic measure along the time axis. Another extension to the edit distance has been proposed in [Muhammad Fuad and Marteau 2008], it has been called the extended edit distance (EED). Following the observation that the edit distance penalizes all change operations with the same cost, it includes an additional term reflecting whether the operation implied characters that are more frequent, therefore closer in distance. A different approach for constraining the edit operations has been proposed in [Chhieng and Wong 2010]; it is based on the Constraint Continuous Editing Distance (CCED) that adjusts the potential energy of each sequence to achieve optimal similarity. As CCED does not satisfy triangle inequality, a lower bounding distance is provided for efficient indexing.

4.3.3. Feature-based. These measures rely on the computation of a feature set reflecting various aspects of the series. Features can be selected by using coefficients from DFT [Shatkay and Zdonik 1996] or DWT decompositions (cf. section 4.2.2) In [Janacek et al. 2005], a likelihood ratio for DFT coefficients has been shown to outperform Euclidean distance. In [Vlachos et al. 2005], a combination of periodogram and autocorrelation functions allows to select the most important periods of a series. This can be extended to carrying out local correlation tracking as proposed in [Papadimitriou et al. 2006].

Concerning symbolic representations, [Mannila and Seppnen 2001] represent each symbol with a random vector and a symbolic sequence by the sum of the vectors weighted by the temporal distance of the symbols. In [Flanagan 2003] weighted histograms of consecutive symbols are used as features. The similarity search based on Threshold Queries (TQuEST) [Aßfalg et al. 2006] use a given threshold parameter τ in order to transform a time series into a sequence of *threshold-crossing* time intervals. It has however been shown to be highly specialized with mitigated results on classical datasets [Ding et al. 2008]. [Bartolini et al. 2005] proposed a Fourier-based approach, called WARP and making the using of the DFT phase possible, this being more accurate for a description of object boundaries.

An approach using ideas from shape and feature-based representations has been described in [Megalooikonomou et al. 2005]. Typical local shapes are extracted with vector quantization and the time series are represented by histograms counting the occurrences of these shapes at several resolutions. Multiresolution Vector Quantized (MVQ) approximation keeps both local and global information about the original time series, so that defining a multi-resolution and hierarchical distance function is made possible.

4.3.4. Structure-based. Even if the previously cited approaches have been useful for short time series or subsequences applications, they often fail to produce meaningful results on longer series. This is mostly due to the fact that these distances are usually defined to find *local* similarities between patterns. However, when handling very long time series, it might be more profitable to find similarities on a more *global* scale. Structure-based distances [Lin and Li 2009] are thus designed to identify higher-level structures in series.

Model-based. Model-based distances offer the additional advantage that prior knowledge about the generating process can be incorporated in the similarity measurement. The similarity can be measured by modeling one time series and determining the likelihood that one series was produced by the underlying model of another. Any type of parametric temporal model may be used. HMM with continuous output values or ARMA models are common choices [Xiong and Yeung 2004]. However, best results are obtained if the model selected is related to the type of production that generated the data available. In [Ge and Smyth 2000], HMMs are combined with a piecewise linear representation. In [Panuccio et al. 2002] the distance between HMM is normalized to take into account the quality of fit of the series producing the model. [Bicego et al. 2003] use the similarity-based paradigm where HMM is used to determine the similarity between each object and a pre-

determinate set of other objects. For short time series, it is also possible to use regression models as proposed by [Gaffney and Smyth 1999].

Among other common choices for symbolic representations, we may cite MC [Reinert et al. 2000], HMM with discrete output distributions [Law and Kwok 2000], and grammar based models [Antunes and Oliveira 2001]. Alternatively to pairwise likelihood, the Kullback-Leibler divergence allows to have direct comparison of models [Sebastiani et al. 1999].

Compression-based. [Keogh et al. 2004], inspired by results obtained in bioinformatics, defined a distance measure based on the Kolmogorov complexity called Compression-Based Dissimilarity Measure (CDM). The underlying idea is that concatenating and compressing similar series should produce higher compression ratios than when doing so with very different data. This approach appears to be particularly efficient for clustering; it has been applied to fetal heart rate tracings [Costa Santos et al. 2006]. Following the same underlying ideas, [Degli Esposti et al. 2009] recently proposed a parsing-based similarity distance in order to distinguish healthy patients from hospitalized ones on the basis of various symbolic codings of ECG signals. By comparing the performances of several data classification methods, this distance is shown to be a good compromise between accuracy and computational efforts. Similar approaches have been undertaken earlier in bioinformatics [Chen et al. 2000] and several compression techniques – such as the Lempel-Ziv complexity [Otu and Sayood 2003] – have been successfully applied to compute similarity between biological sequences.

4.3.5. Comparison of distance measures. The choice of an adequate similarity measure highly depends on the nature of the data to analyze as well as application-specific properties that could be required. If the time series are relatively short and visual perception is a meaningful description, shape-based methods seems to be the appropriate choice. If the application is targeting a very specific dataset or any kind of prior knowledge about the data is available, model-based methods may provide a more meaningful abstraction. Feature-based methods seem more appropriate when periodicities is the central subject of interest and causality in the time series is not relevant. Finally, if the time series are long and little knowledge about the structure is available, the compression-based and more generally structure-based approaches have the advantage of being a more generic and parameter-free solution for the evaluation of similarity. Even with these general recommendations and comparisons for the selection of an appropriate distance measure, the accuracy of the similarity chosen still has to be evaluated. Ironically, it seems almost equally complex to find a good accuracy measure to evaluate the different similarities. However (cf. section 4.4), a crucial result when indexing is that any distance measure should lower bound the true distance between time series in order to preclude false dismissals [Faloutsos et al. 1994]. Therefore the tightness of lower bound [Keogh and Kasetty 2003] appears to be the most appropriate option to evaluate the performance of distance measures as it is a completely hardware and implementation independent measure and offers a good prediction concerning the indexing performance. The accuracy of distance measures are usually evaluated within a 1-NN classifier framework. It has been shown by [Ding et al. 2008] that, despite all proposals regarding different kinds of robustness, the forty year old DTW usually performs better. Table I summarizes the properties of every distance measures reviewed in this paper, based on our formalization of four types of robustness. It also determines whether the distance is a metric and indicates the computational cost and the number of parameters required.

4.4. Indexing

An indexing scheme allows to have an efficient organization of data for quick retrieval in large databases. Most of the solutions presented involve a dimensionality reduction in order to index this representation using a spatial access method. Several studies suggest that the various representations differ but slightly in terms of indexing power [Keogh and Kasetty 2003]. However, wider differences arise concerning the quality of results and the speed of querying. There are two main issues when

Table I. Comparison of the distance measures surveyed in this paper with the four properties of robustness. Each distance measure is thus distinguished as *scale* (amplitude), *warp* (time), *noise* or *outliers* robust. The next column shows whether the proposed distance is a metric. The cost is given as a simplified factor of computational complexity. The last column gives the minimum number of parameters setting required by the distance measure.

Distance measure	Scale	Warp	Noise	Outliers	Metric	Cost	Param
Shape-based							
L_p norms					√	$O(n)$	0
Dynamic Time Warping (DTW)		√				$O(n^2)$	1
LB_Keogh (DTW)		√	√		√	$O(n)$	1
Spatial Assembling (SpADe)	√	√	√			$O(n^2)$	4
Optimal Bijection (OSB)		√	√	√		$O(n^2)$	2
DISSIM		√	√		√	$O(n^2)$	0
Edit-based							
Levenshtein				√	√	$O(n^2)$	0
Weighted Levenshtein				√	√	$O(n^2)$	3
Edit with Real Penalty (ERP)		√		√	√	$O(n^2)$	2
Time Warp Edit Distance (TWED)		√		√	√	$O(n^2)$	2
Longest Common SubSeq (LCSS)		√	√	√		$O(n)$	2
Sequence Weighted Align (Swale)		√	√	√		$O(n)$	3
Edit Distance on Real (EDR)		√	√	√	√	$O(n^2)$	2
Extended Edit Distance (EED)		√	√	√	√	$O(n^2)$	1
Constraint Continuous Edit (CCED)		√	√	√		$O(n)$	1
Feature-based							
Likelihood			√	√	√	$O(n)$	0
Autocorrelation			√	√	√	$O(nlogn)$	0
Vector quantization		√	√	√	√	$O(n^2)$	2
Threshold Queries (TQuest)		√	√	√		$O(n^2logn)$	1
Random Vectors		√	√	√		$O(n)$	1
Histogram			√	√	√	$O(n)$	0
WARP	√	√	√		√	$O(n^2)$	0
Structure-based							
Model-based							
Markov Chain (MC)			√	√		$O(n)$	0
Hidden Markov Models (HMM)	√	√	√	√		$O(n^2)$	1
Auto-Regressive (ARMA)			√	√		$O(n^2)$	2
Kullback-Leibler			√	√	√	$O(n)$	0
Compression-based							
Compression Dissimilarity (CDM)		√	√	√		$O(n)$	0
Parsing-based		√	√	√		$O(n)$	0

designing an indexing scheme: *completeness* (no false dismissals) and *soundness* (no false alarms). In an early paper, [Faloutsos et al. 1994] list the properties required for indexing schemes:

(1) It should be much faster than sequential scanning.
(2) The method should require little space overhead.
(3) The method should be able to handle queries of various lengths.
(4) The method should allow insertions and deletions without rebuilding the index.
(5) It should be correct, i.e. there should be no false dismissals.

As noted by [Keogh et al. 2001] there are two additional desirable properties:

(1) It should be possible to build the index within "reasonable time".
(2) The index should be able to handle different distance measures.

A time series X can be considered as a point in an n-dimensional space. This immediately suggests that time series could be indexed by Spatial Access Methods (SAMs). These allow to partition

space into regions along a hierarchical structure for efficient retrieval. B-trees [Bayer and McCreight 1972] on which most hierarchical indexing structures are based, were originally developed for one-dimensional data. They use prefix separators, thus no overlap for unique data objects is guaranteed. Multidimensional indexing structures – such as the R-tree [Beckmann et al. 1990] – use data organized in minimum bounding rectangles (MBR). However, when summarizing data in minimum bounding regions, the sequential nature of time series cannot be captured. Their main shortcoming is that wide MBR produce large overlap with a majority of empty space. Queries therefore intersect with many of these MBRs.

Typical time series contain over thousand datapoints and most SAM approaches are known to degrade quickly at dimensionality greater than 8-12 [Chakrabarti and Mehrotra 1999]. The degeneration with high dimensions caused by overlapping can result in having to access almost the entire dataset by random I/O. Therefore, any benefit gained when indexing is lost. As R-trees and their variants are victims of the phenomenon known as the '*dimensionality curse*' [Bohm et al. 2001], a solution for their usage is to first perform dimensionality reduction. The X-tree (extended node tree), for example, uses a different split strategy to reduce overlap [Berchtold et al. 2002]. The A-tree (approximation tree) uses VA-file-style (vector approximation file) quantization of the data space to store both MBR and VBR (virtual bounding rectangle) lower and upper bounds [Sakurai et al. 2000]. The TV-tree (telescopic vector tree) is an extension of the R-tree. It uses minimum bounding regions (spheres, rectangles or diamonds, depending on the type of L_p norm used) restricted to a subset of active dimensions. However, not all methods rely on SAM to provide efficient indexing. [Park et al. 2000] proposed the use of suffix trees [Gusfield 1997] to index time series. The idea is that distance computation relies on comparing prefixes first, so it is possible to store every series with identical prefixes in the same nodes. The subtrees will therefore only contain the suffixes of the series. However, this approach seems hardly scalable for longer time series or more subtle notions of similarity. In [Faloutsos et al. 1994] the authors introduced the GEneric Multimedia INdexIng method (GEMINI) which can apply any dimensionality reduction method to produce efficient indexing. [Yi and Faloutsos 2000] studied the problem of multi-modal similarity search in which users can choose between multiple similarity models depending on their needs. They introduced an indexing scheme for time series where the distance function can be any $\mathscr{L}_p$ norm. Only one index structure is needed for all $\mathscr{L}_p$ norms. To analyze the efficiency of indexing schemes, [Hellerstein et al. 1997] considered the general problem of database indexing workloads (combinations of data sets and sets of potential queries). They defined a framework to measure the efficiency of an indexing scheme based on two characterizations: *storage redundancy* (how many times each item in the data set is stored) and *access overhead* (how many unnecessary blocks are retrieved for a query). For indexing purposes, envelope-style upper and lower bounds for DTW have been proposed [Keogh and Ratanamahatana 2005]; the indexing procedure of short time series is efficient but similarity search typically entails more page reads. This framework has been extended [Vlachos et al. 2006] in order to index multidimensional time series with DTW as well as LCSS. [Assent et al. 2008] proposed the TS-tree, an indexing method offering efficient similarity search on time series. It avoids overlap and provides compact meta data information on the subtrees, thus reducing the search space. In [Kontaki et al. 2007], the use of an Incremental DFT Computation index (IDC-Index) has been proposed to handle streams based on a deferred update policy and an incremental computation of the DFT at different update speeds. However, the maintenance of the R*-tree for the whole streaming series might cause a constantly growing overhead and the latter could result in performance loss. It is also possible to use indexing methods to speed up DTW calculation; however, it induces a tradeoff between efficiency and I/O cost. However, [Shieh and Keogh 2008] recently showed that for datasets that are large enough, the benefits of using DTW instead of Euclidean distance is almost null, as the larger the dataset, the higher the probability to find an exact match for any time series. They proposed an extension of the SAX representation – called indexable SAX (iSAX) – allowing to index time series with zero overlap at leaf nodes.

5. RESEARCH TRENDS AND ISSUES

Time series data mining has been an ever growing and stimulating field of study that has continuously raised challenges and research issues over the past decade. We discuss in the following open research issues and trends in time series data mining for the next decade.

Stream analysis. The last years of research in hardware and network research has witnessed an explosion of streaming technologies with the continuous advances of bandwidth capabilities. Streams are seen as continuously generated measurements which have to be processed in massive and fluctuating data rates. Analyzing and mining such data flows are computationally extreme tasks. Several papers review research issues for data streams mining [Gaber et al. 2005] or management [Golab and Ozsu 2003]. Algorithms designed for static datasets have usually not been sufficiently optimized to be capable of handling such continuous volumes of data. Many models have already been extended to control data streams, such as clustering [Domingos and Hulten 2000], classification [Hulten et al. 2001], segmentation [Keogh et al. 2003] or anomaly detection [Chuah and Fu 2007]. Novel techniques will be required and they should be designed specifically to cope with the ever flowing data streams.

Convergence and hybrid approaches. A lot of new tasks can be derived through a relatively easy combination of the already existing tasks. For instance, [Lian and Chen 2007] proposed three approaches, polynomial, DFT and probabilistic, to predict the unknown values that have not fed into the system and answer queries based on forecast data. This approach is a combination of prediction (cf. section 3.5) and query by content (cf. section 3.1) over data streams. This work shows that future research has to rely on the convergence of several tasks. This could potentially lead to powerful hybrid approaches.

Embedded systems and resource-constrained environments. With the advances in hardware miniaturization, new requirements are imposed on analysis techniques and algorithms. Two main types of constraints should absolutely be met when hardware is inherently limited. First, embedded systems have a very limited memory space and cannot have permanent access to it. However, most method use disk-resident data to analyze any incoming informations. Furthermore, sensor networks (which are frequently used in embedded systems) usually generate huge amounts of streaming data. So there is a vital need to design space efficient techniques, in terms of memory consumption as well as number of accesses. An interesting solution has been recently proposed in [Ye et al. 2009]. The algorithm is termed *autocannibalistic*, meaning that it is able to dynamically delete parts of itself to make room for new data. Second, as these resource-constrained environments are often required to be autonomous, minimizing energy consumption is another vital requirement. [Bhargava et al. 2003] has shown that sending measurements to a central site in order to process huge amounts of data is energy inefficient and lack scalability.

Data mining theory and formalization. A formalization of data mining would drastically enhance potential reasoning on design and development of algorithms through the use of a solid mathematical foundation. [Faloutsos and Megalooikonomou 2007] examined the possibility of a more general theory of data mining that could be as useful as relational algebra is for database theory. They studied the link between data mining and Kolmogorov complexity by showing their close relatedness. They conclude from the undecidability of the latter that data mining will never be automated, and therefore stating that "*data mining will always be an art*". However, a mathematical formalization could lead to global improvements of both reasoning and the evaluation of future research in this topic.

Parameter-free data mining. One of the major problems affecting time series systems is the large numbers of parameters induced by the method. The user is usually forced to "fine-tune" the settings in order to obtain best performances. However, this tuning highly depends on the dataset and parameters are not likely to be explicit. Thus, parameter-free systems is one of the key issue that has to be addressed. [Keogh et al. 2004] proposed a first step in this direction by introducing a

compression-based algorithm which does not require any parameter. As underlined by [Faloutsos and Megalooikonomou 2007], this approach could lead to elegant solutions free from the parameter setting problem.

User interaction. Time series data mining is starting to be highly dedicated to application specific systems. The ultimate goal of such methods is to mine for higher-order knowledge and propose a set of solutions to the user. It could therefore seem natural to include an user interaction scheme to allow for dynamic exploration and refinement of the solutions. An early proposal by [Keogh and Pazzani 1998] allows for relevance feedback in order to improve the querying process. From the best results of a query, the user is able to assign positive or negative influences to the series. A new query is then created by merging the series with respect to the user factors on which the system iterates. Few systems have tried to follow the same direction. However, an interactive mining environment allowing dynamic user exploration could increase the accessibility and usability of such systems.

Exhaustive benchmarking. A wide range of systems and algorithms has been proposed over the past few years. Individual proposals are usually submitted together with specific datasets and evaluation methods that prove the superiority of the new algorithm. As noted by [Keogh and Kasetty 2002], selecting those datasets may lead to *data bias* and showed that the performance of time series systems is highly data-dependent. The superiority of an algorithm should be tested with a whole range of datasets provided by various fields [Ding et al. 2008]. There is still a need for a common and exhaustive benchmarking system to perform objective testing. Another highly challenging task is to develop a procedure for real-time accuracy evaluation procedure. This could provide a measure of the accuracy achieved, thus allowing to interact with the system in real-time to improve its performance.

Adaptive mining algorithm dynamics. Users are not always interested in the results of a simple mining task and prefer to focus on evolution of these results in time. This actually represents the *dynamics* of a time series data mining system. This kind of study is of particular relevance in the context of data streams. [Dong et al. 2003] studied what are the distinctive features of analyzing streams are, rather than other kinds of data. They argued that one of the core issues is to mine *changes* in data streams. As they are of constantly evolving nature, a key aspect of the analysis of such data is to establish how an algorithm is able to adapt dynamically to such continuous changes. Furthermore, this could lead to ranking changes on the basis of relevance measures and contribute to the elaboration of methods to summarize and represent changes in the system. By finding a way to measure an approximate accuracy in real-time, it should be possible to imagine more "morphable" algorithms that could adapt dynamically to the nature of the data available on the basis of their own performances.

Link to shape analysis. Shape analysis has also been matter for discussion over the past few years. There is an astonishing resemblance between the tasks that have been examined; such as query by content [Berretti et al. 2000], classification [Kauppinen et al. 1995], clustering [Liew et al. 2000], segmentation [Sebastian et al. 2003] and even motif discovery [Xi et al. 2007]. As a matter of fact, there is a deeper connection between these two fields as recent work shows the numerous inherent link existing between these. [Barone et al. 2009] studied the problem of classifying ordered sequences of digital images. When focusing on a given pixel, it is possible to extract the time series representing the evolution of the information it contains. As this series is morphologically related to the series of the neighboring pixels, it is possible to perform a classification and segmentation based on this information. As presented above, [Ye and Keogh 2009] proposed to extract a time series from the contour of an image. They introduced the time series shapelets that represents the most informative part of an image and allows to easily discriminate between image classes. We can see from these works that both fields could benefit from each other. Even if only modest progress has been made in that direction, a convergence of both approaches could potentially lead to powerful systems.

6. CONCLUSION

After almost two decades of research in time series data mining, an incredible wealth of systems and algorithms has been proposed. The ubiquitous nature of time series led to an extension of the scope of applications simultaneously with the development of more mature and efficient solutions to deal with problems of increasing computational complexity. Time series data mining techniques are currently applied to an incredible diversity of fields ranging from economy, medical surveillance, climate forecasting to biology, hydrology, genetics, or musical querying. Numerous facets of complexity emerge with the analysis of time series, due to the high dimensionality of such data, in combination with the difficulty to define an adequate similarity measure based on human perception.

We have reviewed throughout this paper the field of time series data mining by first giving an overview of the tasks that have occupied most of the research devoted to this topic. We then presented the three core implementation components that constitute most of time series systems, namely representation techniques, similarity measures and indexing methods. We then proposed a categorization of each aspect in order to classify the existing literature. By formalizing four types of robustness, we were able to compare existing similarity measures and provided general guidelines for choosing the best fit similarity according to the nature of analyzed data as well as the desired types of robustness.

As for most scientific research, trying to find the solution to a problem often leads to raising more questions than finding answers. We have thus outlined several trends and research directions as well as open issues for the near future. The topic of time series data mining still raises a set of open questions and the interest of such research sometimes lies more in the open questions than the answers that could be provided.

ACKNOWLEDGMENT

We wish to thank Prof. Jean Claude Lejosne, Professor of English for Special Purposes (ESP) for having improved the English wording of the manuscript.

Literatur

ABONYI, J., FELL, B., NEMETH, S., AND ARVA, P. 2003. Fuzzy clustering based segmentation of time-series. In *Proceedings of the 5th International Symposium on Intelligent Data Analysis, IDA 2003, August 28-30*. Springer-Verlag, New York Inc, Berlin, Germany, 275–285.

AGRAWAL, R., FALOUTSOS, C., AND SWAMI, A. 1993. Efficient Similarity Search In Sequence Databases. In *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*. Springer, Chicago, Illinois, USA, 69–84.

AGRAWAL, R., LIN, K.-I., SAWHNEY, H. S., AND SHIM, K. 1995. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *VLDB '95: Proceedings of the 21th International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 490–501.

AHMED, N., ATIYA, A., EL GAYAR, N., EL-SHISHINY, H., AND GIZA, E. 2009. An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews 29,* 5, 594–621.

AHMED, T., ORESHKIN, B., AND COATES, M. 2007. Machine learning approaches to network anomaly detection. In *Proceedings of the 2nd USENIX workshop on Tackling computer systems problems with machine learning techniques*. USENIX Association, Cambridge, MA, USA, 1–6.

AN, J., CHEN, H., FURUSE, K., OHBO, N., AND KEOGH, E. 2003. Grid-based indexing for large time series databases. *Intelligent Data Engineering and Automated Learning, Lecture Notes in Computer Science 1983,* 1, 614–621.

ANTUNES, C. AND OLIVEIRA, A. 2001. Temporal data mining: An overview. In *KDD Workshop on Temporal Data Mining*. San Francisco, CA, USA, 1–13.

ARGYROS, T. AND ERMOPOULOS, C. 2003. Efficient subsequence matching in time series databases under time and amplitude transformations. In *3rd IEEE International Conference on Data Mining*. 481–484.

ASSENT, I., KRIEGER, R., AFSCHARI, F., AND SEIDL, T. 2008. The TS-tree: efficient time series search and retrieval. In *Proceedings of the 11th International Conference on Extending Database Technology*. 25–29.

ASSENT, I., WICHTERICH, M., KRIEGER, R., KREMER, H., AND SEIDL, T. 2009. Anticipatory DTW for efficient similarity search in time series databases. *Proceedings of the VLDB Endowment 2,* 1, 826–837.

Aßfalg, J., Kriegel, H., Kroger, P., Kunath, P., Pryakhin, A., and Renz, M. 2006. Similarity search on time series based on threshold queries. In *Advances in database technology: EDBT 2006: 10th International Conference on Extending Database Technology, March 26-31*. Vol. 3896. Springer-Verlag New York Inc, Munich, Germany, 276.

Aßfalg, J., Kriegel, H., Kröger, P., Kunath, P., Pryakhin, A., and Renz, M. 2008. Similarity search in multimedia time series data using amplitude-level features. In *Proceedings of the 14th international conference on Advances in multimedia modeling*. Springer-Verlag, 123–133.

Bagnall, A. and Janacek, G. 2005. Clustering time series with clipped data. *Machine Learning 58,* 2, 151–178.

Bagnall, A., Janacek, G., De la Iglesia, B., and Zhang, M. 2003. Clustering time series from mixture polynomial models with discretised data. In *Proceedings of the 2nd Australasian Data Mining Workshop*. 105–120.

Bagnall, A., Ratanamahatana, C., Keogh, E., Lonardi, S., and Janacek, G. 2006. A bit level representation for time series data mining with shape based similarity. *Data mining and knowledge discovery 13,* 1, 11–40.

Bai, J. and Ng, S. 2008. Forecasting economic time series using targeted predictors. *Journal of Econometrics 146,* 2, 304–317.

Bakshi, B. and Stephanopoulos, G. 1994. Representation of process trends–IV. Induction of real-time patterns from operating data for diagnosis and supervisory control. *Computers & Chemical Engineering 18,* 4, 303–332.

Bakshi, B. and Stephanopoulos, G. 1995. Reasoning in time: Modeling, analysis, and pattern recognition of temporal process trends. *Advances in Chemical Engineering 22*, 485–548.

Bandera, J., Marfil, R., Bandera, A., Rodríguez, J., Molina-Tanco, L., and Sandoval, F. 2009. Fast gesture recognition based on a two-level representation. *Pattern Recognition Letters 30,* 13, 1181–1189.

Barone, P., Carfora, M., and March, R. 2009. Segmentation, Classification and Denoising of a Time Series Field by a Variational Method. *Journal of Mathematical Imaging and Vision 34,* 2, 152–164.

Barreto, G. 2007. Time Series Prediction with the Self-Organizing Map: A Review. *Perspectives of neural-symbolic integration 77,* 1, 135–158.

Bartolini, I., Ciaccia, P., and Patella, M. 2005. Warp: Accurate retrieval of shapes using phase of fourier descriptors and time warping distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27,* 1, 142–147.

Bayer, R. and McCreight, E. 1972. Organization and maintenance of large ordered indexes. *Acta informatica 1,* 3, 173–189.

Beckmann, N., Kriegel, H., Schneider, R., and Seeger, B. 1990. The R*-tree: an efficient and robust access method for points and rectangles. *ACM SIGMOD Record 19,* 2, 322–331.

Berchtold, S., Keim, D., and Kriegel, H. 2002. The X-tree: An index structure for high-dimensional data. *Readings in multimedia computing and networking 4,* 1, 451–463.

Berkhin, P. 2006. A survey of clustering data mining techniques. *Grouping Multidimensional Data*, 25–71.

Berndt, D. and Clifford, J. 1994. Using dynamic time warping to find patterns in time series. In *AAAI-94 workshop on knowledge discovery in databases*. 229–248.

Berretti, S., Del Bimbo, A., and Pala, P. 2000. Retrieval by shape similarity with perceptual distance and effective indexing. *IEEE Transactions on multimedia 2,* 4, 225–239.

Bhargava, R., Kargupta, H., and Powers, M. 2003. Energy consumption in data analysis for on-board and distributed applications. In *Proceedings of the ICML*. Vol. 3.

Bicego, M., Murino, V., and Figueiredo, M. 2003. Similarity-based clustering of sequences using hidden Markov models. *Lecture Notes in Computer Science 2743*, 95–104.

Bohm, C., Berchtold, S., and Keim, D. 2001. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys 33,* 3, 322–373.

Bollobas, B., Das, G., Gunopulos, D., and Mannila, H. 1997. Time-series similarity problems and well-separated geometric sets. In *Proceedings of the 13th symposium on computational geometry*. 454–456.

Box, G., Jenkins, G., and Reinsel, G. 1976. *Time series analysis: forecasting and control*. Holden-day San Francisco.

Brockwell, P. and Davis, R. 2002. *Introduction to time series and forecasting*. Springer Verlag.

Brockwell, P. and Davis, R. 2009. *Time series: theory and methods*. Springer Verlag.

Buhler, J. and Tompa, M. 2002. Finding motifs using random projections. *Journal of computational biology 9,* 2, 225–242.

Burkom, H., Murphy, S., and Shmueli, G. 2007. Automated time series forecasting for biosurveillance. *Statistics in Medicine 26,* 22, 4202–4218.

Cai, Y. and Ng, R. 2004. Indexing spatio-temporal trajectories with Chebyshev polynomials. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. ACM, Paris, France, 599–610.

Cao, L. and Tay, F. 2009. Feature selection for support vector machines in financial time series forecasting. *Intelligent Data Engineering and Automated Learning. Lecture Notes in Computer Science 1983*, 41–65.

Chakrabarti, K. and Mehrotra, S. 1999. The hybrid tree: an index structure for high dimensional feature spaces. In *Data Engineering, 1999. Proceedings., 15th International Conference on*. 440–447.

CHAN, F., FU, A., AND YU, C. 2003. Haar wavelets for efficient similarity search of time-series: with and without time warping. *IEEE Transactions on knowledge and data engineering 15,* 3, 686–705.

CHAN, K. AND FU, A. 1999. Efficient time series matching by wavelets. In *Proceedings of the 15th IEEE International conference on data engineering*. Sydney, Australia, 126 – 133.

CHANDOLA, V., BANERJEE, A., AND KUMAR, V. 2009. Anomaly detection: A survey. *ACM Computing Surveys 41,* 3, 15.

CHAPPELIER, J. AND GRUMBACH, A. 1996. A Kohonen map for temporal sequences. In *Proceedings of the Conference on Neural Networks and Their Applications.* 104–110.

CHEN, L. AND NG, R. 2004. On the marriage of Lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 792–803.

CHEN, L., OZSU, M., AND ORIA, V. 2005. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. ACM, Baltimore, Maryland, USA, 491–502.

CHEN, Q., CHEN, L., LIAN, X., LIU, Y., AND YU, J. 2007. Indexable PLA for efficient similarity search. In *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, 435–446.

CHEN, X., KWONG, S., AND LI, M. 2000. A compression algorithm for DNA sequences and its applications in genome comparison. In *Proceedings of the fourth annual international conference on Computational molecular biology*. 107.

CHEN, X. AND ZHAN, Y. 2008. Multi-scale anomaly detection algorithm based on infrequent pattern of time series. *Journal of Computational and Applied Mathematics 214,* 1, 227–237.

CHEN, Y., NASCIMENTO, M., OOI, B., AND TUNG, A. 2007. Spade: On shape-based pattern detection in streaming time series. In *IEEE 23rd International Conference on Data Engineering, 2007*. 786–795.

CHHIENG, V. AND WONG, R. 2010. Adaptive distance measurement for time series databases. *Lecture Notes in Computer Science 4443*, 598–610.

CHIU, B., KEOGH, E., AND LONARDI, S. 2003. Probabilistic discovery of time series motifs. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, Washington, D.C, USA, 493–498.

CHUAH, M. AND FU, F. 2007. ECG anomaly detection via time series analysis. In *Frontiers of High Performance Computing and Networking ISPA 2007 Workshops*. Springer, 123–135.

CORDUAS, M. AND PICCOLO, D. 2008. Time series clustering and classification by the autoregressive metric. *Computational statistics & data analysis 52,* 4, 1860–1872.

CORMODE, G., MUTHUKRISHNAN, S., AND ZHUANG, W. 2007. Conquering the divide: Continuous clustering of distributed data streams. In *IEEE 23rd International Conference on Data Engineering, 2007*. 1036–1045.

COSTA SANTOS, C., BERNARDES, J., VITANYI, P., AND ANTUNES, L. 2006. Clustering fetal heart rate tracings by compression. In *19th International Symposium on Computer-Based Medical Systems*. 685–690.

DAS, G., GUNOPULOS, D., AND MANNILA, H. 1997. Finding similar time series. In *Principles of data mining and knowledge discovery: First European Symposium, PKDD'97, June 24-27*. Vol. 1263. Springer Verlag, Trondheim, Norway, 88–100.

DEGLI ESPOSTI, M., FARINELLI, C., AND MENCONI, G. 2009. Sequence distance via parsing complexity: Heartbeat signals. *Chaos, Solitons & Fractals 39,* 3, 991–999.

DENG, K., MOORE, A., AND NECHYBA, M. 1997. Learning to recognize time series: combining ARMA models with memory-based learning. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, 1997. CIRA'97*. 246–251.

DENTON, A. 2005. Kernel-density-based clustering of time series subsequences using a continuous random-walk noise model. In *Proceedings of the fifth IEEE International Conference on Data Mining*. 122–129.

DING, H., TRAJCEVSKI, G., SCHEUERMANN, P., WANG, X., AND KEOGH, E. 2008. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment 1,* 2, 1542–1552.

DOMINGOS, P. AND HULTEN, G. 2000. Mining high-speed data streams. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 71–80.

DONG, G., HAN, J., LAKSHMANAN, L., PEI, J., WANG, H., AND YU, P. 2003. Online mining of changes from data streams: Research problems and preliminary results. In *Proceedings of the 2003 ACM SIGMOD Workshop on Management and Processing of Data Streams*. San Diego, CA.

FALOUTSOS, C. AND MEGALOOIKONOMOU, V. 2007. On data mining, compression, and kolmogorov complexity. *Data Mining and Knowledge Discovery 15,* 1, 3–20.

FALOUTSOS, C., RANGANATHAN, M., AND MANOLOPULOS, Y. 1994. Fast subsequence matching in time-series databases. *SIGMOD Record 23*, 419 – 429.

FERREIRA, P., AZEVEDO, P., SILVA, C., AND BRITO, R. 2006. Mining approximate motifs in time series. In *Lecture Notes in Computer Science*. Vol. 4265. Springer, 89–101.

FLANAGAN, J. 2003. A non-parametric approach to unsupervised learning and clustering of symbol strings and sequences. In *Proceedings of the 4th Workshop on Self-Organizing Maps (WSOM03)*. 128–133.

FRENTZOS, E., GRATSIAS, K., AND THEODORIDIS, Y. 2007. Index-based most similar trajectory search. In *IEEE 23rd International Conference on Data Engineering, 2007. ICDE 2007*. 816–825.

FRÖHWIRTH-SCHNATTER, S. AND KAUFMANN, S. 2008. Model-based clustering of multiple time series. *Journal of Business and Economic Statistics 26,* 1, 78–89.

FU, A., KEOGH, E., LAU, L., RATANAMAHATANA, C., AND WONG, R. 2008. Scaling and time warping in time series querying. *The VLDB Journal - The International Journal on Very Large Data Bases 17,* 4, 921.

FUCHS, E., GRUBER, T., PREE, H., AND SICK, B. 2010. Temporal data mining using shape space representations of time series. *Neurocomputing 74,* 1-3, 379–393.

GABER, M., ZASLAVSKY, A., AND KRISHNASWAMY, S. 2005. Mining data streams: a review. *ACM Sigmod Record 34,* 2, 18–26.

GAFFNEY, S. AND SMYTH, P. 1999. Trajectory clustering with mixtures of regression models. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 63–72.

GE, X. AND SMYTH, P. 2000. Deformable Markov model templates for time-series pattern matching. In *Proceedings of the 6th ACM International conference on Knowledge Discovery and Data Mining*. 81–90.

GEURTS, P. 2001. Pattern extraction for time series classification. In *Proceedings of the 5th European conference on principles of data mining and knowledge discovery*. Freiburg, Germany, 115 – 127.

GOLAB, L. AND OZSU, M. 2003. Issues in data stream management. *ACM Sigmod Record 32,* 2, 5–14.

GOLDIN, D. AND KANELLAKIS, P. 1995. On similarity queries for time-series data: Constraint specification and implementation. In *Principles and Practice of Constraint Programming - CP95*. Springer, 137–153.

GOLDIN, D., MILLSTEIN, T., AND KUTLU, A. 2004. Bounded similarity querying for time-series data. *Information and Computation 194,* 2, 203–241.

GULLO, F., PONTI, G., TAGARELLI, A., AND GRECO, S. 2009. A time series representation model for accurate and fast similarity detection. *Pattern Recognition 42,* 11, 2998–3014.

GUPTA, S., RAY, A., AND KELLER, E. 2007. Symbolic time series analysis of ultrasonic data for early detection of fatigue damage. *Mechanical Systems and Signal Processing 21,* 2, 866–884.

GUSFIELD, D. 1997. *Algorithms on strings, trees, and sequences: computer science and computational biology*. Cambridge Univ Pr.

HAN, J. AND KAMBER, M. 2006. *Data mining: concepts and techniques*. Morgan Kaufmann.

HARRIS, R. AND SOLLIS, R. 2003. *Applied time series modelling and forecasting*. J. Wiley.

HEBRAIL, G. AND HUGUENEY, B. 2000. Symbolic representation of long time-series. In *Symbolic Data Analysis at the 4th European Conference on Principles of Data Mining and Knowledge Discovery*. 56–65.

HELLERSTEIN, J., KOUTSOUPIAS, E., AND PAPADIMITRIOU, C. 1997. On the analysis of indexing schemes. In *Proceedings of the 16th ACM Symposium on Principles of Database Systems*. 249–256.

HERRERA, L., POMARES, H., ROJAS, I., GUILLÉN, A., PRIETO, A., AND VALENZUELA, O. 2007. Recursive prediction for long term time series forecasting using advanced models. *Neurocomputing 70,* 16-18, 2870–2880.

HIMBERG, J., KORPIAHO, K., TIKANMAKI, J., AND TOIVONEN, H. 2001. Time series segmentation for context recognition in mobile devices. In *Proceedings of the 1st IEEE International Conference on Data Mining*. 203–210.

HIMBERG, J., MANTYJARVI, J., AND KORPIPAA, P. 2001. Using PCA and ICA for exploratory data analysis in situation awareness. In *Proceedings of the International Conference on Multisensor Fusion and Integration for Intelligent Systems*. 127–131.

HUANG, Y. AND YU, P. 1999. Adaptive query processing for time-series data. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 282–286.

HULTEN, G., SPENCER, L., AND DOMINGOS, P. 2001. Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 97–106.

INDYK, P., KOUDAS, N., AND MUTHUKRISHNAN, S. 2000. Identifying representative trends in massive time series data sets using sketches. In *Proceedings of the 26th International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers Inc., 363–372.

JANACEK, G., BAGNALL, A., AND POWELL, M. 2005. A likelihood ratio distance measure for the similarity between the Fourier transform of time series. *Lecture Notes in Computer Science 3518*, 737–743.

JENG, S. AND HUANG, Y. 2008. Time Series Classification Based on Spectral Analysis. *Communications in Statistics-Simulation and Computation 37,* 1, 132–142.

KALPAKIS, K., GADA, D., AND PUTTAGUNTA, V. 2001. Distance measures for effective clustering of ARIMA time-series. In *Proceedings of the IEEE International Conference on Data Mining*. 273–280.

KAUPPINEN, H., SEPPANEN, T., AND PIETIKAINEN, M. 1995. An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence 17,* 2, 201–207.

KEHAGIAS, A. 2004. A hidden Markov model segmentation procedure for hydrological and environmental time series. *Stochastic Environmental Research and Risk Assessment 18,* 2, 117–130.

KEOGH, E., CHAKRABARTI, K., AND PAZZANI, M. 2001. Locally adaptive dimensionality reduction for indexing large time series databases. In *Proceedings of ACM conference on management of data*. 151 – 162.

KEOGH, E., CHAKRABARTI, K., PAZZANI, M., AND MEHROTRA, S. 2001. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems 3,* 3, 263–286.

KEOGH, E., CHU, S., HART, D., AND PAZZANI, M. 2003. Segmenting time series: A survey and novel approach. *Data mining in time series databases*, 1–21.

KEOGH, E. AND KASETTY, S. 2002. On the need for time series data mining benchmarks : a survey and empirical demonstration. In *Proceedings of the 8th ACM SIGKDD International conference on knowledge discovery and data mining*. Edmonton, Alberta, Canada, 102 – 111.

KEOGH, E. AND KASETTY, S. 2003. On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Mining and Knowledge Discovery 7,* 4, 349–371.

KEOGH, E., LIN, J., LEE, S., AND HERLE, H. 2007. Finding the most unusual time series subsequence: algorithms and applications. *Knowledge and Information Systems 11,* 1, 1–27.

KEOGH, E., LIN, J., AND TRUPPEL, W. 2003. Clustering of time series subsequences is meaningless: implications for previous and future research. In *3rd IEEE International Conference on Data Mining*. 115–122.

KEOGH, E., LONARDI, S., AND RATANAMAHATANA, C. 2004. Towards parameter-free data mining. In *Proceedings of 10th ACM international conference on Knowledge discovery and data mining*. 206–215.

KEOGH, E. AND PAZZANI, M. 1998. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *Proceedings of the 4th International Conference of Knowledge Discovery and Data Mining*. AAAI Press, 239–241.

KEOGH, E. AND RATANAMAHATANA, C. 2005. Exact indexing of dynamic time warping. *Knowledge and Information Systems 7,* 3, 358–386.

KERR, G., RUSKIN, H., CRANE, M., AND DOOLAN, P. 2008. Techniques for clustering gene expression data. *Computers in Biology and Medicine 38,* 3, 283–293.

KIM, S., PARK, S., AND CHU, W. 2001. An Index-Based Approach for Similarity Search Supporting Time Warping in Large Sequence Databases. In *Proceedings of the 17th International Conference on Data Engineering*. IEEE Computer Society, 607–614.

KONTAKI, M., PAPADOPOULOS, A., AND MANOLOPOULOS, Y. 2007. Adaptive similarity search in streaming time series with sliding windows. *Data & Knowledge Engineering 63,* 2, 478–502.

KONTAKI, M., PAPADOPOULOS, A., AND MANOLOPOULOS, Y. 2009. Similarity Search in Time Series. *Handbook of Research on Innovations in Database Technologies and Applications*, 288–299.

KORN, F., JAGADISH, H., AND FALOUTSOS, C. 1997. Efficiently supporting ad hoc queries in large datasets of time sequences. In *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*. ACM, 289–300.

KOSKELA, T. 2003. Neural network methods in analysing and modelling time varying processes. Ph.D. thesis, Helsinki University of Technology Laboratory of Computational Engineering.

KUMAR, N., LOLLA, N., KEOGH, E., LONARDI, S., RATANAMAHATANA, C., AND WEI, L. 2005. Time-series bitmaps: a practical visualization tool for working with large time series databases. In *SIAM 2005 Data Mining Conference*. 531–535.

LATECKI, L., MEGALOOIKONOMOU, V., WANG, Q., LAKAEMPER, R., RATANAMAHATANA, C., AND KEOGH, E. 2005. Elastic partial matching of time series. *Knowledge Discovery in Databases*, 577–584.

LATECKI, L., WANG, Q., KOKNAR-TEZEL, S., AND MEGALOOIKONOMOU, V. 2007. Optimal subsequence bijection. In *IEEE Int. Conf. on Data Mining (ICDM)*. Omaha, USA, 565–570.

LAW, M. AND KWOK, J. 2000. Rival penalized competitive learning for model-based sequence clustering. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*. Vol. 2. 2186–2195.

LI, C., YU, P., AND CASTELLI, V. 1998. MALM: a framework for mining sequence database at multiple abstraction levels. In *Proceedings of the seventh international conference on Information and knowledge management*. ACM, 267–272.

LIAN, X. AND CHEN, L. 2007. Efficient similarity search over future stream time series. *IEEE Transactions on Knowledge and Data Engineering 20,* 1, 40–54.

LIAN, X., CHEN, L., AND WANG, B. 2010. Approximate similarity search over multiple stream time series. *Lecture Notes in Computer Science 4443*, 962–968.

LIAO, T. 2005. Clustering of time series data–a survey. *Pattern Recognition 38,* 11, 1857–1874.

LIEW, A., LEUNG, S., AND LAU, W. 2000. Fuzzy image clustering incorporating spatial continuity. *IEEE Proceedings on Vision, Image and Signal Processing 147,* 2, 185–192.

LIN, J. AND KEOGH, E. 2005. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and information systems 8,* 2, 154–177.

LIN, J., KEOGH, E., LONARDI, S., AND CHIU, B. 2003. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*. ACM New York, NY, USA, 2–11.

LIN, J., KEOGH, E., LONARDI, S., LANKFORD, J., AND NYSTROM, D. 2004. Visually mining and monitoring massive time series. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 460–469.

LIN, J. AND LI, Y. 2009. Finding structural similarity in time series data using bag-of-patterns representation. In *Scientific and Statistical Database Management: 21st International Conference, SSDBM 2009 New Orleans, La, USA, June 2-4, 2009 Proceedings*. Springer, 461–477.

LIN, T., KAMINSKI, N., AND BAR-JOSEPH, Z. 2008. Alignment and classification of time series gene expression in clinical studies. *Bioinformatics 24,* 13, 147–155.

LIU, Z., YU, J., LIN, X., LU, H., AND WANG, W. 2005. *Locating motifs in time-series data*. Springer, 343–353.

LOTTE, F., CONGEDO, M., LÉCUYER, A., LAMARCHE, F., AND ARNALDI, B. 2007. A review of classification algorithms for EEG-based brain–computer interfaces. *Journal of Neural Engineering 4*, 1–13.

LOWITZ, T., EBERT, M., MEYER, W., AND HENSEL, B. 2009. Hidden Markov Models for Classification of Heart Rate Variability in RR Time Series. In *World Congress on Medical Physics and Biomedical Engineering*. Springer, Munich, Germany, 1980–1983.

MA, J. AND PERKINS, S. 2003. Online novelty detection on temporal sequences. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 613–618.

MANNILA, H. AND SEPPNEN, J. 2001. Recognizing similar situations from event sequences. In *First SIAM Conference on Data Mining*. Chicago, IL, USA, 1–16.

MARTEAU, P. 2008. Time warp edit distance with stiffness adjustment for time series matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence 31,* 2, 306–318.

MEGALOOIKONOMOU, V., LI, G., AND WANG, Q. 2004. A dimensionality reduction technique for efficient similarity analysis of time series databases. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*. ACM, Washington, D.C., USA, 160–161.

MEGALOOIKONOMOU, V., WANG, Q., LI, G., AND FALOUTSOS, C. 2005. A multiresolution symbolic representation of time series. In *Proceedings. 21st International Conference on Data Engineering*. 668–679.

MOHAMMAD, Y. AND NISHIDA, T. 2009. Constrained Motif Discovery in Time Series. *New Generation Computing 27,* 4, 319–346.

MORSE, M. AND PATEL, J. 2007. An efficient and accurate method for evaluating time series similarity. In *Proceedings of the 2007 ACM international conference on Management of data*. 569–580.

MUEEN, A., KEOGH, E., ZHU, Q., CASH, S., AND WESTOVER, B. 2009. Exact discovery of time series motifs. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*. 473–484.

MUHAMMAD FUAD, M. AND MARTEAU, P. 2008. Extending the Edit Distance Using Frequencies of Common Characters. In *Proceedings of the 19th International Conference on Database and Expert Systems Applications*. Springer, Turin, Italy, 150–157.

NANOPOULOS, A., ALCOCK, R., AND MANOLOPOULOS, Y. 2001. Feature-based classification of time-series data. In *Information processing and technology*. 49–61.

OGRAS, Y. AND FERHATOSMANOGLU, H. 2006. Online summarization of dynamic time series data. *The VLDB Journal - The International Journal on Very Large Data Bases 15,* 1, 84–98.

OTU, H. AND SAYOOD, K. 2003. A new sequence distance measure for phylogenetic tree construction. *Bioinformatics 19,* 16, 2122–2130.

OUYANG, R., REN, L., CHENG, W., AND ZHOU, C. 2010. Similarity search and pattern discovery in hydrological time series data mining. *Hydrological Processes 24,* 9, 1198–1210.

PALPANAS, T., KEOGH, E., ZORDAN, V., GUNOPULOS, D., AND CARDLE, M. 2004. Indexing large human-motion databases. In *Proceedings of the 13th international conference on Very large data bases*. 780–791.

PALPANAS, T., VLACHOS, M., KEOGH, E., AND GUNOPULOS, D. 2008. Streaming time series summarization using user-defined amnesic functions. *IEEE Transactions on Knowledge and Data Engineering 20,* 7, 992–1006.

PALPANAS, T., VLACHOS, M., KEOGH, E., GUNOPULOS, D., AND TRUPPEL, W. 2004. Online amnesic approximation of streaming time series. In *20th International Conference on data engineering*. 338–349.

PANUCCIO, A., BICEGO, M., AND MURINO, V. 2002. A Hidden Markov Model-based approach to sequential data clustering. *Lecture Notes in Computer Science 2396*, 734–743.

PAPADIMITRIOU, S., SUN, J., AND YU, P. 2006. Local correlation tracking in time series. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*. 456–465.

PAPADIMITRIOU, S. AND YU, P. 2006. Optimal multi-scale patterns in time series streams. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*. Chicago, IL, USA, 647–658.

PARK, S., CHU, W., YOON, J., AND HSU, C. 2000. Efficient searches for similar subsequences of different lengths in sequence databases. In *Proceedings. 16th International Conference on Data Engineering*. 23–32.

PARK, S., LEE, D., AND CHU, W. 1999. Fast retrieval of similar subsequences in long sequence databases. In *In 3rd IEEE Knowledge and Data Engineering Exchange Workshop*. 60–67.

PATEL, P., KEOGH, E., LIN, J., AND LONARDI, S. 2002. Mining Motifs in Massive Time Series Databases. In *Proceedings of IEEE International Conference on Data Mining (ICDM02)*. 370–377.

PERNG, C., WANG, H., ZHANG, S., AND PARKER, D. 2000. Landmarks : a new model for similarity-based pattern querying in time series databases. In *Proceedings of the 16th International Conference on Data Engineering*. 33–42.

PESARAN, M., PETTENUZZO, D., AND TIMMERMANN, A. 2006. Forecasting time series subject to multiple structural breaks. *Review of Economic Studies 73,* 4, 1057–1084.

POPIVANOV, I. AND MILLER, R. 2002. Similarity search over time-series data using wavelets. In *Proceedings of the International Conference on Data Engineering*. 212–224.

POVINELLI, R., JOHNSON, M., LINDGREN, A., AND YE, J. 2004. Time series classification using Gaussian mixture models of reconstructed phase spaces. *IEEE Transactions on Knowledge and Data Engineering 16,* 6, 779–783.

RAFIEI, D. AND MENDELZON, A. 1998. Efficient Retrieval of Similar Time Sequences Using DFT. In *Proceedings. 5th International Conference of Foundations of Data Organization and Algorithms*. 249–257.

RATANAMAHATANA, C. AND KEOGH, E. 2004a. Everything you know about dynamic time warping is wrong. In *Third Workshop on Mining Temporal and Sequential Data*. Seattle, WA, USA, 1–11.

RATANAMAHATANA, C. AND KEOGH, E. 2004b. Making time-series classification more accurate using learned constraints. In *Proceedings of SIAM International Conference on Data Mining*. 11–22.

RATANAMAHATANA, C., KEOGH, E., BAGNALL, A., AND LONARDI, S. 2005. A novel bit level time series representation with implication of similarity search and clustering. *Advances in Knowledge Discovery and Data Mining*, 771–777.

RATANAMAHATANA, C. AND WANICHSAN, D. 2008. Stopping Criterion Selection for Efficient Semi-supervised Time Series Classification. *Studies in Computational Intelligence 149*, 1–14.

RAVI KANTH, K., AGRAWAL, D., AND SINGH, A. 1998. Dimensionality reduction for similarity searching in dynamic databases. *ACM SIGMOD Record 27,* 2, 166–176.

REEVES, G., LIU, J., NATH, S., AND ZHAO, F. 2009. Managing massive time series streams with multi-scale compressed trickles. *Proceedings of the VLDB Endowment 2,* 1, 97–108.

REINERT, G., SCHBATH, S., AND WATERMAN, M. 2000. Probabilistic and statistical properties of words: an overview. *Journal of Computational Biology 7,* 1-2, 1–46.

RODRIGUEZ, J. AND KUNCHEVA, L. 2007. Time series classification: Decision forests and SVM on interval and DTW features. In *Proc Workshop on Time Series Classification, 13th International Conference on Knowledge Discovery and Data mining*.

SAKURAI, Y., YOSHIKAWA, M., AND FALOUTSOS, C. 2005. FTW: fast similarity search under the time warping distance. In *Proceedings of the 24th ACM Symposium on Principles of database systems*. 326–337.

SAKURAI, Y., YOSHIKAWA, M., UEMURA, S., AND KOJIMA, H. 2000. The A-tree: An index structure for high-dimensional spaces using relative approximation. In *Proceedings of the 26th International Conference on Very Large Data Bases*. 516–526.

SALVADOR, S. AND CHAN, P. 2007. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis 11,* 5, 561–580.

SALVADOR, S., CHAN, P., AND BRODIE, J. 2004. Learning states and rules for time series anomaly detection. In *Proc. 17th International FLAIRS Conference*. 300–305.

SEBASTIAN, T., KLEIN, P., AND KIMIA, B. 2003. On aligning curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25,* 1, 116–125.

SEBASTIANI, P., RAMONI, M., COHEN, P., WARWICK, J., AND DAVIS, J. 1999. Discovering dynamics using Bayesian clustering. *Lecture Notes in Computer Science 1642*, 199–209.

SFETSOS, A. AND SIRIOPOULOS, C. 2004. Time series forecasting with a hybrid clustering scheme and pattern recognition. *IEEE Transactions on Systems, Man and Cybernetics, Part A 34,* 3, 399–405.

SHASHA, D. AND ZHU, Y. 2004. *High performance discovery in time series: techniques and case studies*. Springer-Verlag New York Inc.

SHATKAY, H. AND ZDONIK, S. 1996. Approximate queries and representations for large data sequences. In *Data Engineering, 1996. Proceedings of the Twelfth International Conference on*. 536–545.

SHIEH, J. AND KEOGH, E. 2008. isax : indexing and mining terabyte sized time series. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 623–631.

SMYTH, P. 1997. Clustering sequences with hidden Markov models. *Advances in Neural Information Processing Systems*, 648–654.

SONG, H. AND LI, G. 2008. Tourism demand modelling and forecasting–A review of recent research. *Tourism Management 29,* 2, 203–220.

SORJAMAA, A., HAO, J., REYHANI, N., JI, Y., AND LENDASSE, A. 2007. Methodology for long-term prediction of time series. *Neurocomputing 70,* 16-18, 2861–2869.

SRISAI, D. AND RATANAMAHATANA, C. 2009. Efficient Time Series Classification under Template Matching Using Time Warping Alignment. In *Proceedings of the Fourth International Conference on Computer Sciences and Convergence Information Technology*. IEEE, 685–690.

STIEFMEIER, T., ROGGEN, D., AND TROSTER, G. 2007. Gestures are strings: Efficient online gesture spotting and classification using string matching. In *Proceedings of the ICST 2nd international conference on Body area networks*. Florence, Italy, 1–8.

STRUZIK, Z., SIEBES, A., AND CWI, A. 1999. Measuring time series similarity through large singular features revealed with wavelet transformation. In *Proceedings of the Tenth International Workshop on Database and Expert Systems Applications*. 162–166.

SUBASI, A. 2007. EEG signal classification using wavelet feature extraction and a mixture of expert model. *Expert Systems with Applications 32,* 4, 1084–1093.

TANG, H. AND LIAO, S. 2008. Discovering original motifs with different lengths from time series. *Knowledge-Based Systems 21,* 7, 666–671.

TSAY, R. 2005. *Analysis of financial time series*. Wiley-Interscience.

VASKO, K. AND TOIVONEN, H. 2002. Estimating the number of segments in time series data using permutation tests. In *Proceedings of the IEEE International Conference on Data Mining*. 466–473.

VLACHOS, M., GUNOPOULOS, D., AND KOLLIOS, G. 2002. Discovering similar multidimensional trajectories. In *Proceedings of the 18th International Conference on Data Engineering*. IEEE Computer Society, 673–684.

VLACHOS, M., GUNOPULOS, D., AND DAS, G. 2004. Indexing time-series under conditions of noise. *Data mining in time series databases*, 67–100.

VLACHOS, M., HADJIELEFTHERIOU, M., GUNOPULOS, D., AND KEOGH, E. 2006. Indexing multidimensional time-series. *The VLDB Journal 15,* 1, 1–20.

VLACHOS, M., LIN, J., KEOGH, E., AND GUNOPULOS, D. 2003. A wavelet-based anytime algorithm for k-means clustering of time series. In *Proc. Workshop on Clustering High Dimensionality Data and Its Applications*. 23–30.

VLACHOS, M., YU, P., AND CASTELLI, V. 2005. On periodicity detection and structural periodic similarity. In *SIAM International Conference on Data Mining*. Newport Beach, CA, 449–460.

WAGNER, N., MICHALEWICZ, Z., KHOUJA, M., AND MCGREGOR, R. 2007. Time series forecasting for dynamic environments: the DyFor genetic program model. *IEEE transactions on evolutionary computation 11,* 4, 433–452.

WEIGEND, A. AND GERSHENFELD, N. 1994. *Time Series Prediction: forecasting the future and understanding the past*. Addison Wesley.

WEISS, G. 2004. Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter 6,* 1, 7–19.

XI, X., KEOGH, E., SHELTON, C., WEI, L., AND RATANAMAHATANA, C. 2006. Fast time series classification using numerosity reduction. In *Proceedings of the 23rd international conference on Machine learning*. 1040.

XI, X., KEOGH, E., WEI, L., AND MAFRA-NETO, A. 2007. Finding Motifs in Database of Shapes. In *Proc. of SIAM International Conference on Data Mining*. Minneapolis, Minnesota, USA, 249–260.

XIE, J. AND YAN, W. 2007. Pattern-based characterization of time series. *International Journal of Information and Systems Science 3,* 3, 479–491.

XIONG, Y. AND YEUNG, D. 2004. Time series clustering with ARMA mixtures. *Pattern Recognition 37,* 8, 1675–1689.

YADAV, R., KALRA, P., AND JOHN, J. 2007. Time series prediction with single multiplicative neuron model. *Applied soft computing 7,* 4, 1157–1163.

YANKOV, D., KEOGH, E., MEDINA, J., CHIU, B., AND ZORDAN, V. 2007. Detecting time series motifs under uniform scaling. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 844–853.

YANKOV, D., KEOGH, E., AND REBBAPRAGADA, U. 2008. Disk aware discord discovery: Finding unusual time series in terabyte sized datasets. *Knowledge and Information Systems 17,* 2, 241–262.

YE, D., WANG, X., KEOGH, E., AND MAFRA-NETO, A. 2009. Autocannibalistic and Anyspace Indexing Algorithms with Applications to Sensor Data Mining. In *The SIAM International Conference on Data Mining (SDM 2009)*. Sparks, Nevada, 85–96.

YE, L. AND KEOGH, E. 2009. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 947–956.

YI, B. AND FALOUTSOS, C. 2000. Fast time sequence indexing for arbitrary Lp norms. In *Proceedings of the 26th International Conference on Very Large Data Bases*. 385–394.

YI, B., JAGADISH, H., AND FALOUTSOS, C. 1998. Efficient retrieval of similar time sequences under time warping. In *Data Engineering, 1998. Proceedings., 14th International Conference on*. 201–208.

YOON, H., YANG, K., AND SHAHABI, C. 2005. Feature subset selection and feature ranking for multivariate time series. *IEEE transactions on knowledge and data engineering*, 1186–1198.

YPMA, A. AND DUIN, R. 1997. Novelty detection using self-organizing maps. *Progress in Connectionist-Based Information Systems 2*, 1322–1325.

ZHAN, Y., CHEN, X., AND XU, R. 2007. Outlier detection algorithm based on pattern representation of time series. *Application Research of Computers 24,* 11, 96–99.

ZHANG, X., WU, J., YANG, X., OU, H., AND LV, T. 2009. A novel pattern extraction method for time series classification. *Optimization and Engineering 10,* 2, 253–271.

ZHONG, S. AND GHOSH, J. 2002. HMMs and coupled HMMs for multi-channel EEG classification. In *Proceedings of the IEEE International Joint Conference on Neural Networks*. 1154–1159.

ZHONG, S., KHOSHGOFTAAR, T., AND SELIYA, N. 2007. Clustering-based network intrusion detection. *International Journal of Reliability Quality and Safety Engineering 14,* 2, 169–187.

Received Month Year; revised Month Year; accepted Month Year