RULE EXTRACTION FROM TIME SERIES DATABASES USING CLASSIFICATION TREES[1]

PAUL COTOFREI
Université Neuchâtel
Pierre-à-Mazel 7, 2000, Neuchâtel
Suisse

KILIAN STOFFEL
Université Neuchâtel
Pierre-à-Mazel 7, 2000, Neuchâtel
Suisse

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. Having as a final goal the extraction of temporal rules from time series databases, we proposed in this article a methodology permitting the application of a classification tree on sequential raw data by the use of a flexible approach of the main terms as "classification class", "training set", "attribute set", etc. We described also a first implementation of this methodology and presented some results on a synthetic time series database.

KEY WORDS: data mining, temporal reasoning, classification trees, C4.5, temporal rules

1. INTRODUCTION

Data mining is the process of discovering interesting knowledge, such as patterns, associations, changes, anomalies and significant structures, in large amounts of data stored in databases, data warehouses, or other information repositories. Due to the wide availability of huge amounts of data in electronic form, and the need for turning such data into useful information and knowledge for broad applications, data mining has attracted a great deal of attention in the 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, signals from biomedical sources, demographic data, etc. Although traditional statistical time series techniques can sometimes produce accurate results, few can provide easily understandable results. However, a drastically increasing number of users with a limited statistical background would like to use these tools. Therefore, it becomes more and more important to be able to produce results that can be interpreted by a domain expert without special statistical training. In the same time, we have a limited amount of tools proposed by researchers in the field of artificial intelligence, which produce, in principle, rules that are easier to understand.

The main tasks concerning the information extraction from time series database and on which the researchers concentrated their efforts may be divided in several directions. *Similarity/Pattern Querying* concerns the measure of similarity between two sequences or sub-sequences respectively. Different methods were developed, as window stitching [1] or dynamic time warping based matching [2] [3]. *Clustering/Classification* direction concentrates on optimal algorithms for clustering/classifying sub-sequences of time series into groups/classes of similar sub-sequences. Different techniques were proposed: Hidden Markov Model (HMM) [4], Dynamic Bayes Networks (DBNs) [5], Recurrent Neural Networks [6], supervised classification using piecewise polynomial modeling [7] and agglomerative clustering based on enhancing the time series with a line segment representation [8]. *Pattern finding/Prediction* methods concern the search for periodicity patterns (full or partial periodic) in time series databases. For full periodicity search there is a rich collection of statistic methods, like FFT [9]. For partial periodicity searching, different algorithms were developed, which explore properties related to partial periodicity such as the a-priori property, the max-subpattern-hit-set property [10] or point-wise periodicity [11]. *Rule extraction* approach concentrated to the extraction of explicit rules from time series, like inter-transaction association rules [12] or cyclic association rules [13]. Adaptive methods for finding rules whose conditions refer to patterns in time series were described in [14] and a general architecture for classification and extraction of comprehensible rules was proposed in [15].

The approaches concerning the information extraction from time series, described above, have mainly two shortcomings, which we try to overcome:

1. The first problem involves 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

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

understandable for a user. Therefore, we developed inference methods that will produce knowledge that can be represented in a 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 are of a much more restricted form than the rules we are proposing.

2. The second problem involves 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 at the time. However, we think this is not sufficient in order to produce knowledge usable for decision-making. Therefore, the methods we proposed to develop would be able to handle multi-dimensional data.

To overcome these problems we propose a methodology that integrates techniques developed both in the field of machine learning and in the field of statistics. The machine learning approaches will be used to extract symbolic knowledge and the statistical approaches will be used to perform numerical analysis of the raw data. The overall goal includes developing a series of methods able to extract/generate/describe temporal rules. These rules may have 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)

The main steps of the proposed methodology may be 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 may be extracted using statistical methods.
2. Inferring temporal rules: We may apply an inference process, using sets of events extracted from an event database as training sets, to obtain several classification trees. Then temporal rules may be extracted from these classification trees.

The rest of the paper is structured as follows. In Section 2, the main steps of the methodology are detailed, including also a brief description of the concept of classification trees. Some problems concerning the implementation of the training sets and of the temporal rules are treated in the next section. Section 4 presents some results of a synthetic time series database and the final section summarizes our work and points to future research.

2. THE METHODOLOGY

An event is in the core of our methodology and so a formal definition seems necessary. So, an event represents the "translation" of a sequence of raw data, of fixed predefined length, into a bracketed n-tuple, ($n \geq 1$) where the first term is the event label and all other terms represents features extracted from the sequence.

The Phase One. We will now describe the procedure we are proposing for the extraction of events. This procedure can be divided into two steps: time series discretisation, which captures the discrete aspect, and global feature calculation, which captures the continuous aspect

Time series discretisation. In the literature, different methods were proposed for the discretisation of times series (window's clustering method [14], ideal prototype template [8]). We adopted a simple solution, with an easy implementation. Starting with the sequence $s = (x_1, x_2, \ldots, x_n)$, we calculate the sequence of the differences between two consecutive values. The sorted list of these differences is then divided into *k* intervals, such that in each interval exactly 1/*k* proportion of values is found. The parameter *k* 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. Each interval may be then labeled using a symbol (a_i for the ith interval). Therefore, the discretisation version of *s*, *D(s)*, is simply the "translation" of the sequence of differences into the sequence of corresponding symbols. The event label, for an initial sequence of *m* points, is thus formed by the concatenation of $m-1$ corresponding symbols from *D(s)*. Of course, there are also other choices for establishing the intervals. If we are interested in a clear separation between the positive differences and the negative ones the zero value must be chosen as an interval extremity. On the other hand, if we are interested only in uncommon events (big differences, positive or negative) we give up the condition of equal distribution in each interval and conveniently choose the interval extremities.

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 important 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 define in advance, but as most of these features are simple descriptive statistics, they can easily be added or removed from the process.

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

1. Application of a first inference process, using the event database as a training database, to obtain a set of *classification trees* and
2. Application of a second inference process using the previously inferred classifications 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. A classification tree may be constructed by recursively partitioning a learning sample of data in which the class label and the value of the predictor variables for each case are known. Each partition is represented by a node in the tree. The hierarchical nature of a classification tree means that the relationship of a leaf to the tree on which it grows can be described by the hierarchy of splits of branches (starting from the root) leading to the last branch from which the leaf hangs. A variety of classification tree programs have been developed and we may mention QUEST [16], CART [17], FACT [18], THAID [19], CHAID [20] and last, but not least, C4.5 [21]. For our methodology we selected the C4.5 like approach. The tree resulting by applying this algorithm minimizes the observed error rate, using equal priors. To select a split, the C4.5 algorithm examines all possible splits for each predictor variable at each nod to find the split producing the largest improvement in goodness of fit. As goodness of fit, the C4.5 algorithm uses the gain criterion or, to rectify the inherent bias, the gain ratio criterion. The spitting process continues until all terminal nodes are pure or contain no more than a specified minimum number of cases or objects. After the tree which best classifies the training set is obtained, it is pruned at the "right-size". For this operation, the C4.5 algorithm estimates the predicted error rate in a leaf as the upper confidence limit for the probability of error (*E/N*, E-number of errors, N-number of covered training cases) multiplied by N.

Second inference process. Different classification trees, constructed starting with different training sets, generate finally different rules implying the same class. The natural way in which we may combine these rules is the application of the logical operator "OR". If we denote by $\{\overline{R}_i\}$ the set of implication clauses of the rule R_i, then from the set of rules $\{R_1,\ldots,R_k\}$ one infers a general rule having a body (the implication) of the form $(C_1 \wedge \ldots \wedge C_s) \wedge (D_1 \vee \ldots \vee D_p)$, where $C_i \in \bigcap_i \{\overline{R}_i\}$ and $D_i \in \bigcup_i (\{\overline{R}\}_i - \bigcap_j \{\overline{R}_j\})$. The confidence level of the inferred rule will be at least equal with the minimum confidence level for the initial rules.

3. IMPLEMENTATION PROBLEMS

The training set construction. Before we can start to apply the decision tree algorithm on the event database obtained in phase one, 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 represent the label of the class.

There are two different approaches on how the sequence of class labels is obtained. In a supervised methodology, the sequence that contains the classification (the values of the categorical dependent variable) is done by an expert. The situation becomes more difficult when we do not dispose of prior knowledge about the possible classifications. As an example, suppose that our database contains a set of time series representing the evolution of stock prices for different companies. We are interested in seeing if a given stock value depends on other stock values. Because the dependent variable (the stock price) is not categorical, it can't represent a classification used to create a classification tree. The idea is to use the sequence of labels of events extracted from the continuous time series as class labels.

Suppose we have *k* time series representing the predictor variables, $s_1, s_2, \ldots s_k$, where $s_i = \{s_{i1}, \ldots, s_{in}\}$ and each s_{ij} is an event. We have also a time series containing the class labels (i.e. the classification) $s_c = s_{c1}, \ldots s_{cn}$. The training set will be constructed using a procedure depending on three parameters. The first, t_0, represents the *present time*. Practically, the first tuple contains the class label 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, controls the class label $s_{c(t_0-t_p)}$ included in the last tuple. Consequently, the number of tuples in the training set is $t_p + 1$. The third parameter, *h*, controls the influence of the past events $s_{i(t-1)}, s_{i(t-2)}, \ldots, s_{i(t-h)}$ on the actual event s_{it} and 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*. Consequently, each tuple contains $k(h+1)$ events and one class label. The parameter *h* has also a side effect on the final temporal rules: because a tuple contains events dispersed on a time interval of length *h*, the predictability horizon in time for a temporal rule (named the *time window*) is limited on *h*.

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

The temporal rules. A tuple in the training set includes events that, by definition, have no explicit information on the time the event started. Practically, there is no time value processed during the creation of the classification tree. The solution we chose to "encode" the temporal information in the process of creation the classification tree is to establish 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 $\{A_0, A_1, \ldots, A_h, \ldots A_{k(h+1)-1}\}$. A rule extracted from a classification tree based on these attributes will have the

form "$(A_i = e_1) \wedge \cdots \wedge (A_s = e_p) \rightarrow C_j$", where $\{e_i\}$ are events and C_j is a label of a class or of an event. Suppose the set of the attributes that appear in the body of the rule is $\{A_{i_0}, \ldots, A_{i_m}\}$. The set of indexes $\{i_0, \ldots, i_m\}$ is transformed into the set $\{\bar{i}_0, \ldots, \bar{i}_m\}$, where "$\bar{i}$" means "i modulo *(h+1)*". If we denote by *t* the moment in time 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 $t - \bar{i}_j$.

4. EXPERIMENTAL RESULTS

For our experiments we used a synthetic database. Three time series represent the predictive variables and, choosing a supervised situation, we dispose also of a sequence of class labels, representing the classification. Each series had the length 500 and each value of the series was generated randomly, in a first phase, between 0 and 30. In a second phase, we modified some values such that the first series had, from time to time, decreasing sequences of length 5 (let's call it 'decrease'), the second series, sequences of 5 almost equal values ('stable') and the third series, increasing sequences of length 5 ('increase'). As we may observe in Fig.1, where only the first 33 values of the three time series were represented, such particular sequences start at time $t = 8$ and $t = 24$. If a 'decrease' starts at time *t* in the first time series, a 'stable' in the second and an 'increase' in the third series, then at time $t + 4$ the expert puts, in the classification sequence, the label "1". For all other situations, the label class will be "0". There are 39 classes labeled "1" among all 500 cases, which represents 7.8% of all cases. The reason for this particular labeling process is that we preferred a classification that is independent on the numerical values of the series, but depends on some particular behavior of the time series. A classification tree which would be constructed using only numerical values of the series in the training set would have a high error rate, due to the random character of the data.

During the discretisation phase we used a very simple approach that included defining five intervals, [-30, -10), [-10, -1), [-1, 1], (1, 10] and (10, 30] and encoding them with the letters {a, b, c, d, e}. Each sequence of length two, (s_i, s_{i+1}), is thus labeled depending on which interval the difference $s_i - s_{i+1}$ falls into. In this way the sequence 'decrease' may be labeled with a 4-character word using the letters {a, b}, the sequence 'stable' will be labeled *cccc* and the sequence 'increase' with a 4-character word using the letters {d, e}.

Different trees were constructed with the same parameters $t_0 = 280$ and $t_p = 274$ (the training set contains almost half of the data), but with different *h*. As we may observe in Fig.2, as long as the parameter *h* increases, the observed errors (the number of misclassified cases in the training set) and the prediction errors (the number of misclassified cases when the classification tree is applied to the remaining events in the database) diminish. This can be explained by the fact that past events influence the classification at present time. The more information from the past we take into consideration, the more the classification tree becomes precise. On the other hand one can see that this influence is limited to a time window of length 4 (the classification trees for *h* greater than four remaining unchanged). The rule implying the class "1", extracted from the classification tree with $h = 4$, is:

A0 in {a, b}, A4 in {b, a}, A5 in {c}, A6 in {c}, A8 in {c}, A9 in {c}, A14 in {e, d} -> class "1" ,

with a confidence of 93.8%. It is interesting to observe that the rule is not using all possible conditions (e.g. A1 in {a, b}, A2 in {a, b}, A3 in {a, b}), which means that not all events have an influence on the classification.

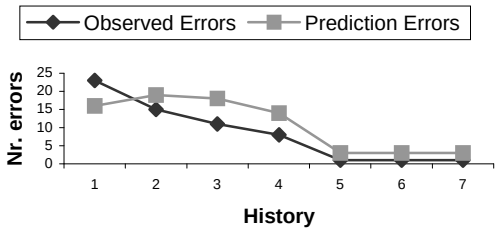

Fig. 2 The number of errors vs. history parameter

On the other hand we can see that for each time series the event farthest back in time, (A4, A9 and respectively, A14) is used. Applying the procedure for transforming the ordinary rules into temporal rules we obtain, into a more or less "natural language", the following rule:

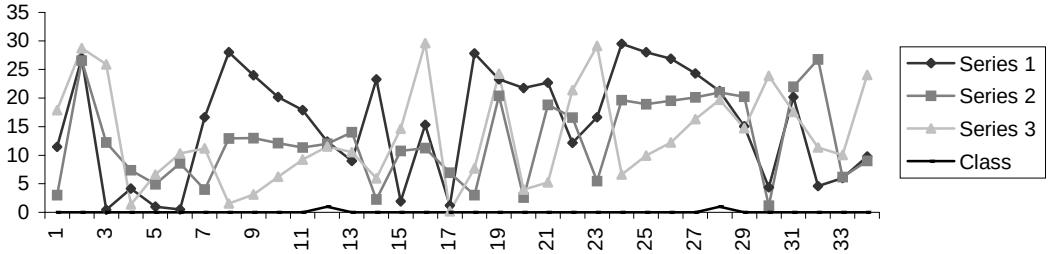

Fig. 1 The first 33 values of the time series : class "1" at t=12 and t=28

«If at time $t_0 - 5$ and $t_0 - 1$ the first time series decreases by more than one unit and at time $t_0 - 5$, $t_0 - 4$, $t_0 - 2$ and $t_0 - 1$ the second time series varies by maximum one unit and at time $t_0 - 5$ the third time series increases by more than one unit then at time t_0 we will have the class "1" ».

To analyze how the rules depend on initial training sets, we generated trees by varying the parameter t_0 and keeping $t_p = 145$ and $h = 4$. Table 1 presents the conditions from the body of the rule implying the class "1" extracted from different trees. The last line contains the conditions of the same rule when the classification tree was constructed using the largest possible training set.

t_0	Conditions	Conf.
250	A0 in {a,b}, A4 in {a,b}, B0 in {c}, B2 in {c}, B4 in {c}, C4 in {e,d}	84.3%
350	A0 in {a,b}, A1 in {a,b}, B0 in {c}, B3 in {c}, B4 in {c}, C4 in {e,d}	77.1%
450	A0 in {a,b}, B0 in {c}, B1 in {c}, B3 in {c}, B4 in {c}, C0 in {e,d}, C4 in {e,d}	79.5%
$t_0 = 500$ $t_p = 495$ $h = 4$	A0 in {a,b}, A4 in {a,b}, B0 in {c}, B1 in {c}, B3 in {c}, B4 in {e,d}, C0 in {e, d}, C4 in {e, d}	96.5%

Table 1.

It is obvious that the conditions do not remain the same, because the classification tree captures, in fact, a particular distribution of the events included in the training set. Applying the second inference process on this particular set of rules, we obtain the following general rule:

((A0 in {a ,b}) and (B0 in {c}) and (B4 in {c}) and (C4 in {e, d})) and ((A1 in {a, b}) or (A4 in {a, b}) or (B2 in {c}) or (B3 in {c}) or (C0 in {e, d})) -> class "1"

which will be then transformed into a temporal rule.

Because the initial classification was done such that it depends only on the labels of events ({a, b, c, d, e}), all the training sets constructed in the precedent examples did not included any of the possible features of the events. To analyze the influence of the features' presence in the training set, we modified the class labels by considering the following supplementary condition: if in the third series the value at the moment t +2 is greater than 15 then the class label at moment t + 4 is "1". The feature that must be considered here is simply the left extremity of the interval (s_i, s_{i+1}). In the training set we add new attributes for each added feature. Using the mechanism of indexing, it is possible to keep the connection between an event and his corresponding feature. The temporal rule implying the class "1", when the initial parameters are set to $t_0 = 499$, $t_p = 495$ and $h = 4$ is:

«If at time $t_0 - 5$ the first time series decreases by more than one unit and at time $t_0 - 5$, $t_0 - 4$, $t_0 - 2$ and $t_0 - 1$ the second time series varies by maximum one unit and at time $t_0 - 5$, $t_0 - 1$ the third time series increases by more than one unit and at time $t_0 - 2$ the value in the third series is greater than 14 then at time t_0 we will have the class "1" ».

As we already mentioned, in an unsupervised situation we take as the sequence of class labels the sequence of event labels, more precisely, of those events considered as dependent from the others. For our database, we considered the sequence of events extracted from the third time series as being implied by the events extracted from series one and two. The parameters for the training set are setting at $t_0 = 300$, $t_p = 280$ and h taking values between 0 and 18. Of course, due to the fact that the initial values of the time series were generated randomly, we do not expect the resulting classification trees to find some "nice" rules implying the corresponding events. But it is very interesting to see that the observed error rate goes down even in this "non-dependence context" when the parameter *h* increases (see Fig. 3). On the other hand, the prediction error rate does not have the same behavior. This is obvious because the remaining data events have nothing in common with the events of the training set (again because of the random generator process) and thus the rules have little chances to be applicable.

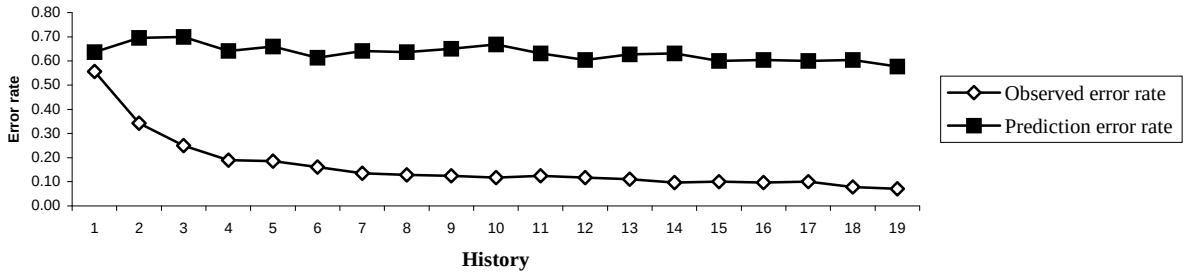

Fig. 3 The number of errors vs. the history for the unsupervised classification

5. CONCLUSION

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. To obtain what we called "temporal rules", a discretisation phase that extracts "events" from raw data may be applied first, followed by an inference phase, which constructs classification trees from these events. The discrete and continuous characteristics of an "event", according to its definition, 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 (used later to obtain the classification tree) is proposed. This procedure may depend on three parameters, among others, the so-called *h*istory that controls the time window of the temporal rules. The experiments we conducted on a synthetic database showed that the process of event extraction has a major influence on the observed error rate when the classification depends rather on the shape of the time series than on their numerical values. As long as the parameter h increases, the observed error rate decreases, until the time window is large enough to capture (almost) all the relations between events. This dependence between the observed error rates and the parameter *h* permits us to stop the process of adding new attributes as soon as the structure of the classification tree becomes stable and thus prevents us from overfitting the tree.

We wish to emphasis that our methodology is, at this time, just an alternative to other procedures and methods for the extraction of knowledge from time series databases. In order to analize the quality of the final temporal rules, it is necessary to conduct comparative experiments, using different methods, on the same databases. Finally, we would like to mention an open question, which is common to all these methods and 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] G. Das, D. Gunopulos, H. Mannila, Finding Similar Time Series, *Proc. of First Conference PKDD*, Trondheim, 1997, 88-100

[2] R. McConnell, Ψ-S Correlation and dynamic time warping: Two methods for tracking ice floes in SAR images, *IEEE Transactions on Geoscience and Remote sensing*, *29*(6): 1004-1012, 1991

[3] D. J. Berndt, J. Clifford: Using dynamic time warping to find patterns in time series, *Proc. of the First International Conference on Knowledge Discovery and Data Mining*, New York, 1994, 359-370

[4] L. Rabiner, B. Juang, An introduction to Hidden Markov Models, *IEEE Magazine on Acoustics, Speech and Signal Processing*, *3*, 1986, 4-16

[5] N. Friedman, K. Murphy, S. Russel, Learning the structure of dynamic probabilistic networks, *Proc. of 14th Conference on Uncertainty in Artificial Intelligence*, 1998, AAAI Press, pg. 139-147

[6] Y. Bengio, *Neural Networks for Speech and Sequence Recognition* (International Thompson Publishing Inc., London, 1996)

[7] S. Mangaranis, *Supervised Classification with temporal data* (Ph.D. Thesis, Computer Science Department, School of Engineering, Vanderbilt University, 1997)

[8] E. Keogh, M. J. Pazzani, An Enhanced Representation of time series which allows fast and accurate classification, clustering and relevance feedback, *Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining*, 1998, 239-243.

[9] H. Loether, D. McTavish, *Descriptive and Inferential Statistics: An introduction* (Allyn and Bacon, 1993)

[10] J. Han, W. Gong, Y. Yin, Mining Segment-Wise Periodic Patterns in Time-Related Databases, Proc. *of the Fourth International Conference on Knowledge Discovery and Data Mining*, 1998, 214-218.

[11] J. Han, G. Dong, Y. Yin, Efficient Mining of Partial Periodic Patterns in Time Series Database, *Proc. of International Conference on Data Engineering*, 1999, Sydney, Australia, 106-115

[12] H. Lu, J. Han, L. Feng, 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

[13] B. Ozden, S. Ramaswamy, A. Silberschatz, Cyclic association rules, *Proc. of International Conference on Data Engineering*, Orlando, 1998, 412-421

[14] G. Das, K. Lin, H. Mannila, G Renganathan, P Smyth, Rule Discovery from Time Series, *Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining*, 1998, 16-22.

[15] M. Waleed. Kadous, Learning Comprehensible Descriptions of Multivariate Time Series, *Proc. of the Sixteenth International Conference on Machine Learning*, 1999, 454-463

[16] W. Loh, Y. Shih, Split Selection Methods for Classification Trees, *Statistica Sinica*, *7*, 1997, 815-840

[17] L. Breiman, J.H. Friedman, R. A. Olshen, C. J. Stone *Classification and regression trees* (Wadsworth & Brooks/ Cole Advanced Books & Software, 1984)

[18] W. Loh, N. Vanichestakul, Tree-structured classification via generalized discriminant analysis (with discussion), *Journal of American Statistical Association*, *83*, 1998, 715-728.

[19] J. Morgan, R. Messenger, *THAID:* A sequential analysis program for the analysis of nominal scale dependent variables, *Technical report*, University of Michigan, 1973

[20] G. V. Kass, An exploratory technique for investigating large quantities of categorical data, *Applied Statistics*, *29*, 1980, 119-127

[21] J. R. Quinland, *C4.5: Programs for Machine Learning* (Morgan Kauffmann Publishers, San Mateo, California, 1993)