Time Series Data Mining Methods: A Review

Master's Thesis submitted

to

Prof. Dr. Wolfgang Karl Härdle

Humboldt-Universität zu Berlin

School of Business and Economics

Ladislaus von Bortkiewicz Chair of Statistics

by

Caroline Kleist

(533039)

In partial fulfillment of the requirements for the degree of

Master of Science in Statistics

Berlin, March 25, 2015

Abstract

Today, real world time series data sets can take a size up to a trillion observations and even more. Data miners' task is it to detect new information that is hidden in this massive amount of data. While well known techniques for data mining in cross sections have been developed, time series data mining methods are not as sophisticated and established yet. Large time series bring along problems like very high dimensionality and up to today, researchers haven't agreed on best practices in this regard.

This review gives an overview of the challenges of large time series and the proposed problem solving approaches from time series data mining community. We illustrate the most important techniques with Google trends data. Moreover, we review current research directions and point out open research questions.

Heutzutage sind die Möglichkeiten der Datensammlung und -Speicherung unvorstellbar weitreichend und somit können Zeitreihendatensätze mittlerweile bis zu einer Billion Beobachtungen enthalten. Die Aufgabe von Data Mining ist es, versteckte Informationen aus dieser Datenschwemme herauszufiltern. Während es für Querschnittsdaten viele verschiedene und sehr gut entwickelte Techniken gibt, hinken die Zeitreihen Data Mining Methoden weit hinterher. Die Forschungspraxis hat sich in diesem Bereich noch nicht auf standardisierte Vorgehensweisen geeinigt.

Dieser Literaturüberblick stellt zunächst die typischen Probleme, die Zeitreihen mit sich bringen, dar und systematisiert daraufhin die von der Forschungsgemeinde vorgeschlagenen Lösungsansätze hierfür. Die wichtigsten Ansätze werden anhand von Google Trends Daten illustriert. Darüber hinaus werfen wir einen Blick auf aktuelle Forschungsströme und zeigen noch offene Forschungsfragen auf.

Contents

List of Abbreviations

ACF	Autocorrelation Function
AIC	Akaike Information Criterion
APCA	Adaptive Piecewise Constant Approximation
AR(I)MA	Autoregressive (Integrated) Moving Average
CDM	Compression Based Dissimilarity Measure
DFT	Discrete Fourier Transform
DTW	Dynamic Time Warping
DWT	Discrete Wavelet Transform
ECG	Electro Cardiogram
ED	Euclidean Distance
EDR	Edit Distance on Real Sequence
ERP	Edit Distance with Real Penalty
fMRI	Functional Magnetic Resonance Imaging
ForeCA	Forecastable Component Analysis
FTW	Fast Dynamic Time Warping
GIR	Global Iterative Replacement
HMM	Hidden Markov Model
ICA	Independent Component Analysis
iSAX	Indexable Symbolic Aggregate Approximation
LCSS	Longest Common Subsequence
LDA	Linear Discriminant Analysis
MCMC	Markov Chain Monte Carlo
MDL	Minimum Description Length
MVQ	Multiresolution Vector Quantized
NN	Nearest Neighbor
OGLE	Optical Gravitational Lensing Experiment
PAA	Piecewise Aggregate Approximation
PACF	Partial Autocorrelation Function
PCA	Principal Component Analysis
PFA	Predictable Feature Analysis
PIP	Perceptually Important Points
PLR	Piecewise Linear Representation

QbH	Query by Humming
SARIMA	Seasonal Autoregressive Integrated Moving Average
SAX	Symbolic Aggregation Approximation
SDL	Shape Definition Query
SOM	Self Organizing Maps
SpADe	Spatial Assembling Distance
SPIRIT	Streaming Pattern Discovery in Multiple Time Series
SVD	Singular Value Decomposition
SVDD	Singular Value Decomposition with Deltas
SVI	Search Volume Index
SVM	Support Vector Machines
SVR	Support Vector Regression
SWAB	Sliding Window And Bottom Up
TB	Terabyte
TQuEST	Threshold Query Execution
TSDM	Time Series Data Mining
TWED	Time Warp Edit Distance

List of Figures

List of Tables

1 Introduction

The buzzwords *data mining* and *big data* are of strongly growing importance and found their way into mass media a few years ago. Moreover, big data is according to Forbes magazine (IBM, 2014) even labeled as the new natural resource of the century. Considering the various sources of big data in real life, this trend is not surprising. With the vast amount and variety of data available, the capacity of manual data analysis has been exceeded longly. So, the explosion of information available brings about not only ample opportunities but raises new challenges on data analysis methods. Therefore, data mining as a set of techniques for the analysis of massive datasets is of ever increasing importance as well.

In spite of these developments, it might sound surprising that *time series data mining* is far behind cross sectional data mining techniques. While cross section techniques have been well developed, the time series data mining methods are not that sophisticated yet.

Generally speaking, data mining is the analytic process of knowledge discovery in large and complex data sets. It is a discipline at the very intersection of statistics and computer science as the search for hidden information is partly made automatic by employing computers for this task. To be more precise, data mining is the result of the hybridization of statistics, computer science, artificial intelligence and machine learning (Fayyad et al., 1996). Compared to times before computers and sensors were able to collect and store such bulky data sets, the paradigm of the methods of statistical operations has changed and partly gone into reverse. Today, a data scientist wants to find the needle in the hay and operates top down. In data mining, no a priori intended model is calibrated using known data. Mainly, data miners search for hidden information in the data like frequently recurring patterns, anomalies or natural groupings. Generally speaking, typical data mining tasks include knowledge discovery, clustering, classification, rule discovery, summarization and visualization.

In this paper, we review time series data mining methods. We present an overview of state of the art time series data mining techniques which become gradually established in the data mining community. Moreover, we point out recent and still open research areas in time series data mining. The search for real world applications and data sources yields numerous examples. One well known data giant is Google. They processes on average over 40,000 search queries per second and store over 1.2 trillion search queries per year (Internetlivestats.com, 2014). We illustrate the most popular time series data mining techniques with Google trends data (Google, 2014).

The remainder of this paper is organized as follows. Section 2 shows characteristic properties and thereof resulting challenges and problems of large time series. Section 3 discusses crucial preprocessing methods for time series including representation and indexing techniques, segmentation, visualization and similarity measures. Hereafter, Section 4 proceeds with typical data mining tasks tailored to time series: clustering, knowledge discovery, classification, data streams, rule discovery and prediction. Section 5 points out recent research directions and Section 6 highlights the broad range of applications for time series data mining. Section 7 concludes.

2 Properties and Challenges of Large Time Series

Before we come up with time series data mining methods, we itemize which problems need to be tackled. As a general rule, large time series come along with super-high dimensionality, noise along characteristic patterns, outliers and dynamism. Moreover, the most crucial challenge in time series data mining is the comparison of two or more time series which are shifted or scaled through time or in amplitude.

The problems that need to be tackled in time series data mining arise from typical properties of large time series. Firstly, as one observation of a time series is viewed as one dimension, the dimensionality of large time series is typically very high (Rakthanmanon et al., 2012). The visualization alone of time series which are larger than a several ten thousand observations can be challenging Lin et al. (2005). Working with super-high dimensional raw data can be very costly with respect to processing and storage costs (Fu, 2011). Therefore, a high level representation of the data or abstraction is required. Besides, the basic philosophy of data mining implies that avoiding a potential information loss by studying the raw data is not convenient and too slow. In the context of time series data mining, noise along characteristic patterns are additive white noise components (Esling and Agon, 2012). Provided that we are interested in global characteristics, the time series data mining techniques need to be robust against noisy components. If such massive amounts of data are collected, the sensitivity towards measurement errors and outliers can be high. At the same time, long time series enable us to better differentiate between outliers and rare outcomes. Rare outcomes which would be categorized as outliers in small subsamples help us to better understand heterogeneity (Fan et al., 2014).

Moreover, time series data mining aims at the comparison of two or more time series or subsequences. However, time series are frequently not aligned in the time axis (Berndt and

Clifford, 1994) or the amplitude (Esling and Agon, 2012). Besides temporal and amplitude shifting differences, time series can have scaling or acceleration differences while still having very similar characteristics. Time series data mining methods need to be robust against these transformations and combinations of them.

Furthermore, we up front clarify what "large" means in the context of *large* time series. The manageable dimensions can reach up to 1 trillion = $1.000,000,000,000$ time series objects. For settlement, 1 trillion time series objects need roughly 7.2 terabyte storage space. Rakthanmanon et al. (2012) include a brief discussion of *a trillion* time series object. They illustrate that a time series with one trillion observations would correspond to each and every heartbeat of a 123 year old human being.

Another compelling reason for the application of time series data mining methods is the emergence of such a massive amount of data that is too big to store. The incoming time series data is growing faster than our ability to process and store the raw data. Hence, the data has to be reduced immediately in order to achieve reasonable storage size of the data. A typical example for data that is too big to store is streaming data, or data streams. Data streams are continuously and at very fluctuating rates generated observations (Gaber et al., 2005). For instance, computer network traffic data or physical network sensors deliver such non stopping streams of information.

2.1 Streaming Data

Besides the already discussed properties of large time series data sets, continuous *stream data* brings along further challenges. Streaming data is characterized by a tremendous volume of temporally ordered observations arriving at a steady high-speed rate, it is fast changing, and potentially infinite (Gaber et al., 2005). In some cases, the entire original data stream is even too big to be stored. Typically, data mining methods require multiple scans through the data at hand. But, constantly flowing data requires single-scan and online multidimensional analysis methods for pattern and knowledge discovery. Resulting from that, the use of data reduction and indexing methods is not only necessary but inevitable. Initial research on streaming data is primarily concerned with data stream management systems and Golab and Özsu (2003) provide a review in this regard. As technological boundaries are constantly pushed outwards and even more massive and complex data sets are collected day by day, the need for data mining techniques for potentially infinite volumes of streaming data is becoming more urgent. The trade-off between storage size and accuracy is for streaming data methods even more important than for time series data mining in general. As the discussion of streaming data reaches far beyond the scope of this review, please refer to the survey by Gaber et al. (2005). Streaming data techniques are still in the early stages of development and they have a high relevance in real world applications. Therefore, this research area is labeled as the next hottest topic in time series data mining.

3 Preprocessing Methods

Before jumping into actual data mining, it is essential to preprocess the data at hand. Firstly, large time series data is often very bulky. Thus, directly dealing with such data in its raw format is expensive with respect to processing and storage costs. Secondly, we are dealing with time series which are no more comprehensible with the unaided eye in its raw format. Therefore we beforehand reduce dimensionality or segment the time series and then index them. In the light of lacking natural clarity of the raw data, visualization techniques and tools for large time series emerged and are presented here. Moreover, similarity measures are the backbone of all data mining applications and need to be discussed.

3.1 Representation

As already discussed in Section 2, large time series are super-high dimensional. Each observation of a time series is viewed as one dimension. So, in order to achieve effectiveness and efficiency in managing time series, representation techniques that reduce dimensionality of time series are crucial and still a basic problem in time series data mining. If we reduce the dimension of a time series X of original length n to $k \ll n$, we can reduce computational complexity from $\mathcal{O}(n)$ to $\mathcal{O}(k)$. At the same time, we do not want to lose too much information and aim to preserve fundamental characteristics of the data. Furthermore, we require an intuitive interpretation and visualization of the reduced data set. So, desired properties of representation approaches are dimensionality reduction, a short computing time, preservation of local and global shape characteristics and insensitivity towards noise (additive white noise components). The time series data mining community developed many different representation and indexing techniques that aim to satisfy these requirements. The approaches differ with respect to their ability to address these properties. According to the various available representation techniques, a large number of different systematizations of representation approaches exists. One common approach is to seperate representation techniques corresponding to their domain and indexing form. Many approaches transform the original data to another domain for dimensionality reduction and then apply an indexing mechanism. But, a more practically orientated systematization of representation techniques is proposed by Ding et al. (2008): For practical purposes, it is important to know whether the representation techniques at hand are *data adaptive*, *non data adaptive* or *model based.*

3.1.1 Non Data Adaptive Representation Techniques

Non data adaptive representation techniques are stiff and have always the same transformation parameters regardless the features of the data at hand. So, the transformation parameters are fixed a priori. One subgroup of non data adaptive representation techniques are operating in the frequency domain. Their logic follows the basic idea of spectral decomposition: any time series can be represented by a finite number of trigonometric functions. Generally speaking, operating in the frequency domain is valid as the Euclidean distances between two time series is the same in the time domain and in the frequency domain and hereby preserve distances. For example, *D*iscrete *F*ourier *T*ransform (DFT) as proposed by Agrawal et al. (1993a) for mining sequence databases preserves the essential characteristics of time series in the first few Fourier coefficients which are single complex numbers representing a finite number of sine and cosine waves. Only the first “strongest” coefficients of the DFT are kept for lower bounding of the actual distances.

Among others, Graps (1995), Burrus et al. (1998) and Chan and Fu (1999) point out that *D*iscrete *W*avelet *T*ransform (DWT) is an effective replacement for DFT. Opposed to the DFT representation, in DWT we consider not only the first few global shape coefficients, but we use both, coefficients representing the global shape as well as “zoom in” coefficients representing smaller, local subsections. The DWT consists of wavelets (which are functions) representing the data and is calculated by computing the differences and sum of a benchmark “mother” wavelet. Popivanov and Miller (2002) demonstrate that a large class of wavelets are applicable for time series dimension reduction. A major drawback of using wavelets is the necessity to have data with its length being an integer power of two. One popular wavelet is the so called “Haar” wavelet proposed to use in the time series data mining context by Struzik and Siebes (1999).

Other, more recently proposed non data adaptive representation techniques are especially tailored to time series data mining. The most popular approach from this group was introduced by Keogh and Pazzani (2000b) and is called *P*iecewise *A*ggregate *A*pproximation

(PAA) since the research by Keogh et al. (2001a). Originally, Keogh and Pazzani (2000b) called the PAA method *P*iecewise *C*onstant *A*pproximation (PCA) and the name was afterwards changed to PAA as the abbreviation *PCA* is already reserved for Principal Component Analysis. The PAA coefficients are generated by dividing the time series into ω equi-sized windows and calculating averages of the subsequences in the corresponding windows. The averages of each window are stacked into a vector and called "PAA coefficients". Hence, the dimension reduction of a time series X of length n into a string $\widehat{X} = \hat{x}_1, \ldots, \hat{x}_\omega$ of arbitrary length ω with $\omega \ll n$ is performed by the following transformation:

$$\hat{x}_i = d^{-1} \sum_{j=d\,(i-1)+1}^{d\,i} x_j, \qquad i = 1, \ldots, \omega, \quad j = 1, \ldots, n, \quad d = n\,w^{-1} \tag{1}$$

Figure 1 shows the PAA dimension reduction of the Google search volume index for the query term "Data Mining" with $\omega = 40$. Although the PAA dimension reduction approach is very simple, it is well comparable to more complex competitors (see for example Ding et al. (2008)). Furthermore, a very strong advantage is the fact that each PAA window is of the same length. This facilitates the indexing technique enormously (see Subsection 3.2). An extension of the PAA approach is *A*daptive *P*iecewise *C*onstant *A*pproximation (APCA) by Keogh et al. (2001b). APCA aims to approximate a time series by PAA segments of varying length, i.e. it allows the different windows to have an arbitrary size. In this way, we try to minimize the individual reconstruction error of the reduced time series. For the APCA, we do not only store the mean for each window but its length as well.

*P*rincipal *C*omponent *A*nalysis (PCA) is a further dimension reduction technique adopted from static data approaches (see for instance Yang and Shahabi (2005b) or Raychaudhuri et al. (2000)). PCA disregards less significant components and therefore gives a reduced representation of the data. PCA computations are based on orthogonal transformations with the goal to obtain linearly uncorrelated variables, the so called principal components. So, in order to reduce dimensionality, PCA requires the covariance matrix of the corresponding time series. However, for the covariance calculations it is ignored whether the considered time series are similar or correlated at different points in time. If two time series are evolving very similar and are only shifted in time, the traditional PCA approach would lead to false conclusions. To avoid this ineffectiveness, Li (2014) present an asynchronism-based principal component analysis. In order to improve PCA, Li (2014) embed *D*ynamic *T*ime *W*arping (DTW, see Section 3.5.1) before applying PCA. By doing that, they employ asynchronous and not only synchronous covariances. Furthermore, DTW does not require the time series to be of the

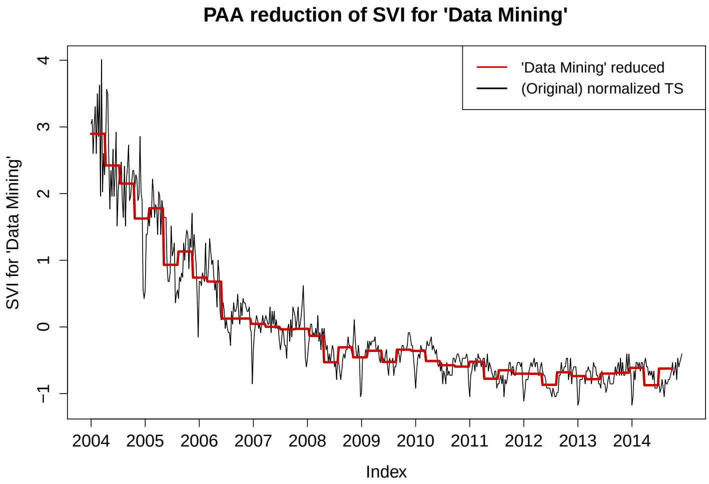

Figure 1: PAA dimension reduction of the Google search volume index (SVI) for the query term "Data Mining" with $n = 569$ and $\omega = 40$. Averages are calculated for all observations falling into one window. TSDMPaa

same length as PCA does.

Opposed to all representation methods presented so far, Ratanamahatana et al. (2005) and Bagnall et al. (2006) hold the view that we do not have to compress the raw data by aggregation of dimensions. Instead, they store a proxy replacing each real valued time series object. Namely, Ratanamahatana et al. (2005) and Bagnall et al. (2006) propose to store bits instead of all real valued time series objects and therefore obtain a so called clipped representation of the time series which needs less storage space and still has the same length n.

3.1.2 Data Adaptive Representation Techniques

Data adaptive representation techniques are (more) sensitive to the nature of the data at hand. The transformation parameters are chosen depending on the available data and not fixed a priori as for non data adaptive techniques. However, almost all non data adaptive techniques can be turned into data adaptive approaches by adding data-sensitive proceeding schemes.

The *S*ingular *V*alue *D*ecomposition (SVD) which is also known as Latent Semantic Indexing is said to be an optimal transform with respect to reconstruction error (see for example Ye (2005)). For SVD construction, we linearly combine the basis shapes best representing the original data. This is done by a global transformation of the whole dataset in order to maximize the variance carried by the axes. Finally, if we try to reconstruct the original data from the SVD representation, the error we make is relatively low. Korn et al. (1997) even aim to further minimize the reconstruction error and further improve the SVD representation. For this purpose they introduce *S*ingular *V*alue *D*ecomposition with *D*eltas (SVDD).

A very frequently used and probably the most popular dimensional reduction and indexing technique is based on a symbolic representation and called SAX. It was introduced by Lin et al. (2003). The *S*ymbolic *A*ggregate *A*pproximation (SAX) represenation is said to outperform all other dimensionality reduction techniques (see for instance Ratanamahatana et al. (2010) or Kadam and Thakore (2012)). In order to illustrate how the SAX algorithm works, Figure 2 shows the SAX representation including the PAA representation which is the preceding interim dimension reducing stage of the SAX algorithm. The illustrating example is computed for the trajectories of the Google SVI for the query terms "Data Mining" and "Google". As already indicated, the transformation of a time series X of length n into a string $\overline{X} = \overline{x}_1, \ldots, \overline{x}_\omega$ of arbitrary length ω with $\omega \ll n$ is performed in two steps. In a first step, the z-normalized time series is converted to a PAA representation, i.e. the *P*iecewise *A*ggregate *A*pproximation. As a reminder, the PAA coefficients are derived by slicing the data at hand along the temporal axis into ω equidistant windows and thereupon calculating sample means for all observations falling into one window (see Equation 1). Hereafter, the PAA coefficients are mapped to symbols (mostly letters). For doing so, the cutoff lines dividing the distribution space into α equiprobable segments need to be specified. Assuming that the z-normalized time series are standard normally distributed, these cutoff lines are tabulated for α different symbols as reported in Table 1. Hence, α is a hyperparameter that has to be chosen a priori. Equation 2 depicts how the PAA coefficients from the vector $\widehat{X} = \hat{x}_1, \ldots, \hat{x}_\omega$ are mapped into α different symbols yielding the SAX string $\overline{X} = \overline{x}_1, \ldots, \overline{x}_\omega$:

$$\overline{x}_\omega = \alpha_j \quad \text{if} \quad \hat{x}_i \in [\beta_{j-1}, \beta_j) \tag{2}$$

with $\beta_1, \ldots, \beta_i$ being the corresponding tabulated breakpoints as reported in Table 1 for $i, j = 1, \ldots, \alpha - 1$. Finally, as symbols require fewer bits than real valued numbers, symbols are stored instead of the original time series. If we chose the hyperparameter to be $\alpha = 3$, the three used symbols are "a", "b" and "c". Instead of the PAA coefficients or even the real valued time series objects, the symbols are stored in the corresponding order as a "word".

In order to speed the SAX algorithm dramatically up, Shieh and Keogh (2008) propose a modification of the SAX representation called *i*ndexable *S*ymbolic *A*ggregate *A*pproximation (iSAX). This multi-resolution symbolic representation allows extensible hashing and hence

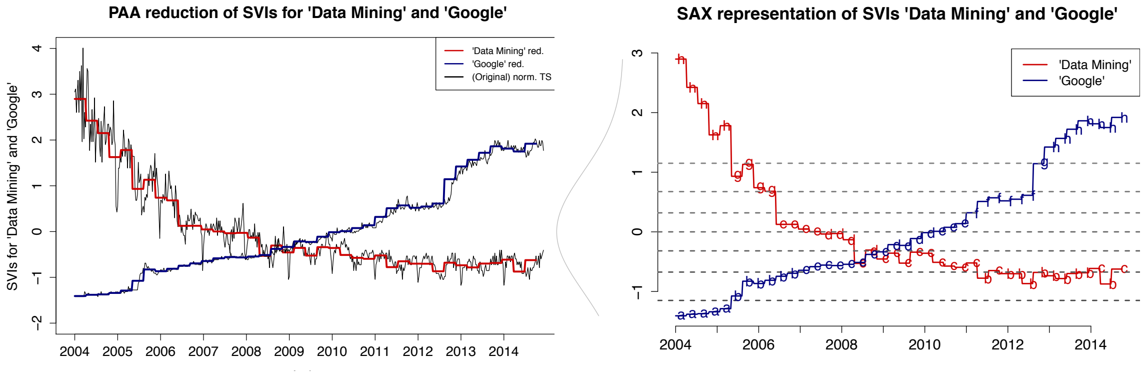

Figure 2: PAA dimension reduction (left) of Google search volume indices (SVIs) for the query terms “Data Mining” and “Google” with $n = 569$ and $\omega = 40$ respectively. Corresponding SAX representation (right) with $\omega = 40$ and $\alpha = 8$. In case of the SVI for the query term “Google”, the raw data is replaced by the word “aaaaabbbbbbccccccccddddeeeffffffgh-hhhhhh”.

TSDMSax

indexing of terabyte sized time series. Extensible hashing is a database management system technique (see for example Fagin et al. (1979)).

If we want to automatize the decision on the quantity of the hyperparameter α, we can use the Minimum Description Length (MDL) algorithm by Rissanen (1978). MDL is a computational learning concept and extracts regularities which lead to the best compressed representation of the data. In our context, MDL chooses that SAX representation depending on α which suffices and optimizes the trade-off between goodness-of-fit and model complexity.

Moreover, other symbolic representations than SAX exist. Megalooikonomou et al. (2005) propose the Multiresolution Vector Quantized (MVQ) approximation which is a multiresolution symbolic representation of time series. Mörchen and Ultsch (2005) develop a symbolic representation through an unsupervised discretization process incorporating temporal information. Li et al. (2000) arrive at a symbolic representation using fuzzy neural networks for clustering prediscretized sequences. The prediscretizing step is performed using a piecewise linear segmentation representation first.

Ye and Keogh (2009) propose a new representation approach called *Shapelets*. Shapelets are capturing the shape of small time series subsequences. Ye and Keogh (2009) propose Shapelets especially for classification tasks. Aside from that, Zhao and Zhang (2006) take recent-biased time series into account. They embed traditional dimension reduction techniques into their framework for recent-biased approximations.

β_i	3	4	5	6	7	8
	α = number of different symbols					
β_1	−0.43	−0.67	−0.84	−0.97	−1.07	−1.15
β_2	0.43	0.00	−0.25	−0.43	−0.57	−0.67
β_3		0.67	0.25	0.00	−0.18	−0.32
β_4			0.84	0.43	0.18	0.00
β_5				0.97	0.57	0.32
β_6					1.07	0.67
β_7						1.15

Table 1: Tabulated SAX breakpoints for the corresponding cutoff lines for 3 to 8 different symbols as reported in Lin et al. (2003). Corresponding to the hyperparameter α, the breakpoint parameters are chosen data independently. The breakpoints are determined based on the assumption that the z-normalized time series are standard normally distributed.

Lastly, Wen et al. (2014) recently proposed an adaptive sparse representation for massive spatial-temporal remote sensing.

3.1.3 Model Based Representation Techniques

The paradigm of model based representation techniques builds on the idea that the data at hand has been produced by an underlying model. Therefore, these techniques try to find the parametric form of the corresponding underlying model. One simple approach is to represent subsequences with best fitting linear regression lines (see for example Shatkay and Zdonik (1996)). Or we can employ more complex and better suited models as the *A*uto*R*egressive *M*oving *A*verage (ARMA) model and moreover, we can adduce *H*idden *M*arkov *M*odels (HMM, see for example Azzouzi and Nabney (1998)).

Unfortunately, there is no general answer to the question which representation technique is *the best*. The answer is: it depends. It depends on the data at hand and the purposes we are pursuing with our analysis. For example, if we want to get an idea of overall shapes and trends, the more globally focused approaches like DFT suite well. Or, if we want to incorporate wavelets, we need to have data with its length being an integer power of two. Highly periodic data is most likely treated best by spectral methods. Generally, the paper by Keogh and

Kasetty (2003) and Wang et al. (2013) both show that the different representation methods are overall performing very similar in terms of speed and accuracy. They use the tightness of lower bounding as measure for comparison of the manifold representation techniques. Furthermore, we have to regard that the choice of the dimensionality reduction technique determines the choice of an indexing technique (see Section 3.2). PAA, DFT, DWT and SVD naturally lead to an indexable representation.

3.2 Indexing

Representation and indexing techniques for time series work hand in hand with each other. Time series indexing schemes are designed for efficient time series data organization and especially for a quick processing request in large databases. If we want to find the closest match to a given query time series X in a database, a sequential or linear scan of the whole database is very costly. The access to the raw data is inefficient as it can take quite a long time. Therefore, indexing schemes are required to be much faster than sequential or linear scanning of the time series database. Hence, we store two representation levels of the data: on the one hand the raw data and on the other a compressed high level representation version of the data (see Section 3.1 for the different representation techniques). Then, we perform a linear scan for our query on the compressed data and compute a lower bound to the original distances to the query time series X. To put it in simpler words, the indexing is used to retrieve a rough first result from the database at hand. This quick and dirty result is used for a further, more detailed search for certain (sub)sequences. So, we try to avoid scanning the whole database but only examine certain sequences further which come into question.

Moreover, we build an index structure for an even more efficient similarity search in a database. We group similar indexed time series into clusters and access only the most promising clusters for further investigations. Index structure approaches can be classified into vector based and metric based index structures. Vector based index structures firstly reduce the data's dimensionality and then clusters the vector based compressed sequences into similar groups. The clustering technique can be hierarchical and non-hierarchical. R-Tree by Guttman (1984) is the most popular non-hierarchical vector based index structure. For indexing the first few DFT coefficients, Agrawal et al. (1993a) adopt the R*-tree by Beckmann et al. (1990). Faloutsos et al. (1994) focus on subsequence matching and employ R*-trees as well.

Opposed to this clustering based on the compressed features, metric based index structures cluster the sequences with respect to relative distances to each other. Yang and Shahabi (2005a) propose a multilevel distance-based index structure for multivariate time series. Vlachos et al. (2003a) index multivariate time series incorporating multiple similarity measures at the same time. The index by Vlachos et al. (2006) can accommodate multiple similarity measures and can be used for indexing multidimensional time series.

As mentioned above, the choice of the indexing structure can depend on the a priori made decision on the corresponding representation technique. In this context, we recite more representation specific indexing structures. Rafiei and Mendelzon (1997) build an similarity based index structure based on Fourier transformations. Agrawal et al. (1993a) build an index structure based on DFT coefficients and Kahveci and Singh (2004) focus on wavelets based index structures. Chen et al. (2007a) introduce indexing mechanisms for *P*iecewise *L*inear *R*epresentation, (PLR, see Section 3.3) and Keogh et al. (2001d) introduce the ensemble-index which combines two or more time series representation techniques for more effective indexing.

Lastly, Aref et al. (2004) present an algorithm for partial periodic patterns in time series databases and address the problem of so called *merge mining*. In merge mining, discovered patterns of two or more databases that are mined independently are merged, see for example Aref et al. (2004).

3.3 Segmentation

Segmentation is a discretization problem and aims to accurately approximate time series. As representation techniques (see Section 3.1) strive for similar purposes, the boundaries between segmentation and shape representation techniques are blurred. Bajcsy et al. (1990) even make the point that both preprocessing steps should not be handled separately. Yet historically grown, still other discretization and dimension reduction techniques than those discussed in Section 3.1 are subsumed under the concept of segmentation.

Time series are characterized by their continuous nature. Segmentation approaches reduce the dimensionality of the time series while preserving essential features and characteristics of the original time series. The general approach is to firstly segment a time series into subsequences (windows) and to secondly choose primitive shape patterns which represent the original time series best. The most intuitive time series segmentation technique is called PLR, the *P*iecewise *L*inear *R*epresentation (see for example Zhu et al. (2007)). The idea is to approximate a time series X of length n by k linear functions which are the segments then. But this segmentation technique highly depends on the choice of k and slices the time series in equidistant windows. This fixed-length segmentation comes along with obvious disadvantages. Therefore, we are interested in more flexible and data-responsive algorithms. Keogh and Pazzani (1998) introduce a PLR technique which uses weights as well accounting for the relative importance of each individual linear segment and Keogh and Pazzani (1999) add relevance feedback from the user.

Following Keogh et al. (2004a), the segmentation algorithms which result in a piecewise linear approximation can be categorized into three major groups of approaches: top down, bottom up and sliding windows approaches. Top down approaches recursively segment the raw data until some stopping criteria are met. Top down algorithms are used in many research areas and known under several names. The machine learning community for example knows it under the name "Iterative End-Points Fit" as named by Duda et al. (1973). Park et al. (1999) modify the top down algorithm by firstly scanning the whole time series for extreme points. These peaks and valleys are used as segmental starting points and then the top down approach refines the segmentation. Opposed to top down approaches, bottom up approaches start with the finest possible approximation and join segments until some stopping criteria are met. The finest possible approximation of a n-length time series are $n/2$ segments. Both, the top down and bottom up approaches are offline and need to scan the whole data set. Therefore, they operate with a global view on the data. Sliding windows anchor the left point of a potential segment and try to approximate the data to the right with increasing longer segments. A segment increases until it exceeds some predefined error bound. The next time series object not included in the newly approximated segment is the new left anchor and the process repeats. Sliding windows algorithms are especially attractive as they are online algorithms. However, sliding window algorithms are according to Keogh et al. (2004a) and Shatkay and Zdonik (1996) producing poor segmentation results if the time series at hand contains abrupt level changes as they cannot look ahead. The bottom up and top down approaches which operate with a global view on the data produce better results than sliding windows (see e.g. Keogh et al. (2004a)). By combining the bottom up approach and sliding windows, Keogh et al. (2001c) try to offset the disadvantages of the respective technique mutually. They propose the *S*liding *W*indow *A*nd *B*ottom Up (SWAB) segmentation algorithm which allows an online approach with "semi-global" view.

Within the top down, bottom up or sliding window approaches, the choices of techniques we can apply are manifold. All these techniques aim at the identification of in some way prominent points which are used for decisions in the three segmentation approaches. A common method is according to e.g. Jiang et al. (2007) or Fu et al. (2006) the *P*erceptually *I*mportant *P*oints (PIP) method as introduced for time series by Chung et al. (2001). Besides that, a plethora of different techniques are proposed. Oliver and Forbes (1997) pursue a change point detection approach, Bao and Yang (2008) propose turning points sequences applied to financial trading strategies, Guralnik and Srivastava (1999) present a special event detection, Oliver et al. (1998) and Fitzgibbon et al. (2002) use minimum message length approaches, and Fancourt and Principe (1998) tailor PCA to locally stationary time series. Duncan and Bryant (1996) suggest to use dynamic programming for time series segmentation. Himberg et al. (2001) speed the dynamic programming approaches up by approximating them with *G*lobal *I*terative *R*eplacement (GIR) algorithm results and they illustrate their segmentation technique with mobile phone applications in context recognition. Wang and Willett (2004) use a piecewise generalized likelihood ratio for a rough, first segmentation and then elaborate the segments further. Fancoua and Principe (1996) perform a piecewise segmentation with an offline approach and furthermore map similar segments as neighbors in a neighborhood map. Recently, Cho and Fryzlewicz (2012) segment a piecewise stationary time series with unknown number of breakpoints using a nonparametric locally stationary wavelet model.

Moreover, the segmentation of multivariate time series is an active research area. Dobos and Abonyi (2012) combine recursive and dynamic *P*rincipal *C*omponent *A*nalysis (PCA) for multivariate segmentation. Lastly, in a very recent paper by Guo et al. (2015), dynamic programming is applied in order to tackle multivariate time series segmentation automatically.

3.4 Visualization

Due to the massive size of the data, an actually simple task like visualization can very fast become anything but trivial. As a result of the very high dimensionality of large time series, plotting a univariate time series using a usual line plot is unrewarding. Accordingly, the need for manageable and intuitive data visualization gives rise to several visualization tools. The most popular representatives of the recent visualization approaches or interfaces are Calendar-Based visualization, Spiral, TimeSearcher and VizTree.

Calender-Based visualization by Van Wijk and Van Selow (1999) is based on the idea of showing cluster averages of similar sequences as an aggregated representation of the data.

The time series are chunked into daily sequences and clusters are computed for similar daily sequential patterns. Furthermore, a calendar with color-coded clusters is shown.

The *Spiral* by Weber et al. (2001) is mainly used for detecting periodic patterns and structures in the data. Periodic patterns are mapped onto rings and assigned to colors and the line width is corresponding to their features. But implicitly, the Spiral is only useful for data with periodic structures.

Time Searcher by Hochheiser and Shneiderman (2004) and Keogh et al. (2002a) requires a priori knowledge about the data at hand as we need to have at least an idea of what we are searching for. We need to insert query orders which are called "TimeBoxes" for zooming in on certain patterns.

The most recent tool and most promising of these four is *VizTree* by Lin et al. (2005). The VizTree interface provides both, a global visual summary of the whole time series and the possibility to zoom in for interesting subsequences. So, VizTree is suited best for data mining tasks as we can discover hidden patterns without previous knowledge about the data. VizTree firstly computes a symbolic representation of the data and then builds a suffix tree. In the suffix tree, characteristic features of patterns and frequencies are mapped onto colors. VizTree is suited for the discovery of frequently appearing patterns as well as the detection of outliers and anomalies.

Moreover, Kumar et al. (2005) propose a user friendly visualization tool which employs similarities and differences of subsequences within a collection of bitmaps. Lastly, Li et al. (2012) introduce a motif visualization system based on grammar induction. For this visualization system, no a priori knowledge about motifs is required and the motif discovery can take place for time series with variable lengths.

3.5 Similarity Measures

Similarity measures indicate the level of (dis)similarity between time series. They are at the same time the backbone and the bottleneck of time series data mining. As similarity measures are needed for almost all data mining tasks (i.e. pattern discovery, clustering, classification, rule discovery and novelty detection) they are the backbone of time series data mining. Coincidentally, similarity measures impose the major capacity constraints on time series data mining algorithms (Rakthanmanon et al., 2012). The faster the similarity measure computation algorithm, the faster is the whole time series data mining procedure as the main computing time is needed for the similarity measure calculations.

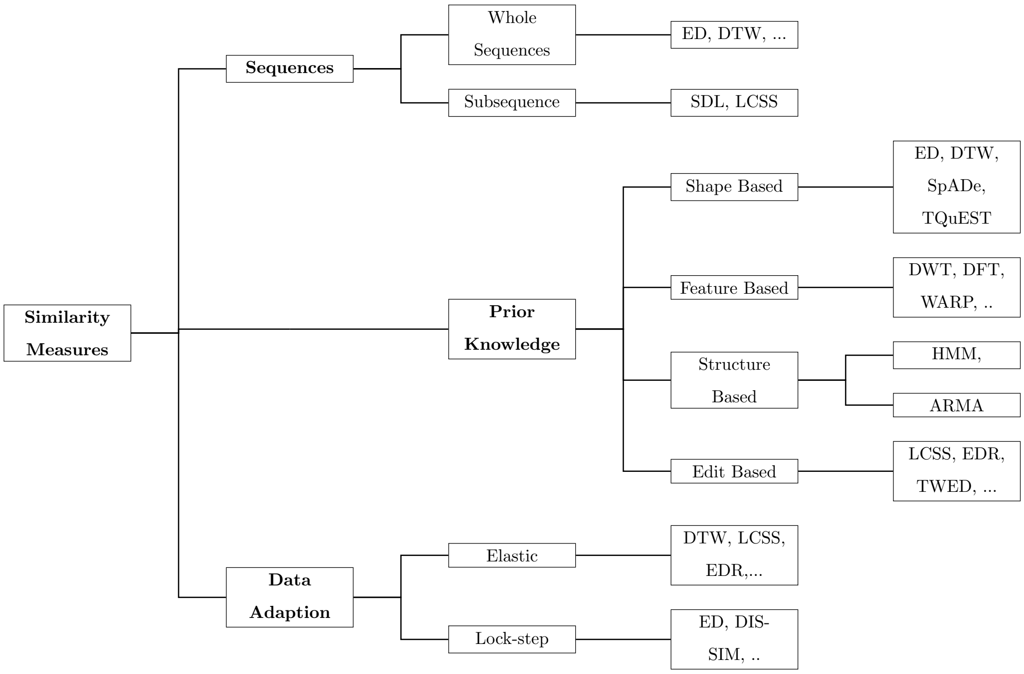

Figure 3: Possible systematizations of time series similarity measures.

Similarity measures are required to be robust against scaling differences between time series, warping in time, noise along characteristic patterns and outliers (Esling and Agon, 2012). Scaling robustness includes robustness against amplitude modifications and warping robustness corresponds to robustness against temporal modifications. A “noisy” time series is here interpreted as a time series with an additive white noise component.

Similarity measures are not only the capacity bottleneck of the time series data mining process with respect to time, but they govern the number of dimensions we can deal with, too. So, the processable dimensions of time series datasets depend on the manageable dimensions for the similarity search. Rakthanmanon et al. (2012) were the first to develop a similarity search algorithm based on dynamic time warping (see Section 3.5.1) that allows mining a *trillion* time series objects. Up to that paper, time series data mining algorithms were limited to a few million observations (if one requires an acceptable computing time). But at the same time industry possesses massive amounts of time series waiting to be explored; speeding similarity search up in order to mine trillions of time series objects is a breakthrough in time series data mining.

Time series data mining community proposed a plethora of different similarity measures and distance computation algorithms. Also, many different systematizations of time series similarity measures exist (see Figure 3). We take over the systematization by Esling and Agon (2012) who classify similarity measures into four categories: shape based, edit based, feature

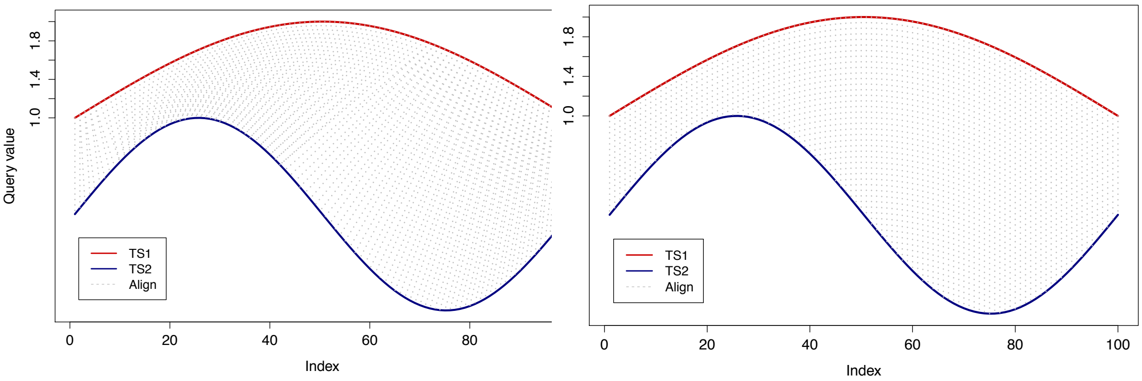

Figure 4: Dynamic time warping (left) vs. Euclidean distance (right): DTW searches the optimal alignment path through the distance matrix consisting of all pairwise Euclidean distances between the two time series X and Y.

TSDMDtw

based and structure based similarity measures. Furthermore, we distinguish between elastic and lock-step similarity measures. Lock-step measures compare the i-th point of time series X to the i-th point of time series Y. In contrast to that, elastic similarity measures allow a flexible comparison and additionally compare one-to-many or one-to-none points of X to Y. The also useful systematization of similarity measures into sequence and subsequence matching approaches is discussed in Fu (2011).

3.5.1 Shape Based Similarity Measures

Shape based similarity measures compare the global shape of time series. All L_p norms, especially the popular Euclidean distance, are widely used similarity measures (Yi and Faloutsos, 2000). Nevertheless, in time series similarity computations, L_p norms deliver poor results (see for example Keogh and Kasetty (2003) or Ding et al. (2008)) as they are not robust against temporal or scale shifting. As they are lock-step measures, the length and position in time of the two time series which we want to compare need to be the same.

Opposed to that, the most popular elastic shape based similarity measure is Dynamic Time Warping (DTW) and especially proposed to handle warps in the temporal dimension. Temporal *warping* corresponds to shifting and further modifications in the temporal axis. DTW is said to be the most accurate similarity search measure (see for example Rakthanmanon et al. (2012), Ding et al. (2008)). After its introduction it was firstly popular in speech recognition (Sakoe and Chiba (1978)) and nowadays it is used in manifold domains, for example for online signature recognition. Berndt and Clifford (1994) were the first to use DTW in data mining processes. Figure 4 shows a DTW alignment plot and for comparison Euclidean distances between two time series. DTW is proposed to overcome inconveniences of rigid distances and to handle warping and shifting in the temporal dimension, i.e. the two temporal sequences may vary in time or speed. Furthermore, DTW allows to compare time series of different length. Even an acceleration or deceleration during the course of events is manageable. In order to compute a DTW distance measure between two time series $X = \{x_1, x_2, \ldots, x_N\}$ and $Y = \{y_1, y_2, \ldots, y_M\}$ with $N, M \in \mathbb{N}$, two general steps are necessary. Firstly, a cost (or "distance") matrix $D_{(N \times M)}$ as shown in Figure 5 has to be calculated. For this purpose, predefined distances (most often Euclidean distances) between *all* components of X and Y have to be computed. Each entry of the cost matrix $D_{(N \times M)}$ corresponds to the Euclidean distance between a pair of points (x_i, y_j). Secondly, we look for the optimal alignment path between X and Y as a mapping between the two time series. The optimal alignment between two time series results in minimum overall costs, i.e. the minimum cumulative distance. The basic search for the optimal alignment path through the cost matrix is subject to three general conditions: (i) boundary, (ii) monotonicity and (iii) step size conditions. The boundary condition ensures that the first and the last observations of both time series are compared to each other. So, the start of the alignment path in the cost matrix is fixed as $D(0, 0)$ and the end as $D(N, M)$. The monotonicity and step size conditions ensure that the alignment path moves always up or right or both at once but never backwards. Using dynamic programming, the computing time of DTW has complexity $\mathcal{O}(n^2)$ (Ratanamahatana and Keogh, 2004).

Additionally, many extensions to the classical DTW exist. Further constraints (especially lower bounding measures) aim to speed up the matching process. Ding et al. (2008) recommend to generally use constrained DTW measures instead of plain DTW. Constraining the warping window size can reduce computation costs and enable effective lower-bounding while resulting in the same or even better accuracy. The most frequently applied global constraint is the Sakoe-Chiba band. Sakoe and Chiba (1978) place a symmetric band around the cost matrix' main diagonal. The optimal alignment path is forced to stay inside this band. The Itakura parallelogram is a further very frequently used global path constraint. Itakura (1975) place a parallelogram around the cost matrix' main diagonal constraining the warping range. A further common constraint is the *lower bounding* of the DTW distance. Lower bounding conditions require that the approximated DTW distance is at least as large

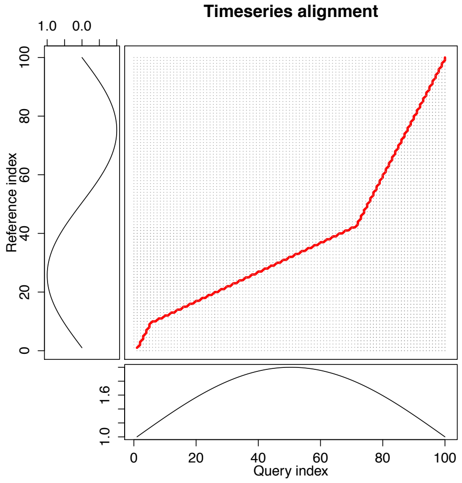

Figure 5: Corresponding DTW cost matrix with the query index (reference index) corresponding to time series X (Y). The diagonal would represent the case for fixed Euclidean distances in the right panel of Figure 4. The red line is the optimal alignment path for time series X and Y in the left panel of Figure 4. By minimizing overall costs, i.e. by minimizing the cumulative distance through the cost matrix, we find the optimal alignment path.

TSDMDtw

as the actual DTW distance. The most common lower bound condition is proposed by Keogh and Ratanamahatana (2005). They introduce the upper and lower envelope representing the maximum allowed warping which reduces computing complexity to $\mathcal{O}(n)$.

Moreover, modifications and extensions of DTW exist. Yi et al. (1998) use a FastMap technique for an approximate indexing of DTW. Salvador and Chan (2007) introduce the approximation of DTW, called FastDTW. This algorithm recursively projects a solution from a coarse solution to a higher resolution and then refines it. Fast DTW is on the one hand only approximative but on the other hand, it enables linear computing time. Chu et al. (2002) introduce an iterative deepening DTW approach and Sakurai et al. (2005) present their Fast Dynamic Time Warping (FTW) similarity search. Fu et al. (2008) combine the locally flexible DTW with globally flexible uniform scaling which leads to search pruning and speeds up the search as well. Furthermore, Keogh and Pazzani (2000a) show that operating with the DTW algorithm on the higher level PAA representation instead of the raw data does not lead to a loss of accuracy.

Until 2012, the main disadvantage of DTW is said to be the computational complexity. DTW computation was too slow to be used for truly massive databases. But Rakthanmanon et al.

(2012) proposed a new DTW based exact subsequence search suite of four novel ideas they are calling the *UCR suite*. They normalize the time series subsequences, reorder abandoning, reverse the query/data role in lower bounding and cascade lower bounds. Rakthanmanon et al. (2012) hereby facilitate mining of time series with up to a trillion observations.

Besides DTW and its modifications, many other shape based similarity measures exist. Frentzos et al. (2007) introduce the index based DISSIM metric and Chen et al. (2007b) introduce the shape based *Sp*atial *A*ssembling *D*istanc*e* (SpADe) algorithm which is able to handle shifting in the temporal and amplitude dimensions. Aßfalg et al. (2006) propose TQuEST, a similarity search based on threshold queries in time series databases which report those sequences exceeding a query threshold at similar time frames as the query time series. Goldin and Kanellakis (1995) impose constraints on similarity queries formalizing the notion of exact and approximate similarity.

3.5.2 Edit Based Similarity Measures

The main idea of edit based similarity measures is to assemble the minimum number of operations needed to transform time series X into time series Y.

Lin and Shim (1995) introduce a similarity concept that captures the intuitive notion that two sequences are considered as similar if they have enough non-overlapping time-ordered pairs of subsequences that are similar. They allow amplitude scaling of one of the two time series for their similarity search. Hereafter, Chu and Wong (1999) introduce the idea of similarity based on scaling and shifting transformations. Following their idea, the time series X is similar to time series Y if suitable scaling and shifting transformations can turn X into Y. The *L*ongest *C*ommon *S*ub*S*equence (LCSS) measure is the most popular edit based similarity measure (Das et al., 1997) and probably the biggest competitor of DTW. Generally, LCSS is the minimum number of elements that should be transferred from time series X to Y in order to transform Y into X. It is said to be very elastic as it allows unmatched elements in the matching process between two time series. Therefore, the handling of outliers is very elegant in LCSS. For the LCSS approach, extensions or modifications exist. For example, Vlachos et al. (2002) extend LCSS for multivariate similarity measures for more than two time series. Morse and Patel (2007) introduce the *F*ast *T*ime *S*eries *E*valuation method which is used to evaluate threshold values of LCSS and its modifications. Moreover, using this method we can evaluate the *E*dit *D*istance on *R*eal sequences (EDR) by Chen et al.

(2005). The edit based similarity measure EDR is robust against any data imperfections and corresponds to the number of insert, delete and replace operations needed to change X into Y.

Moreover, combinations and extensions of edit based similarity measures with other similarity measures exist. Marteau (2009) extend edit based similarity measures and present Time Warp Edit Distance (TWED) which is a dynamic programming algorithm for edit operations. Chen and Ng (2004) combine edit based similarity measures and L_p norms and name the resulting similarity measure Edit distance with Real Penalty (ERP).

3.5.3 Feature Based Similarity Measures

The subcategory of feature based similarity measures is not as broadly developed as for example shape based similarity measures. In order to compare two time series X and Y based on feature similarity measures, we firstly select characteristic features of both time series. The features are extracted coefficients, for example stemming from representation transformations (see Section 3.1).

With respect to feature based similarity, Vlachos et al. (2005) focus on periodic time series and want to detect structural periodic similarity utilizing the autocorrelation and the periodogram for a non-parametric approach. While Chan and Fu (1999) simply apply the Euclidean distance to DWT coefficients, Agrawal et al. (1993a) apply it to DFT coefficients. Janacek et al. (2005) construct a likelihood ratio for testing the null hypothesis that the series are from the same underlying process. For the likelihood ratio construction they use Fourier coefficients. WARP is a Fourier-based feature similarity measure by Bartolini et al. (2005). They apply the DWT similarity measure to the phase of Fourier coefficients. Papadimitriou and Yu (2006) estimate eigenfunctions incrementally for the detection of trends and periodic patterns in streaming data. Lastly, Aßfalg et al. (2008) extract sequences of local features from amplitude-wise scanning of time series for similarity search.

3.5.4 Structure Based Similarity Measures

In contrast to all kinds of distance based similarity measures, structure based measures utilize probabilistic similarity measures. Structure based similarity measures are especially useful for very long and complex time series and the focus is on a global scale comparison of time series. In the majority of structure based similarity measures, prior knowledge about the data generating process is factored in. The first step is to model one time series X as the reference time series parametrically. Then, the likelihood that the time series Y is produced by the underlying model of X corresponds to the similarity between X and Y. For the parametrical modeling any type of all well known time series models as for example the *A*uto*R*egressive *M*oving *A*verage (ARMA) model can be applied. The Kullback-Leibler divergence for example measures then the difference between probability distributions (Kullback and Leibler, 1951). Gaffney and Smyth (1999) suggest to use a mixture of regression models (including non-parametric techniques) and use the expectation maximization approach. *H*idden *M*arkov *M*odel (HMM) approaches are frequently used for structure based similarity measurement. Panuccio et al. (2002) as well as Bicego et al. (2003) use HMM for a probabilistic model-based approach for proximity distance construction. Ge and Smyth (2000) model the distance between segments as semi Markov processes and hereby allow for flexible deformation of time. Finally, Keogh et al. (2004b) motivate their *C*ompression Based *D*issimilarity *M*easure (CDM) approach with the need for parameter free data mining approaches. The CDM dissimilarity measure is based on the Kolmogorov complexity. Based on the Lempel-Ziv complexity, Otu and Sayood (2003) construct a sequence similarity measure for phylogenetic tree construction.

As for representation techniques, the question which similarity measure is most suitable depends highly on the data at hand. For example shape based similarity measures are the best to use if the time series is short and still overseeable by the unaided eye. Within this class, DTW is said to be the best performing similarity measure (see for example Rakthanmanon et al. (2012), Ding et al. (2008)). And if we know a lot about the data a priori, we can use this knowledge by incorporating it in structure based similarity measures. If a central feature of the available data is periodicity, feature-based methods can be most suitable. Besides theoretical considerations about the appropriability of the different similarity measure classes, we want to quantify the accuracy of the potential similarity measures. Most frequently, a 1-NN classifier is used to evaluate the accuracy of similarity measures as for example in Wang et al. (2013). They find in a comprehensive experimental comparison of similarity measures that the elastic measures DTW, LCSS, EDR and ERP are for small data sets significantly more accurate than lock-step measures. Wang et al. (2013) make the point that the performance of similarity measures depends on the size of the data set. In small data sets, DTW is the most accurate one - but for massively long time series, the performance of the simple ED converges to the DTW performance. Furthermore they find that the performance of edit distance based similarity measures as LCSS, EDR and ERP are very similar to DTW. DTW however is much simpler and therefore preferable. Lastly, they find novel similarity measures as TQuEST and SpADe to perform inferior.

4 Mining in Time Series

After preprocessing the raw data at hand and turning bulky time series data sets into manageable and overseeable data, we can proceed with the typical data mining tasks. We aim to cluster the data and detect frequently appearing patterns and anomalies. Moreover, classification is a designated time series data mining task. Finally, we shed light on rule discovery and forecasting.

4.1 Clustering

Clustering of unlabeled data is one important step in the pattern discovery process. In the machine learning jargon, clustering is assigned to unsupervised or semisupervised learning algorithms depending on whether we have hyperparameters or not. The aim of clustering is to find natural groupings in the data at hand. The *natural* groups are desired to be homogeneous groups and found by maximizing the dissimilarity between groups and minimizing the dissimilarity within the groups. Apparently, similarity measures (see Section 3.5) are required for the clustering of time series. Figure 6 shows the Euclidean distance based hierarchical clustering for weekly Google search volume index data. The clusters of similar time series are marked by red rectangles and one exemplary cluster is the rectangle containing the SVIs for the query terms "Data Mining", "Clustering" and "UNO". Figure 7 shows these three SVIs which are detected as one cluster and indeed, the pathways look similar. Moreover, the closeness of the query terms "Data Mining" and "Clustering" are intuitively evident.

For static data, plenty of clustering approaches exist. Not all clustering procedures for static data can be overtaken or translated into the task of finding groups of similar time series. Only three major classes of clustering approaches are utilized for time series clustering: hierarchical, partitioning and model-based methods. Hierarchical clustering operates in a bottom-up way as it merges similar clusters beginning with pairwise distances. A strong shortfall of hierarchical clustering is the limited number of time series we can cluster as the computational complexity is $\mathcal{O}(n^2)$. Partitional clustering aims to minimize the sum of

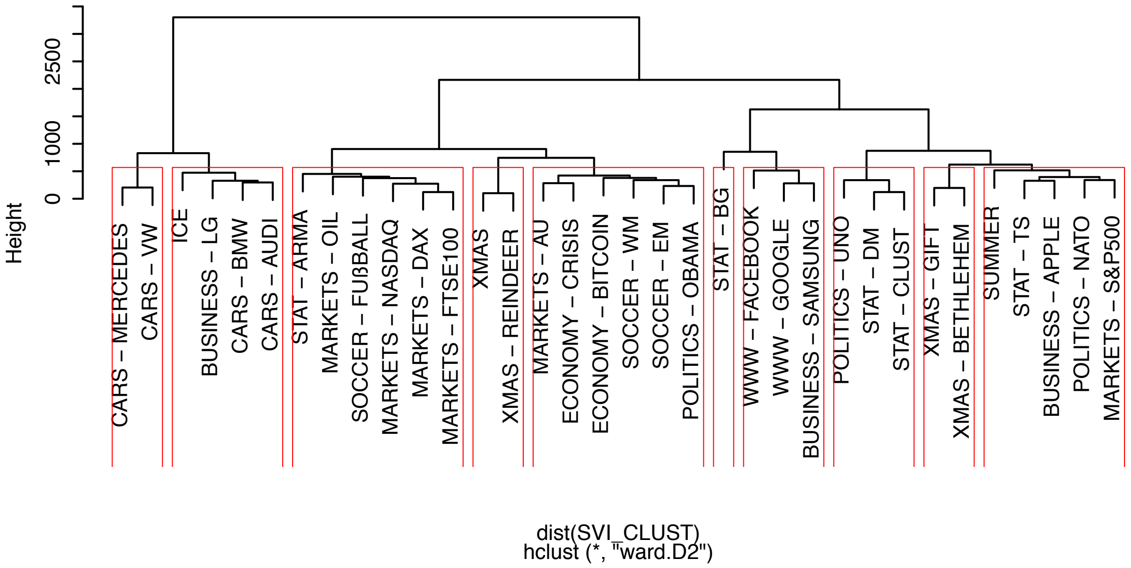

Figure 6: Hierarchical clustering based on Euclidean Distances for weekly Google SVI data (01/2004 - 11/2014) on 34 different query terms.

TSDMClust

squared errors within one cluster using k-means. But, using a k-means algorithm, we have to prespecify the number of clusters k in this semisupervised learning procedure. Many different partitional clustering methods were proposed in recent years. For example Vlachos et al. (2003b) focus on k-means clustering and Cormode et al. (2007) implement k-center clustering. Möller-Levet et al. (2003) focus on fuzzy clustering of short time series as they modify the fuzzy c-means algorithm for time series. Lin et al. (2004) adopt the multi-resolution property of wavelets for a partitioned clustering algorithm. Dai and Mu (2012) especially tailor the k-means clustering approach to a symbolic time series representation. Moreover, Rakthanmanon et al. (2011) focus on clustering of streaming time series data and for this purpose utilize the computational learning *M*inimum *D*escription *L*ength (MDL) framework from Rissanen (1978).

Another general clustering idea stems from artificial neural networks and are *S*elf *O*rganizing *M*aps (SOM) by Kohonen (2001). Euliano and Principe (1996) and Ultsch (1999) adopted SOM for time series and use self-organizing feature maps.

Lastly, model-based methods are frequently used clustering approaches. The most commonly known time series model is the *A*uto*R*egressive *I*ntegrated *M*oving *A*verage (ARIMA) model. For example Kalpakis et al. (2001) cluster ARIMA time series. An ARMA mixture model for time series clustering is proposed by Xiong and Yeung (2004). They derive an expectation maximization approach for learning the mixing coefficients and model parameters. Moreover, HMM approaches are popular in time series clustering and pattern recognition applications. Panuccio et al. (2002), Law and Kwok (2000) as well as Bicego et al. (2003) develop HMM-based approaches for sequential clustering. Shalizi et al. (2002) uses HMM for pattern recognition as well, but they make no a priori assumptions about the causal architecture of the data but starting from a minimal structure, they refer the number of hidden states and their transition structure from the data. Ge and Smyth (2000) model the distance between segments deformable Markov model templates and address hereby the problem of automatic pattern matching between time series. Oates et al. (1999) introduce an unsupervised clustering approach of time series with hidden Markov models and DTW (see Section 3.5.1). They aim at an automatic detection of K, the number of generating HMMs and learning the HMM parameters. For this purpose, they use the DTW similarity as an initial estimate of K. Related to model-based clustering methods, Wang et al. (2006) describe a characteristic-based clustering approach for time series data, i.e. they cluster time series with respect to global features extracted from the data. Additionally, Denton (2005) proposes a Kernel-based clustering approach for time series subsequences. However, Denton et al. (2009) show that its performance degrades fast for increasing window sizes. Lastly, Fröhwirth-Schnatter and Kaufmann (2008) use Markov Chain Monte Carlo (MCMC) methods for estimating the appropriate grouping of time series simultaneously. For a more detailed review on time series clustering methods, please refer for example to Liao (2005) or Berkhin (2006).

In the context of time series data mining, the most common shortfall of time series clustering techniques is the inability to handle longer time series. Keogh and Lin (2005) even claim that clustering of time series can be meaningless with randomly extracted clusters. Chen (2005) however argue against that claim using other similarity measures than the Euclidean distance as Keogh and Lin (2005) do. Time series increase mostly linear with time and this slows the pattern discovery process exponentially down (Fu, 2011). These facts plead again for the previously discussed preprocessing methods (Section 3). An effective compression of the data speeds up all subsequent tasks.

4.2 Knowledge Discovery: Pattern Mining

Generally speaking, knowledge discovery is referred to as the detection of frequently appearing patterns, novelties and outliers or deviants in a time series database. Novelties are referred to anomalies or surprising patterns. Pattern discovery (or 'motif discovery') comes

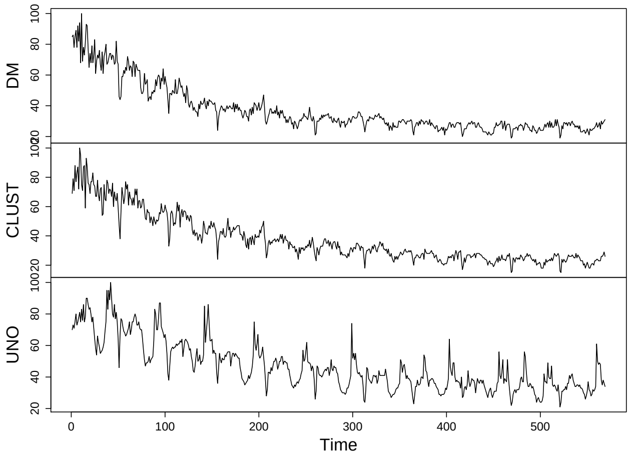

Figure 7: Three exemplary Google SVI time series detected to be similar according to hierarchical clustering on the query terms: “UNO”, “Clustering” and “Data Mining”

TSDMClust

along hand in hand with clustering methods as the occurrence frequency of patterns in time series subsequences can naturally be found by clustering.

As a general rule, *motifs* are seen as frequently appearing patterns in a time series database (Patel et al., 2002). Frequently appearing patterns are subsequences of time series which are very similar to each other. In recent years, motif mining in time series is of ever growing importance. Therefore, the available literature and approaches are manifold.

The most popular pattern discovery algorithm is from Berndt and Clifford (1994) and uses *D*ynamic *T*ime *W*arping (DTW, see Section 3.5.1). They use a dynamic programming approach for this knowledge discovery task. Yankov et al. (2007) propose a motif discovery algorithm which is invariant to uniform scaling (stretching of the pattern length) and Chiu et al. (2003) approach time series motif discovery in a probabilistic way as they aim to circumvent the inability to discover motifs in the presence of noise and poor scalability. Lonardi and Patel (2002) address the problem of finding frequently appearing patterns which are previously unknown. Most motif discovery approaches require a predefined motif length parameter. Nunthanid et al. (2012) and Yingchareonthawornchai et al. (2013) approach this problem and propose parameter free motif discovery routines. Heading in a similar direction, Li et al. (2012) focus on the visualization of time series motifs with variable lengths and no a priori knowledge about the motifs. Moreover, Hao et al. (2012) rely on visual motif discovery, too. As time series can hide a high variety of different recurring motifs, they aim to support visual exploration.

The importance of motif discovery techniques which can handle multivariate time series is ever growing. Papadimitriou et al. (2005) developed the popular Streaming Pattern dIscove$R$y in mult$I$ple Time series (SPIRIT) algorithm which can incrementally find correlations and hidden variables summarizing the key trends of the entire multiple time series stream. Tanaka et al. (2005) adduce the Minimum Description Length (MDL) principle and further use Principal Component Analysis (PCA) for the extraction of motifs from multi-dimensional time series.

Naturally, many time series datasets have an inherent periodic structure. Therefore, detecting periodicity is another classical pattern discovery task. Besides classical time series analysis methods for handling seasonality and periodicity (see for example Brockwell and Davis (2009)), time series data mining community produced techniques for massive data sets. Han et al. (1998) and Han et al. (1999) address mining for partial periodic patterns as in many data applications full periodic patterns appear not that frequently. Elfeky et al. (2005) aim to mine for the periodicity rate of a time series database and Vlachos et al. (2004) use power spectral density estimation for the periodic pattern detection. Similarly, the detection of trends is a classical time series analysis task as well (see for example Brockwell and Davis (2009)). However, the detection of trend behavior is here subsumed under general pattern detection tasks described in this chapter. Only a few time series data mining papers explicitly address the identification of frequently appearing trends like for example Indyk et al. (2000) or Udechukwu et al. (2004).

Besides, a great assemblage of pattern discovery approaches are developed especially for financial time series data. The overall goal of all these papers is obvious: they want to forecast financial operating numbers. Lee et al. (2006) transfer financial time series data to fuzzy patterns and model them with fuzzy linguistic variables for pattern recognition. Fu et al. (2001) use self-organizing maps for pattern discovery in stock market data.

The massive amount of recently proposed motif discovery approaches for time series data mining reveals the yet unsatisfied demand for suitable techniques. For example the first algorithms for exact discovery of time series motifs are delivered by Mueen et al. (2009a) and Mueen et al. (2009b). Floratou et al. (2011) utilize suffix-trees to find frequent occurring patterns. Another more innovative approach is the particle swarm based multimodal optimization algorithm by Serrà and Arcos (2015). Particle swarm is an computational optimizer and for example Kennedy (2010) provides more details on it.

Surprising patterns have to be seen separately from the similar problem of outlier detection. Outliers are individually surprising datapoints and surprising patterns are collections of time series datapoints which are collectively surprising or show anomalies. Keogh et al. (2002b) are the first to properly define the term of "surprise" in the context of pattern recognition in massive time series databases. They classify a pattern as surprising if the frequency of its occurrence differs significantly from the expected one. For building expectations, they apply a suffix tree to encode the frequency of all observed patterns. Yet, Keogh et al. (2007) introduce a further related concept for anomalies in time series data namely "discords" which are defined as subsequences which are maximally different from all remaining subsequences. Shahabi et al. (2000) introduce a wavelet based tree structure for multi-level surprise and level queries on time series data. Synonymously, surprising patterns are often referred to as anomalies (see for example Wei et al. (2005)). Ma and Perkins (2003) aim to detect anomalies based on a *S*upport *V*ector *R*egression (SVR) approach. Ypma and Duin (1997) use *S*elf *O*rganizing *M*aps (SOM) for the novelty detection. Wei et al. (2006) firstly convert the data to the SAX representation and then try to find anomalies. As the detection of anomalies usually requires domain specific expertise, Wei et al. (2005) propose an assumption-free anomaly detection approach which is based on time series bitmaps. Chan and Mahoney (2005) produce anomaly scores in an online manner for multiple time series. Yankov et al. (2008) aim to scale the detection for unusual time series up to terabyte sized datasets with a disk aware algorithm.

Also in the field of anomaly detection and discovery of surprising patterns in time series, recent research compulsion has not been satisfied yet. Therefore, possible solutions on this topic were proposed very recently. For example Leng et al. (2013) introduce an anomaly detection algorithm based on pattern densities and Begum and Keogh (2014) try to find rare motifs in time series streams and propose an approximate brute force algorithm. Very recently, Izakian and Pedrycz (2014) firstly divide the time series in subsequences with sliding windows and then apply a fuzzy *c*-means clustering for anomaly detection.

4.3 Classification

Classification of unlabeled time series to existing classes is a further traditional data mining task. All classification approaches first build a classification model based on labeled time series. In this case, "labeled time series" means that we use a training data set with correctly classified observations or time series sequences for model building. Then, the built models are used to predict the label of a new, unlabeled observation or sequence of a time series. Classification as data mining task is assigned to the supervised learning algorithms in the machine learning jargon. For example Geurts (2001) illustrate one possible idea and the necessary steps to that end of time series classification. The first, essential step is to find local properties and patterns (see Section 4.2 for pattern discovery techniques). In a second step, these patterns are combined to build classification rules. In the following, the different classifiers are discussed only briefly as they are well known and not uniquely used in time series data mining (Lotte et al., 2007).

The group of nearest neighbor classifiers has the simplest classification idea: we assign a new time series object or time series sequence to the most common class among its neighborhood. As indicated by the name, the k-Nearest Neighbor (NN) classifier takes the k nearest neighbors into account. Another popular nearest neighbor classifier is the Mahalanobis distance. A further big group of classifiers is using linear functions to separate classes. Popular examples of such linear classifiers are Linear Discriminant Analysis (LDA) and Support Vector Machines (SVM). In both cases, the aim is to separate the observations falling into different classes using hyperplanes. For LDA (see for example McLachlan (2004)), we need to assume normality of the data with equal covariance matrix for all classes. These very strong assumptions and the predetermined linearity weaken the applicability of LDA to time series data mining problems. On the other hand, the computational costs are very low and therefore, LDA is scalable to larger data set problems. SVM (see for example Cortes and Vapnik (1995)) implies linearity for the discriminating hyperplanes as well. The separating hyperplanes are obtained by maximizing the so called margins which are the "gaps" between the classes. In addition, SVM can be enhanced to tackle nonlinear classification problems. Thus, if the classes may not be separable by a hyperplane in the original space of the data, we apply the Kernel trick for mapping the data to an n-dimensional space where we can separate the classes with a linear decision surface. SVM has only a few hyperparameters that need to be chosen (2 parameters maximally) and is insensitive towards the curse of dimensionality and overtraining. At the same time, it has a relatively long computational time. Neural Networks by contrast produce nonlinear decision boundaries. See for example Haykin (2004) for a comprehensive introduction to neural networks. Moreover, nonlinear bayesian classifiers are a frequently used group of time series classifiers. The basic idea of bayesian classifiers is to maximize the probability that the new observation Y belongs to class $c = 1, ..., k$ conditional on time series X. Besides, Hidden Markov Models (HMM) are according to Rabiner (1989) especially useful for the classification of time series. We can build individual hidden Markov models for each class of interest and use the HMM as time series classifier.

Finally, researchers proposed classification algorithms which are especially adopted or tailored to time series data mining. Povinelli et al. (2004) for example use Gaussian mixture models and Rodríguez and Alonso (2004) apply DTW-based decision trees for time series classification. Wei and Keogh (2006) with good reason criticize that all classifiers discussed above need prelabeled training data and hence expert knowledge and supervision. Semisupervised time series classification includes both, learning from labeled and unlabeled data and requires therefore less human intervention and effort. Wei and Keogh (2006) propose a semisupervised time series classifier which incorporates a self-training step. In this step, they build an adequately sized training set out of a priori unlabeled data by applying nearest neighbor methods as a first rough approximation until some stopping criterion is reached. Moreover, Xi et al. (2006) claim that the 1-NN-DTW classifier which combines dynamic time warping as similarity measure and a 1-NN classifier is the best time series classification approach. They try to offset the shortfall of the very high computational effort by speeding the DTW calculations up by numerosity reduction.

For a more detailed review on time series classification methods, please refer for example to Lotte et al. (2007).

4.4 Rule Discovery

Rule discovery is at the very heart of knowledge discovery through data mining. To put it in simple words, we try to detect relations between variables, time series sequences or patterns that typically appear in a very large database.

Association rules as investigated by Agrawal et al. (1993b) indicate which variables or items in a very large database are linked with each other. Agrawal et al. (1993b) examine market basket association rules for discovering regularities about which products are typically bought together. The very basic idea of association rules is to search for correlations between variables. For a more detailed overview of the association rule algorithms like the Apriori, Eclat or FP-growth algorithm, see for example Hipp et al. (2000). Beyond that, Ale and Rossi (2000) not only investigate association rules but take additionally the temporal dimension into account. Association rule mining theoretically does not control for associations which are very far away from each other over time. Ale and Rossi (2000) introduce “age” as an obsolescence factor for rules. Hence, nontemporal association rules only suggest that two variables are associated with each other or in terms of events that two events frequently occur jointly. Temporal association rules additionally indicate in which order in time these jointly occurring events take place. Roddick and Spiliopoulou (2002) for instance vividly illustrate the difference between association and temporal association rule mining.

Up to here, the rule mining approaches are not especially tailored to time series data. In rule mining in time series data, we can broadly distinguish between the discovery of *temporal rules* and *temporal patterns*. Temporal rules are cause-effect relationships between events over time and temporal patterns are groups of events ordered by time.

Discovering rules relating temporal patterns can be leant onto well known cross sectional rule mining methods. Das et al. (1998) proceed in that way and suggest a very straightforward technique. They firstly discretize the time series by sliding windows in order to form subsequences. Then, they cluster these subsequences with respect to a similarity measure and thereupon apply simple rule discovery techniques for finding hidden rules for temporal patterns. Last et al. (2001) propose a fuzzy association rule algorithm in time series databases which incorporates the computational theory of perception and signal processing algorithms. Lu et al. (1998) aim to drive up the dimensions of mining capacities for inter-transaction association rules along time. They reduce the search space by a maximum span along time which works in the 1-dimensional case like a sliding window. Lu et al. (1998) illustrate their n-dimensional inter-transaction association rule discovery approach with stock market price movement predictions. Besides Lu et al. (1998), many authors aim to apply rule mining approaches to stock market data. For example Allen and Karjalainen (1999) use a genetic algorithm from machine learning for learning technical trading rules. Ting et al. (2006) assess inter- and intra-stock patterns by employing symbolic representation techniques (see Section 3.1.2) before associative classification. Leigh et al. (2002) chime in stock market data mining and use technical charting heuristics for trading rule discovery.

Decision trees are in addition to association rules another frequently used rule mining method. Cotofrei and Stoffel (2002b) initially applied classification trees on sequential data and extract rules with a temporal dimension. Then, Cotofrei and Stoffel (2002a) propose temporal rules based on temporal first logic-order formalism. Ohsaki et al. (2002) discretize sequential medical data and then extract temporal patterns. Hereafter, they use decision trees for rule discovery. Wang and Chan (2006) use two-layer bias decision trees for stock market trading rule discovery. Lastly, Lai et al. (2009) employ fuzzy decision trees.

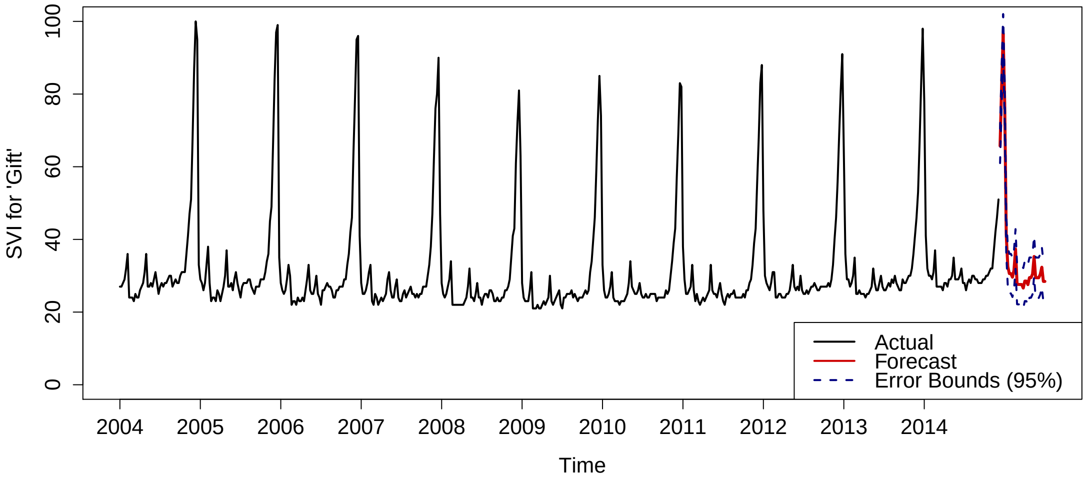

Figure 8: Prediction of search volume index for the query term “gift” with $n = 569$ by fitting a SARIMA$(2, 0, 4) \times (1, 0, 0)_{52}$ and predicting 30 weeks ahead. Error bounds at 95% confidence level are included.

TSDMPred

4.5 Prediction

Prediction or forecasting of the next few values of a time series is an extensively applied task. A postulated causal relationship between variables is modeled and estimated from the data in order to forecast the next values of a time series. Prediction of time series values is a distinct and extensive research area uncoupled from data mining and many reviews and standard references exist (see for example Hamilton (1994) or Brockwell (2002)). Nevertheless, prediction is one of the major ends of data science and needs to be discussed here shortly. The most frequently used prediction techniques are ARMA models and more specifically *S*easonal *A*uto*R*egressive *I*ntegrated *M*oving *A*verage (SARIMA) models (Brockwell, 2002). For illustration, Figure 8 shows a 30 week ahead prediction of the search volume index for the query term “gift”. Firstly, we identify possible SARIMA model candidates considering e.g. the (P)ACF and then apply model selection criteria like AIC. As Christmas is *the* season for making gifts, most people tend to google for gift ideas in December. Therefore, we can clearly see a seasonal structure in the SVI for the query term “gift”. In order to account for this seasonality, we fit a SARIMA model to the data and predict the SVI 30 weeks ahead.

Besides SARIMA models, frequently used forecasting approaches are neural networks (Hill et al., 1996), *S*elf *O*rganizing *M*aps, (SOM, see Barreto (2007) for a review) or hidden Markov models (Hassan and Nath, 2005). Moreover, there are prediction techniques especially tailored to time series data mining methods. Ahmed et al. (2010) review all machine learning approaches applicable to time series forecasting, including multilayer perceptron, Bayesian neural networks, radial basis functions, Kernel regression, k-nearest neighbor regression, regression trees, support vector regression and Gaussian process regression. They show that Gaussian process regressions and multilayer perceptrons are the best methods. Liu et al. (2004) use wavelets and radial basis functions for prediction. Sorjamaa et al. (2007) aim to tackle the problem of long-term prediction and combine a direct prediction strategy with sophisticated input selection criteria including k-nearest neighbor approximation and nonparametric noise estimation. Wagner et al. (2007) present a dynamic genetic program model which is especially designed for forecasting streaming data. Goerg (2013) proposes a dimension reduction technique for multivariate time series which focuses on predictability. The *Fore*castable *C*omponent *A*nalysis (ForeCA) approach reduces the dimension of multivariate time series with the constraint for the most forecastable subspace. Goerg (2013) shows that a lower entropy of the spectral density implies a better predictable signal and therefore minimizes the entropy of the spectral density. The self-organizing *P*redictable *F*eature *A*nalysis (PFA) by Richthofer and Wiskott (2013) bases on similar ideas but searches for best predictable systems and not best predictable single components as ForeCA. Shah (2012) propose to use fuzzy based methods for the prediction task for non-stationary time series. Tsinaslanidis and Kugiumtzis (2014) firstly segment the time series into subsequences using *P*erceptually *I*mportant *P*oints (PIP) and then search similar subsequences using *D*ynamic *T*ime *W*arping (DTW). The mapping of the most similar subsequences are used for prediction afterwards. Lastly, unsupervised learning algorithms for time series data mining are not very developed yet. Kattan et al. (2015) recently proposed an unsupervised learning algorithm which is based on genetic programming for an event based prediction of time series objects.

5 Recent Research

In spite of the ever growing importance of data mining techniques and the massive information flow and data collection, it is rather surprising that time series data mining methods are in the early stages of development. Desirable data mining techniques like un- or semisupervised learning are not well achieved yet. For that reason, recent research activities in the time series data mining community are pulsing.

Firstly, recent research strives for more elaborated *multivariate time series data mining* techniques. Many of the reviewed time series data mining approaches only work well for univariate time series. Especially clustering and classification approaches for mining multivariate time series were proposed manifoldly in recent years. Plant et al. (2009), Wang et al. (2007) and Singhal and Seborg (2005) address the clustering problem of finding natural groupings in multivariate time series. Formisano et al. (2008), Batal et al. (2009), Esmael et al. (2012) and Batal et al. (2013) propose classification methods for multivariate time series. Lee et al. (2000) perform similarity searches in multivariate time series and Kahveci et al. (2002) in multiattribute time series respectively. Segmentation tasks in multivariate time series are demonstrated by Chamroukhi et al. (2013), Omranian et al. (2013) and Guo et al. (2015). Minnen et al. (2007a), Minnen et al. (2007b), Mörchen and Ultsch (2007), Lee et al. (2009) and Batal et al. (2011) inspect multivariate motif discovery and pattern mining. Tatavarty et al. (2007) and Batal et al. (2012) especially investigate temporal pattern mining in multivariate time seris. Moreover, Povinelli and Feng (1999) design an identification approach for temporal patterns in multiple nonstationary time series. Finally, Shibuya et al. (2009) search for causalities between multivariate time series for prediction purposes.

Un- or semisupervised learning techniques for time series data mining are in an earlier stage of development. Well known un- or semisupervised learning techniques are for example hidden Markov models, state-space models (see for example Ghahramani (2004)) and self-organizing maps (Kohonen, 1990). Naftel and Khalid (2006) use self-organizing maps for learning similarities between object trajectories in motion classification. Moreover, there are newly proposed semi-supervised classification algorithms for time series data mining which can detect patterns with no predefined labels. Wei and Keogh (2006) for instance propose a semi-supervised time series classifier. Kasabov (2001) and Kasabov and Song (2002) aim to take a step towards a hybrid (supervised and unsupervised) learning through dynamic evolving fuzzy neural networks. Unsupervised learning algorithms for time series data mining are even less developed yet. Cao et al. (2003) strive for unsupervised learning for facial motions in speech recognition based on *I*ndependent *C*omponent *A*nalysis (ICA). Kattan et al. (2015) recently proposed an unsupervised learning algorithm which is based on genetic programming for an event based prediction of time series objects.

In the era of big data, massive amounts of personal data is collected and traded with only a few effective controls over how it is used or secured. Therefore, a multidisciplinary examination of *data security* and *privacy concerns* is imperiously essential. And besides legislation and other concerned parties, the data mining community picks up privacy concerns as well. Agrawal and Srikant (2000) are the first to suggest data mining methods with privacy-preserving features. They aim to develop accurate models without the access to precise individual data records. Zhu et al. (2008) outline the needed transition of privacy concerned methods from cross sectional to time series data mining. In particular, they point out that attackers could compromise time-domain privacy by a so called data flow separation attack. Lian et al. (2008) perform pattern matching over cloaked time series. Nin and Torra (2009) propose a framework to evaluate time series privacy protection methods. In recent years, privacy concerns are very pronounced in cases where internet users are not aware of their exposure or simply cannot protect their information. Hence, Nguyen and Jung (2014) focus on the detection of hot topics on Twitter with privacy concerns in mind. Zhang et al. (2014a) are concerned with time series privacy protection on cloud computing by noise obfuscation, i.e. a noise generation strategy for deception. Beyond that, medical research areas are highly concerned with privacy issues. People with certain diseases might not want to disclose their non-anonymized medical time series. Mi-Jung et al. (2014) work with *E*lectro*C*ardio*G*ram (ECG) time series from cardiology and try to balance privacy concerns as well as mining accuracy.

6 A Thousand Applications

Big Data is a buzzword which found its way into mass media a few years ago and its relative importance increases heavily. Newspaper articles are labeling data science as the hottest career choice. Varian (2009), chief economist of Google Inc., commented that "the sexy job in the next ten years will be statistician". Considering the various sources of big data in real life, this trend is not surprising. Big data is produced and present in almost every area of life and industry. The technological progress allows the collection and storage of information in massive data sets. In almost every application domain, not only cross sectional but above all time series data is generated at an unprecedented speed. From biological experiments, stock markets, up to records of sensor networks, innumerable data origins exist.

One common group of applications is concerned with genomics. In the meanwhile, whole genome sequencing is less expensive and a high throughput is possible. Thus, massive genome sequencing datasets are available and are used to uncover common genetic patterns of rare disorders. For example, Chen et al. (2000) construct a time series similarity measure originally designed for genetic sequences.

Rapid developments in neuroimaging techniques such as fMRI allow to monitor various gene and protein functions simultaneously. fMRI is a technique in cognitive and psychological neuroscience and delivers massive sized high-resolution brain images. These fMRI images are “3D over time” and contain hundreds of thousands of voxels. For example Goutte et al. (1999), Norman et al. (2006), Formisano et al. (2008), van Bömmel et al. (2014) and Zhang et al. (2014b) address the statistical analysis of fMRI data.

A further group of real world examples stems from economics and finance. Financial markets produce data about stock prices over time, currencies, derivative trades, unstructured news, business sentiments, consumer choices and many more. Partly, this data has a very high frequency and new observations are made during a tiny fraction of a second. The availability of such high frequency financial data shaped its very own research directions in statistics and econometrics. The interested reader may consult for example Hautsch (2011).

Sensors are well-known data recording devices which deliver a plethora of information. The data is arriving ordered in time or even as data streams, i.e. continuous rapid data records. And still, the data collection rates continue to improve with improving sensor technologies. The fields in which sensors or sensor networks are employed range from medicine to physics or even music. For example in astrophysics, telescopes trace burst activities in a far-away galaxy. In this field, a more particularized example is the astronomical project *O*ptical *G*ravitational *L*ensing *E*xperiment (OGLE, see Udalski (2004) and Szymanski (2006)). In medicine, sensors facilitate real-time surgery with magnetic resonance. The analysis and mining of data stemming from sensor networks is a very own subdiscipline of (time series) data mining. Firstly, the analysis of streaming data needs its own rules (see Section 2.1). And secondly, the correlation between the time series of sensors which are located close to each other in the network is a crucial characteristic that has to be accounted for. Deligiannakis et al. (2004) for instance propose a dimension reduction approach exclusively dedicated to data stemming from sensor networks.

A final example stems from a music application. The music retrieval system *Q*uery *b*y *H*umming (QbH) allows a (probably little musical) user to find a song by humming parts of it. Lerner et al. (2004) present a short list of the special QbH problems from the time series data mining point of view. QbH is an especially interesting example for time series data mining as most users might hum at inconsistent tempos (too slow, too fast, accelerating or decelerating) and at a wrong altitude. Therefore, the aim is to compare patterns and this is achieved best by *D*ynamic *T*ime *W*arping (DTW, see Section 3.5.1) as DTW is specifically designed to tackle these challenges.

7 Conclusion

In times where data or *big data* is labeled as the new natural resource of the century, the importance of data mining and according techniques is ever growing. The furious development of technology enables us to collect and store massive sized and complex data sets. Therefore, real world time series data sets can take a size up to a trillion observations and even more. The overall goal is to detect new information that is hidden in these massive data sets.

This review gives an overview of the challenges of large time series and the proposed problem solving approaches from time series data mining community. The lack of well established (best) practices in time series data mining becomes evident as a plethora of sparsely systematized different approaches exist and they are accepted alike in spite of their very distinct strategies. This review contributes to that aspect as we collect, systemize and describe all generally accepted time series data mining methods.

References

Agrawal, R., Faloutsos, C., and Swami, A. (1993a). *Efficient similarity search in sequence databases*. Springer.

Agrawal, R., Imieliński, T., and Swami, A. (1993b). Mining association rules between sets of items in large databases. In *ACM SIGMOD Record*, volume 22, pages 207–216. ACM.

Agrawal, R. and Srikant, R. (2000). Privacy-preserving data mining. *ACM Sigmod Record*, 29(2):439–450.

Ahmed, N. K., Atiya, A. F., Gayar, N. E., and El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29(5-6):594–621.

Ale, J. M. and Rossi, G. H. (2000). An approach to discovering temporal association rules. In *Proceedings of the 2000 ACM symposium on Applied computing*, volume 1, pages 294–300. ACM.

Allen, F. and Karjalainen, R. (1999). Using genetic algorithms to find technical trading rules. *Journal of financial Economics*, 51(2):245–271.

Aref, W. G., Elfeky, M. G., and Elmagarmid, A. K. (2004). Incremental, online, and merge mining of partial periodic patterns in time-series databases. *Knowledge and Data Engineering, IEEE Transactions on*, 16(3):332–342.

Aßfalg, J., Kriegel, H.-P., Kröger, 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*, pages 276–294. Springer.

Aßfalg, J., Kriegel, H.-P., Kröger, P., Kunath, P., Pryakhin, A., and Renz, M. (2008). Similarity search in multimedia time series data using amplitude-level features. In *Advances in multimedia modeling*, pages 123–133. Springer.

Azzouzi, M. and Nabney, I. T. (1998). Analysing time series structure with hidden markov models.

Bagnall, A., 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.

Bajcsy, R., Solina, F., and Gupta, A. (1990). *Segmentation versus object representation—are they separable?* Springer.

Bao, D. and Yang, Z. (2008). Intelligent stock trading system by turning point confirming and probabilistic reasoning. *Expert Systems with Applications*, 34(1):620–627.

Barreto, G. A. (2007). Time series prediction with the self-organizing map: A review. In *Perspectives of neural-symbolic integration*, pages 135–158. Springer.

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

Batal, I., Fradkin, D., Harrison, J., Moerchen, F., and Hauskrecht, M. (2012). Mining recent temporal patterns for event detection in multivariate time series data. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 280–288. ACM.

Batal, I., Sacchi, L., Bellazzi, R., and Hauskrecht, M. (2009). Multivariate time series classification with temporal abstractions. In *FLAIRS Conference*.

Batal, I., Valizadegan, H., Cooper, G. F., and Hauskrecht, M. (2011). A pattern mining approach for classifying multivariate temporal data. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 358–365. IEEE.

Batal, I., Valizadegan, H., Cooper, G. F., and Hauskrecht, M. (2013). A temporal pattern mining approach for classifying electronic health record data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(4):63.

Beckmann, N., Kriegel, H.-P., Schneider, R., and Seeger, B. (1990). *The R*-tree: an efficient and robust access method for points and rectangles*, volume 19. ACM.

Begum, N. and Keogh, E. (2014). Rare time series motif discovery from unbounded streams. *Proceedings of the VLDB Endowment*, 8(2).

Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer.

Berndt, D. J. and Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA.

Bicego, M., Murino, V., and Figueiredo, M. A. (2003). Similarity-based clustering of sequences using hidden markov models. In *Machine learning and data mining in pattern recognition*, pages 86–95. Springer.

Brockwell, P. J. (2002). *Introduction to time series and forecasting*, volume 1. Taylor & Francis.

Brockwell, P. J. and Davis, R. A. (2009). *Time series: theory and methods*. Springer Science & Business Media.

Burrus, C. S., Gopinath, R. A., and Guo, H. (1998). *Introduction to wavelets and wavelet transforms*, volume 998. Prentice Hall New Jersey.

Cao, Y., Faloutsos, P., and Pighin, F. (2003). Unsupervised learning for speech motion editing. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 225–231. Eurographics Association.

Chamroukhi, F., Mohammed, S., Trabelsi, D., Oukhellou, L., and Amirat, Y. (2013). Joint segmentation of multivariate time series with hidden process regression for human activity recognition. *Neurocomputing*, 120:633–644.

Chan, K.-P. and Fu, A.-C. (1999). Efficient time series matching by wavelets. In *Proceedings of the 15th International Conference on Data Engineering*, pages 126–133. IEEE.

Chan, P. K. and Mahoney, M. V. (2005). Modeling multiple time series for anomaly detection. In *Fifth IEEE International Conference on Data Mining*, pages 90–97. IEEE.

Chen, J. R. (2005). Making subsequence time series clustering meaningful. In *Fifth IEEE International Conference on Data Mining*, pages 114–121. IEEE.

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, pages 792–803. VLDB Endowment.

Chen, L., Özsu, M. T., 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*, pages 491–502. ACM.

Chen, Q., Chen, L., Lian, X., Liu, Y., and Yu, J. X. (2007a). Indexable pla for efficient similarity search. In *Proceedings of the 33rd international conference on Very large data bases*, pages 435–446. VLDB Endowment.

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*, page 107. ACM.

Chen, Y., Nascimento, M. A., Ooi, B. C., and Tung, A. K. (2007b). Spade: On shape-based pattern detection in streaming time series. In *ICDE IEEE 23rd International Conference on Data Engineering*, pages 786–795. IEEE.

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*, pages 493–498. ACM.

Cho, H. and Fryzlewicz, P. (2012). Multiscale and multilevel technique for consistent segmentation of nonstationary time series. *Statistica Sinica*, 22(1):207–229.

Chu, K. K. W. and Wong, M. H. (1999). Fast time-series searching with scaling and shifting. In *Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 237–248. ACM.

Chu, S., Keogh, E. J., Hart, D. M., Pazzani, M. J., et al. (2002). Iterative deepening dynamic time warping for time series. In *SDM*, pages 195–212. SIAM.

Chung, F.-L., Fu, T.-C., Luk, R., and Ng, V. (2001). Flexible time series pattern matching based on perceptually important points. In *International joint conference on artificial intelligence workshop on learning from temporal and spatial data*, pages 1–7.

Cormode, G., Muthukrishnan, S., and Zhuang, W. (2007). Conquering the divide: Continuous clustering of distributed data streams. In *ICDE IEEE 23rd International Conference on Data Engineering*, pages 1036–1045. IEEE.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Cotofrei, P. and Stoffel, K. (2002a). Classification rules+ time= temporal rules. In *Computational Science—ICCS 2002*, pages 572–581. Springer.

Cotofrei, P. and Stoffel, K. (2002b). Rule extraction from time series databases using classification trees. In *APPLIED INFORMATICS-PROCEEDINGS*, number 2, pages 327–332.

Dai, D.-m. and Mu, D.-j. (2012). A fast approach to k-means clustering for time series based on symbolic representation. *International Journal of Advancements in Computing Technology*, 4(5).

Das, G., Gunopulos, D., and Mannila, H. (1997). Finding similar time series. In *Principles of Data Mining and Knowledge Discovery*, pages 88–100. Springer.

Das, G., Lin, K.-I., Mannila, H., Renganathan, G., and Smyth, P. (1998). Rule discovery from time series. In *KDD*, volume 98, pages 16–22.

Deligiannakis, A., Kotidis, Y., and Roussopoulos, N. (2004). Compressing historical information in sensor networks. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 527–538. ACM.

Denton, A. (2005). Kernel-density-based clustering of time series subsequences using a continuous random-walk noise model. In *Fifth IEEE International Conference on Data Mining*, pages 122–129. IEEE.

Denton, A. M., Besemann, C. A., and Dorr, D. H. (2009). Pattern-based time-series subsequence clustering using radial distribution functions. *Knowledge and Information Systems*, 18(1):1–27.

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.

Dobos, L. and Abonyi, J. (2012). On-line detection of homogeneous operation ranges by dynamic principal component analysis based time-series segmentation. *Chemical Engineering Science*, 75:96–105.

Duda, R. O., Hart, P. E., et al. (1973). *Pattern classification and scene analysis*, volume 3. Wiley New York.

Duncan, S. R. and Bryant, G. F. (1996). A new algorithm for segmenting data from time series. In *Proceedings of the 35th IEEE Conference on Decision and Control*, volume 3, pages 3123–3128. IEEE.

Elfeky, M. G., Aref, W. G., and Elmagarmid, A. K. (2005). Periodicity detection in time series databases. *Knowledge and Data Engineering, IEEE Transactions on*, 17(7):875–887.

Esling, P. and Agon, C. (2012). Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):12.

Esmael, B., Arnaout, A., Fruhwirth, R. K., and Thonhauser, G. (2012). Multivariate time series classification by combining trend-based and value-based approximations. In *Computational Science and Its Applications–ICCSA 2012*, pages 392–403. Springer.

Euliano, N. R. and Principe, J. C. (1996). Spatio-temporal self-organizing feature maps. In *IEEE International Conference on Neural Networks*, volume 4, pages 1900–1905. IEEE.

Fagin, R., Nievergelt, J., Pippenger, N., and Strong, H. R. (1979). Extendible hashing—a fast access method for dynamic files. *ACM Transactions on Database Systems (TODS)*, 4(3):315–344.

Faloutsos, C., Ranganathan, M., and Manolopoulos, Y. (1994). *Fast subsequence matching in time-series databases*, volume 23. ACM.

Fan, J., Han, F., and Liu, H. (2014). Challenges of big data analysis. *National Science Review*, pages 293–314.

Fancoua, C. and Principe, J. C. (1996). A neighborhood map of competing one step predictors for piecewise segmentation and identification of time series. In *IEEE International Conference on Neural Networks*, volume 4, pages 1906–1911. IEEE.

Fancourt, C. L. and Principe, J. C. (1998). Competitive principal component analysis for locally stationary time series. *Signal Processing, IEEE Transactions on*, 46(11):3068–3081.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (1996). Advances in knowledge discovery and data mining.

Fitzgibbon, L. J., Dowe, D. L., and Allison, L. (2002). Change-point estimation using new minimum message length approximations. In *PRICAI 2002: Trends in Artificial Intelligence*, pages 244–254. Springer.

Floratou, A., Tata, S., and Patel, J. M. (2011). Efficient and accurate discovery of patterns in sequence data sets. *Knowledge and Data Engineering, IEEE Transactions on*, 23(8):1154–1168.

Formisano, E., De Martino, F., and Valente, G. (2008). Multivariate analysis of fmri time series: classification and regression of brain responses using machine learning. *Magnetic resonance imaging*, 26(7):921–934.

Frentzos, E., Gratsias, K., and Theodoridis, Y. (2007). Index-based most similar trajectory search. In *ICDE IEEE 23rd International Conference on Data Engineering*, pages 816–825. IEEE.

Fröhwirth-Schnatter, S. and Kaufmann, S. (2008). Model-based clustering of multiple time series. *Journal of Business & Economic Statistics*, 26(1):78–89.

Fu, A. W.-C., Keogh, E., Lau, L. Y., Ratanamahatana, C. A., and Wong, R. C.-W. (2008). Scaling and time warping in time series querying. *International Journal on Very Large Data Bases*, 17(4):899–921.

Fu, T.-c. (2011). A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181.

Fu, T.-c., Chung, F.-l., and Ng, C.-m. (2006). Financial time series segmentation based on specialized binary tree representation. In *DMIN*, pages 3–9.

Fu, T.-c., Chung, F.-l., Ng, V., and Luk, R. (2001). Pattern discovery from stock time series using self-organizing maps. In *Workshop Notes of KDD2001 Workshop on Temporal Data Mining*, pages 26–29. Citeseer.

Gaber, M. 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*, pages 63–72. ACM.

Ge, X. and Smyth, P. (2000). Deformable markov model templates for time-series pattern matching. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 81–90. ACM.

Geurts, P. (2001). Pattern extraction for time series classification. In *Principles of Data Mining and Knowledge Discovery*, pages 115–127. Springer.

Ghahramani, Z. (2004). Unsupervised learning. In *Advanced Lectures on Machine Learning*, pages 72–112. Springer.

Goerg, G. (2013). Forecastable component analysis. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 64–72.

Golab, L. and Özsu, M. T. (2003). Issues in data stream management. *ACM Sigmod Record*, 32(2):5–14.

Goldin, D. Q. and Kanellakis, P. C. (1995). On similarity queries for time-series data: constraint specification and implementation. In *Principles and Practice of Constraint Programming—CP'95*, pages 137–153. Springer.

Google (2014). Google trends data. `http://www.google.com/trends/`. Accessed: 2014-12-12.

Goutte, C., Toft, P., Rostrup, E., Nielsen, F. Å., and Hansen, L. K. (1999). On clustering fmri time series. *NeuroImage*, 9(3):298–310.

Graps, A. (1995). An introduction to wavelets. *Computational Science & Engineering, IEEE*, 2(2):50–61.

Guo, H., Liu, X., and Song, L. (2015). Dynamic programming approach for segmentation of multivariate time series. *Stochastic Environmental Research and Risk Assessment*, 29(1):265–273.

Guralnik, V. and Srivastava, J. (1999). Event detection from time series data. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 33–42. ACM.

Guttman, A. (1984). *R-trees: a dynamic index structure for spatial searching*, volume 14. ACM.

Hamilton, J. D. (1994). *Time series analysis*, volume 2. Princeton University Press.

Han, J., Dong, G., and Yin, Y. (1999). Efficient mining of partial periodic patterns in time series database. In *Proceedings of the 15th International Conference on Data Engineering*, pages 106–115. IEEE.

Han, J., Gong, W., and Yin, Y. (1998). Mining segment-wise periodic patterns in time-related databases. In *KDD*, pages 214–218.

Hao, M. C., Marwah, M., Janetzko, H., Dayal, U., Keim, D. A., Patnaik, D., Ramakrishnan, N., and Sharma, R. K. (2012). Visual exploration of frequent patterns in multivariate time series. *Information Visualization*, 11(1):71–83.

Hassan, M. R. and Nath, B. (2005). Stock market forecasting using hidden markov model: a new approach. In *ISDA Proceedings of the 5th International Conference on Intelligent Systems Design and Applications*, pages 192–196. IEEE.

Hautsch, N. (2011). *Econometrics of financial high-frequency data*. Springer.

Haykin, S. (2004). *Neural Networks: A comprehensive foundation*, volume 2. Prentice Hall PTR.

Hill, T., O'Connor, M., and Remus, W. (1996). Neural network models for time series forecasts. *Management science*, 42(7):1082–1092.

Himberg, J., Korpiaho, K., Mannila, H., Tikanmaki, J., and Toivonen, H. T. (2001). Time series segmentation for context recognition in mobile devices. In *ICDM Proceedings IEEE International Conference on Data Mining*, pages 203–210. IEEE.

Hipp, J., Güntzer, U., and Nakhaeizadeh, G. (2000). Algorithms for association rule mining—a general survey and comparison. *ACM sigkdd explorations newsletter*, 2(1):58–64.

Hochheiser, H. and Shneiderman, B. (2004). Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18.

IBM (2014). Why big data is the new natural resource. `http://www.forbes.com/sites/ibm/2014/06/30/why-big-data-is-the-new-natural-resource/`. Accessed: 2015-03-21.

Indyk, P., Koudas, N., and Muthukrishnan, S. (2000). Identifying representative trends in massive time series data sets using sketches. In *VLDB*, pages 363–372.

Internetlivestats.com (2014). `www.internetlivestats.com/google-search-statistics/`. Accessed: 2015-03-21.

Itakura, F. (1975). Minimum prediction residual principle applied to speech recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 23(1):67–72.

Izakian, H. and Pedrycz, W. (2014). Anomaly detection and characterization in spatial time series data: A cluster-centric approach.

Janacek, G. J., Bagnall, A. J., and Powell, M. (2005). A likelihood ratio distance measure for the similarity between the fourier transform of time series. In *Advances in Knowledge Discovery and Data Mining*, pages 737–743. Springer.

Jiang, J., Zhang, Z., and Wang, H. (2007). A new segmentation algorithm to stock time series based on pip approach. In *WiCom, International Conference on Wireless Communications, Networking and Mobile Computing*, pages 5609–5612. IEEE.

Kadam, S. and Thakore, D. (2012). High dimensional data mining in time series by reducing dimensionality and numerosity. *International Journal of Computer Technology & Applications*, 3(3).

Kahveci, T., Singh, A., and Gurel, A. (2002). Similarity searching for multi-attribute sequences. In *Proceedings of the 14th International Conference on Scientific and Statistical Database Management*, pages 175–184. IEEE.

Kahveci, T. and Singh, A. K. (2004). Optimizing similarity search for arbitrary length time series queries. *Knowledge and Data Engineering, IEEE Transactions on*, 16(4):418–433.

Kalpakis, K., Gada, D., and Puttagunta, V. (2001). Distance measures for effective clustering of arima time-series. In *ICDM, Proceedings IEEE International Conference on Data Mining*, pages 273–280. IEEE.

Kasabov, N. (2001). Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 31(6):902–918.

Kasabov, N. K. and Song, Q. (2002). Denfis: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *Fuzzy Systems, IEEE Transactions on*, 10(2):144–154.

Kattan, A., Fatima, S., and Arif, M. (2015). Time-series event-based prediction: An unsupervised learning framework based on genetic programming. *Information Sciences*.

Kennedy, J. (2010). Particle swarm optimization. In *Encyclopedia of Machine Learning*, pages 760–766. Springer.

Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S. (2001a). Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems*, 3(3):263–286.

Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S. (2001b). Locally adaptive dimensionality reduction for indexing large time series databases. *ACM SIGMOD Record*, 30(2):151–162.

Keogh, E., Chu, S., Hart, D., and Pazzani, M. (2001c). An online algorithm for segmenting time series. In *ICDM, Proceedings IEEE International Conference on Data Mining*, pages 289–296. IEEE.

Keogh, E., Chu, S., Hart, D., and Pazzani, M. (2004a). Segmenting time series: A survey and novel approach. *Data mining in time series databases*, 57:1–22.

Keogh, E., Chu, S., and Pazzani, M. (2001d). Ensemble-index: A new approach to indexing large databases. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 117–125. ACM.

Keogh, E., Hochheiser, H., and Shneiderman, B. (2002a). An augmented visual query mechanism for finding patterns in time series data. In *Flexible Query Answering Systems*, pages 240–250. Springer.

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

Keogh, E., Lin, J., Lee, S.-H., and Van Herle, H. (2007). Finding the most unusual time series subsequence: algorithms and applications. *Knowledge and Information Systems*, 11(1):1–27.

Keogh, E., Lonardi, S., and Chiu, B.-c. (2002b). Finding surprising patterns in a time series database in linear time and space. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 550–556. ACM.

Keogh, E., Lonardi, S., and Ratanamahatana, C. A. (2004b). Towards parameter-free data mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 206–215. ACM.

Keogh, E. and Ratanamahatana, C. A. (2005). Exact indexing of dynamic time warping. *Knowledge and information systems*, 7(3):358–386.

Keogh, E. J. and Pazzani, M. J. (1998). An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *KDD*, volume 98, pages 239–243.

Keogh, E. J. and Pazzani, M. J. (1999). Relevance feedback retrieval of time series data. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 183–190. ACM.

Keogh, E. J. and Pazzani, M. J. (2000a). Scaling up dynamic time warping for datamining applications. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 285–289. ACM.

Keogh, E. J. and Pazzani, M. J. (2000b). A simple dimensionality reduction technique for fast similarity search in large time series databases. In *Knowledge Discovery and Data Mining. Current Issues and New Applications*, pages 122–133. Springer.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480.

Kohonen, T. (2001). *Self-organizing maps*, volume 30. Springer Science & Business Media.

Korn, F., Jagadish, H. V., and Faloutsos, C. (1997). Efficiently supporting ad hoc queries in large datasets of time sequences. *ACM SIGMOD Record*, 26(2):289–300.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, pages 79–86.

Kumar, N., Nishanth, V., Eamonn, L., Stefano, K., Chotirat, L., Ratanamahatana, A., and Wei, L. (2005). Time-series bitmaps: a practical visualization tool for working with large time series databases. In *Proceedings of SIAM International Conference on Data Mining*. Citeseer.

Lai, R. K., Fan, C.-Y., Huang, W.-H., and Chang, P.-C. (2009). Evolving and clustering fuzzy decision tree for financial time series data forecasting. *Expert Systems with Applications*, 36(2):3761–3773.

Last, M., Klein, Y., and Kandel, A. (2001). Knowledge discovery in time series databases. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 31(1):160–169.

Law, M. H. and Kwok, J. T. (2000). Rival penalized competitive learning for model-based sequence clustering. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume 2, pages 195–198. IEEE.

Lee, A. J., Wu, H.-W., Lee, T.-Y., Liu, Y.-H., and Chen, K.-T. (2009). Mining closed patterns in multi-sequence time-series databases. *Data & Knowledge Engineering*, 68(10):1071–1090.

Lee, C.-H. L., Liu, A., and Chen, W.-S. (2006). Pattern discovery of fuzzy time series for financial prediction. *Knowledge and Data Engineering, IEEE Transactions on*, 18(5):613–625.

Lee, S.-L., Chun, S.-J., Kim, D.-H., Lee, J.-H., and Chung, C.-W. (2000). Similarity search for multidimensional data sequences. In *Proceedings of the 16th International Conference on Data Engineering*, pages 599–608. IEEE.

Leigh, W., Modani, N., Purvis, R., and Roberts, T. (2002). Stock market trading rule discovery using technical charting heuristics. *Expert Systems with Applications*, 23(2):155–159.

Leng, M., Yu, W., Wu, S., and Hu, H. (2013). Anomaly detection algorithm based on pattern density in time series. In *Emerging Technologies for Information Systems, Computing, and Management*, pages 305–311. Springer.

Lerner, A., Shasha, D., Wang, Z., Zhao, X., and Zhu, Y. (2004). Fast algorithms for time series with applications to finance, physics, music, biology, and other suspects. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 965–968. ACM.

Li, B., Tan, L., Zhang, J., and Zhuang, Z. (2000). Using fuzzy neural network clustering algorithm in the symbolization of time series. In *IEEE APCCAS*, pages 379–382. IEEE.

Li, H. (2014). Asynchronism-based principal component analysis for time series data mining. *Expert Systems with Applications*, 41(6):2842–2850.

Li, Y., Lin, J., and Oates, T. (2012). Visualizing variable-length time series motifs. In *SDM*, pages 895–906.

Lian, X., Chen, L., and Yu, J. X. (2008). Pattern matching over cloaked time series. In *IEEE 24th International Conference on Data Engineering*, pages 1462–1464. IEEE.

Liao, T. W. (2005). Clustering of time series data—a survey. *Pattern recognition*, 38(11):1857–1874.

Lin, J., Keogh, E., and Lonardi, S. (2005). Visualizing and discovering non-trivial patterns in large time series databases. *Information visualization*, 4(2):61–82.

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*, pages 2–11. ACM.

Lin, J., Vlachos, M., Keogh, E., and Gunopulos, D. (2004). Iterative incremental clustering of time series. In *Advances in Database Technology-EDBT 2004*, pages 106–122. Springer.

Lin, R. A. K.-l. and Shim, H. S. S. K. (1995). Fast similarity search in the presence of noise, scaling, and translation in time-series databases. *Very Large Data Bases. Zurich, Switzerland: University of Zurich, Swiss Federal Institute of Technology*, pages 490–501.

Liu, J. N., Kwong, R. W., and Bo, F. (2004). Chart patterns recognition and forecast using wavelet and radial basis function network. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 564–571. Springer.

Lonardi, J. L. E. K. S. and Patel, P. (2002). Finding motifs in time series. In *Proc. of the 2nd Workshop on Temporal Data Mining*, pages 53–68.

Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., Arnaldi, B., et al. (2007). A review of classification algorithms for eeg-based brain–computer interfaces. *Journal of neural engineering*, 4.

Lu, H., Han, J., and Feng, L. (1998). Stock movement prediction and n-dimensional inter-transaction association rules. In *Proc. of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, page 12.

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*, pages 613–618. ACM.

Marteau, P.-F. (2009). Time warp edit distance with stiffness adjustment for time series matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):306–318.

McLachlan, G. (2004). *Discriminant analysis and statistical pattern recognition*, volume 544. John Wiley & Sons.

Megalooikonomou, V., Wang, Q., Li, G., and Faloutsos, C. (2005). A multiresolution symbolic representation of time series. In *ICDE Proceedings. 21st International Conference on Data Engineering*, pages 668–679. IEEE.

Mi-Jung, C., Hea-Suk, K., Sanghun, L., and Moon, Y.-S. (2014). Secure publishing of ecg time series for privacy-preserving data mining.

Minnen, D., Isbell, C., Essa, I., and Starner, T. (2007a). Detecting subdimensional motifs: An efficient algorithm for generalized multivariate pattern discovery. In *ICDM Seventh IEEE International Conference on Data Mining*, pages 601–606. IEEE.

Minnen, D., Isbell, C. L., Essa, I., and Starner, T. (2007b). Discovering multivariate motifs using subsequence density estimation and greedy mixture learning. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, page 615.

Möller-Levet, C. S., Klawonn, F., Cho, K.-H., and Wolkenhauer, O. (2003). Fuzzy clustering of short time-series and unevenly distributed sampling points. In *Advances in Intelligent Data Analysis V*, pages 330–340. Springer.

Mörchen, F. and Ultsch, A. (2005). Optimizing time series discretization for knowledge discovery. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 660–665. ACM.

Mörchen, F. and Ultsch, A. (2007). Efficient mining of understandable patterns from multivariate interval time series. *Data Mining and Knowledge Discovery*, 15(2):181–215.

Morse, M. D. and Patel, J. M. (2007). An efficient and accurate method for evaluating time series similarity. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 569–580. ACM.

Mueen, A., Keogh, E., and Bigdely-Shamlo, N. (2009a). Finding time series motifs in disk-resident data. In *ICDM Ninth IEEE International Conference on Data Mining*, pages 367–376. IEEE.

Mueen, A., Keogh, E. J., Zhu, Q., Cash, S., and Westover, M. B. (2009b). Exact discovery of time series motifs. In *SDM*, pages 473–484. SIAM.

Naftel, A. and Khalid, S. (2006). Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space. *Multimedia Systems*, 12(3):227–238.

Nguyen, D. T. and Jung, J. E. (2014). Privacy-preserving discovery of topic-based events from social sensor signals: An experimental study on twitter. *The Scientific World Journal*, 2014.

Nin, J. and Torra, V. (2009). Towards the evaluation of time series protection methods. *Information Sciences*, 179(11):1663–1677.

Norman, K. A., Polyn, S. M., Detre, G. J., and Haxby, J. V. (2006). Beyond mind-reading: multi-voxel pattern analysis of fmri data. *Trends in cognitive sciences*, 10(9):424–430.

Nunthanid, P., Niennattrakul, V., and Ratanamahatana, C. A. (2012). Parameter-free motif discovery for time series data. In *9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–4. IEEE.

Oates, T., Firoiu, L., and Cohen, P. R. (1999). Clustering time series with hidden markov models and dynamic time warping. In *Proceedings of the IJCAI-99 workshop on neural, symbolic and reinforcement learning methods for sequence learning*, pages 17–21. Citeseer.

Ohsaki, M., Sato, Y., Yokoi, H., and Yamaguchi, T. (2002). A rule discovery support system for sequential medical data, in the case study of a chronic hepatitis dataset. In *Proc of the ECML/PKDD-2003 Discovery Challenge Workshop*, pages 154–165.

Oliver, J. J., Baxter, R. A., and Wallace, C. S. (1998). Minimum message length segmentation. In *Research and Development in Knowledge Discovery and Data Mining*, pages 222–233. Springer.

Oliver, J. J. and Forbes, C. S. (1997). *Bayesian approaches to segmenting a simple time series*. Citeseer.

Omranian, N., Klie, S., Mueller-Roeber, B., and Nikoloski, Z. (2013). Network-based segmentation of biological multivariate time series. *PloS one*, 8(5):e62974.

Otu, H. H. and Sayood, K. (2003). A new sequence distance measure for phylogenetic tree construction. *Bioinformatics*, 19(16):2122–2130.

Panuccio, A., Bicego, M., and Murino, V. (2002). A hidden markov model-based approach to sequential data clustering. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 734–743. Springer.

Papadimitriou, S., Sun, J., and Faloutsos, C. (2005). Streaming pattern discovery in multiple time-series. In *Proceedings of the 31st international conference on Very large data bases*, pages 697–708. VLDB Endowment.

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*, pages 647–658. ACM.

Park, S., Lee, D., and Chu, W. W. (1999). Fast retrieval of similar subsequences in long sequence databases. In *Knowledge and Data Engineering Exchange Proceedings*, pages 60–67. IEEE.

Patel, P., Keogh, E., Lin, J., and Lonardi, S. (2002). Mining motifs in massive time series databases. In *ICDM Proceedings IEEE International Conference on Data Mining*, pages 370–377. IEEE.

Plant, C., Wohlschlager, A., and Zherdin, A. (2009). Interaction-based clustering of multivariate time series. In *ICDM Ninth IEEE International Conference on Data Mining*, pages 914–919. IEEE.

Popivanov, I. and Miller, R. J. (2002). Similarity search over time-series data using wavelets. In *Proceedings of the 18th International Conference on Data Engineering*, pages 212–221. IEEE.

Povinelli, R. J. and Feng, X. (1999). Data mining of multiple nonstationary time series. In *Artificial Neural Networks in Engineering*, pages 511–516.

Povinelli, R. J., Johnson, M. T., Lindgren, A. C., and Ye, J. (2004). Time series classification using gaussian mixture models of reconstructed phase spaces. *Knowledge and Data Engineering, IEEE Transactions on*, 16(6):779–783.

Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Rafiei, D. and Mendelzon, A. (1997). Similarity-based queries for time series data. In *ACM SIGMOD Record*, volume 26, pages 13–25. ACM.

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., and Keogh, E. (2012). Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 262–270. ACM.

Rakthanmanon, T., Keogh, E. J., Lonardi, S., and Evans, S. (2011). Time series epenthesis: clustering time series streams requires ignoring some data. In *ICDM IEEE 11th International Conference on Data Mining*, pages 547–556. IEEE.

Ratanamahatana, C., Keogh, E., Bagnall, A. J., and Lonardi, S. (2005). A novel bit level time series representation with implication of similarity search and clustering. In *Advances in knowledge discovery and data mining*, pages 771–777. Springer.

Ratanamahatana, C. A. and Keogh, E. (2004). Everything you know about dynamic time warping is wrong. In *Third Workshop on Mining Temporal and Sequential Data*, pages 22–25.

Ratanamahatana, C. A., Lin, J., Gunopulos, D., Keogh, E., Vlachos, M., and Das, G. (2010). Mining time series data. In *Data Mining and Knowledge Discovery Handbook*, pages 1049–1077. Springer.

Raychaudhuri, S., Stuart, J. M., and Altman, R. B. (2000). Principal components analysis to summarize microarray experiments: application to sporulation time series. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, page 455. NIH Public Access.

Richthofer, S. and Wiskott, L. (2013). Predictable feature analysis. *arXiv preprint arXiv:1311.2503*.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14(5):465–471.

Roddick, J. F. and Spiliopoulou, M. (2002). A survey of temporal knowledge discovery paradigms and methods. *Knowledge and Data Engineering, IEEE Transactions on*, 14(4):750–767.

Rodríguez, J. J. and Alonso, C. J. (2004). Interval and dynamic time warping-based decision trees. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 548–552. ACM.

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

Sakurai, Y., Yoshikawa, M., and Faloutsos, C. (2005). Ftw: fast similarity search under the time warping distance. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 326–337. ACM.

Salvador, S. and Chan, P. (2007). Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580.

Serrà, J. and Arcos, J. L. (2015). Particle swarm optimization for time series motif discovery. *arXiv preprint arXiv:1501.07399*.

Shah, M. (2012). Fuzzy based trend mapping and forecasting for time series data. *Expert Systems with Applications*, 39(7):6351–6358.

Shahabi, C., Tian, X., and Zhao, W. (2000). Tsa-tree: A wavelet-based approach to improve the efficiency of multi-level surprise and trend queries on time-series data. In *Proceedings of the 12th International Conference on Scientific and Statistical Database Management*, pages 55–68. IEEE.

Shalizi, C. R., Shalizi, K. L., and Crutchfield, J. P. (2002). An algorithm for pattern discovery in time series. *arXiv preprint cs/0210025*.

Shatkay, H. and Zdonik, S. B. (1996). Approximate queries and representations for large data sequences. In *Proceedings of the Twelfth International Conference on Data Engineering*, pages 536–545. IEEE.

Shibuya, T., Harada, T., and Kuniyoshi, Y. (2009). Causality quantification and its applications: structuring and modeling of multivariate time series. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 787–796. ACM.

Shieh, J. and Keogh, E. (2008). i sax: indexing and mining terabyte sized time series. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–631. ACM.

Singhal, A. and Seborg, D. E. (2005). Clustering multivariate time-series data. *Journal of chemometrics*, 19(8):427–438.

Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., and Lendasse, A. (2007). Methodology for long-term prediction of time series. *Neurocomputing*, 70(16):2861–2869.

Struzik, Z. R. and Siebes, A. (1999). The haar wavelet transform in the time series similarity paradigm. In *Principles of Data Mining and Knowledge Discovery*, pages 12–22. Springer.

Szymanski, M. (2006). The optical gravitational lensing experiment. internet access to the ogle photometry data set: Ogle-ii bvi maps and i-band data. *arXiv preprint astro-ph/0602018*.

Tanaka, Y., Iwamoto, K., and Uehara, K. (2005). Discovery of time-series motif from multi-dimensional data based on mdl principle. *Machine Learning*, 58(2-3):269–300.

Tatavarty, G., Bhatnagar, R., and Young, B. (2007). Discovery of temporal dependencies between frequent patterns in multivariate time series. In *Computational Intelligence and Data Mining, CIDM, IEEE Symposium on*, pages 688–696. IEEE.

Ting, J., Fu, T.-C., and Chung, F.-l. (2006). Mining of stock data: intra-and inter-stock pattern associative classification. *Threshold*, 5(100):5–99.

Tsinaslanidis, P. E. and Kugiumtzis, D. (2014). A prediction scheme using perceptually important points and dynamic time warping. *Expert Systems with Applications*.

Udalski, A. (2004). The optical gravitational lensing experiment. real time data analysis systems in the ogle-iii survey. *arXiv preprint astro-ph/0401123*.

Udechukwu, A., Barker, K., and Alhajj, R. (2004). Discovering all frequent trends in time series. In *Proceedings of the winter international synposium on Information and communication technologies*, pages 1–6. Trinity College Dublin.

Ultsch, A. (1999). Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series. *Kohonen maps*, 46:33–46.

van Bömmel, A., Song, S., Majer, P., Mohr, P. N., Heekeren, H. R., and Härdle, W. K. (2014). Risk patterns and correlated brain activities. multidimensional statistical analysis of fmri data in economic decision making study. *Psychometrika*, 79(3):489–514.

Van Wijk, J. J. and Van Selow, E. R. (1999). Cluster and calendar based visualization of time series data. In *Information Visualization, Proceedings IEEE Symposium on*, pages 4–9. IEEE.

Varian, H. (2009). `http://www.mckinsey.com/insights/innovation/hal_varian_on_how_the_web_challenges_managers`. Accessed: 2015-03-20.

Vlachos, M., Hadjieleftheriou, M., Gunopulos, D., and Keogh, E. (2003a). Indexing multi-dimensional time-series with support for multiple distance measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 216–225. ACM.

Vlachos, M., Hadjieleftheriou, M., Gunopulos, D., and Keogh, E. (2006). Indexing multidimensional time-series. *International Journal on Very Large Data Bases*, 15(1):1–20.

Vlachos, M., Kollios, G., and Gunopulos, D. (2002). Discovering similar multidimensional trajectories. In *Proceedings. 18th International Conference on Data Engineering*, pages 673–684. IEEE.

Vlachos, M., Lin, J., Keogh, E., and Gunopulos, D. (2003b). A wavelet-based anytime algorithm for k-means clustering of time series. In *In Proc. Workshop on Clustering High Dimensionality Data and Its Applications*. Citeseer.

Vlachos, M., Meek, C., Vagena, Z., and Gunopulos, D. (2004). Identifying similarities, periodicities and bursts for online search queries. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 131–142. ACM.

Vlachos, M., Philip, S. Y., and Castelli, V. (2005). On periodicity detection and structural periodic similarity. In *SDM*, volume 5, pages 449–460. SIAM.

Wagner, N., Michalewicz, Z., Khouja, M., and McGregor, R. R. (2007). Time series forecasting for dynamic environments: the dyfor genetic program model. *Evolutionary Computation, IEEE Transactions on*, 11(4):433–452.

Wang, J.-L. and Chan, S.-H. (2006). Stock market trading rule discovery using two-layer bias decision tree. *Expert Systems with Applications*, 30(4):605–611.

Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., and Keogh, E. (2013). Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2):275–309.

Wang, X., Smith, K., and Hyndman, R. (2006). Characteristic-based clustering for time series data. *Data Mining and Knowledge Discovery*, 13(3):335–364.

Wang, X., Wirth, A., and Wang, L. (2007). Structure-based statistical features and multivariate time series clustering. In *ICDM Seventh IEEE International Conference on Data Mining*, pages 351–360. IEEE.

Wang, Z. J. and Willett, P. (2004). Joint segmentation and classification of time series using class-specific features. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(2):1056–1067.

Weber, M., Alexa, M., and Müller, W. (2001). Visualizing time-series on spirals. In *Information Visualization, IEEE Symposium on*, pages 7–7. IEEE Computer Society.

Wei, L. and Keogh, E. (2006). Semi-supervised time series classification. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 748–753. ACM.

Wei, L., Keogh, E., and Xi, X. (2006). Saxually explicit images: Finding unusual shapes. In *ICDM Sixth International Conference on Data Mining*, pages 711–720. IEEE.

Wei, L., Kumar, N., Lolla, V. N., Keogh, E. J., Lonardi, S., and Ratanamahatana, C. A. (2005). Assumption-free anomaly detection in time series. In *SSDBM*, volume 5, pages 237–242.

Wen, J., Liu, P., Chen, L., and Wang, L. (2014). Sparse representation for remote sensing images of long time sequences. In *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*, pages 1293–1296. IEEE.

Xi, X., Keogh, E., Shelton, C., Wei, L., and Ratanamahatana, C. A. (2006). Fast time series classification using numerosity reduction. In *Proceedings of the 23rd international conference on Machine learning*, pages 1033–1040. ACM.

Xiong, Y. and Yeung, D.-Y. (2004). Time series clustering with arma mixtures. *Pattern Recognition*, 37(8):1675–1689.

Yang, K. and Shahabi, C. (2005a). A multilevel distance-based index structure for multivariate time series. In *Temporal Representation and Reasoning, 2005. TIME 2005. 12th International Symposium on*, pages 65–73. IEEE.

Yang, K. and Shahabi, C. (2005b). On the stationarity of multivariate time series for correlation-based data analysis. In *Fifth IEEE International Conference on Data Mining*, pages 4–pp. IEEE.

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*, pages 844–853. ACM.

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, J. (2005). Generalized low rank approximations of matrices. *Machine Learning*, 61(1-3):167–191.

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*, pages 947–956. ACM.

Yi, B.-K. and Faloutsos, C. (2000). Fast time sequence indexing for arbitrary lp norms. VLDB.

Yi, B.-K., Jagadish, H., and Faloutsos, C. (1998). Efficient retrieval of similar time sequences under time warping. In *Proceedings of the 14th International Conference on Data Engineering*, pages 201–208. IEEE.

Yingchareonthawornchai, S., Sivaraks, H., Rakthanmanon, T., and Ratanamahatana, C. A. (2013). Efficient proper length time series motif discovery. In *IEEE 13th International Conference on Data Mining*, pages 1265–1270. IEEE.

Ypma, A. and Duin, R. P. (1997). Novelty detection using self-organizing maps. *Progress in connectionist-based information systems*, 2:1322–1325.

Zhang, G., Liu, X., and Yang, Y. (2014a). Time-series pattern based effective noise generation for privacy protection on cloud.

Zhang, L., Guindani, M., Versace, F., and Vannucci, M. (2014b). A spatio-temporal nonparametric bayesian variable selection model of fmri data for clustering correlated time courses. *NeuroImage*, 95:162–175.

Zhao, Y. and Zhang, S. (2006). Generalized dimension-reduction framework for recent-biased time series analysis. *Knowledge and Data Engineering, IEEE Transactions on*, 18(2):231–244.

Zhu, Y., Fu, Y., and Fu, H. (2008). On privacy in time series data mining. In *Advances in Knowledge Discovery and Data Mining*, pages 479–493. Springer.

Zhu, Y., Wu, D., and Li, S. (2007). A piecewise linear representation method of time series based on feature points. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 1066–1072. Springer.

Declaration of Authorship

I hereby confirm that I have authored this Master's thesis independently and without use of others than the indicated sources. All passages which are literally or in general matter taken out of publications or other sources are marked as such.

Berlin, March 25, 2015

Caroline Kleist