DISCOVERY TIME- SERIES DATA USING TIME INTERVALS CLUSTERING IN TRAFFIC CONTROL SYSTEM

Sitesh kumar sinha
Department of Information Technology
AISECT University, Bhopal
(siteshkumarsinha@gmail.com)

Rajiv Saxena
Department of Information Technology
Sagar Institute of Research & Technology, Bhopal
(rajeev_saxena10@yahoo.com)

Abstract

This is proposed research on sequence mining of transactional data. However, there are many applications where it is important to find significant intervals in which some events occur with specified strength. We study approaches to convert point-based data into intervals, thereby predicting the next occurrence of the event.. We compare the performances of various approaches in terms of computation time, number of passes, coverage and interval statistics like density, interval-length and interval-confidence. We propose an approach to clustering using the significant intervals produced. Furthermore, we use these intervals, which serve as representative areas of the dataset as input to a Hybrid A-priori algorithm to mine for sequential patterns. We present the two types of interval semantics that can be used with sequential mining. We formulate Hybrid A-priori sequential algorithm that accepts intervals as input. Finally, we summarize the results and use these results in traffic control system.

I. INTRODUCTION

Discovering the hidden knowledge is not a straightforward task. To compete effectively in today's market, decision makers need to identify and utilize this information buried in the collected data and take advantage of the high return opportunities in a timely fashion. The key here is the generation of previously unknown knowledge from huge datasets. The process of mining is driven by the outcome requirements. Based on what we want, a specific data mining technique is employed. The different data mining techniques and their outcomes are briefly discussed below [1].

The focus of our approach lies between clustering and sequential mining since both kinds of information are required to discover frequent patterns and answer queries related to intelligent environments. As mentioned above, our approach has enormous potential in intelligent environments where the key is to continuously learn from the surroundings and automate the inhabitant's activities.

II. CLASSIFICATION

This is a process of grouping items based on a classifying attribute. A model is then built based on the values of other attributes to classify each item to a particular class. A training dataset is typically used for validating and tuning the model. The classification technique may be used, for example, to identify the most probable consumers for a product, based on their spending patterns.

III. CLUSTERING

The process of clustering tries to group the data set in such a way that the data points in one cluster are more similar to one another while the data points in different clusters are more dissimilar. A similarity measure needs to be defined and the quality of the outcome, to a large extent, depends on the appropriateness of the similarity measure for the data set. The technique of clustering, for example, can be used to divide the market into distinct groups, so that each group can be targeted with a different strategy.

The basic difference between classification and clustering is that in classification, the classifying class is known previously (also known as supervised), whereas clustering does not assume any knowledge of clusters (unsupervised).

IV. PREDICTION

Data mining typically makes use of statistical analysis when it comes to predicting the next value of a continuous variable rather than a categorical label. Prediction of continuous values can be modeled by statistical techniques of regression [6]. Many problems can be solved by linear regression and even more by applying transformations to the variables so that a non- linear problem can be converted into a linear one. Linear regression is the simplest form of regression where the data is modeled as a straight line. Bi-variant linear regression models a random value Y, as a linear function of another random value X, that is,

$Y = a + bX$

Where a and b are regression coefficients specifying the Y intercept and slope of the line respectively. Many times, although the application cannot be modeled as a straight line to predict the value of Y given X, it can be viewed as Y=F(X). We focus on prediction of Y given X, which is a series of events changing with time. This can be described as a time-series database. Sometimes, there are several other variables that affect Y as it can have multiple values over a range of X. Multiple variables and the classification of the variables given below can make regression overtly complicated.

There are four major characteristics that are used to categorize time- series data [6]

Long- term or trend movement: This indicates the general direction over which the time- series graph is moving over a long interval of time.

Cyclic movements or cyclic variations: These refer to the cycles, or long-term oscillations about a trend line or curve, which may or may not be periodic.

Seasonal movements or seasonal variations: These movements are due to the events that recur annually. In other words, seasonal movements are the nearly identical patterns that a time-series appears to follow during corresponding months of each year.

Irregular or Random movements: These characterize the sporadic motion of the time - series due to random or chance events.

The common method of determining trend is to calculate the moving average, also referred to as smoothing of time-series. The concept of seasonal index is introduced to show the relative values of the variables in each group. To form the index, the data is divided into a set of partitions such as groups of months or groups of hours and the variation of the variable is monitored over each group to identify recurring patterns. However without any predefined knowledge, the above grouping is very arbitrary. An interesting solution would be to identify the groups from the data and look for patterns.

V. MINING SEQUENTIAL PATTERNS

The sequential associations or sequential patterns can be represented as follows:

When A occurs, B also occurs within a certain time. The difference between traditional association rules mining and sequence mining is that the time information is included both in the rule and also in the mining process in the form of constraints. In general three attributes characterize the sequence data: object, timestamp, and event. Hence, the corresponding input records consist of occurrences of events on an object at a particular time. Depending on the data and the problem in hand, various definitions of the objects and events can be used. As an example, an object can be a customer in a book store and events are the books bought by the customer. As another example, an object can be a day and the events a switch- alarm pair of telecommunications network.

The major task associated with this kind of data is to discover sequential relationships or patterns present in these data. This can be very useful for prediction of future events. Several approaches have been proposed to tackle the problem. However the problems they assess, and the resulting solutions are very much problem dependent and often not suitable for other types of sequential data.

VI. FOCUS OF THIS PAPER

The predominant problem domain for this work is a traffic control system where discovery of patterns and their automation is of prime interest. However, the solution proposed by this concept is not restricted to traffic control system alone but can be used for other domains where the need is to extract useful segments from the data based on user specified parameters. Some of the characteristics of a traffic control system.. to accomplish this the traffic control system reacts to the changes in inhabitant's behavior by automating the operations of traffic light system instead of waiting for the inhabitants to manually interact with them. Following are some of the questions for which the answer is needed to accomplish the goals of a traffic control system.

- When does A Traffic Light turn on?
- When does A Traffic Light turn on for particular days or between particular time periods.
- When does A Traffic Light turn on every day for long period for particular route?
- When does A Traffic Light turn on Sunday for long period for particular route?
- Of these, which are the most frequently occurring patterns?
- What are the times during which the patterns occur?
- How many times patterns occur during a given time interval.

This work proposes to answer most if not all of the above questions raised by traffic control system. In addition to intelligent environments, many applications such as telephone logs, security logs and other time or numerical applications want to know, 'Illustrate intervals in terms of groups of time or activity which best represents the data' or 'Illustrate intervals in terms of groups of time or activity which have the following characteristics'. With telephone logs, periods of high activity are useful information for

making informed business decisions. Magazine subscription logs can also be mined to determine the age groups that subscribe the most to the magazine. Security logs can also be mined to extract intervals with certain characteristics. These intervals can be compared to values associated with normal conditions, to raise alerts when abnormal conditions are discovered.

The characteristics of an interval can be its density, length or the strength. Given the above general problem domain, we divide our task into two phases: Identify the intervals which best represent based on interval characteristics provided by the user and use the intervals to identify frequently occurring patterns of different sizes and strengths

VII. RELATED WORK

The following sections provide a survey of the existing algorithms. WINEPI [3], MINEPI [3], GSP [4] are described in detail in sections. Provides a brief overview of several other algorithms, which are related to the current domain. Finally, we provide a brief introduction to our proposed solution in section

A lot of work has also been done on prediction, from Markov's mth order model to using statistical techniques in time series analysis. Markov's model [5] predicts which event will occur next, or when an event occurs using probabilities. This model is primarily used for pre- fetching of pages in computer architecture and other applications (e.g., speech recognition) from an input sequence; the next event is predicted using probability distribution functions

WinEpi [3] is an algorithm, designed for discovering serial, parallel or composite sequences. Serial sequences require a temporal order of events whereas parallel sequences do not. Composite sequences are generated from the combination of parallel and serial sequences. In addition to the above, events of the sequences must be close to each other, which is determined by the window parameter. A time window is slid over the input data and only the sequences within the window are considered. The support for the sequence is determined by counting the number of windows in which it occurred. Referring to the timing constraints described above, the algorithm finds all sequences that satisfy the time constraints ms and whose support exceeds a user-defined

Minimum min_sup, counted with the CWIN method. The algorithm makes multiple passes over the data. The first pass determines the support for all individual events. In other words, for each event the number of windows containing the event is counted. Each subsequent pass k starts with generating the k- event long candidate sequences Ck from the set of frequent sequences of length k- 1 found in the previous pass. This approach is based on the subset property of apriori principle that states that a sequence cannot be frequent unless its subsequences are also frequent. The algorithm terminates when no frequent sequences are generated at the end of the pass. WinEpi uses set of counters and sequence length for support counting of parallel sequences and finite state automata for serial.

MinEpi uses the same algorithm for candidate generation as WinEpi with a different support counting technique. In the first round of the main algorithm mo(s) is computed for all sequences of length one. In the subsequent rounds the minimal occurrences of s are located by first selecting its two suitable subsequences s1 and s2 and then performing a temporal join on their minimal occurrences. Frequent rules and patterns can be enumerated by looking at all the frequent sequences and then its subsequences

The GSP (Generalized Sequential Patterns) by [4] is designed for transactional data where each sequence is a list of transactions ordered by transaction time and each transaction is a set of items. It extends their previous work [9] by enabling specification of the maximum time difference between the earliest and latest event in an element as well as the minimum and maximum gaps between adjacent elements of the sequential patterns. Thus the timing constraints included are ws, xg and ng. Support is counted using COBJ method. The algorithm works the same way as WinEpi described in the previous section. The difference is in the way the candidates are generated and their support counted. GSP introduces the notion of contiguous subsequences. The sequence

c is a subsequence of s if any of the following holds:

- c is derived from s by dropping an event from its first or last event- set.
- c is derived from s by dropping an event from any of its event- sets that
- Have at least 2 elements
- c is a contiguous subsequence of c', which is a contiguous subsequence Of s.

The determination of the support of the candidates is done by reading one data sequence at a time and incrementing the support count of the candidates contained in the data sequence. Given a set of candidate sequences C and a data sequence d, all sequences in C that are subsequences of d are found. Our domain considers data to be a series of events with timestamps with frequent patterns discovered between various events, in contrast to GSP, which discovers sequential relationships between items within a set of transactions

CSpade [10] has the same application domain as GSP but involves more constraints that are versatile. CSpade is an extension of the earlier Spade [11] algorithm, which efficiently integrates constraint into the algorithm. The key

features of Spade are the use of vertical layout and idealists, which include the object timestamp tupples of the events. Equivalence classes partition the data set into several classes, which are processed independently. Problem decomposition using equivalence classes is decoupled from pattern search. Depth- first search is used for enumerating the frequent subsequences within each equivalence class. Our approach also considers a vertical database layout similar to that of Spade, partitions the database on the number of events and identifies intervals of occurrences based on user specified 'measure' independently.

Cyclic association rules [12] attempt to find rules, which are very prominent in a segment of data but are lost when the entire dataset is considered for mining. Partitioning the data correctly plays a crucial role in the discovery of these hidden rules. In addition to the mining techniques, many mathematical and statistical models [6, 7] also attempt to predict or discover the intervals by formulating an equation, which best describes the data. However these models have the drawback that they predict one answer based on historical data. One answer may not be adequate in several situations. In order to get multiple answers the data needs to be partitioned thereby predicting the best answer for each partition. This however would introduce some arbitrariness in the choice of best partition in the absence of appropriate guidelines. [13] uses data cubes and Apriori mining techniques for mining segment- wise periodicity with respect to a fixed length period. In [14] MDL (minimum description length) principle, instead of support, is used to find candidate item-sets. The merit of this approach lies in the application of the periodicity of the event to prune unwanted sequences. This approach has some similarity to the first approach of [3] in the use of a sliding window defined by the user to find frequent episodes. Defining the periodicity, however, can be an error prone task. As for [13], the algorithm discovers rules based on different measures for each time partition. Our primary interest is to find partitions which best describe the nature of the data.

One of the distinct disadvantages of using traditional k-means [15, 16] or density based clustering algorithms [17, 18] is the determination of input parameters such as k or threshold density. Determination of the values of these input parameters either requires proficient domain knowledge or sufficient time for re- running algorithms with different inputs. Even though the primary aim of the present study is not cluster identification, to decide a better value for k and the threshold density, the number of clusters identified at the end of interval discovery algorithm along with their density and length can be used as input to the traditional clustering algorithms. Salient points of the work and discussing additional work that can be performed to improve its utility, efficiency and scalability.

VIII. INTERVAL DISCOVERY

Here discuss our proposed solution with Not many data mining algorithms discuss the formation of intervals on time series data based on the interaction of events. The data collected from traffic control exhibits the interactions between the inhabitant. This results in large amount of information stored over a period of time for each route, with data value at every point in the time scale. The primary aim is to coalesce the points and convert them to intervals. Start and end times associated with an event signifies the occurrence of the event within it with certain characteristics of the interval such as its strength, length and density. With large numerical and time series data, events occur with a high degree of certainty not at specific points but within tight intervals (sets of points). Therefore intervals give us more information on the total strength of the device activity during a period as compared to points. Based on this observation, the data related to each route is mined separately to identify the intervals with maximum strength. Traffic system can greatly benefit from an algorithm that can infer the usage patterns of each route as well as interactions between different routs. This system consists of numerous sensors (or manually) deployed around the squire of road that monitor different arrivals of vehicle

Every change large change is recorded in the database. Abstractly such data can be viewed as a collection of events, where each event has an associated time of occurrence Multiple events can occur at the same time, which means different events can have the same timestamp. Discovery of the frequent sequences and automation of the traffic using discovered sequences could reduce the interaction between the inhabitant and the traffic. The crux of the study is to find, when each route load is increase and decrease to determine the interaction between the routes load (such as the causality of their usage), using the intervals. This gives the answer to the exact time of occurrences of each route/event as well as of the frequent patterns (sets of devices).

Even though interval discovery can be used with various applications the traffic control system scenario is used this type of table

Table Sample of traffic control Input Data for a particular route R1.

Status Time (per 10 minutes)		Support
R1	08:00 Am	10
R1	09:00 Am	14
R1	10:00 Am	25
R1	11:00 Am	45
R1	12:00 Am	40
R1	01:00 Pm	35
R1	02:00 Pm	30
R1	03:00 Pm	30
R1	04:00 Pm	30
R1	05:00 Pm	45
R1	06:00 Pm	50
R1	07:00Pm	35
R1	08:00 Pm	20

The significant interval discovery algorithm proposed in this study can be partitioned into 3 phases:

- Preprocessing (one time processing)
- Interval Formation (Iterative process)
- Cluster Formation (one time processing)

The interval discovery algorithm accepts a number of parameters from the user (from a configuration file) to compute the set of significant intervals and clusters. The input parameters accepted by the algorithm are:

- Minimum Strength
- Window
- Measure
- Measure Value
- Period
- Interval Semantics
- Sequential Window
- Number of Threads

Minimum Strength and Window are parameters used in the preprocessing phase to prevent the formation of certain first level intervals from the point- based data. They are domain specific and optional, they make the process more efficient and accurate when provided. Minimum Strength ensures that only intervals with strength greater than specified value (threshold) form an interval.

IX. CONCLUSION

It is evident that number of algorithms have been proposed for solving the problem of frequent pattern discovery. Approaches that work for one domain do not necessarily form the best solution for another. The focus of our approach lies between clustering and sequential mining since both kinds of information are required to discover frequent patterns and answer queries related to intelligent environments. As mentioned above, our approach has enormous potential in intelligent environments where the key is to continuously learn from the surroundings and automate the inhabitant's activities. The traffic control system (Managing An Intelligent and Versatile light system) patterns enables us to automate device usage and reduce human interaction. For finding patterns, the algorithm uses the intervals derived from various routes based on a user- defined confidence, density or interval length to predict the time of operation of each route. This information is used to answer user queries as well as to find sequential patterns. Representative intervals can be classified as the smallest intervals with highest density satisfying the desired interval-confidence.

X. REFERENCES

[1] Thuraisingham, B., A Primer for Understanding and Applying Data Mining. IEEE, 2000. Vol. 2, No.1. p. 28- 31.

[2] Thomas, S., Architectures and optimizations for integrating Data Mining algorithms with Database Systems, in CSE. 1998, University of Florida: Gainesville.

[3] Mannila, H., H. Toivonen, and I. Verkamo. Discovering Frequent Episodes in Sequences. in Proc of the 1st Intl. Conference on Knowledge Discovery and Data Mining. 1995. Montreal, Canada.

[4] Srikant, R. and R. Agrawal. Mining Sequential Patterns:Generalizations and Performance Improvements. in In 5th Intl. Conf. Extending DatabaseTechnology. 1995. Avignon, France: IBM.

[5] Bell, T.C., J.C. Cleary, and I.H. Witten, eds. Text Compression. Advance" Reference. 1990, Prentice Hall.

[6] Bruce L. Bowerman, R.T.O.C., Time Series Forecasting. Second Edition ed. 1990: PWS Publishers. 25- 120.

[7] Gilchrist, W., Statistical Forecasting. 1976: John Wiley and Sons. 115-148,77-90.

[8] Joshi, M., G. Karypis, and V. Kumar, A Universal Formulation of Sequential Patterns. 1999, Department of Computer Science, University of Minnesota: Minnesota. p. 20.

[9] Agarawal, R. and R. Srikant. Mining Sequential Patterns. in Proc. of 11th Int'l Conference on Data Engineering. 1995. Taipei, Taiwan.

[10] Zaki, M.J., SPADE:An Efficient Algorithm for Mining Frequent Sequences. Machine Learning

Journal,special issue on Unsupervised Learning (Doug Fisher; ed.), 2001. 42 No 1/2: p. 31-60.

[11] Zaki, M.J. Sequence Mining in Categorical Domains: Incorporating Constraints. in 9th Int'l Conference on Information and Knowledge Management. 2000. Washington DC.

[12] B. Ozden, S. Ramaswamy, and A. Silberschatz. Cyclic Association Rules. in Proceedings of the IEEE International Conference on Data Engineering. 1998. Orlando, FL.

[13] Han, J., W. Gong, and Y. Yin. Segment-Wise Periodic Patterns in Time Related Database. in Proc 1998 Int'l Conference on Knowledge Discovery and Data Mining. 1998. New York City: AAAI Press.

[14] Das, S.K., et al., The Role of Prediction Algorithms in the MavHome Smart Home Architecture, in IEEE Wireless Communications Communications Special Issue on Smart Homes. 2002. p. 77- 84.

[15] Kanungo, T., et al., An Efficient k-Means Clustering Algorithm: Analysis and Implementation, in IEEE TRANSACTIONS ON PATTERN ANALYSIS ANDMACHINE INTELLIGENCE. 2002. p. 881-892.

[16] Han, R.N.a.J., Clarans: A method for clustering objects for spatial data mining, in IEEE Transactions on Knowledge and Data Engineering. 2002. p. 1003--1016.

[17] A. Hinneburg and D. A. Keim. An Efficient Approach to Clustering in Large Multimedia Databases with Noise. in Proc. 4rd Int. Conf. on Knowledge Discovery and Data Mining. 1998. New York.

[18] Ester, M., et al. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. in In Proceeding of the 2nd International Conference on Knowledge Discovery and Data Mining. 1996. Portland, OR.

[19] D. J. Cook, M.Y., E. Heierman, K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja. MavHome: An Agent-Based Smart Home. in to appear in Proceedings of the Conference on Pervasive Computing. 2003.

[20] Anwar, E., L. Maugis, and S. Chakravarthy, A New Perspective on Rule Support for Object-Oriented Databases, in 1993 ACM SIGMOD Conf. on Management of Data. 1993. Washington D.C. p. 99-108.

[21] Chakravarthy, S., et al. ECA Rule Integration into an OODBMS: Architecture and Implementation. in ICDE. 1995.