Volume 3, Issue 12, December 2013 ISSN: 2277 128X
International Journal of Advanced Research in
Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Mining Temporal Sequences Using Transform Techniques

M.Yasodha, P.Usha, S.Saranya, N.Tamilselvi
Department of Computer
Science & Bharathiar
University, India

Abstract—One of the main unresolved problems that arise during the data mining process is treating data that contains temporal information. In this situation, to get a good knowledge of entire data requires that the data should be viewed as sequence of events. Over the last decade many interesting techniques of temporal data mining were proposed and shown to be useful in many applications. Temporal Data Mining is a rapidly evolving area of research that is at the intersection of several disciplines, including statistics, temporal pattern recognition, temporal databases, optimization, visualization, high-performance computing, and parallel computing. This paper is first intended to serve as an overview of the temporal data mining in research and applications.

Keywords—APRIORI, CAPSUL, DFT, DWT, SDL.

I. INTRODUCTION

A huge amount of data is collected everyday and a real challenge is to find a good knowledge from such huge amount of data. Data mining is an emerging research direction to meet this challenge. Data mining techniques can be deployed to search large databases to discover useful information that might unknown for the user. Many data mining problems involve temporal aspects. Examples range from transaction databases in health care and insurance, stock exchange and customer goods in market sectors, sensor data collected from sensor networks, to scientific databases in geophysics and astronomy. Mining this temporal data poses interesting challenges than mining static data. There are two factors that contribute to the popularity of temporal data mining. The first factor is an increase in the volume of temporal data stored, as many real-world applications deal with huge amount of temporal data. The second factor is the mounting recognition in the value of temporal data.

Since temporal data mining is relatively a new field of research, there is no widely accepted taxonomy yet. Several approaches have been used to classify data mining problems and algorithms. Roddick & Spiliopoulou have presented the mining of temporal data using three dimensions: data type, mining operations and type of timing information (ordering). On the other hand, Antunes & Oliviera base their classification on representation, similarity and operations. Based on its main tasks, temporal data mining can be grouped into five broad categories: prediction, classification, clustering, search and retrieval, and pattern discovery (Laxman & Sastry 2006).

A. Definition and Tasks of Temporal Data Mining

Temporal data mining deals with the problem of mining patterns from temporal data, which can be either symbolic sequences or numerical time series.

B. Definition

Temporal Data Mining is a single step in the process of Knowledge Discovery in Temporal Databases that enumerates structures temporal patterns or models) over the temporal data, and any algorithm that enumerates temporal patterns from, or fits models to, temporal data is a Temporal Data Mining Algorithm. The ultimate goal of temporal data mining is to discover hidden relations between sequences and sub sequences of events. The discovery of relations between sequences of events involves mainly three steps: the representation and modeling of the data sequence in a suitable form; the definition of similarity measures between sequences; and the application of models and representations to the actual mining problems.

Depending on the nature of the event sequence, the approaches to solve the problem may be quite different. A sequence composed by a series of nominal symbols from a particular alphabet is usually called a temporal sequence and a sequence of continuous, real-valued elements, is known as a time series.

II. TEMPORAL DATA MINING TASKS

Data mining has been used in a wide range of applications. Data mining tasks presented by Han and Kamber, extended to temporal data mining.

- Prediction
- Classification
- Clustering
- Searching and Retrieval
- Pattern Discovery

A. Prediction

Prediction is the task of explicitly modeling variable dependencies to predict a subset of the variables from others. The task of time series prediction is to forecast future values of the time series based on its past samples. In order to perform the prediction, one needs to build a predictive model from the data. Koskela et al. (1996) have studied neural networks for nonlinear modeling of time series data. The prediction problem for symbolic sequences has been addressed in AI research by Dietterich and Michalski (1985).

B. Classification

Classification is the task of assigning class labels to the data according to a model learned from the training data where the classes are known. Classification is one of the most common tasks in supervised learning, but it has not received much attention in temporal data mining (Antunes & Oliveira 2001). In sequence classification, each sequence presented to the system is assumed to belong to one of predefined classes and the goal is to automatically determine the corresponding category for a given input sequence. Examples of sequence classification applications include signature verification (Nalwa 1997), gesture ecognition (Yamato, Ohya & Ishii 1992), and hand-written word recognition (Kundu, He & Bahl 1988).

C. Clustering

Clustering1 is the process of finding intrinsic groups, called clusters, in the data. Clustering of time series (or sequences) is concerned with grouping a collection of time series (or sequences) based on their similarity. Time series clustering has been shown effective in providing useful information in various domains (Liao 2005). For example, in financial data, clustering can be used to group stocks that exhibit similar trends in price movements. Another example could be clustering of fMRI time series for identifying regions with similar patterns of activation (Goutte, Toft & Rostrup 1999). Clustering of sequences is relatively less explored but is becoming increasingly important in data mining applications such as web usage mining and bioinformatics Laxman & Sastry 2006). A survey on clustering time series has been presented by Liao (2005).

D. Searching and Retrieval

Searching and Retrieval Searching and retrieval are concerned with efficiently locating subsequences or sub-series (often referred to as queries) in large databases of sequences or time series. In data mining, query based searches are more concerned with the problem of efficiently locating approximate matching than exact matching, known as content-based retrieval.

An example of a time series retrieval application is to find out all the days of the year in which a particular stock had similar movements to those of today. Another example is finding products with similar demand cycles. An example of sequence retrieval is finding gene expression patterns that are similar to the expression pattern of a given gene. In order to address the time series retrieval problem, different notions of similarity between time series and indexing techniques have been proposed. There is considerably less work in the area of sequence retrieval, and the problem is more general and difficult. For more detail about time series and sequence retrieval can be found in Das and Gunopulos (2003).

E. Pattern Discovery

Unlike in search and retrieval applications, in the pattern discovery there is no specific query in hand with which to search the database. The objective is simply to discover all patterns of interest. While the other tasks described earlier have their origins in other disciplines like statistics, machine learning or pattern recognition, the pattern discovery task has its origin in data mining itself. A pattern is a local structure in the data. There are many ways of defining what constitutes a pattern. There is no universal notion for interestingness of a pattern either. However, one concept that is normally used in data mining is that of frequent patterns, that is, patterns that occurs many times in the data. Much of data mining literature is concerned with formulating useful pattern structures and developing efficient algorithms for discovering frequent patterns.

Methods for finding frequent patterns are important because they can be used for discovering useful rules, which in turn can be used to infer some interesting regularities in the data. A rule usually consists of a pair of a left-hand side proposition (the antecedent) and a right-hand side proposition (the consequent). The rule states that when the antecedent is true, then the consequent will be true as well. It was mentioned above that temporal data can be symbolic sequences or time series.

The pattern discovery task typically assumes an underlying symbolic representation. Therefore, to apply the pattern discovery methods on time series data, the time series should be first converted into a discrete representation, for example by first forming subsequences (using a sliding window) and then clustering these subsequences using a suitable measure of pattern similarity (Das, Lin, Mannila, Renganathan & Smyth 1998). Another method can be used by quantizing the time series into levels and representing each level (e.g., high, medium, etc.) by a symbol (Aref, Elfeky & Elmagarmid 2004). A survey on time series abstraction methods can be found in H¨oppner (2002).

F. Association Rule

One of the most common approaches to mining frequent patterns is the apriori method and when a transactional database represented as a set of sequences of transactions performed by one entity is used (as described in section 3.5), the manipulation of temporal sequences requires that some adaptations be made to the apriori algorithm. The most important modification is on the notion of support: support is now the fraction of entities, which had consumed the itemsets in any of their possible transactions , i.e. an entity could only contribute one time to increment the support of each itemset, beside it could had consumed that itemset several times.

After identifying the large itemsets, the itemsets with support greater than the minimum support allowed, they are translated to an integer, and each sequence is transformed in a new sequence, whose elements are the large itemsets of the previous- one. The next step is to find the large sequences. For achieve this, the algorithm acts iteratively as apriori: first it generates the candidate sequences and then it chooses the large sequences from the candidate ones, until there are no candidates. One of the most costly operations in apriori-based approaches is the candidate generation. A proposal to frequent pattern mining states that it is possible to find frequent patterns avoiding the candidate generation-test. Extending this to deal with sequential data is presented in. The discovery of relevant association rules is one of the most important methods used to perform data mining on transactional databases. An effective algorithm to discover association rules is the apriori. Adapting this method to deal with temporal information leads to some different approaches. Common sub-sequences can be used to derive association rules with predictive value, as is done, for instance, in the analysis of discretized, multi-dimensional time series.

Another method consists on considering cyclic rules. A cyclic rule is one that occurs at regular time intervals, i.e. Transaction that support specific rules occur periodically, for example at every first Monday of a month. In order to discover these rules, it is necessary to search for them in a restrict portion of time, since they may occur repeatedly at specific time instants but on a little portion of the global time considered. A method to discover such rules is applying an algorithm similar to the apriori, and after having the set of traditional rules, detects the cycles behind the rules. A more efficient approach to discover cyclic rules consists on inverting the process: first discover the cyclic large item sets and then generate the rules. A natural extension to this method consists in allowing the existence of different time units, such as days, weeks or months, and is achieved by defining calendar algebra to define and manipulate groups of time intervals. Rules discovered are designated calendric association rules.

A different approach to the discovery of relations in multivariate time sequences is based on the definition of N-dimensional transaction databases. Transactions in these databases are obtained by discretizing, if necessary, continuous attributes. This type of databases can then be mined to obtain association rules. However, new definitions for association rules, support and confidence are necessary. The great difference is the notion of address, which locates each event in a multi-dimensional space and allows for expressing the confidence and support level in a new way.

III. MINING TEMPORAL SEQUENCES

One possible definition of data mining is "the nontrivial extraction of implicit, previously unknown and potential useful information from data". The ultimate goal of temporal data mining is to discover hidden relations between sequences and subsequences of events. The discovery of relations between sequences of events involves mainly three steps: the representation and modeling of the data sequence in a suitable form; the definition of similarity measures between sequences; and the application of models and representations to the actual mining problems. Other authors have used a different approach to classify data mining problems and algorithms. Roddick has used three dimensions: data type, mining operations and type of timing information. Although both approaches are equally valid, we preferred to use representation, similarity and operations, since it provided a more comprehensive and novel view of the field. Depending on the nature of the event sequence, the approaches to solve the problem may be quite different. A sequence composed by a series of nominal symbols from a particular alphabet is usually called a temporal sequence and a sequence of continuous, real-valued elements, is known as a time series.

Time series or, more generally, temporal sequences, appear naturally in a variety of different domains, from engineering to scientific research, finance and medicine. In engineering matters, they usually arise with either sensor-based monitoring, such as telecommunications control or log-based systems monitoring. In scientific research they appear, for example, in spatial missions.

In finance, applications on the analysis of product sales or inventory consumptions are of great importance to business planning . Another very common application in finance is the prediction of the evolution of financial data. In healthcare, temporal sequences are a reality for decades; with data originated by complex data acquisition systems like ECG's, or even with simple ones like measuring the patient temperature or treatments effectiveness . In the last years, with the development of medical informatics, the amount of data has increased considerably, and more than ever, the need to react in real-time to any change in the patient behavior is crucial. In general, applications that deal with temporal sequences serve mainly to support
diagnosis and to predict future behaviors.

A. Representation of Temporal Sequences

A fundamental problem that needs to be solved in the representation of the data and the pre-processing that needs to be applied before actual data mining operations take place. The representation problem is especially important when dealing with time series, since direct manipulation of continuous, high-dimensional data in an efficient way is extremely difficult. This problem can be addressed in several different ways. One possible solution is to use the data with only minimal transformation, either keeping it in its original form or using windowing and piecewise linear approximations to obtain manageable sub-sequences. One issue that is relevant for all the representation methods addressed is the ability to discover and represent the subsequences of a sequence. A method commonly used to find subsequences is to use a sliding window of size w and place it at every possible position of the sequence. Each such window defines a new subsequence, composed by the elements inside the window.

B. Time-Domain Continuous Representations

A simple approach to represent a sequence of real-valued elements (time series) is using the initial elements, ordered by their instant of occurrence without any preprocessing alternative consists in finding a piecewise linear function able to approximately describe the entire initial sequence. In this way, the initial sequence is partitioned in several segments and each segment is represented by a linear function. In order to partition the sequence there exist two main possibilities: define beforehand the number of segments or discover that number and then identify the correspondent segments. The objective is to obtain a representation amenable to the detection of significant changes in the sequence. While this is a relatively straightforward representation, not much is gained in terms of our ability to manipulate the generated representations. One possible application of this type of representations is on change-point detection, where one wants to locate the points where a significant change in behavior takes place.

Another proposal, based on the idea that the human visual system partitions smooth curves into linear segments, has been proposed. It mainly consists on segmenting a sequence by iteratively merging two similar segments. Choosing which segments are to be merged is done based on the squared error minimization criteria. An extension to this model consists in associating with each segment a weight value, which represents the segment importance relatively to the entire sequence. In this manner, it is possible to compare sequences mainly by looking at their most important segments . With this representation, a more effective similarity measure can be defined, and consequently, mining operations may be applied.

One significant advantage of these approaches is the ability to reduce the impact of noise. However, problems with amplitude differences (scaling problems) and the existence of gaps or other time axis distortion are not addressed easily.

C. Transformation Based Representations

The main idea of Transformation Based Representations is to transform the initial sequences from time to another domain, and then to use a point in this new domain to represent each original sequence.

One proposal uses the Discrete Fourier Transform (DFT) to transform a sequence from the time domain to a point in the frequency domain. Choosing the k first frequencies, and then representing each sequence as a point in the k dimensional space, achieves this goal. The DFT has the attractive property that the amplitude of the Fourier coefficients is invariant under shifts, which allows extending the method to find similar sequences ignoring shifts. Other possibilities that have been proposed represent each sequence as a point in the n-dimensional space, with n being the sequence length. This representation can be viewed as a time-domain continuous representation. However, if one maps each subsequence to a point in a new space with coordinates that are the derivatives of each original sequence point, this should be considered as a transformation based representation. In this case, the derivative at each point is determined by the difference between the point and its preceding one.

A more recent approach uses the Discrete Wavelet Transform (DWT) to translate each sequence from the time domain into the time/frequency domain. The DWT is a linear transformation, which decomposes the original sequence into different frequency components, without loosing the information about the instant of the elements occurrence. The sequence is then represented by its features, expressed as the wavelet coefficients. Again, only a few coefficients are needed to approximately represent the sequence.

With these kinds of representations, time series became a more manageable object, which permit the definition of efficient similarity measures and an easier application to common data mining operations.

D. Discretization Based Methods

A third approach to deal with time series is the translation of the initial sequence (with real-valued elements) to a discretized sequence, this time composed of symbols from an alphabet. One language to describe the symbols of an alphabet and the relationship between two sequences was proposed as Shape Definition Language (SDL) is also able to describe shape queries to pose to a sequence database and it allows performing 'blurry' matching. A blurry match is one where the important thing is the overall shape, therefore ignoring the specific details of a particular sequence. The first step in the representation process is defining the alphabet of symbols and then translating the initial sequence to a sequence of symbols. The translation is done by considering transitions from an instant to the following one, and then assigning a symbol of the described alphabet to each transition.

Constraint-Based Pattern Specification Language (CAPSUL) is another language used to specify patterns, which similarly to SDL, describes patterns by considering some abstractions of the values at time intervals. The key difference between these two languages is that CAPSUL uses a set of expressive constraints upon temporal objects, which allows describing more complex patterns, for instance, periodic patterns.

A particular case is when change ratios are classified into one of two classes: large fluctuations or small fluctuations. This way, each time series is converted into a binary sequence, perfectly suited to be manipulated by genetic algorithms. A different approach to convert a sequence into a discrete representation is using clustering. First, the sequence originates a set of subsequences with length w, by sliding a window of width w. Then, the set of all subsequences is clustered, originating k clusters, and a different symbol is associated with each cluster. The discretized version of the initial sequence is obtained by substituting each subsequence by the symbol associated to the cluster. Another method to convert time series into a sequence of symbols is based on the use of self-organizing maps.

This conversion consists on three steps:

first, a new series composed by the differences between consecutive values of the original time series is derived; second windows with size d (called delay embedding are inputted to the SOM, which finally outputs the winner node. Each node is associated with a symbol, which means that the resultant sequence may be viewed as a sequence of symbols. The advantage of these methods is that the time series is partitioned in a natural way, depending on its values. However, the symbols of the alphabet are usually chosen externally, which means that they are imposed by the user, who has to know the most suitable symbols, or are established in an artificial way. The discretized time series is more amenable to manipulation and processing than the original, continuous valued, time series.

IV. GENERATIVE MODELS

A significantly different approach consists in obtaining a model that can be viewed as a generator for the sequences obtained. Several methods that use probabilistic generators have been proposed. One such proposal uses semi-Markov models to model a sequence and presents an efficient algorithm that can find appearances of a particular sub-sequence in another sequence. A different alternative is to find partial orders between symbols and build complex episodes out of serial and parallel compositions of basic events.

A further development builds a mixture model that generates sequences similar to the one that one wishes to model, with high probability. In this approach, an intuitively appealing model is derived from the data, but no particular applications in classification, prediction or similarity-based mining are presented. These models can be viewed as graph-based models, since one or more graphs that describe the relationships between basic events are used to model the observed sequences. A class of other approaches aim at inferring grammars from the given time sequence. Extensive research in grammatical inference methods has led to many interesting results but has not yet found their way into a significant number of real applications. Usually, grammatical inference methods are based on discrete search algorithms that are able to infer very complex grammars. Neural net based approaches have also been tried but have been somewhat less successful when applied to complex problems, . However, some recent results have shown that neural-network based inference of grammars can be used in realistic applications.

The inferred grammars can belong to a variety of classes, ranging from deterministic regular grammars to stochastic and context free grammars. It is clear that the grammar models obtained by induction from examples can be used for prediction and classification, but, so far, these methods have not been extensively applied to actual problems in data mining.

A. Transactional databases with timing information

A type of time sequences that does not easily match any of the classes described above are transactional datasets that incorporate timing information. For example, one might have a list of items bought by customers and would like to consider the timing information contained in the database. This type of high-dimensional discrete data cannot be modeled effectively by any of the previously described approaches.

Although the modeling and representation of this type of data is an important problem in temporal data mining, to our knowledge, there exist only a few proposals for the representation of data that has this kind of characteristics. In one of the proposed representations each particular set of items is considered a new symbol of the alphabet and a new sequence is created from the original one, by using this new set of symbols. These symbols are then processed using methods that aim of finding common occurrences of sub-sequences in an efficient way.

This representation is effective if the objective is to find common events and significant correlations in time, but fails in applications like prediction and classification. Other authors have used a similar representation , although they have proposed new methods to manipulate it.

B. Pattern discovery

In pattern discovery there is no specific query in hand with which to search the database. The objective is simply to unearth all patterns of interest. It is worthwhile to note at this point that whereas the other temporal data mining tasks discussed earlier in (i. e. sequence prediction, classification, clustering and association rule) had their origins in other disciplines like estimation theory, machine learning or pattern recognition, the pattern discovery task has its origins in data mining itself. In that sense, pattern discovery, with its exploratory and unsupervised nature of operation, is something of a sole preserve of data mining. For this reason, this review lays particular emphasis on the temporal data mining task of pattern discovery.

A pattern is a local structure in the data. It would typically be like a substring or a substring with some 'don't care' characters in it etc. The problem of pattern discovery is to unearth all 'interesting' patterns in the data. There is no universal notion for interestingness of a pattern either. However, one concept that is found very useful in data mining is that of frequent patterns. A frequent pattern is one that occurs many times in the data.

Much of data mining literature is concerned with formulating useful pattern structures and developing efficient algorithms for discovering all patterns which occur frequently in the data. Methods for finding frequent patterns are considered important because they can be used for discovering useful rules. These rules can in turn be used to infer some interesting regularities in the data. A rule consists of a pair of Boolean-valued propositions, namely, a left-hand side proposition (the antecedent) and a right-hand side proposition (the consequent). The rule states that when the antecedent is true, then the consequent will be true as well. Rules have been popular representations of knowledge in machine learning and AI for many years.

Decision tree classifiers, for example, yield a set of classification rules to categorize data. In data mining, association rules are used to capture correlations between different attributes in the data (Agrawal & Srikant 1994). In such cases, the (estimate of) conditional probability of the consequent occurring given the antecedent is referred to as confidence of the rule. For example, in a sequential data stream, if the pattern "B follows A" appears f1 times and the pattern "C follows B follows A" appears f2 times, it is possible to infer a temporal association rule "whenever B follows A, C will follow too" with a confidence (f2/f1). A rule is usually of interest, only if it has high confidence and it is applicable sufficiently often in the data, i. e., in addition to the confidence (f2/f1) being high, frequency of the consequent (f2) should also be high.

One of the earliest attempts at discovering patterns (of sufficiently general interest) in sequential databases is a pattern discovery method for a large collection of protein sequences (Wang et al 1994). A protein is essentially a sequence of amino acids. There are 20 amino acids that commonly appear in proteins, so that, by denoting each amino acid by a distinct letter, it is possibly to describe proteins (for computational purposes) as symbolic sequences over an alphabet of size twenty. As was mentioned earlier, protein sequences that are similar or those that share similar subsequences are likely to perform similar biological functions. Wang et al (1994) consider a large database of more than 15000 protein sequences. Biologically related (and functionally similar) proteins are grouped together into around 700 groups.

The problem now is to search for representative (temporal) patterns within each group. Each temporal pattern is of the form $* X1 * X2 \cdots * XN$ where the Xi 's are the symbols defining the pattern and $*$ denotes a variable length "don't care" sequence. A pattern is considered to be of interest if it is sufficiently long and approximately matches sufficiently many protein sequences in the database. The minimum length and minimum number of matches are user defined parameters. The method by Wang et al (1994) first finds some candidate segments by constructing a generalized suffix tree for a small sample of the sequences from the full database. These are then combined to construct candidate patterns and the full database is then searched for each of these candidate patterns using an edit distance based scoring scheme.

The number of sequences (in the database) which are within some user-defined distance of a given candidate pattern is its final occurrence score and those patterns whose score exceeds a user-defined threshold are the output temporal patterns.

V. CONCLUSION

Temporal data mining is a very fast expanding field with many new research results reported and many new temporal data mining analysis methods or prototypes developed recently. Some articles of overview of temporal data mining have discussed in different frameworks for coveing research and application in temporal data mining. For example, Roddick and Spiliopoulou have presented a comprehensive overview of techniques for the mining of temporal data.In this report we have provided an overview of the temporal data mining process and some background to Temporal Data Mining. Also we discussed a difficult and fundamental problem, a general analysis theory of temporal data mining and provided some answers to the problem. This leads intoa discussion on why there was a need for Temporal Data Mining in industry, which has been a major factor in the efforts that have gone into building the present generation of Temporal Data Mining Systems. We have presented a number of areas which are related to Temporal Data mining in their objectives and compared and contrasted these technologies with Temporal Data Mining.

REFERENCES

[1] B. R. Gaines and P. Compton. Induction of meta knowledge about knowledge discovery. IEEE Trans. On Knowledge And Data Engineering, 5:990–992, 1993.

[2] Cláudia M. Antunes 1, and Arlindo L. Oliveira 2- Temporal Data Mining: an overview

C. Glymour, D. Madigan, D. Pregibon, and P. Smyth. Statistical inference and data mining. Communications of the ACM, 39(11):35–41, Nov. 1996.

[3] D. Brillinger, editor. Time Series: Data Analysis and Theory. Holt, Rinehart and Winston, New York, 1975.

[4] D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, editors. Learning bayesian networks: the combination of knowledge and statistical data. AAAI Press, 1994.

[5] D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, editors. Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97). AAAI Press, 1997.

[6] F. H. Grupe and M. M. Owrang. Data-base mining - discovering new knowledge and competitive advantage. Information Systems Management, 12:26–31, 1995.

[7] G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining. AAAI Press / MIT Press, 1995.

[8] Gwadera R, Atallah M J, Szpankowski W 2005 Markov models for identification of significant episodes. In Proc. 2005 SIAM Int. Conf. on Data Mining (SDM-05), Newport Beach, California

[9] Han J, Kamber M 2001 Data mining: Concepts and techniques (San Fransisco, CA: Morgan Kauffmann)

[10] Han J, GongW, Yin Y 1998 Mining segment-wise periodic patterns in time-related databases. In Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining (KDD'98), New York, pp 214–218

[11] J. Han, Y. Cai, and N. Cercone. Knowledge discovery in databases: An attribute-oriented approach. In Proceedings of the 18th VLDB Conference, pages 547–559, Vancouver, British Columbia, Canada, Aug. 1992.

[12] J. W. Han, Y. D. Cai, and N. Cercone. Data-driven discovery of quantitative rules in relational databases. Ieee Trans. On Knowledge And Data Engineering, 5:29– 40, Feburary 1993.

[13] J. W. Han, Y. Yin, and G.Dong. Efficient mining of partial periodic patterns in time series database. IEEE Trans. On Knowledge And Data Engineering, 1998.

[14] P. Cheeseman and J. Stutz. Bayesian classification (AUTOCLASS):Theory and results. In U. M. Fayyad,

[15] SRIVATSAN LAXMAN and P S SASTRY - A survey of temporal data mining

[16] T. Fulton, S. Salzberg, S. Kasif, and D. Waltz. Local induction of decision trees: Towards interactive data mining. In Simoudis et al. [21], page 14.

[17] Weiqiang Lin, Mehmet A. Orgun Graham J. Williams - An Overview of Temporal Data Mining Wolfer's sunspot numbers. Philos. Trans. R. Soc. London A226

[18] Yan X, Han J, Afshar R 2003 CloSpan: Mining closed sequential patterns in large datasets. In Proc. 2003 Int. SIAM Conf. on Data Mining (SDM03), San Fransisco, CA

[19] Yule G 1927 On a method of investigating periodicity in distributed series with special reference to Zaki M J 1998 Efficient enumeration of frequent sequences. In Proc. ACM 7th Int. Conf. Informationand Knowledge Management (CIKM)

