Volume 2, Issue 4, April 2012 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Finding Maximal Periodic Patterns and Pruning Strategy in Spatiotemporal Databases

O.Obulesu
Research Scholar in CSE
JNTUA, Anantapur, A.P., India
obulesh194@gmail.com

Dr.A.Rama Mohan Reddy
Professor of CSE, SVUCE
Tirupati, A.P., India

K.Suresh
Asst.Professor of IT
AITS, Rajampet, A.P.

Abstract-**In many applications that track and analyze spatiotemporal data, movements obey periodic patterns; the objects follow the same routes (approximately) over regular time intervals. Periodic Pattern Mining or Periodicity Detection has numerous applications such as Prediction, Forecasting, Detection of Unusual events, etc. The periodic patterns are detected in a Time-Series database depending on the time intervals. Existing approaches could not detect all types of periodicity such as Symbol, Sequence and Segment at a time. In this Paper, we propose an approach to detect all types of the periodicity in time series Databases. It also finds the periodicity in the subsections of the Time-Series very effectively. Actually the periodicity detection results in the redundant data. To remove redundant data there are pruning techniques to apply and to get the desired pattern as an output. The comprehensive study demonstrates the effectiveness of the proposed approach. This is very time efficient, accurate approach than many existing approaches.**

Keywords—**Periodicity Detection, Symbol Periodicity, Sequence Periodicity, Segment Periodicity, Timeseries Database.**

I. INTRODUCTION

Data mining is the process of extracting patterns from data. It is the process of applying the methods like neural networks, clustering, genetic algorithms (1950s), decision trees (1960s) and support vector machines (1980s) to data with the intention of uncovering hidden patterns. It has been used for many years by businesses, scientists and governments to transfer large volumes of data such as Airline Passenger Trip Records, Census data and Supermarket Scanner data to produce market research reports. Data mining, "The extraction of hidden predictive information from large databases", is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions. Data mining tools can answer business questions that traditional tools are too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations.

There are various Steps those involved in Knowledge discovery process given below:

1. *Data Cleaning:* The data collected are not clean and may contain errors, missing values, noisy or inconsistent data, so that different techniques needed to get error-free data before applying Mining method such as Clustering/Classification or Prediction.
2. *Data Integration:* Data are collected and integrated from all the different sources.
3. *Data Selection:* In this step, select only those data useful for data mining.
4. *Data Transformation:* The data even after cleaning are not ready for mining so transform them into forms appropriate for mining. The techniques used to accomplish this are smoothing, aggregation, normalization.
5. *Data Mining:* In this step, apply data mining techniques on the data to discover the interesting patterns. Various techniques like clustering and association analysis are used for data mining.
6. *Pattern Evaluation and Knowledge Presentation:* This step involves visualization, transformation, removing redundant patterns from the generated patterns .
7. *Decisions / Use of Discovered Knowledge:* This step helps user to make use of the knowledge acquired to take better decisions.

A. *TimeSeries Database:*

A Time-Series Database is a collection of data values gathered generally at uniform interval of time to reflect certain behavior of an entity. In Real World, There are several examples of Time-Series such as Weather Conditions of a particular location, Spending Patterns, Stock Growth, Transactions in a Supermarket, Network Delays, Power Consumption, Computer Network Fault Analysis and Security Breach detection, Earthquake Prediction. The periodicity detection is a process of finding temporal regularities within the Time-Series and the goal of analyzing a Time-Series Database is to find how frequent a periodic pattern (full or partial) is repeated within time intervals. In general, there are three types of

periodic patterns can be detected in a time series Database. They are mentioned given below:
1) Symbol Periodicity
2) Sequence Periodicity or Partial Periodic Patterns
3) Segment or Full-Cycle Periodicity

B. Symbol Periodicity

A Time-Series is said to be a Symbol periodicity, if at least one *symbol* is repeated periodically. For example, in a Time-Series, let *T* = *abdacbabdabc*, symbol *a* is periodic within periodicity p = 3, starting at position zero (StPos=0).

C. Sequence Periodicity

A Time-Series is said to be a Sequence Periodicity, if more than one symbol may be periodic and it is also called as partial periodic patterns. For instance, in a Time-Series Database let *T* = *bbaaabbdabcaabbcabcd* then the sequence *ab* is periodic within periodicity p = 4 starting at position 4 (StPos = 4).

D. Segment Periodicity

A Time-Series is said to be a Segment Periodicity, if the whole Time-Series can be mostly represented as a repetition of a pattern or segment and it is also known as full-cycle periodicity. For instance, in a Time-Series Database let *T*=*abcababcababcab* has Segment Periodicity of 5 (p = 5) starting at the first position (stPos = 0), i.e., T consists of only three occurrences of the segment abcab.

E. Subsection

A Time-Series T possesses symbol, sequence, or segment periodicity within period p between positions StPos and EndPos (Where 0<=StPos<EndPos<=|T|), if the investigated period satisfies the Definition 3, Definition 4, or Definition 5 respectively by considering only subsection [StPos, EndPos] of T. For instance, StPos=5, EndPos=12 represents a period in T = bbcdbababababccdaccab; Here, it is worth noting because the analysis considers only subsection of the series, from positions 5 to 12.

F. Pruning Strategy

For a given Time-Series there may exist repeated patterns and thus leads to the redundant data. By applying the pruning strategy the redundant data is eliminated and this returns the first occurrences of all the possible patterns.

II. DATABASE TRANSFORMATION

The database consists of a single column table, which stores the Time-Series events on which Pattern Matching, Detecting Periodicity Types, Pruning Strategy can be performed. The Time-Series should be specified first and then table name should be specified next in which that has to be stored.

The Database will be updated and verified by using simple SQL Query Statements like UPDATE, INSERT,DELETE and SELECT Statements(DML Commands).

III. SUBSEQUENCE TRANSFORMATION

Based on the given Time-Series Database, the subsequences will be generated. For instance, aTime-Series T=abac then the subsequences generated are a,ab,aba,abac,b,ba,bac,a,ac,c. After the generation of subsequences these are categorized by the periodicity types like Symbol, Sequence and Segment Periodicity.

IV. PERIODICITY TYPES

In a Time-Series Database, let T=abac the subsequences are categorized as

Symbol: a
Sequence: ab
Sequence: aba
Segment: abac
Symbol: b
Sequence: ba
Sequence: bac
Symbol: a
Sequence: ac
Symbol: c

A. Positions of subsets of patterns

A Time-Series Database include many operations which contain same symbols at multiple positions, so that the main task is to find the positions of the patterns which are generated as subsequences.

Finding Positions: Let, a Time-Series Database has T=abad then

The position of *a* in the timeseries *abad* is 0,2
The position of *ab* in the timeseries *abad* is 0
The position of *aba* in the timeseries *abad* is 0
The position of *abad* in the timeseries *abad* is 0
The position of *b* in the timeseries *abad* is 1
The position of *ba* in the timeseries *abad* is 1
The position of *bad* in the timeseries *abad* is 1
The position of *a* in the timeseries *abad* is 0,2
The position of *ad* in the timeseries *abad* is 2
The position of *d* in the timeseries *abad* is 3

V. SUBSECTION PATTERN MATCHING

In the previous section the pattern matching has been performed to the entire Time-Series. In the same way the pattern matching can be performed in the subsection also. A Subsection is a part of the Time-Series. For the subsection pattern matching the StPos and EndPos should be specified along with the timeseries as mentioned below.

Let, T=abacdaf and StPos=2 and EndPos=5

When these parameters are specified automatically the Database will be updated with the the subsection of the Time-Series and the periodicity types are also synchronized to the subsection as well. The positions of the patterns are found in the entire Time-Series, in the same way the positions are found for the subsection also.

VI. PRUNING STRATEGY

The pruning strategy defines the elimination of redundant data and obsolete data. A Time-Series Database may contain redundant data or repeated data which increase the space or memory and the time to search the data will be more and Complex.

A. Pruning Technique in the full time series:

Let, T=abac then the subsequences generated are:

Symbol: a
Sequence: ab
Sequence: aba
Segment: abac
Symbol: b
Sequence: ba
Sequence: bac
Symbol: a
Sequence: ac
Symbol: c

The starting index of *a* is 0
The starting index of the string *ab* is 0
The starting index of the string *aba* is 0
The starting index of the string *abac* is 0
The starting index of the string *b* is 1
The starting index of the string *ba* is 1
The starting index of the string *bac* is 1
The starting index of the string *a* is 0
The starting index of the string *ac* is 2
The starting index of the string *c* is 2

In the above example ,the symbol "a" is repeated at index 2. Since then the index of a is given as 0.

B. Pruning Technique in the subsection time series:

The pruning technique can also be applied in the subsection of a Time-Series Database. Here we need to specify the StPos, EndPos along with the Time-Series values.

Let, T=abada and StPos=2 and EndPos=4

The results of Pruning technique after specifying these parameters are given below:

Let, T=ada then

The starting index of *a* is 0
The starting index of *ad* is 0
The starting index of *ada* is 0
The starting index of *d* is 1
The starting index of *da* is 1
The starting index of *a* is 0

VII. PATTERN COLLECTION

The complete set of patterns is the union of all the subsets of patterns detected above.

A. Advantages of Proposed System:

- Faster
- Efficient
- Less levels of recursion
- Less memory
- Periodicity types can be found at a time

VIII. CONCLUSION

Several important Time-Series data mining problems reduce to the core task of finding approximately repeated subsequences in a longer time series. Mining frequent patterns in Transactional Databases, Time-Series Databases and many other kinds of databases has been studied popularly in data mining research. In this paper, we have presented the pattern matching technique to find the patterns that were repeated in a Time-Series Database. The three types of patterns such as Symbol, Sequence and Segment Periodicity are also discovered. We have also presented how to reduce the redundant data there by reducing the memory usage and complexities through the pruning strategy. These sequences can be applicable anywhere in any field according to the requirements of the user, particularly in Earthquake Prediction, Weather Forecasting and Fraud detection Applications. Further improvements can be done in the areas of Biological and DNA Sequences and various fields of Science.

REFERENCES

[1] Faraz Rasheed, Mohammed Alshalalfa and Reda Alhajj, "Efficient Periodicity Mining in Time Series Databases Using Suffix Trees", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 1, JANUARY 2010.

[2] C. Berberidis, W. Aref, M. Atallah, I. Vlahavas, and A. Elmagarmid, "Multiple and Partial Periodicity Mining in TimeSeries Databases," Proc. European Conf. Artificial Intelligence, July 2002.

[3] M.G. Elfeky, W.G. Aref, and A.K. Elmagarmid, "Periodicity Detection in Time Series Databases," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 7, pp. 875-887, July 2005.

[4] J. Han, Y. Yin, and G. Dong, "Efficient Mining of Partial Periodic Patterns in Time Series Database," Proc. 15th IEEE Int'l Conf. Data Eng., p. 106, 1999.

[5] S. Ma and J. Hellerstein, "Mining Partially Periodic Event Patterns with Unknown Periods," Proc. 17th IEEE Int'l Conf. Data Eng., Apr. 2001.

[6] F. Rasheed and R. Alhajj, "STNR: A Suffix Tree Based Noise Resilient Algorithm for Periodicity Detection in Time Series Databases," Applied Intelligence, vol. 32, no. 3, pp. 267-278, 2010.

[7] F. Rasheed and R. Alhajj, "Using Suffix Trees for Periodicity Detection in Time Series Databases," Proc. IEEE Int'l Conf. Intelligent Systems, Sept. 2008.

[8] C. Sheng, W. Hsu, and M.-L. Lee, "Mining Dense Periodic Patterns in Time Series Data," Proc. 22nd IEEE Int'l Conf. Data Eng., p. 115, 2005.

[9] R. Kolpakov and G. Kucherov, "Finding Maximal Repetitions in a Word in Linear Time," Proc. Ann. Symp. Foundations of Computer Science, pp. 596-604, 1999.

[10] Jiawei Han, Micheline Kamber and Jian Pei, "Data Mining Concepts and Techniques", 2nd Edition, M.K Publishers.

[11] Ian H. Witten, Eibe Frank and Mark A. Hall, "Data Mining: Practical Machine Learning Tools and Techniques", 2nd Edition, M.K Publishers.

[12] Data Mining Introductory and Advanced Topics, Margaret Dunham.

[13] M.G. Elfeky, W.G. Aref, and A.K. Elmagarmid, "Periodicity Detection in Time Series Databases," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 7, pp. 875-887, July 2005.

[14] J. Han, Y. Yin, and G. Dong, "Efficient Mining of Partial Periodic Patterns in Time Series Database," Proc. 15th IEEE Int'l Conf. Data Eng., p. 106, 1999.

[15] R. Agrawal and R. Srikant, "Mining Sequential Patterns," Proc.Int'l Conf. Data Eng. (ICDE'95), pp. 3-14, 1995.

[16] Z. Zhang, Y. Wang, and M. Kitsuregawa, "Effective Sequential Pattern Mining Algorithms for Dense Database," Proc. Japanese Nat'l DatavEng. Workshop (DEWS 06), 2006.

[17] J. Chen and T. Cook, "Mining Contiguous Sequential Patterns from Web Logs," Proc. World Wide Web Conf. (WWW '07) Poster Session, May 2007.

[18] J.Han, G.Dong and Y.Yin, " Efficient Mining of Partial Periodic Patterns in Time-Series Database".

[19] Zhenhui Li, J.Han, Ming Ji, Lu-An Tang, Y. Yu, Bolin Ding,R.Kays and J.Lee, "MoveMine:Mining Moving Object Data for Discovery of Animal Movement Patterns:, ACM Journal Name,Vol.,No., 05 2010, Pages 111–077.

[20] http://www.programmingsimplified.com/java-source-codes.

[21] http://docs.oracle.com/javase/tutorial/java/data/manipstrings.html

[22] http://www.java-examples.com/