IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

SIGNATURE BASED INDEXING METHOD FOR EFFICIENT CONTENT BASED RETRIVAL OF RELATIVE TEMPORAL PATTERNS

[1]M Gayathri, [2]Dr N Syed Siraj Ahmed

[1]PG Scholar, [2]Assistant Professor

[1]Master of Computer Applications

[1]Madanapalle Institute of Technology and Science, Madanapalle, India

Abstract: A number of algorithms have been proposed for the discovery of data from the large database. However, since the number of generated patterns can be large, selecting which patterns to analyse can be nontrivial. There is thus a need for algorithms and tools that can assist in the selection of discovered patterns so that subsequent analysis can be performed in an efficient and, ideally, interactive manner. In this project, we propose a signature-based indexing method to optimize the storage and retrieval of a relative data from the large database

Index Terms Data Mining, OODB, Index, Retrieval

I. INTRODUCTION

Many rule discovery algorithms in data mining generate a large number of patterns/rules, sometimes even exceeding the size of the underlying database, with only a small fraction being of interest to the user It is generally understood that interpreting the discovered patterns/rules to gain insight into the domain is an important phase in the knowledge discovery process. However, when there are a large number of generated rules, identifying and analysing those that are interesting becomes difficult. For example, providing the user with a list of association rules ranked by their confidence and support might not be a good way of organizing the set of rules as this method would overwhelm the user and not all rules with high confidence and support are necessarily interesting for a variety of reasons.

Therefore, to be useful, a data mining system must manage the generated rules by offering flexible tools for rule selection. In the case of association rule mining, several approaches for the post processing of discovered association rules have been discussed. One approach is to group “similar” rules which works well for a moderate number of rules. However, for a larger number of rules it produces too many clusters. A more flexible approach is to allow the identification of rules that are of special importance to the user through templates or data mining queries. This approach can complement the rule grouping approach and has been used to specify interesting and uninteresting classes of rules (for both association and episodic rules). The importance of data mining queries has been highlighted by the introduction of the inductive database concept, which allows the user to both query the data and query patterns, rules, and models extracted from these data.

II. Literature survey:

Temporal Patterns

The temporal patterns described in this paper consist of two components: a set of states and a set of relationships between those states that represent the order of states within the pattern. In order to retrieve such patterns efficiently, any indexing method should deal with both temporal concepts—states and state relationships. The problem of indexing has been studied in depth in the database literature, (for example, B+ trees, R trees etc.). However, studies on set-based indexes that support queries on set-valued attributes (i.e., attributes that are sets of items) are limited. Ishikawa et al. apply the signature file technique to support the processing of queries involving set-valued attributes in OODBs. Two signature file organizations are considered: the sequential signature file and the bit-slice file. The bit-slice approach still needs to examine every signature in the file but only a part of it. In order to avoid reading every signature in the signature file, the hierarchical file organization uses several levels of signatures. The higher levels perform coarse filtering before the signatures on the lower levels are consulted. Examples of the hierarchical file organization include the S-tree and the SG-tree. The partitioned file organization approach avoids reading every signature by grouping the signatures into several partitions such that all signatures in a given partition possess the same component part, called the signature key. The signature key used is usually a substring of the signature. By partitioning the signatures, some of the partitions need not to be searched during the execution of a query so that the number of accesses can be reduced. Helmer and Marcotte study the performance of four index structures for set-valued attributes (sequential signature files, signature trees, extensible signature hashing, and inverted lists). The indexes are evaluated on three forms of set-valued queries—equality queries, subset queries, and superset queries. It was observed that the inverted file index structure outperformed other index structures for subset and superset queries with respect to query processing time. Morzy and Markiewicz generalize the problem of association rule and item set retrieval as a subset search problem. Two types of queries are examined: first, the retrieval of item sets that contain a given subset of items and, second, the retrieval of rules that contain a given subset of items in their antecedent or consequent. In order to speed up the query processing, a group bitmap index is proposed in which the group bitmap key represents a set of items in the database. These set-based indexing methods do not consider the order of items within the sets, as is required in the case of the indexing and retrieval of sequential patterns. To overcome this limitation, new indexing techniques have been proposed the general idea of which is to convert the sequential patterns into equivalent sets that accommodate the ordering of the items. After that, set based indexing methods can be applied to the equivalent sets. A partitioning technique is proposed to divide large equivalent sets into a collection of smaller subsets so that the probability of collision is reduced.

Proposed System

- Efficiently retrieving subsets of a large collection of previously discovered temporal patterns.
- Focuses on supporting content-based queries of temporal patterns, as opposed to point- or range-based queries.

Modules:

- Data collection from the user end.
- Constructing raw database into Signature Files
- Query process
- Answering the query from database

Module Description:

Module 1:

In this module we are getting the data from the user end which is maintained as a raw database for our project.

Module 2:

1. Converting the raw database into equivalent id-based file.
2. Id based database is converted into equivalent signature file.

Module 3:

In this module the query is converted into equivalent signature form i.e. into bit form In this module the (content-based queries), Let D be a temporal pattern database and q be a query pattern. The four forms of content-based queries that this research supports include the following:

1. Sub pattern queries. Find those patterns in D that contain q.
2. Super pattern queries. Find those patterns in D that are a sub-pattern of q.
3. Equality queries. Find those patterns in D equal to q.
4. K-nearest sub pattern queries. Find the k most similar patterns in D to q.

Methodology

The source of information for developing the proposed system is gathered directly from clients of end user who is going to use the package becomes the primary source to give information.

Here we used the "Linear sequential Model" for developing this module because it is static and all the requirements are defined at beginning of the project.

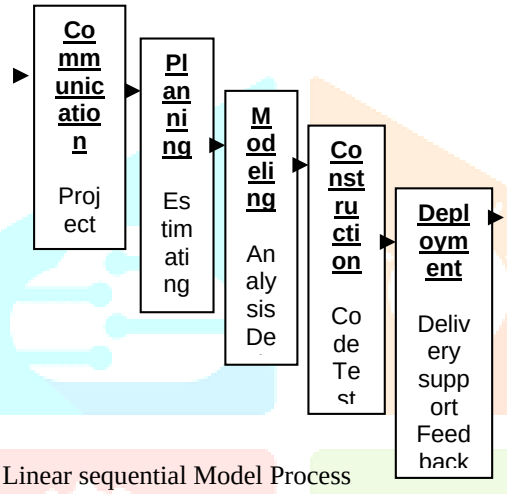

Fig 1: Linear sequential Model Process

1. Real Projects rarely follow the sequential flow that the model proposes. Although, the linear model can accommodate iteration, it does so indirectly. As a result, changes can cause confusion as the project team proceeds.
2. It is often difficult for the customer to state all requirements explicitly .The Waterfall model requires this and has difficult accommodating the natural uncertainty that exists at the beginning of many projects.
3. The customer must have patience. A working version of the program will not be until late in the project time-span. A major blunder, if Undetected until the working program is reviewed, can be disastrous.

Implementation

Implementation is the process of converting a new or revised system design into an operational one. Apart from planning, the major tasks of preparing for implementation or education and training of users. Implementation includes following activities:

- Obtaining and installing the system hardware
- Providing user access to the system
- Creating and updating the database
- Training the users on the new system
- Documenting the system for its users
- Evaluating the operation and use of the system

Implementation Methods

There are four basic methods of implementation:

- Review Reports
- Analyse Reports
- Analyse Timing
- Implementation results

Parallel Conversion:

Description:

In this method, the old system is operated along with the new system.

The Present system has been done by using Parallel Conversion in which the old one is replaced with the newly developed system. In this method, we can run both the systems and we can decide which method to follow based on results.

Conclusions

The use of a signature-based index for content-based retrieval of temporal patterns has been presented. The signatures of temporal patterns are created by first converting temporal patterns into equivalent sets and then generating the signatures from the equivalent sets. The study focused on the sequential and BSSF organizations, and a series of experiments compared the performance of both signature files in processing sub pattern and super pattern queries. In conclusion, the use of signature files improves the performance of temporal pattern retrieval. The bit-slice signature file performs better than the SSF and is a good choice for content-based retrieval of temporal patterns. This retrieval system is currently being combined with visualization techniques for monitoring the behaviour of a single pattern or a group of patterns over time.

References

1. E. Win Arko and J.F. Roddick, "ARMADA—An Algorithm for Discovering Richer Relative Temporal Association Rules from Interval-Based Data," Data and Knowledge Eng.,
2. D. Comer, "The Ubiquitous B-Tree," Computing Surveys
3. A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," Proc. ACM SIGMOD
4. S. Helmer and G. Marcotte, "A Performance Study of Four Index Structures for Set-Valued Attributes of Low Cardinality," VLDB J.,
5. Y. Ishikawa, H. Kitagawa, and N. Ohno, "Evaluation of Signature Files as Set Access Facilities in OODBs," Proc. ACM SIGMOD '93, P. Benemann and S. Jahoda, eds., pp.
6. T. Moray and M. Markiewicz, "Group Bitmap Index: A Structure for Association Rules Retrieval," Proc. ACM SIGKDD '98, R. Agrawal, P. Stolz, and G. Petosky-Shapiro,
7. U. Deepish, "S-Tree: A Dynamic Balanced Signature Index for Office Retrieval," Proc. ACM SIGIR
8. N. Mamou Lis, D.W. Cheung, and W. Lian, "Similarity Search in Sets and Categorical Data Using the Signature Tree," Proc. 19th Int'l Conf. Data Eng. (ICDE '03), U. Da
9. D. Comer, "The Ubiquitous B-Tree," Computing Surveys, vol. 11, no. 2, pp. 121-137, 1979.
10. A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," Proc. ACM SIGMOD '84, pp. 47-57, 1984.
11. S. Helmer and G. Marcotte, "A Performance Study of Four Index Structures for Set-Valued Attributes of Low Cardinality," VLDB J., vol. 12, no. 3, pp. 244-261, 2003.
12. M. Markiewicz, "Sequential Index Structure for Content-Based Retrieval," Proc. Fifth Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD '01), pp. 306-311, 2001.
13. J. Allen, "Maintaining Knowledge about Temporal Intervals," Comm. ACM, vol. 26, no. 11, pp. 832-843, 1983.
14. J. Xiao, Y. Zhang, X. Jia, and T. Li, "Measuring Similarity of Interests for Clustering Web-Users," Proc. 12th Australasian Database Conf. (ADC '01), M. Orlowski and J. Roddick, eds., pp. 107-114, 2001.
15. C. Fallouts and S. Hristopoulos, "Signature Files: An Access Method for Documents and Its Analytical Performance Evaluation," ACM Trans. Office Information Systems, vol. 2, no. 4, pp. 267- 288, 1984.
16. Y. Chen, "On the General Signature Trees," Proc. 16th Int'l Conf. Database and Expert Systems Applications (DEXA '05), pp. 207-219, 2005.
17. H. Kitagawa, Y. Fukushima, Y. Ishikawa, and N. Ohno, "Estimation of False Drops in Set-Valued Object Retrieval with Signature Files," Proc. Fourth Int'l Conf. Foundations of Data Organization and Algorithms (FODO '93), pp. 146-163, 1993.

18. Y. Chen, “Building Signature Trees into OODBs,” J. Information Science and Eng., vol. 20, no. 2, pp. 275-304, 2004.

19. J. Yang and M. Hu, “Trumpeter: Mining Sequential Patterns from Imprecise Trajectories of Mobile Objects,” Proc. 10th Int’l Conf. Extending Database Technology (EDBT ’06), pp. 664-681, 2006.

20 . T. Morty and M. Markiewicz, “Group Bitmap Index: A Structure for Association Rules Retrieval,” Proc. ACM SIGKDD ’98, R. Agrawal, P. Stolz, and G. Petosky-Shapiro, eds., pp. 284- 288, 1998. [23] U. Deepish, “S-Tree: A Dynamic Balanced Signature Index for Office Retrieval,” Proc. ACM SIGIR ’86, pp. 77-87, 1986.

21 .N. Mamou Lis, D.W. Cheung, and W. Lian, “Similarity Search in Sets and Categorical Data Using the Signature Tree,” Proc. 19th Int’l Conf. Data Eng. (ICDE ’03), U. Dayak, K. Ramakrishna, and T. Vijaya Raman, eds., pp. 75-86, 2003