Classification Rules + Time = Temporal Rules[1]

Paul Cotofrei, Kilian Stoffel

Université Neuchâtel,
Pierre-à-Mazel 7, 2000, Neuchâtel, Suisse
paul.cotofrei@unine.ch, kilian.stoffel@unine.ch

Abstract. Due to the wide availability of huge data collection comprising multiple sequences that evolve over time, the process of adapting the classical data-mining techniques, making them capable to work into this new context, becomes today a strong necessity. In [1] we proposed a methodology permitting the application of a classification tree on sequential raw data and the extraction of the rules having a temporal dimension. In this article, we propose a formalism based on temporal first logic-order and we review the main steps of the methodology through this theoretical frame. Finally, we present some solutions for a practical implementation.

1. Introduction

Data mining is defined as an analytic process designed to explore large amounts of (typically business or market related) data, in search for consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. Due to the wide availability of huge amounts of data in electronic form, and the imminent need for turning such data into useful information and knowledge for broad applications including market analysis, business management and decision support, data mining has attracted a great deal of attention in information industry in recent years.
In many applications, the data of interest comprises multiple sequences that evolve over time. Examples include financial market data, currency exchange rates, network traffic data, sensor information from robots, signals from biomedical sources like electrocardiographs, demographic data from multiple jurisdictions, etc. Although traditional time series techniques can sometimes produce accurate results, few can provide easy understandable results. However, a drastically increasing number of users with a limited statistical background would like to use these tools. Therefore, it becomes increasingly important to be able to produce results that can be interpreted by domain expert without special statistical training. In the same time, we have a number of tools developed by researchers in the field of artificial intelligence, which produce understandable rules. However, they have to use ad-hoc, domain-specific techniques for "flattering" the time series to a “learner-friendly” representation. These

[1] This work was supported by the Swiss National Science Foundation (grant N° 2100-063 730).

techniques fail to take into account both the special problems and special heuristics applicable to temporal data and so often unreadable concept description are obtained.

1.1 State of the Art

The main tasks concerning the information extraction from time series database and on which the researchers concentrated their efforts over the last years may be divided into four directions.

1. *Similarity/Pattern Querying*. The main problem addressed by this body of research concerns the measure of similarity between two (sub-) sequences. Different models of similarity were proposed, based on different similarity measures (Euclidean metric [2], [3], envelope metric ($|X - Y| \leq \varepsilon$)). Different methods were developed (window stitching [4], dynamic time warping based matching [5]) to allow matching similar series despite gaps, translation and scaling. For all methods, the quality of the results and the performance of the algorithms are intrinsically tied to the subjective parameter that is the user-specified tolerance ε.
2. *Clustering/Classification*. In this direction, researchers studied optimal algorithms for clustering/classifying sub-sequences of time series into groups/classes of similar sub-sequences. Different techniques were developed, as the Hidden Markov Model [6], Dynamic Bayes Networks [7] or the Recurrent Neural Networks [8]. Recently, the machine learning approach opened new directions. A system for a supervised classification of signals using piecewise polynomial modeling was developed in [9] and a technique for agglomerative clustering of univariate time series representation was studied in [10].
3. *Pattern finding/Prediction.* These methods, concerning the search for periodicity patterns in time series databases, may be divided into two groups: those that search full periodic patterns and those that search partial periodic patterns, which specify the behavior at some but not all points in time. For full periodicity search there is a rich collection of statistic methods, like FFT [11]. For partial periodicity search, different algorithms were developed which explore properties related to partial periodicity, like the a-priori property and the max-sub-pattern-hit-set property [12] or the segment-wise and the point-wise periodicity [13].
4. *Rule extraction.* Besides these, some researches were devoted to the extraction of explicit rules from time series. Inter-transaction association rules, proposed by Lu, [14] are implication rules whose two sides are totally ordered episodes with time-interval restriction on the events. Cyclic association rules were considered in [15], adaptive methods for finding rules whose conditions refer to patterns in time series were described in [16] and a general architecture for classification and extraction of comprehensible rules was proposed in [17].

1.2 Aims and Scopes

The real-world problem we tried to solve and which practically "pushed" us to develop our methodology consisted in the analysis of an insurance company's database. The question was if it is possible, looking to data containing medical

information, to find/extract rules of the form "If a person contracted an influenza necessitating more than five days of hospitalization, then with a confidence p the same person will contract, in the next three month, a serious lung-infection". The size of the database (several Gigabytes) made the application of statistical tools very difficult. Therefore, we searched for alternative techniques to solve these kinds of problems. However, most approaches dealing with the information extraction from time series, described above, have mainly two shortcomings.

The first problem is the type of knowledge inferred by the systems, which is very difficult to be understood by a human user. In a wide range of applications, (e.g. almost all decision making processes) it is unacceptable to produce rules that are not understandable for an end user. Therefore, we decided to develop inference methods that produce knowledge that can be represented in form of general Horn clauses, which are at least comprehensible for a moderately sophisticated user. In the fourth approach, (*Rule extraction*), a similar representation is used. However, the rules inferred by these systems have a more restricted form than the rules we are proposing.

The second problem consists in the number of time series considered during the inference process. Almost all methods mentioned above are based on uni-dimensional data, i.e. they are restricted to one time series. The methods we propose are able to handle multi-dimensional data.

To overcome these problems we developed [1] a methodology that integrates techniques developed both in the field of machine learning and in the field of statistics. The machine learning approach is used to extract symbolic knowledge and the statistical approach is used to perform numerical analysis of the raw data. The overall goal consists in developing a series of methods able to extract/generate/ temporal rules, having the following characteristics:

- Contain explicitly a temporal (or at least a sequential) dimension
- Capture the correlation between time series
- Predict/forecast values/shapes/behavior of sequences (denoted events)

In this article, we extended the methodology with a formalism based on first-order temporal logic, which permits an abstract view on temporal rules. The formalism allows also the application of an inference phase in which "general" temporal rules are inferred from "local" rules. The latter are extracted from the sequences of data. Using this strategy, we can guarantee the scalability of our system to handle huge databases, applying standard statistical and machine learning tools.

The rest of the paper is structured as follows. In the next section, the formalism for the representation of events and temporal rules based on temporal logic is proposed. In Section 3, the main steps of the methodology are summary presented but in connection with the proposed formalism. Some specific problems concerning the practical implementation of the training sets and of the temporal rules are treated in the following section. The final section summarizes our work and points to future research.

2. The Formalism of Temporal Rules

Time is ubiquitous in information systems, but the mode of representation/perception varies in function of the purpose of the analysis [18], [19]. Firstly, there is a choice of a *temporal ontology*, which can be based either on *time points* (instants) or on *intervals* (periods). Secondly, time may have a *discrete* or a *continuous* structure. Finally, there is a choice of *linear* vs. *nonlinear* time (e.g. acyclic graph). For our methodology, we chose a temporal domain represented by linearly ordered discrete instants. This option is imposed by the discrete representation of all databases.

DEFINITION 1. *A single-dimensional linearly ordered temporal domain is a structure* $T_P = (T, <)$*, where T is a set of time instants and "<" a linear order on T.*

Because we may always find an isomorphism between T and the set of natural numbers, N, the temporal domain will be usually represented as (N, <). As consequence, there is an initial time moment (the first time instant), which is 0.

Databases being first-order structures, the first-order logic represents a natural formalism for their description. Consequently, the first-order temporal logic expresses the formalism of temporal databases. For a better rigor and for a cleaner description of the used terms one may consider the following set of definitions.

A *first-order language* L is constructed over an alphabet containing function symbols, predicate symbols, variables, logical connectives, temporal connectives and qualifier symbols. A *constant* is a zero-ary function symbol and a zero-ary predicate is a *proposition symbol*. There is a particular variable, T, representing the time. There are several special binary predicate symbols ($=, <, \leq, >, \geq$) known as *relational symbols*. The basic set of logical connectives is $\{\wedge, \neg\}$ from which one may express $\vee, \rightarrow$ and $\leftrightarrow$. The basic set of temporal connectives are F ("sometime"), G ("always"), X ("next") and U ("until"). From these connectives, we may derive X_k, where a k positive means "next k" and a k negative means "previous k".

The syntax of L defines terms, atomic formulae and compound formulae. A *term* is either a variable or a function symbol followed by a bracketed *n*-tuple of terms ($n \geq 0$). A predicate symbol followed by a bracketed *n*-tuple of terms ($n \geq 0$) is called an *atomic formula*, or *atom.* If the predicate is a relational symbol, the atom is called a *relation*. A *compound formula* or *formula* is either an atom or *n* atoms ($n \geq 1$) connected by logical (temporal) connectives and/or qualifier symbols.

A Horn clause is a formula of the form $\forall X_1 \forall X_2 \ldots \forall X_s (B_1 \wedge B_2 \cdots \wedge B_m \rightarrow B_{m+1})$ where each B_i is a positive (non-negated) atom and $X_1, \ldots, X_s$ are all the variables occurring in $B_1, \ldots, B_{m+1}$. The atoms $B_i, i = 1 \ldots m$ are also called implication clauses, whereas B_{m+1} is also known as the implicated clause.

DEFINITION 2. *An event (or temporal atom) is an atom formed by the predicate symbol TE followed by a bracketed* $(n+1)$*-tuple of terms* $(n \geq 1)$ $TE(T, t_1, t_2, \ldots, t_n)$*. The first term of the tuple is the time variable, T, the second* t_1 *is a constant representing the name of the event and all others terms are function symbols. A short temporal atom (or the event's head) is the atom* $TE(T, t_1)$*.*

DEFINITION 3. *A constraint formula for the event* $TE(T, t_1, t_2, \ldots t_n)$ *is a conjunctive compound formula,* $C_1 \wedge C_2 \wedge \cdots \wedge C_k$, *where each* C_j *is a relation implying one of the terms* t_i.

DEFINITION 4. *A temporal rule is a Horn clause of the form*

$$H_1 \wedge \cdots \wedge H_m \rightarrow H_{m+1} \quad (1)$$

where H_{m+1} *is a short temporal atom and* H_i *are constraint formulae, prefixed or not by the temporal connectives* $X_{-k}, k \geq 0$. *The maximum value of the index k is called the time window of the temporal rule.*

The semantics of L is provided by an interpretation I that assigns an appropriate meaning over a domain D to the symbols of L. The set of symbols is divided into two classes, the class of global symbols, having the same interpretation over all time instants (global interpretation) and the class of local symbols, for which the interpretation depends on the time instant (local interpretation). In the framework of temporal databases, the constants and the function symbols are global symbols, and represent real numbers (with an exception, strings for constants representing the names of events), respectively statistical functions. The events, the constraint formulae and the temporal rules are local symbols, which means they are true only at some time instants. We denote such instants by $i \mapsto P$, i.e. at time instant i the formula P is true. Therefore, $i \mapsto TE(T, t_1, \ldots, t_n)$ means that at time i an event with the name t_1 and characterized by the statistical parameters $t_2, \ldots, t_n$ started (the meaning of the word "event" in the framework of time series will be clarified later). A constraint formula is true at time i iff all relations are true at time i. A temporal rule is true at time i iff either $i \mapsto H_{m+1}$ or $i \mapsto \neg(H_1 \wedge \cdots \wedge H_m)$. (*Remark*: $i \mapsto X_k P$ iff $i + k \mapsto P$).

However, the standard interpretation for a temporal rule is not conforming to the expectation of a final user for two reasons. The first is that a user wants a global interpretation for rules, that is, rules that are "generally" true. The second is tied on the table of true for a formula $p \rightarrow q$: even if p and q are false, the implication is still true. In a real application, a user search temporal rules where both the body, $H_1 \wedge \cdots \wedge H_m$, and the head of the rule, H_{m+1}, are simultaneously true.

DEFINITION 5. *A model for L is a structure* $M = (\tilde{T}_p, D, I)$ *where* $\tilde{T}_p$ *is a finite temporal domain and D, (respectively I), the domain, respectively the interpretation for L. We denote by N the cardinality of* $\tilde{T}_p$.

DEFINITION 6. *Given L, a model M for L and an infinite temporal domain* T_p *that include* $\tilde{T}_p$, *a global interpretation* I_G *for an n-ary atom P is a function defined on* D^n *with values in* $true \times [0,1]$. I_G *assigns to P the boolean value "true" with a confidence equal with the ratio* $card(A)/N$, *where* $A = \{i \in \tilde{T}_p \mid i \mapsto P\}$.

The global interpretation is naturally extended to formulae, using the usual probability calculus to obtain the confidence ($P(A \cap B) = P(A) + P(B) - P(A \cap B)$). There is

only one exception: for temporal rules the confidence is calculated as a ratio between the number of certain applications (time instants where both the body and the head of the rule are true) and the number of potential applications (time instants where only the body of the rule is true). A useful temporal rule is a rule with a confidence greater than 0.5.

DEFINITION 7. *The confidence of a temporal rule* $\forall T(H_1 \wedge \cdots \wedge H_m \rightarrow H_{m+1})$ *is the ratio* $card(A)/card(B)$ *, where* $A = \{i \in \tilde{T}_p \mid i \mapsto H_1 \wedge \cdots \wedge H_m \wedge H_{m+1}\}$ *and* $B = \{i \in \tilde{T}_p \mid i \mapsto H_1 \wedge \cdots \wedge H_m\}$.

3. The Methodology

The two scientific communities which made important contribution relevant to data analysis (the statisticians and database researchers) chose two different approaches: statisticians concentrated on the continuous aspect of the data and the large majority of statistical models are continuous models, whereas the database community concentrated much more on the discrete aspects, and in consequence, on discrete models. We are adopting here a mixture of these two approaches, which represents a better description of the reality of data and which generally allows us to benefit from the advantages of both approaches. However, the techniques applied by the two communities must be adapted in order to be integrated in our approach.

The two main steps of the methodology of temporal rules extraction are structured in the following way:

1. Transforming sequential raw data into sequences of events: Roughly speaking, an event can be regarded as a labeled sequence of points extracted from the raw data and characterized by a finite set of predefined features. The features describing the different events are extracted using statistical methods.
2. Inferring temporal rules: We apply an inference process, using sets of events extracted from event database as training sets, to obtain several classification trees. Then temporal rules are extracted from these classification trees and a final inference process will generate the set of global temporal rules.

3.1. The Phase One

The procedure that creates an event database from the initial raw data can be divided into two steps: time series discretisation, which extracts the discrete aspect, and global feature calculation, which captures the continuous aspect. This phase totally defines the interpretation of the domain D and of the function symbols from L.

Time series discretisation. Formally, during this step, the constant symbols t_1 (the second term of a temporal atom) receive an interpretation. Practically, all contiguous subsequences of fixed length *w* are classified in clusters using a similarity measure and these clusters receive a name (a symbol or a string of symbols). By a misuse of language, an event means a subsequence having a particular shape. There are cases

when the database contains explicitly the names of events, like in our insurance case, where we identified the names of events by the names of diseases.
All the methods which construct/find class of events depend on a parameter k which controls the degree of discretisation: a bigger k means a bigger number of events and finally, less understandable rules. However, a smaller k means a rough description of the data and finally, simple rules but without significance.
Global feature calculation. During this step, one extracts various features from each sub-sequence as a whole. Typical global features include global maxima, global minima, means and standard deviation of the values of the sequence as well as the value of some specific point of the sequence, such as the value of the first or of the last point. Of course, it is possible that specific events will demand specific features, necessary for their description (e.g. the average value of the gradient for an event representing an increasing behavior). The optimal set of global features is hard to be defined in advance, but as long as these features are simple descriptive statistics, they can be easily added or removed form the process. From the formal model viewpoint, this step assigns an interpretation to the terms $t_2,\ldots,t_n$ of a temporal atom. They are considered as statistical functions, which take continuous or categorical values, depending on the interpretation of the domain D.

3.2. The Phase Two

During the second phase, we create a set of temporal rules inferred from the events database. This database was obtained using the procedures described above. Two important steps can be defined here:

1. Application of a first inference process, using the event database as training database, to obtain one or more *classification trees* and
2. Application of a second inference process using the previously inferred classification to obtain the final set of temporal rules.

Classification trees. There are different approaches for extracting rules from a set of events. Associations Rules, Inductive Logic Programming, Classification Trees are the most popular ones. For our methodology, we selected the *classification tree approach.* It is a powerful tool used to predict memberships of cases or objects in the classes of a categorical dependent variable from their measurements on one or more predictor variables (or attributes). A classification tree is constructed by recursively partitioning a learning sample of data in which the class and the values of the predictor variables for each case are known. Each partition is represented by a node in the tree. A variety of classification tree programs has been developed and we may mention QUEST [20], CART [21], FACT [22] and last, but not least, C4.5 [23]. Our option was the C4.5 like approach. The tree resulting by applying the C4.5 algorithm is constructed to minimize the observed error rate, using equal priors. This criterion seems to be satisfactory in the frame of sequential data and has the advantage to not favor certain events. To create the successive partitions, the C4.5 algorithm uses three forms of tests in each node: a "standard" test on a discrete attribute, with one outcome ($A = x$) for each possible value x of the attribute A, a more complex test, also based on a discrete attribute, in which the possible values are allocated to a variate number

of groups with one outcome for each group, and a binary test, for continuos attributes, with outcomes $A \leq Z$ and $A > Z$, where *A* is the attribute and *Z* is a threshold value. The application of a classification tree algorithm is equivalent, as we will detail in the next section, with the choice of a model for the language L and the procedure that extract rules from the classification tree is equivalent with the creation of a global interpretation I_G and with the calculus of the confidence of temporal rules.

Second inference process. Different classification trees, constructed starting with different training sets, generate finally different sets of rules. The process that tries to infer a general temporal rule from a set of rules implying the same class is similar to the pruning strategy for a classification tree. Consider two temporal rules R_1 and R_2 having the same head and obtained from two different classification trees. Formally, this means we have two models, $M_1 = (\tilde{T}_1, D, I)$ and $M_2 = (\tilde{T}_2, D, I)$, and the confidence for each rule was calculated on the corresponding model. If we denote by $\{\overline{R}_i\}$ the set of implication clauses of the rule R_i, $i = 1,2$, then from the set of rules $\{R_1, R_2\}$ on infer a general rule $R_{1,2}$ having a body of the form $(C_1 \wedge \ldots \wedge C_s)$ where $C_i \in \bigcap_i \{\overline{R}_i\}$. The confidence of the inferred rule is then calculated on the extended model $M = (\tilde{T}_1 \cup \tilde{T}_2, D, I)$. If this confidence is at least equal with the minimum confidence for the initial rules, the general rule is kept, if not, it is rejected.

4. Implementation Problems

Before we can start to apply the decision tree algorithms to an event database, an important problem has to be solved first: establishing the training set. An *n*-tuple in the training set contains *n*-1 values of the predictor variables (or attributes) and one value of the categorical dependent variable, which represents the class.

Each time series from the initial database was "translated", during the first phase, into a sequence of events. As the classification tree algorithm demands, one of these event series must represent the dependent variable. There are two different approaches on how the sequence that represents the classification (the values of the categorical dependent variable) is obtained. In a supervised methodology, an expert gives this sequence (e.g., in our insurance case, a physician). The situation becomes more difficult when there exists no prior knowledge about the possible classifications. Suppose that our database contains a set of time series representing the evolution of stock price for different companies. We are interested in seeing if a given stock value depends on other stock values. As the dependent variable (the stock price) is not categorical, it cannot represent a classification used to create a classification tree. The solution is to use the sequence of the names of events extracted from the continuous time series as the sequence of classes.

Lets suppose we have *k* time series representing the predictor variables $s_1, s_2, \ldots s_k$. Each $s_{ij}, i = 1..k, j = 1..n$ is an event. We have also a sequence $s_c = s_{c1}, \ldots s_{cn}$ representing the classification. The training set will be constructed using a procedure depending on three parameters. The first, t_0, represents a time instant considered as

present time. Practically, the first tuple contains the class s_{ct_0} and there is no tuple in the training set containing an event that starts after time t_0. The second, t_p, represents a time interval and controls the further back in time class $s_{c(t_0-t_p)}$ included in the training set. Consequently, the number of tuples in the training set is t_p+1. The third parameter, *h*, controls the influence of the past events $s_{i(t-1)},\ldots,s_{i(t-h)}$ on the actual event s_{it}. This parameter (*h*istory) reflects the idea that the class s_{ct} depends not only on the events at time *t*, but also on the events started before time *t*. Finally, each tuple contains $k(h+1)$ events and one class value. The first tuple is $s_{1t_0},\ldots,s_{1(t_0-h)},\ldots,s_{k(t_0-h)},s_{ct_0}$ and the last $s_{1(t_0-t_p)},\ldots,s_{k(t_0-t_p-h)},s_{c(t_0-t_p)}$. The set of time instants at which the events included in the training set start may be structured as a finite temporal domain $\tilde{T}_p$. The cardinality of $\tilde{T}_p$ is t_p+h+1. Therefore, the selection of a training set is equivalent with the selection of a model for L and, implicitly, of a global interpretation I_G. As a remark, the parameter *h* has also a side effect on the final temporal rules: because the time window of any tuple is *h*, the time window for temporal rules cannot exceed *h*.

Because the training set depends on different parameters, the process of applying the classification tree will consist in creating multiple training sets, by changing these parameters. For each set, the induced classification tree will be "transformed" into a set of temporal rules.

To adopt this particular strategy for the construction of the training set, we made an assumption: the events $s_{ij}, i=1..k$, j a fixed value, start all at the same time instant. Formally, this is equivalent with $j\mapsto s_{ij}$, for all $i=1..k$. The same assumption permits us to solve another implementation problem: the time information is not processed during the classification tree construction, (time is not a predictor variable), but the temporal dimension must be captured by the temporal rules. The solution we chose to "encode" the temporal information is to create a map between the index of the attributes (or predictor variables) and the order in time of the events. The $k(h+1)$ attributes are indexed as $\{A_0,A_1,\ldots,A_h,\ldots,A_{2h},\ldots A_{k(h+1)-1}\}$. Suppose that the set of indexes of the attributes that appear in the body of the rule is $\{i_0,\ldots,i_m\}$. This set is transformed into the set $\{\bar{i}_0,\ldots,\bar{i}_m\}$, where "$\bar{i}$" means "i modulo *(h+1)*". If t_0 represents the time instant when the event in the head of the rule starts, then an event from the rule's body, corresponding to the attribute A_{i_j}, started at time $t_0-\bar{i}_j$.

5. Conclusions and Further Directions

The methodology we proposed in this article tries to respond to an actual necessity, the need to discover knowledge from databases where the notion of "time" represents an important issue. We proposed to represent this knowledge in the form of general

Horn clauses, a more comprehensible form for a final user without sophisticated statistical background. A formalism based on first-order temporal logic tries to give a theoretical support to the main terms “event” and “temporal rule”. To obtain the “temporal rules”, a discretisation phase that extracts “events” from raw data is applied first, followed by an inference phase, which constructs classification trees from these events. The discrete and continuous characteristics of an “event” allow us to use statistical tools as well as techniques from artificial intelligence on the same data.
To capture the correlation between events over time, a specific procedure for the construction of a training set is proposed. This procedure depends on three parameters. Among others, the so-called *h*istory controls the time window of the temporal rules. We wish also to emphasis the fact that our methodology represents, at this moment, just an alternative to others procedures and methods for the extraction of knowledge from time series databases. In order to analyze the quality of the final temporal rules, it is necessary to conduct comparative experiments, using different methods, on the same databases. Finally, we are also interested in an open question, which is strictly connected to the temporal characteristics of the data: At what moment in time, a specific rule is no longer applicable?

References

1. Cotofrei, P, Stoffel K.: Rule Extraction from Time Series Databases using Classification Trees. Proc. of 20th I.A.S.T.E.D. Conference on Applied Informatics, Innsbruck, (2002)
2. Agrawal R., Faloutsos C., Swami A.: Efficient Similarity Search In Sequence Databases. Proc. of IVth Conference F.D.O.A, (1993), 69-84
3. Faloutsos C., Ranganathan M., Manolopoulos Y.: Fast Subsequence Matching in Time-Series Databases. Proc. of ACM SIGMOD, (1994), 419-429
4. Faloutsos C., Jagadish H., Mendelzon A., Milo T.: A Signature Technique for Similarity-Based Queries. Proc. of SEQUENCES97, Salerno, IEEE Press, (1997)
5. Yi B., Jagadish H., Faloutsos C.: Efficient Retrieval of Similar Time Sequences Under Time Warping. IEEE Proc. of Int. Conf. on Data Engineering, (1998), 201-208
6. Rabiner L., Juang B.: An introduction to Hidden Markov Models. IEEE Magazine on Accoustics, Speech and Signal Processing, 3, (1986), 4-16
7. Zweig G., Russel S.: Speech recognition with dynamic Bayesian networks. AAAI, (1998), 173-180
8. Bengio Y.: Neural Networks for Speech and Sequence Recognition. International Thompson Publishing Inc., (1996)
9. Mangaranis S.: Supervised Classification with temporal data. PhD. Thesis, Computer Science Department, School of Engineering, Vanderbilt University, (1997)
10. Keogh E., Pazzani M. J.: An Enhanced Representation of time series which allows fast and accurate classification, clustering and relevance feedback. Proc. of KDD, (1998), 239-243
11. Loether H., McTavish D.: Descriptive and Inferential Statistics: An introduction. Ally and Bacon(ed.), (1993)
12. Han J., Gong W., Yin Y.: Mining Segment-Wise Periodic Patterns in Time-Related Databases. Proc. of IVth Conf. on Knowledge Discovery and Data Mining, (1998), 214-218
13. Han J., Dong G., Yin Y.: Efficient Mining of Partial Periodic Patterns in Time Series Database. Proc. of Int. Conf. on Data Engineering, Sydney, (1999), 106-115

14. Lu H., Han J., Feng L.: Stock movement and n-dimensional inter-transaction association rules. Proc. of SIGMOD workshop on Research Issues on Data Mining and Knowledge Discovery, (1998), 12:1-12:7
15. Ozden B., Ramaswamy S., Silberschatz A.: Cyclic association rules. Proc of International Conference on Data Engineering, Orlando, (1998), 412-421
16. Das G., Lin K., Mannila H., Renganathan G, Smyth P.: Rule Discovery from Time Series. Proc. of IVth Int. Conf. on Knowledge Discovery and Data Mining, (1998), 16-22
17. Waleed Kadous M: Learning Comprehensible Descriptions of Multivariate Time Series. Proc. of Int. Conf. on Machine Learning, (1999), 454-463
18. Chomicki J., Toman D.: Temporal Logic in Information systems, BRICS LS-97-1, (1997)
19. Allen Emerson E.: Temporal and Modal Logic. In Handbook of Theoretical Computer Science, ed. Noth-Holland Pub. Co., (1995)
20. Loh W., Shih Y.: Split Selection Methods for Classification Trees, Statistica Sinica, vol. 7, (1997), 815-840
21. Breiman L., Friedman J.H., Olshen R. A., Stone C. J.: Classification and regression trees, Monterey. Wadsworth & Brooks/Cole Advanced Books & Software, (1984)
22. Loh W., Vanichestakul N.: Tree-structured classification via generalized discriminant analysis (with discussion). Journal of American Statistical Association, (1983), 715-728.
23 Quinland J. R.: C4.5: Programs for Machine Learning. Morgan Kauffmann Publishers, San Mateo, California, (1993)