Applied temporal Rule Mining to Time Series

P.Dafas and A.S. d'Avila Garcez

Department of Computing

City University

Technical Report Series

TR/2006/DOC/01

ISSN 1364-4009

Applied temporal rule mining to time series

Panagiotis A. Dafas Artur S. d'Avila Garcez

December 21, 2005

Abstract

Association rule mining from time series has attracted considerable interest over the last years and various methods have been developed. Temporal rules between discovered episodes provide useful knowledge for the dynamics of the problem domain and the underlying data generating process. However, temporal rule mining has received little attention over the last years. In addition, the proposed methods suffer from two significant drawbacks. First the rules they produce are not robust enough with respect to noise. Second the proposed methods are highly dependent on the choice of the parameters since small perturbations on the parameters lead to significantly different results. In this paper we propose a framework to derive temporal rules from time series. Our approach is based on episode rule mining that discovers temporal rules from time series in the frequency domain using the discrete cosine transform. The rules are then translated to temporal relations between time series patterns of arbitrary length. Experimental results of the proposed framework are presented in the relevant section.

1 Introduction

Traditional association rule mining algorithms are used to generate episodes from which intra-transaction association rules are discovered. Such rules are of the form "people who buy shampoo tend to buy toothpaste". The problem of mining sequential patterns is the discovery of inter-transaction associations across the same or similar data. These problems were initially addressed in [1, 2, 21]. In a similar framework the authors in [14] proposed a method to discover frequent episodes, i.e., a collection of events that occur relatively close to each other in a given partial order, in sequences of events. In the spirit of association rules, sequential patterns and episode rules, the authors in [4] proposed a rule discovery method that aims at finding local relationships from time series using a technique known as time series subsequence clustering. Subsequence clustering was also used by a number of authors in [6, 9, 8, 16, 19] as a subroutine to discover rules from time series. In order to identify rules in time series subsequence clustering is applied to convert the time series into a discrete block representation by first forming subsequences and then clustering these subsequences using a suitable distance metric. The discrete version of the time series is obtained by using the cluster identifiers (e.g., cluster centroids) corresponding to the subsequences. Rule finding algorithms such as episode rule mining methods are then used to infer rules that relate temporal patterns.

Current approaches in rule discovery from time series are intended as an interactive iterative exploratory method usually coupled with human interpretation of the rules. Hence, no fully systematic methods have been presented yet.

In addition, the rules extracted by the rule mining techniques that use subsequence clustering are highly dependent on the correct choice of the parameters. Small changes for example in the number of clusters or the length of the cluster window may lead to a totally different set of rules. This is mainly because the underlying time series subsequence clustering is highly sensitive to the selection of the window length that determines the size of the subsequences. The weaknesses of the subsequence clustering technique are summarized in [11] where the authors demonstrate by examples that subsequence clustering can produce meaningless results.

As an alternative to subsequence clustering the use of motifs was proposed by the authors in [17] to derive a discrete version of the time series. However the method assumes that motifs are patters known a priory and the problem of discovering the motifs remains open. Furthermore, rule mining is usually performed in subsequences that are blocks of the original raw time series data. If the data are distorted by noise the whole rule inference process is no longer reliable.

In this paper, we show that clustering on time series subsequences may be meaningful if

1. Subsequences do not overlap or overlap little
2. Subsequences are allowed to have an arbitrary length
3. Noise is removed considerably

We strongly believe that given the above conditions are met, most of the problems related to the subsequence clustering can be sufficiently addressed and a temporal rule mining technique that produce meaningful rules can be derived based on a generalized time series subsequence clustering.

2 Related Work

The authors in [15] have shown that the use of the Discrete Fourier and Wavelet Transforms as a features extraction technique can improve subsequence clustering and reduce the size of the classification rule sets.

Furthermore they conclude that when the number of the coefficients selected to represent the subsequences in the new domain is constant choosing the coefficients by their mean energy is optimal with respect to energy preservation and in many cases is much better than the method of choosing the first coefficients.

However this method is worse than the optimal procedure which is an adaptive approach by choosing a variable number of coefficients for each transformed subsequence of the original time series.

The authors in [5] have shown that time series subsequence clustering can also produce meaningful patterns by introducing a kernel-density-based algorithm. They conclude that the density based clustering can improve subsequence clustering and lead to meaningful results primarily because of the noise elimination properties that come with that approach.

This paper is structured as follows. In the next section we present a unitary transform that is used throughout the paper namely the Discrete Cosine Transform (DCT). What follows is a description of the proposed generalised temporal rule discovery method from time series. Experimental results that compare and test the proposed method under noisy conditions are presented in the last section.

3 Unitary Discrete Transforms

Transform theory plays a fundamental role in signal processing, as working with the transform of a signal instead of the signal itself may give us more insight into the properties of the signal. In this section we will present the fundamentals on unitary tranforms and we will briefly introduce the Discrete Fourier Transform (DFT) and it's short time representation. The definitions used in this section are taken from [7].

3.1 Unitary Transforms

For a sequence $\{s(t), \quad 0 \leq t \leq N-1\}$ represented as a vector $\boldsymbol{s}$ of size N, a transformation may be written as

$$\boldsymbol{y} = \boldsymbol{A} \cdot \boldsymbol{s} \rightarrow y(u) = \sum_{t=0}^{N-1} A(t,u)s(t), \quad 0 \leq u \leq N-1 \tag{1}$$

where $y(u)$ is the transform of $s(t)$, and $A(t,u)$ is the forward transformation kernel. Similarly, the inverse transform is the relation

$$s(t) = \sum_{u=0}^{N-1} I(t,u)y(u), \quad 0 \leq t \leq N-1 \tag{2}$$

where $I(u,t)$ is the inverse transformation kernel. If

$$\boldsymbol{A}^{-1} = \boldsymbol{A}^{*^T} \tag{3}$$

the matrix $\boldsymbol{A}$ is called unitary, and the transformation is called unitary as well. It can be proven that the columns (or rows) of an NxN unitary matrix are orthonormal and form a complete set of basis vectors in an dimensional vector space. In that case

$$\boldsymbol{s} = \boldsymbol{A}^{*^T} \cdot \boldsymbol{y} \rightarrow s(t) = \sum_{u=0}^{N-1} \boldsymbol{A}^* y(u) \tag{4}$$

The columns of $\boldsymbol{A}^{*^T}$, i.e. the vectors $\boldsymbol{v}_u^* = \left\{v^*(t,u), 0 \leq t \leq N-1\right\}^T$ are called the basis vectors of $\boldsymbol{A}$. In the unitary transformation

$$\boldsymbol{y} = \boldsymbol{A} \cdot \boldsymbol{s} \tag{5}$$

it is easily proven that

$$\|\boldsymbol{y}\|^2 = \|\boldsymbol{s}\|^2 \tag{6}$$

Thus, a unitary transformation preserves the signal energy. This means that every unitary transformation is simply a rotation of the vector s in the N - dimensional vector space. Moreover, most unitary transforms pack a large fraction of the energy of the signal into the first few components of the transform coefficients.

What follows is a revision of the unitary transforms that will be used in this work.

3.2 Discrete Cosine Transform

The Discrete Cosine Transform (DCT) is define as

$$y(u) = c(u) \sum_{t=0}^{N-1} s(t) \cos\left[\frac{(2t+1)u\pi}{2N}\right] \tag{7}$$

where $u \in [0, \cdots, N-1]$ and

$$c(u) = \begin{cases} \sqrt{\frac{1}{N}} & u = 0 \\ \sqrt{\frac{2}{N}} & u = 1, \cdots, N-1 \end{cases} \tag{8}$$

The inverse DCT is

$$s(t) = c(u) \sum_{u=0}^{N-1} y(u) \cos\left[\frac{(2t+1)u\pi}{2N}\right] \tag{9}$$

4 Temporal rule discovery from time series

We propose a novel method to derive temporal rules in time series that are robust with respect to noise and to parameter perturbations. This method is based on a generalised rule discovery technique from time series by mining association rules between subsequences of arbitrary length of the time series. We perform clustering on the time series subsequence in the frequency domain.

This is done by transforming segments of arbitrary length of the original time series to the frequency domain using the discrete cosine transform.

More specifically, we convert the time series into a discrete block representation by first transforming the original subsequences to the frequency domain where the temporal features of the time series are better captured. That is because of the noise reduction archived through an appropriate selection of the coefficients of the transformed subsequences. Clustering is then performed on the transformed coefficients. Using the cluster identifiers we derive a discrete version of the time series. Rule mining is then applied to infer rules that relate local temporal patterns of various sizes in the time series.

4.1 Association rule mining

An association rule is an expression $X \Rightarrow Y$, where X and Y are set of items. The meaning of such rules is the following: Given a database $\mathbf{D}$ of transactions where each transaction $T \in \mathbf{D}$ is a set of items, $X \Rightarrow Y$ means that whenever a transaction T contains X then it contains also Y with some probability. The probability or rule confidence is defined as the percentage of transactions containing Y in addition to X with regard to the overall number of transaction containing X.

Being a very important research area within data mining, association rule mining efforts have been moving in several directions. General association mining algorithms are used to generate sets of items that occur frequently from which intra-transaction associations rules are generated. In [1, 2] the authors first address the problem of sequence mining to infer sequential patterns across similar data from transaction database. The authors in [14] analyze event sequences to effectively discover frequent episodes i.e., collections of events occurring frequently together. Association rule mining query languages are introduced in [22] and constraint-based rule mining languages that meet user-specified constraints such as minimum support and confidence are presented in [3]. In [12] the authors introduce a piece-wise linear representation of the time series which allows fast indexing and accurate classification and clustering.

The authors in [4] proposed a rule discovery method that aims at finding local relationships from time series using a technique known as time series subsequence clustering. Subsequence clustering was also used by a number of authors in [6, 9, 8, 16, 19] as a subroutine to discover rules from time series. However the use of the subsequence clustering was heavily criticized by the authors in [11] where they demonstrate by examples that subsequence clustering may produce meaningless results.

We believe that clustering on time series subsequences can produce meaningful results given that

1. Subsequences do not overlap or overlap little
2. Subsequences are allowed to have an arbitrary length
3. Noise is removed considerably

The first condition means that different segments of the time series should capture different features or local characteristics. For example applying subsequence clustering to a stationary signal i.e. the statistical properties of the signal do not change over time, will probably be meaningless.

The second condition guaranties that all patterns, independent of their size, will be represented by at least one subsequence given that the maximum window size is large enough to capture the longest patter.

Finally the third point is there to highlight the impact of the noise to the subsequence clustering and hence to rule inference. The rule mining process should be robust and invariant to noise.

Inspired by the approaches presented in [4], [15] and [5] we propose a new method to derive temporal rules from time series that is based on a generalised rule discovery technique from time series. The method can be described as follows. We apply a short time unitary transform on the original time series data. In particular the sort time transform is carried out in two steps. The first step involves choosing an appropriate window range and generating a set of subsequences (blocks) of the original time series by shifting a window and selecting the corresponding region of the original time series. This process is repeated for each window size within the window range specified. In the second step the DCT is applied to each subsequence. Hence the original time series can be converted into a discrete block representation by clustering the transformed subsequences using a suitable similarity metric. The discrete version of the time series is obtained by using the cluster identifiers corresponding to the transformed subsequences. Rule finding algorithms such as episode rule mining methods are then used to infer rules.

4.2 Short time transform

The short time transform breaks the time series data (signal) $\{s(t), \quad 0 \leq t \leq N-1\}$ into blocks and apply a unitary transform to each block. For simplicity we will assume that the window length w is constant. Let k be the block index. We create $\lfloor \frac{N-w}{l} \rfloor + 1$ blocks of the original time series by each time shifting the window l samples and selecting the corresponding region. The derived subsequences (blocks) can be written as

$$s_j(t) = \{s(l(j-1)), \cdots, s(l(j-1)+w)\} \qquad j \in [1 \cdots \lfloor \frac{N-w}{l} \rfloor + 1] \tag{10}$$

A unitary transformation is then applied to each block. For a subsequence $\{s_j(t), \quad l(j-1) \leq t \leq l(j-1)+w\}$ represented as a vector $\boldsymbol{s_j}$ of size w, a transformation may be written as

$$\boldsymbol{y_j} = \boldsymbol{A} \cdot \boldsymbol{s_j} \Rightarrow y_j(u) = \sum_{t=0}^{w-1} A(t,u) s_j(t), \quad 0 \leq u \leq w \tag{11}$$

where $y_j(u)$ is the transform of $s_j(t)$, and $A(t,u)$ is the forward transformation kernel. The choice of the unitary transform determines the transformation kernel. In this work we have used the Discrete Cosine Transform (DCT). Most of the unitary transforms pack a large fraction of the energy of the signal into the first few components of the transform coefficients. Hence we only keep the first m_j samples of the subsequences $\boldsymbol{y_j}$ and we set the rest to zero, $\{\widetilde{y}_j(u), \quad l(j-1) \leq u \leq l(j-1)+m_j\}$. The parameter m_j is defined by optimally choosing the percentage of the total energy of the subsequences that we want to preserve [15] for each subsequence. By applying the appropriate inverse cosine transform, using the subsequences $\widetilde{y}_j(u)$, the original subsequences can be reconstructed, $\{\widetilde{s}_j(t), \quad l(j-1) \leq t \leq l(j-1)+w\}$. This process can be repeated for each window length within the range $[w_{min} \cdots w_{max}]$. For simplicity and to make sure that we are capturing different characteristics of the times series we allow windows to grow by exactly the number of samples defined by the shift parameter l.

4.3 Discretisation of the time series

For a given window length $w_i \in [w_{min}, w_{max}]$, $i \in [1 \cdots \lfloor \frac{w_{max}-w_{min}}{l} \rfloor + 1]$ and using an appropriate similarity metric we can cluster the subsequences $\widetilde{y}_j$ into k distinct clusters. We use the Euclidean distance between any two subsequences as a measure of dissimilarity. These distances can be used to cluster the set of all subsequences $\widetilde{y}_j$ into sets $C_1, ..., C_k$. In this work K-means clustering is used and the parameter k is specified in advance. For each cluster C_h we introduce a symbol a_h. Hence the discrete version $D_i(s)$ of the sequence $s(t)$ is over the alphabet $\Sigma_i = \{a_1, ..., a_k\}$. The sequence $D_i(s)$ is obtained by looking for each subsequence $\widetilde{y}_j$ in the cluster C_h such that $\widetilde{y}_j \in C_h$, and using the corresponding symbol a_h. So, we have

$$D_i(s) = a_{h_1}, a_{h_2}, ..., a_{h_{\lfloor \frac{N-w_i}{l} \rfloor + 1}} \tag{12}$$

We apply the same procedure for all window lengths within a given range and we define our alphabet over the union of the alphabets obtained for each window length in the range $[w_{min}, w_{max}]$

$$\Sigma = \bigcup \Sigma_i \tag{13}$$

4.4 Temporal rule mining

Most of the definitions in this section are based on [4]. We use the following rule format: *If A_{τ_i} occurs, then B_{τ_j} occurs in exactly T time units,* where A_{τ_i} and B_{τ_j} are letters from the alphabet Σ. A_{τ_i} can be seen as a pattern that evolves for time τ_i, similarly B_{τ_j} is a pattern that evolves for time τ_j. We write the above rule as $A_{\tau_i} \Rightarrow_T B_{\tau_j}$. We use three measures to qualify these rules i.e. frequency, confidence and J-measure.

Frequency Given a sequence $D(S) = (a_1, a_2, ..., a_n)$, the *frequency* $F(A_{\tau_i})$ of some letter A_{τ_i} is the number of occurrences of $A_{\tau_i} \in D(S)$ divided by n. Frequency of the first element of the rule takes the place of support in standard rule mining, because support itself would be too small for efficient user manipulation, and combination of frequency and confidence provides the same functionality.

Confidence The relative frequency $f(A_{\tau_i})$ of A_{τ_i} is given by (14). The *confidence* $c(A_{\tau_i} \Rightarrow_T B_{\tau_j})$ is defined as the fraction of occurrences of A_{τ_i} which are followed by another shape B_{τ_j} in T time units.

$$f(A_{\tau_i}) = \frac{F(A_{\tau_i})}{n} \tag{14}$$

$$c(A_{\tau_i} \Rightarrow_T B_{\tau_j}) = \frac{F(A_{\tau_i}, B_{\tau_j}, T)}{F(A_{\tau_i})} \tag{15}$$

where $F(A_{\tau_i}, B_{\tau_j}, T)$ is the number of occurrences of A_{τ_i} followed by a B_{τ_j} within T.

J-Measure Finally, as a single measure of rule efficiency, we use the J-measure for rule ranking [20] which is defined as:

$$\begin{aligned} J(B_{\tau_j,T}, A_{\tau_i}) = p(A)(p(B_{\tau_j,T}/A) \log \frac{p(B_{\tau_j,T}/A)}{p(B_{\tau_j,T})}) + \\ + (1 - p(B_{\tau_j,T}/A) \log \frac{1 - p(B_{\tau_j,T}/A)}{1 - p(B_{\tau_j,T})}) \end{aligned} \tag{16}$$

where $p(A_{\tau_i})$ is the probability of the letter A_{τ_i} occurring anywhere in the sequence, $p(B_{\tau_j,T})$ is the probability of at least one $B_{\tau_j,T}$ occurring anywhere in any window of duration T and $p(B_{\tau_j,T}/A_{\tau_i})$ is the probability of at least one $B_{\tau_j,T}$ occurring anywhere in any window of duration T given that the window is just after an occurrence of A_{τ_i}.

As can be seen from the formula, J-measure calculates the entropy of the rule by comparing frequencies and posterior probabilities of pattern occurrences. High J-measure indicates an important rule, balancing out the frequency and confidence - a rule with high frequency will not have a high J-measure if it has a very low confidence, and vice versa.

Note that the extension of the above notation to include rules between different time series is trivial. If we are given q time series then $D(s_h) = a_{h_1}, a_{h_2}, ..., a_{h_n}$ for $h = 1, .., q$ The format of the rule is still the same, only A_{τ_i} and B_{τ_j} may be from different discretisations of the time series.

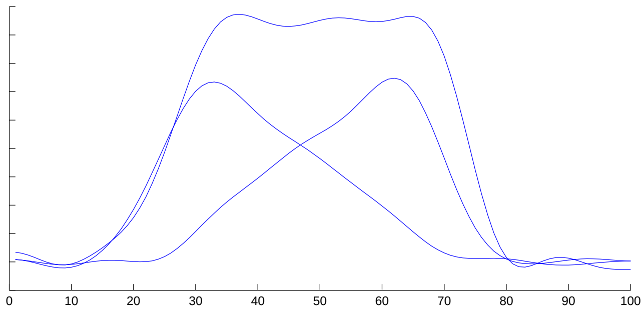

Figure 1: The three reconstructed cluster centers found by subsequence clustering on the DCT domain.

5 Experimental Results

To demonstrate the proposed rule mining framework we used the so called Cylinder-Bell-Funnel data set [18, 13, 10]. We generated a time series by concatenating 90 instances of the cylinder (C), bell (B) and funnel (F) patterns using a arbitrary predefined deterministic order. The time series we constructed is a periodic sequence $FFBCFFFBC$... of the cylinder (C), bell (B) and funnel (F) patterns. We have then applied the proposed temporal rule mining framework trying to identify rules that relate the above three patterns. For simplicity we have used the exact length of the patterns ($d = 100$) for the length of a single window, and $k = 3$ as the number of the desired clusters. The number of the coefficients we kept was set to $m = 15$ and the sliding parameter to $l = 100$. The reconstructed cluster centers are shown in Figure 1.

Note that although the clustering was performed on the DCT transforms of the subsequences we have reconstructed the cluster centers using the inverse discrete cosine transform (IDCT).

Figure 2 shows a part of the time series data used and the discrete version using subsequence clustering on the DCT domain.

We have mined two set of rules that relate the cylinder, bell and funnel patterns on 100 and 200 sample points ahead. The theoretical rules are given in 3. The actual rules mined using the proposed framework are given in 4.

6 Conclusion

In this paper we propose an applied framework to derive temporal rules from time series. Our approach is based on a generalised association rule mining technique that discovers temporal rules from time series by mining association rules in the frequency domain.

In order to obtain meaningful result we make sure that all the following

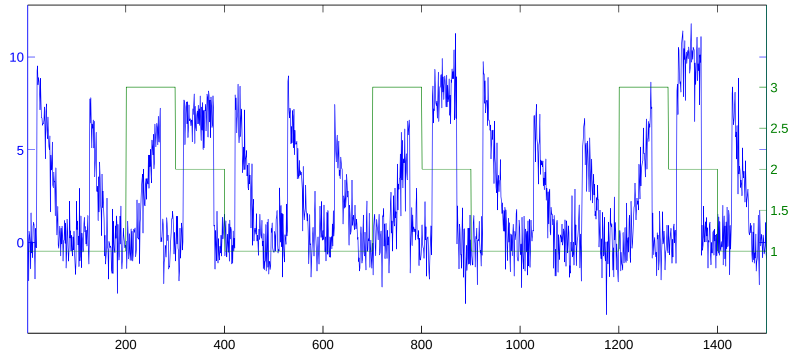

Figure 2: Time series of the cylinder, bell and funnel pattern and the corresponding discrete version using subsequence clustering on the DCT domain.

Rule	Confidence
$B \Rightarrow_{100} C$	1
$C \Rightarrow_{100} F$	1
$F \Rightarrow_{100} F$	0.666
$F \Rightarrow_{100} B$	0.333
$B \Rightarrow_{200} F$	1
$C \Rightarrow_{200} F$	1
$F \Rightarrow_{200} F$	0.333
$F \Rightarrow_{200} B$	0.333
$F \Rightarrow_{200} C$	0.333

Figure 3: Theoretical rules based on an arbitrary deterministic ordering of the CFB patterns.

Rule	Confidence
$B \Rightarrow_{100} C$	1
$C \Rightarrow_{100} F$	0.945
$F \Rightarrow_{100} F$	0.6603
$F \Rightarrow_{100} B$	0.3396
$B \Rightarrow_{200} F$	0.9450
$C \Rightarrow_{200} F$	0.9444
$F \Rightarrow_{200} F$	0.3207
$F \Rightarrow_{200} B$	0.3396
$F \Rightarrow_{200} C$	0.3396

Figure 4: Rules.

conditions are met

1. Subsequences do not overlap or overlap little
2. Subsequences are allowed to have an arbitrary length
3. Noise is removed considerably

In particular to extract local features of the time series we apply the short time Discrete Cosine Transform (DCT) for various window lengths. Noise reduction is obtained by optimally selecting a number of coefficients of the transformed subsequences. We perform subsequence clustering on the transformed subsequences to derive our alphabet that we then use to extract temporal rules.

Experimental results demonstrate that the proposed framework is robust with respect to noise and to parameter perturbations. We have demonstrated the applied rule mining framework by successfully identifying rules in a artificially created time series using the Cylinder-Bell-Funnel data generating process.

References

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C., 26–28 1993.

[2] R. Agrawal and R. Srikant. Mining sequential patterns. In Philip S. Yu and Arbee S. P. Chen, editors, *Eleventh International Conference on Data Engineering*, pages 3–14, Taipei, Taiwan, 1995. IEEE Computer Society Press.

[3] R. J. Bayardo, Jr. R. Agrawal, and D. Gunopulos. Constraint-based rule mining in large, dense databases. In *Proceedings of the 15th International Conference on Data Engineering, Sydney, Australia*, page 188, 1999.

[4] G. Das, K. I. Lin, H. Mannila, G. Renganathan, and P. Smyth. Rule discovery from time series. In *Knowledge Discovery and Data Mining*, pages 16–22, 1998.

[5] Anne Denton. Density-based clustering of time series subsequences. In *Proceedings of the Third Workshop on Mining Temporal and Sequential Data (TDM 04) in conjunction with the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, 2004.

[6] T. C. Fu, F. L. Chung, V. Ng, and R. Luk. Workshop notes of the workshop on temporal data mining, at the 7th acm sigkdd int'l conference on knowledge discovery and data mining. In *Proceedings of National Academy of Sciences USA*, pages 27–37, San Francisco, CA, 2001.

[7] R. C. Gonzalez and R .E. Woods. *Digital Image Processing*. Prentice Hall, 2002.

[8] X. Jin, L. Wang, Y. Lu, and C. Shi. Indexing and mining of the local patterns in sequence database. In *Proceedings of the 3rd Int'l Conference on Intelligent Data Engineering and Automated Learning*, pages 68–73, Manchester, UK, 2002.

[9] S. K., Harms, and J. Deogun T. Tadesse. Discovering sequential association rules with constraints and time lags in multiple sequences. In *Proceedings of the 13th Int'l Symposium on Methodologies for Intelligent Systems*, pages 432–441, Lyon, France, 2002.

[10] E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. In *ACM SIGKDD*, 2002.

[11] E. Keogh, J. Lin, and W. Truppel. Clustering of time series subsequences is meaningless: implications for previous and future research. In *Proceedings of the IEEE International Conference on Data Mining*, pages 115–122, Melbourne, FL, 2003.

[12] E. Keogh and M. Pazzani. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro, editors, *Fourth International Conference on Knowledge Discovery and Data Mining (KDD'98)*, pages 239–241, New York City, NY, 1998. ACM Press.

[13] S. Manganaris. *Supervised Classification with Temporal Data*. PhD thesis, Computer Science Department, School of Engineering, Vanderbilt University, December 1997.

[14] H. Mannila, H. Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3):259–289, 1997.

[15] F. Morchen. Time series feature extraction for data mining using dwt and dft, 2003.

[16] T. Mori and K. Uehara. Extraction of primitive motion and discovery of association rules from human motion. In *Proceedings of the 10th IEEE Int'l Workshop on Robot and Human Communication*, pages 200–206, France, Bordeaux-Paris, 2001.

[17] P. Patel, E.Keogh, J.Lin, and S.Lonardi. Mining motifs in massive time series databases. In *Proceedings of IEEE International Conference on Data Mining (ICDM'02)*, pages 370–377, Maebashi City, Japan, December 2002.

[18] N. Saito. *Local feature extraction and its application using a library of bases*. PhD thesis, Yale University, December 1994.

[19] B. K. Sarker, T. Mori, and K. Uehara. Parallel algorithms for mining association rules in time series data. In *Technical Report*, 2002.

[20] P. Smyth and R. M. Goodman. *Rule induction using information theory*, pages 159–176. AAAI/MIT Press, 1991.

[21] R. Srikant and R. Agrawal. Mining generalized association rules. *Future Generation Computer Systems*, 13(2–3):161–180, 1997.

[22] S. Tsur, J. D. Ullman, S. Abiteboul, C. Clifton, R. Motwani, S. Nestorov, and A. Rosenthal. Query flocks: A generalization of association-rule mining. In Laura M. Haas and Ashutosh Tiwary, editors, *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA*, pages 1–12. ACM Press, 1998.