Temporal Discretization of medical time series - A comparative study

[1]Revital Azulay, [1]Robert Moskovitch, [1]Dima Stopel, [2]Marion Verduijn, [3]Evert de Jonge, and [1]Yuval Shahar
[1]Medical Informatics Research Center, Ben Gurion University, P.O.B. 653, Beer Sheva 84105, Israel
{robertmo,stopel,azorevi,yshahar}@bgu.ac.il
[2]Dept of Medical Informatics, [3]Dept of Intensive Care Medicine, Academic Medical Center - University of Amsterdam, P.O.B. 22700, 1100 DE Amsterdam, The Netherlands {m.verduijn,e.dejonge}@amc.uva.nl

Abstract

Discretization is widely used in data mining as a preprocessing step; discretization usually leads to improved performance. In time series analysis commonly the data is divided into time windows. Measurements are extracted from the time window into a vectorial representation and static mining methods are applied, which avoids an explicit analysis along time. Abstracting time series into meaningful time interval series enables to mine the data explicitly along time. Transforming time series into time intervals can be made through discretization and concatenation of equal value and adjacent time points. We compare in this study five discretization methods on a medical time series dataset. Persist, a temporal discretization method yields with the longest time intervals and lowest error rate.

1 Introduction

Time oriented data presents an exceptional opportunity to analyze data, having a better and more natural analysis. Often, features from time series, such as minimal value, are extracted and represented as vectors for further use in static data mining algorithms. This is made through windowing, in which the data is divided to time windows and measurements are extracted from the window. It is very hard to determine the window size and this approach avoids the explicit time representation. Converting time series to time intervals series presents a more compact representation of the time series, which enables an efficient analysis of the data and further mining operations explicitly along time [Moskovitch and Shahar, 2005]. However, to transform time series to time interval series a temporal abstraction method should be applied. This can be made through discretization and concatenation of the discretized values. In this study we present five types of discretization methods, three are static and two consider the time explicitly. For the task of mining time intervals we are interested in *long time intervals* and *low level of error* relative to the original dataset.

We start with a detailed background of time intervals mining, as the motivation for this study. Later we present temporal abstractions and discretization methods. In the methods section we present the methods we used in the study and finally we discuss the results and present our conclusions.

2 Background

2.1 Mining Time Intervals

The problem of mining time intervals, a relatively young field, is attracting a growing attention recently. Generally, the task is given a database of symbolic time intervals to extract repeating temporal patterns. One of the earliest works was made by Villafane et al [1999], which searches for *containments* of intervals in a multivariate symbolic interval series. Kam and Fu [2000] were the first to use all Allen's relations [Allen, 1983] to compose interval rules, in which the patterns are restricted to right concatenation of intervals to existing extended patterns, called *A1* patterns. Höppner [2001] introduced a method using Allen's relations to mine rules in symbolic interval sequences and the patterns are mined using an Apriori algorithm. Höppner uses a k^2 matrix to represent the relations of a k sized pattern. Additionally, Höppner proposes how to abstract the patterns or make them more specific. Winarko and Roddick [2005] rediscovered Höppner's method, but used only half of the matrix for the representation of a pattern, as well as added the option to discover constrained temporal patterns. Similar to Winarko and Roddick [2005], Papapetrou et al [2005] rediscovered the method of mining time intervals using Allen's relations. Their contribution was in presenting a novel mining method consisting on the SPAM sequential mining algorithm, which results in an enumeration tree; the tree spans all the discovered patterns.

A recent alternative to Allen's relations based methods surveyed earlier was presented by Mörchen [2006], in which time intervals are mined to discover coinciding multivariate time intervals, called Chords, and the repeating partially ordered chords called Phrases.

Mining time intervals offers many advantages over common time series analysis methods commonly applied on the raw time point data. These advantages include mainly, a significant reduction in the amount of data, since we mine summaries of the time series, based on temporal abstraction methods. In addition a restriction of short time window is not needed and unrestricted frequent patterns can be discovered. However, in order to enable mining of time series through time intervals the time series have to be abstracted to time intervals. This can be done based on knowledge acquired from a domain expert [Shahar, 1997] or based on automatic data driven discretization methods.

2.2 Temporal Abstraction

Temporal abstraction is the conversion of a time series to a more abstracted representation. This abstracted representation is usually more comprehensive to human and used as a preprocessing step to many knowledge discovery and data mining tasks. The *Knowledge Based Temporal Abstraction* (KBTA) presented by Shahar [1997], infers domain-specific interval-based abstractions from point-based raw data, based on domain-specific knowledge stored in a formal knowledge-base, e.g. the output abstraction of a set of time stamped hemoglobin measurements, include an episode of *moderate anemia* during the *past 6 weeks*. However, while the KBTA applies the temporal knowledge and creates abstractions that are meaningful to the domain expert, such knowledge is not always available. Moreover, the domain expert knowledge provided is not always the proper one for knowledge discovery and mining tasks, but rather for his routine activities, such as diagnosis. Thus, there are several automatic data driven methods which can be used for this task, which is the focus of this paper.

The task of temporal abstraction corresponds to the task of *segmenting* the time series and characterizing the data in each segment. *Segmenting time series* [Keogh *et al.*, 1993] is the task of representing a time series in a piecewise linear representation, which is the approximation of a time series length *n* with *k* straight lines, usually *k*<<*n*. Three common approaches for segmenting time series are: *Sliding Window* approach, in which a segment is grown until a specified error threshold is reached. *Top Down* approach repeatedly splitting the series according to best splitting point from all considered points, until a stopping criterion is met. *Bottom Up* approach starts by segmenting the series with small segments and then iteratively merges adjacent segments. A survey on temporal abstraction methods is given in [Höppner, 2002].

2.3 Discretization

Many data mining algorithms and tasks can benefit from a discrete representation of the original data set. Discrete representation is more comprehensive to human and can simplify, reduce computational costs and improve accuracy of many algorithms [Liu *et al.*, 2002]. Discretization is the process of transforming continuous space valued series $X = \{x_1,...,x_n\}$ into a discrete valued series $Y = \{y_1,...,y_n\}$. The next step is usually to achieve interval based representation of the discretized series. The main part of the discretization process is choosing the best *cut points* which split the continuous value range into discrete number of bins usually referred to as *states*. Discretization methods are mainly categorized as *supervised* vs. *unsupervised* methods.

Unsupervised discretization does not consider class information or any given label. For time series class information is usually not available and unsupervised methods are needed. Two common methods are *equal width discretization* (EWD) and *equal frequency discretization* (EFD). Another method is *k-means clustering* [MacQueen, 1967], in which the time series values are grouped into k clusters (states) represented by centroids, from which the states and the cut points are deduced.

Supervised discretization considers class information, which for time series is often unavailable. There are many supervised discretization methods available in the literature. Known methods for supervised discretization [Dougherty et al, 1995] are, decision tree discretization, heuristic methods and entropy minimization based methods consisting on Shannon entropy [Shannon, 1948]. Two common decision tree algorithms using entropy measure are ID3, [Quinlan, 1986], and C4.5, [Quinlan, 1993] and error based methods [Kohavi and Sahami, 1996]. A good survey and framework for discretization is given in [Liu et al., 2002]. In this study we will focus on the application of unsupervised discretization methods to time series.

2.4 Temporal Discretization

Temporal discretization refers to the discretization of time series, usually made by unsupervised means, as a preprocessing step in transforming the time series into time intervals series. Most of the discretization methods do not consider the temporal order of the values in a time series since most of them were developed for static data. However, recently several methods were proposed, in which the temporal order is considered. Symbolic Aggregate approXimation (SAX) [Lin et al., 2003] is a method for symbolic representation of time series. ***SAX*** was the first method developed explicitly to discretize time series, based on the Piecewise Aggregate Approximation (PAA) [Keogh et al., 2000] which is a time series segmenting algorithm. However, SAX does not explicitly consider the temporal order of the values. Later Mörchen and Ultsch [2005] introduced ***Persist*** which considers the temporal order of the time series and selects the best cut point based on persisting behavior of the discretized series. We will elaborate later on these two methods in the methods section. Another discretization method for time series is suggested by Dimitrova et al. [2005]. The method combines graph theory to create the initial discretization, and information theory to optimize this discretization. The number of states returned is a number determined by the method. The ***Gecko*** [Salvador, 2004] algorithm for identifying states in a time series is a clustering algorithm which dynamically determines the number of states. Another method is ***HMM*** [Bilmes, 1997], hidden Markov model. In HMM the time series is assumed to have been created by states which are hidden, this hidden model is assumed to be a Markov process. HMM is not a discretization method in the sense of resulting in a set of cut points, in HMM the state sequence directly created.

3 Methods

3.1 Discretization methods

We examined the following five discretization methods on medical time series. Apart from Persist, which considers the temporal order of the values in the time series, and

SAX, which was designed for time series discretization, all methods are static data discretization methods.

3.1.1 Equal Width Discretization

Equal Width Discretization (EWD) determines the cut points by dividing the value range into equal width bins, as shown in figure 1. Note the amount of values in each bin is based on the distribution of the values.

3.1.2 Equal Frequency Discretization

Equal Width Discretization (EFD) divides the value range into bins having equal frequency of values in each bin as shown in figure 1.

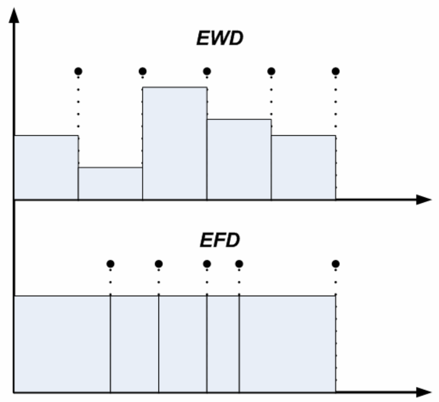

Figure 1. An illustration of the distributed values along the value range after EWD and EFD discretization.

3.1.3 K means Clustering

K-means clustering with Euclidean distance measure is a simple method that was chosen in this study as a representative for discretization method via clustering. K-means clusters the data into *k* clusters which are represented by the centroids. The clustering algorithm begins with a random or more educated choice (more efficient due to the sensitivity of the clustering process to the initial selection) of clusters centroids. The second step is to assign each data point to the cluster that has the closest centroid. After every data point has been assigned, the *k* centroids are recalculated as the mean value of each cluster. These two steps are repeated until no data point is reassigned or the *k* centroids no longer change. The resulting clusters or centroids are used as the states of the discretization process.

All the three methods presented EWD, EFD and K-means are all static methods, which do not consider the temporal order of the time series.

3.1.4 SAX

Symbolic Aggregate approXimation is one of the first discretization methods designed specifically for time series data. SAX consists of two steps, in the first step the time series is converted into a less granular representation and in the second step the abstracted time series is discretized into fixed number of states. The first step is the PAA, Piecewise Aggregate Approximation, in this step the temporal aspect of the data is taken into account. PAA is a representation of time series $X = \{x_1,..., x_n\}$ by the vector $\overline{X} = \{\overline{x}_1,..., \overline{x}_m\}$ ($m<n$), where each $\overline{x}_i$ is the mean value of n/m sequential observations of X. Two important PAA properties are dimensionality reduction and lower bounding. In dimensionality reduction, time series of length n are considered as a point in a n dimensional space, that can be reduced to a m dimensional *space* ($m<n$) after performing PAA dimensionality. In lower bounding, the distance between two PAA represented series is less or equal to the distance between the original two series, which guarantees no false dismissals; the PAA part of SAX is the time oriented part, which considers the temporal aspect. The second and main step of the SAX method, the discretization of the PAA output, is based on the assumption that normalized time series have a Gaussian distribution and the desire to produce equal probability states. Therefore the time series is normalized and discretized into fixed number of states according to predetermined cut points which produce equal-sized areas under Gaussian curve (the cut points chosen respectively to the selected number of states).

3.1.5 Persist

New univariate discretization method designed specifically for the purpose of knowledge discovery in time series, which for the first time explicitly considers the order of the values in the time series. Given a set of possible (discrete) symbols $S = \{S_1,\ldots,S_k\}$ of a time series of length n, Persist computes the marginal probability $P(S_j)$ of a symbol S_j and the transition probabilities given in a $k\times k$ matrix $A(j,m) = P(s_i=S_j|s_{i-1}=S_m)$, in which the self transitions are the values on the main diagonal of A. In this approach the assumption is that if there is no temporal structure in the time series, the symbols can be interpreted as independent observations of a random variable according to the marginal distribution of symbols, thus, the probability of observing each symbol is independent from the previous state, i.e. $P(s_i=S_j|s_{i-1},\ldots,s_{i-m}) = P(S_j|s_{i-1})$. Based on this Markovian model, if there is no temporal structure the transition probabilities should be close to the marginal probabilities. Otherwise if the states show persistence behavior, which is expected to result in long time intervals, the self transition probabilities will be higher than the marginal probabilities. The Persist algorithm is based on a measure based on the Kullback-Leibler Divergence [Kullback & Leibler, 1951], which indicates which cutoffs lead to a discretization which will result eventually in long time intervals. Persist method was compared to common discretization methods, and showed to achieve relatively good results. However Persist only deals with time series that comes from uniform sampling.

3.2 ICU Dataset

An ICU dataset of patients who underwent cardiac surgery at the Academic Medical Center in Amsterdam, the Netherlands, in the period of April 2002-May 2004. Two types of data were measured: *static data* including details on the patient, such as *age*, *gender*, *surgery type*, whether the patient was mechanically ventilated more than 24 hours, and *temporal data* which were used for the study. The temporal data, included two types of variables: *high frequency variables* (measured each minute): mean arterial blood pressure (ABPm), central venous pressure (CVP), heart rate (HR), body temperature (TMP), fraction inspired oxygen (FiO2) and level of positive end-expiratory pressure (PEEP). FiO2 and PEEP variables are parameters of the ventilator. The variables base excess (BE), creatinine kinase MB (CKMB), glucose value (GLUC), and cardiac output (CO) are *low frequency variables* (measured several times a day). The data contains 664 patients, among which 196 patients were mechanically ventilated for more than 24 hours.

3.3 Evaluation measures

Evaluating unsupervised methods, particular discretization methods is a challenging task since there is no clear objective, such as accuracy in supervised methods. Thus, commonly in the evaluation of unsupervised methods, the evaluation measures are derived from the study objectives.
The time series abstraction task we present here includes the process of discretization, which results in corresponding time series labeled with the states representative value. The following process is the concatenation of adjacent points labeled with the same state label. The output of this state is an interval based time series. We hereby define the evaluation measures we used to evaluate the performance of each discretization method. Generally, our goal was to find the method which results with the longest time intervals, which smoothes the time series towards the task of mining. On the other side we also wanted to minimize the error defined by the difference between the state value and the original value.

3.3.1 Mean and Standard deviation of Time Intervals

To measure the length of the time intervals resulted from each method we calculated the mean and standard deviation of the resulting time intervals, as shown in formula 1.

$$\mu = \frac{\sum_{i=1}^{n} |I|}{n}, \sigma = \sqrt{E\left(|I|^2\right) - \left(E\left(|I|\right)\right)^2} \tag{1}$$

Where $|I|$ is an interval length and $E(|I|)$ is the expected value of $|I|$.

3.3.2 Error measures

To define the error or distance measure which measures the states representation relatively to the original values of the time series, we used the Euclidean distance by which we measure the distance among the original value and the state value, as shown in formula 2.

$$E_D(X,Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2}$$

Where $X=x_1 \ldots x_n$ is the original time series and $Y=y_1 \ldots y_n$ is the discretized series, the value of every y_i is one of the discrete states representatives values.
Note that implementing this measure is straightforward, but it isn't clear which value represents in the best way the state which includes an interval of values. Due to the sensitivity of this error measure to the state representative value, we defined two different state representative values. **States mean error** called ***Error1***, in which the state representative value is the mean value the two cut points defining the state. **State observation mean error**, called ***Error2***, in which the state representative value is chosen to be the mean value of all the original values within the state.

4 Evaluation and Results

The goal of this study was to perform an extensive evaluation on the discretization methods on the time series data presented in the ICU dataset. We applied all the five discretization methods, presented in section 3.1, with *3, 4 and 5* number of states. Finally, for each method and each amount of states the evaluation measurements were calculated. Additionally, we provide here statistical properties (standard deviation, mean, and number of observations) to characterize the original time series.

Table 1 – The characteristics of each time series. The amount of time points (n), minimal and maximal values and mean and standard deviation of the values.

	N	Min	max	Mean ± std
TMP	416665	31.00	40.00	36.77 ± 0.85
HR	455613	0.00	230.00	79.62 ± 14.50
PEEP	460146	0	20	8.05 ± 2.76
ABPm	452285	26	193.4	77.34 ± 12.10
FiO2	460774	25.8	100	44.67 ± 8.00
CVP	431885	0	44	13.98 ± 4.61
CI	3216	0.99	5.99	2.54 ± 0.68
GLUC	4234	0	26.6	8.50 ± 2.86
CKMB	2028	1.1	465.2	40.08 ± 47.49
BE	4077	-22.3	13.5	-2.54 ± 2.74

4.1 High frequency variables

The high frequency variables (TMP, HR, PEEP, ABPm, FiO2, and CVP) were evaluated with the proposed measurements, after applying the five discretization methods. Generally, the results were similar in terms of the performance of the methods. Moreover, since we were interested in finding the best method and the best amount of states and due to the lack of space in the paper, we present the mean values of the variables.
Table 2 presents the *mean* and *standard deviation* length of the intervals of all the variables for each amount of states (*s*), having 3, 4 and 5 states, and for each method, as well as *Error1* and *Error2*. EWD and Persist achieved the highest mean interval length. While Persist was the best in the 3 states discretization, EWD was the best in the 4 and 5 states. Usually high mean length had also high standard deviation. In average, while the mean length in the 3 states was higher than in the 4 and 5 states, the last two had the same averaged mean. Error1 and Error2 showed inconsistency in the preferred method, which is reasonable due to the expected sensitivity to the chosen state representative values. The methods having the highest mean length had also the lowest Error1, which is quite surprising since we expected to see a tradeoff. In addition, while in the averaged mean length (according to states) there was no difference between 4 and 5 states, the averaged errors decreased as more states were used.

Table 2 – The mean and standard deviation length, and errors of the high frequency variables. EWD and Persist achieved the highest values of mean length and in general the 3 states had achieved the highest averaged mean.

S	Method	Mean ± std	Error 1	Error 2
3	**Persist**	**132.38 ± 195.75**	**4927.61**	3297.91
	SAX	68.05 ± 115.06	9865.52	2378.70
	EWD	109.73 ± 163.82	6673.67	2804.78
	EFD	66.54 ± 111.27	10209.05	2477.25
	k-means	72.56 ± 122.35	9696.55	**2355.80**
	Avg	68.19 ± 141.65	8274.48	2662.89
4	**Persist**	82.48 ± 152.34	4417.61	2799.02
	SAX	54.49 ± 95.08	8025.43	2042.51
	EWD	**100.64 ± 158.88**	**4212.42**	3013.21
	EFD	45.34 ± 85.96	8880.72	2156.79
	k-means	57.98 ± 99.73	7090.87	**1921.77**
	Avg	55.9 ± 118.40	6525.41	2386.66
5	**Persist**	61.15 ± 121.07	4136.87	2316.51
	SAX	43.53 ± 81.42	6767.22	1767.01
	EWD	**86.48 ± 142.02**	**3527.18**	2658.20
	EFD	38.86 ± 75.22	7892.18	1988.65
	k-means	47.93 ± 86.09	5531.14	**1543.19**
	Avg	55.59 ±101.16	5570.92	2054.71

4.2 Low frequency variables

The low frequency variables were summarized in the same way the high frequency variables, the results shown in table 3. The low frequency variables have two problematic issues, in the context of our work, since they were not measured uniformly, but manually. This is problematic in two phases, the discretization, in which SAX and Persist, which are the more temporal methods are assume the time series have fixed gaps. In addition in the interpolation step we assume the time points can be concatenated.

Persist and EWD achieved the highest mean interval length. However, here Persist outperformed in the 3 and 4 states and EWD in the 5 states. In average, less number of states created longer intervals and higher errors rate, and a tradeoff observed between the mean length and level of error. The longer the mean interval length, the highest the error.

Persist achieved the lowest Error1 in 3 and 4 states and k-means for 5 states. In Error2, Persist achieved the lowest error for 3 states, EWD for 4 states and k-means for 5 states. In general, a higher correlation was observed among the two error measures, unlike in the high frequency variables.

5 Discussion

We presented here the problem of time series discretization, as a preprocessing method in which the time series are transformed to time interval series. Generally, in such process we would like to have the highest mean length of intervals and minimal error when comparing the discretized data to the original values.

Table 3 The mean and standard deviation length, and errors of the low frequency variables. EWD, Persist and k-means achieved the highest values of mean length and in general the 3 states had achieved the highest averaged mean.

S	Method	Mean ± std	Error 1	Error 2
3	**Persist**	**342.15 ± 263.90**	**461.73**	**352.93**
	SAX	170.61 ± 193.19	877.05	405.35
	EWD	307.22 ± 216.45	679.18	365.53
	EFD	165.09 ± 193.41	1355.92	475.92
	k-means	177.61 ± 194.21	725.58	370.24
	Avg	232.54 ± 212.23	819.89	394.00
4	**Persist**	**265.15 ± 239.11**	**455.99**	334.36
	SAX	141.89 ± 171.02	721.17	368.23
	EWD	264.52 ± 215.28	472.58	**306.24**
	EFD	120.76 ± 167.59	1132.04	440.34
	k-means	136.26 ± 172.67	690.84	355.23
	Avg	185.72 ± 193.13	694.53	360.88
5	**Persist**	184.28 ± 200.34	399.09	295.55
	SAX	117.74 ± 156.74	657.75	347.18
	EWD	**233.75 ± 209.90**	354.58	268.94
	EFD	99.82 ± 152.48	980.91	413.92
	k-means	115.08 ± 156.70	**333.13**	**221.62**
	Avg	150.13 ± 175.23	545.09	309.44

We presented five discretization methods. Three are from the traditional static discretization methods, which were not designed specifically for time series and two additional which were designed for time series. We applied the five methods on a dataset from a medical problem aiming in three levels of state abstraction: 3, 4 and 5. We assumed that the more states there will be longer time intervals, which was the desired objective, but also larger error. To measure the error we defined two measures, the first Error1 compares the original values to the middle of the state interval, and the second Error2 compares to the average of the values within the state values intervals. The dataset we used include two types of time series. High-frequency time series which were measured in fixed gaps within each pair of time points, and low frequency in which there were few measurements taken manually in varying gaps. As expected, lower amount of states resulted in longer time intervals and higher rate of error. While in the high frequency there was low correlation between the two error measures, in the low frequency there was a high correlation. This can be explained by the low amount of time points, which probably distribute like the entire state interval. However, we think that Error2 might not be the best measure since it is data driven and thus subjective and influenced by the distribution of the time series. As was shown in the results the Error2 measure was not coherent, although as a state representative mean state value (of Error2) yields smaller distance from the original series.

6 Conclusions and future work

Generally, Persist brought the best outcome. These are very encouraging results indicating that discretization methods for time series which consider the temporal order of the values are required. However, while Persist [Mörchen and Ultsch, 2005] presents a method that explicitly considers the order of the values, it was not designed for time series having varying gaps. We are currently in the process of performing a wider evaluation on additional datasets. In addition we are developing a temporal discretization method which will take into consideration the varying gaps in any type of time series.

Acknowledgment

We used the implementation of the Persist method and other discretization methods in Matlab, provided by Mörchen on his website.

References

[Allen, 1983] J. F. Allen. Maintaining knowledge about temporal intervals, Communications of the ACM, 26(11): 832-843, 1983.

[Bilmes, 1997] J. Bilmes. A Gentle Tutorial on the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. Technical Report ICSI-TR-97-021, University of Berkeley, 1997.

[Dimitrva *et al*., 2005] E. S. Dimitrova, J.J. McGee, and R.C. Laubenbacher. Discretization of Time Series Data, eprint arXiv:q-bio/0505028, 2005.

[Dougherty et al, 1995] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. International Conference on Machine Learning, pages 194-202, 1995.

[Höppner, 2001] F. Höppner, Learning Temporal Rules from State Sequences, Proceedings of WLTSD-01, 2001.

[Höppner, 2002] F. Höppner. Time Series Abstraction Methods - A Survey. In informatik Bewegt: informatik 2002 - 32. Jahrestagung Der Gesellschaft FÜR informatik E.V. (Gi) (September 30 - October 03, 2002). S. Schubert, B. Reusch, and N. Jesse, Eds. LNI, vol. 19. GI, 777-786, 2002.

[Kam and Fu, 2000] P. S. Kam and A. W. C. Fu, Discovering temporal patterns for interval based events, In Proceedings DaWaK-00, 2000.

[Keogh *et al*., 1993] E. Keogh, S. Chu, D. Hart, and M. Pazzani, Segmenting Time Series: A Survey and Novel Approach, Data Mining in Time Series Databases, World Scientific Publishing Company, 1993.

[Keogh et al., 2000] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. Knowledge and Information Systems 3(3), 2000.

[Kohavi and Sahami, 1996] R. Kohavi and M. Sahami. Error-based and entropy-based discretization of continuous features. In Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, pages 114-119, 1996.

[Kullback and Leibler, 1951] S. Kullback and R.A. Leibler. On information and su_ciency. Annals of Mathematical Statistics, 22:79-86, 1951.

[Lin et al., 2003] 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. San Diego, CA. June 13.

[Liu et al., 2002] H. Liu, F. Hussain, C. L. Tan, and M. Dash. Discretization: An enabling technique. Data Mining and Knowledge Discovery, (6):393-423, 2002.

[MacQueen, 1967] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Symposium on Math, Statistics, and Probability (pp. 281{297). Berkeley, CA: University of California Press, 1967.

[Mörchen, 2005] F. Mörchen, and A. Ultsch, Optimizing Time Series Discretization for Knowledge Discovery, In Proceeding of KDD05, 2005.

[Mörchen, 2006] F. Mörchen, Algorithms for Time Series Knowledge Mining, Proceedings of KDD-06, 2006.

[Moskovitch and Shahar, 2005] R. Moskovitch, and Y. Shahar, Temporal Data Mining Based on Temporal Abstractions, (IEEE) ICDM-05 workshop on Temporal Data Mining, Houston, US, 2005.

[Papapetrou, 2005] P. Papapetrou, G. Kollios, S. Sclaroff, and D. Gunopulos, Discovering Frequent Arrangements of Temporal Intervals, Proceedings of ICDM-05, 2005.

[Quinlan, 1986] J. R. Quinlan. Introduction to decision trees. Machine Learning, 1:81-106, 1986.

[Quinlan, 1993] J.R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, California, 1993.

[Salvador and Chan, 2005] S. Salvador and P. Chan. Learning States and Rules for Detecting Anomalies in Time Series. Applied Intelligence 23, 3 (Dec. 2005), 241-255, 2005.

[Shahar, 1997] Y. Shahar, A framework for knowledge-based temporal abstraction, Artificial Intelligence, 90(1-2):79-133, 1997.

[Shannon, 1948] C.E. Shannon. A mathematical theory of communication. Bell System Tech. J. 27, 379-423, 623-656, 1948.

[Villafane et al, 2000] R. Villafane, K. Hua, D Tran, and B. Maulik, Knowledge discovery from time series of interval events, Journal of Intelligent Information Systems, 15(1):71-89, 2000.

[Winarko, and Roddick, 2005] E. Winarko, and J. Roddick, Discovering Richer Temporal Association Rules from Interval based Data, Proceedings of DaWaK-05, 2005.