World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:4, No:10, 2010

Application and Limitation of Parallel Modeling in Multidimensional Sequential Pattern

Mahdi Esmaeili, Mansour Tarafdar

***Abstract*—**The goal of data mining algorithms is to discover useful information embedded in large databases. One of the most important data mining problems is discovery of frequently occurring patterns in sequential data. In a multidimensional sequence each event depends on more than one dimension. The search space is quite large and the serial algorithms are not scalable for very large datasets. To address this, it is necessary to study scalable parallel implementations of sequence mining algorithms.

In this paper, we present a model for multidimensional sequence and describe a parallel algorithm based on data parallelism. Simulation experiments show good load balancing and scalable and acceptable speedup over different processors and problem sizes and demonstrate that our approach can works efficiently in a real parallel computing environment.

***Keywords*—**Sequential Patterns, Data Mining, Parallel Algorithm, Multidimensional Sequence Data

I. Introduction

DATA mining has been defined as "The nontrivial extraction of implicit, previously unknown, and potentially useful information from data." Mining frequent patterns or itemsets is a fundamental and essential problem in many data mining applications. These applications include the discovery of association rules, strong rules, correlations, sequential rules, episodes, multidimensional patterns, and many other important discovery tasks [1].

One of the most important data mining problems is discovery of frequently occurring patterns in sequential data [2]. Sequential pattern mining is applicable in a wide range of applications since many types of data sets are in a time related format. There are many domains where sequence mining has been applied, which include analysis of telecommunication systems, discovering customer buying patterns in retail stores, analysis of Web access databases, and mining DNA sequences and gene structures.

Usually, sequence patterns are associated with different circumstances, and such circumstances form a multiple dimensional space. For example, bank customer sequences are associated with credit, branch, age, and others. It is interesting and useful to mine sequential patterns associated with multidimensional information [3]. Besides mining sequential patterns in a single dimension, mining multidimensional sequential patterns can give us more informative and useful patterns.

M. Esmaeili is with the Department of Computer science, Islamic Azad University (Kashan branch), Kashan, Iran (phone: +98-361-5550055; fax: +98-361-5550056; e-mail: M.Esmaeili@iaukashan.ac.ir).

M. Tarafdar is MS student in Islamic Azad University (Qazvin branch), Qazvin, Iran (e-mail: tarafdar.mansour@gmail.com).

Most of the proposed pattern mining algorithms are a variant of Apriori. Apriori-based algorithms show good performance with sparse datasets such as market-basket data, where the frequent patterns are very short. However, with dense data sets such as telecommunications, where there are many and long frequent patterns, performance of these algorithms degrades incredibly.

On the other side, parallel and distributed computing is expected to relieve current mining methods from the sequential bottleneck, providing the ability to scale to massive datasets, and improving the response time. Achieving good performance on today's multiprocessor systems is a non-trivial task. The main challenges include synchronization and communication minimization, work-load balancing, finding good data layout and data decomposition, and disk I/O minimization, which is especially important for data mining [4]. On the other side, the basic single dimension can not satisfy the requirement of multi-attribute analysis, which is often the case in actual system practice. To address this problem, multidimensional sequence pattern mining is developed.

Due to the huge increase in data volume and also quite large search space, efficient solutions for finding patterns in multidimensional sequence data are nowadays very important. However, the number of sequential patterns grows exponentially and the task of finding all sequential patterns requires a lot of computational resources, which make it an ideal candidate for parallel processing.

In this paper, we present a parallel algorithm follow the level-wise approach and all participating processors or workers generate candidate sequence and count their supports independently. Simulation experiments show good load balancing and scalable and acceptable speedup over different processors and problem sizes.

The rest of the paper is organized as follows: We describe the sequence mining in Section 2. Section 3 describes the problem definition. Algorithm and an experimental study are presented in Section 4,5 and we conclude in Section 6.

II. Sequence Mining

The problem of mining frequent patterns in a set of data sequences together with a few mining algorithms was first introduced in [2]. They also presented three algorithms for solving this problem. The sequence mining task is to discover a sequence of attributes, shared across time among a large number of objects in a given database. Many studies have been contributed to the efficient mining of sequential patterns

in the literature, most of which was focused on developing efficient algorithms for finding all sequential patterns such as AprioriAll [2], GSP [5], SPADE [6], PrefixSpan [7] and so on. In addition, enormous sizes of available databases and possibly large number of mined sequential patterns demand efficient and scalable parallel algorithms. While parallel association mining has attracted wide attention there has been relatively less work on parallel mining of sequential patterns [8]. Several parallel algorithms such as HPSPM [9], pSPADE [5], TPF [10] were proposed and have good performance. Moreover, recent research on sequential pattern mining has progressed closed sequential pattern mining, which can greatly reduce the number of frequent subsequences and improve the efficiency. Yet, the above work all assumes that the database is static, and a database updates requires rediscovering all the patterns by scanning the entire old and new database. Then, there is a need for efficient algorithms to update, maintain and manage the information discovered. Some incremental mining algorithms of sequential patterns were proposed and perform significantly better than the original approach of mining the whole updated database from scratch. Other work contributes on an extension of the problem of sequential pattern mining like constraint-based in sequential pattern mining, mining cyclically repeated patterns, approximate mining of consensus sequential patterns, mining top-k closed sequential patterns, mining frequent max sequential patterns, mining long sequential patterns in a noisy environment, mining hybrid sequential patterns and sequential rules, etc. Mining multidimensional sequential pattern is also one of the central topics in sequence mining as showed by the research efforts produced in recent years [3][11]-[13].

III. Problem Definition

There is a rich variety of sequence terminology but this section defines the couple of concepts according to our model. A table T is a set of tuples $<D,A_1,A_2,\ldots,A_n>$, where D is an attribute whose its domain is totally ordered. A sequence s is denoted by an ordered list $<t_1,t_2,\ldots,t_k>$, where t_i is a tuple, i.e., $D(t_1)\leq D(t_2)\leq\ldots\leq D(t_k)$ for $1\leq i\leq k$ and $D(t_i)$=value of D tuple t_i. Every tuple has n analysis attributes along with an ordered value. A set of analysis attribute values can occur at most once in a same value of D, but can occur multiple times in different values of D attribute. The number of values of D in a sequence is called the length of the sequence. If the length of s is k, then we call it a k-sequence. A sequence $S_1=<a_1a_2\ldots a_n>$ is called a subsequence of another sequence $S_2=<b_1b_2\ldots b_m>$ and S_2 a super sequence S_1, if there exist integers $1\leq j_1\leq j_2\leq\ldots\leq j_n\leq m$ such that $a_1\subseteq b_{j1}$, $a_2\subseteq b_{j2}\ldots$ $a_n\subseteq b_{jn}$.

A multidimensional sequence database is of schema $<D,A_1,\ldots,A_n, R_1,\ldots,R_m>$, where R_i are called relevant dimensions. The schema is partitioned according to relevant attribute values and support computed by number of partitions that contain sequence. As mentioned previously, every partition is similar to table T, a set of tuples $<D,A_1,\ldots,A_n>$.

Given a minimum support threshold min-support, a multidimensional sequence S is called a multidimensional sequential pattern if and only if support(S)≥min-support.

The problem of sequential pattern mining is to find the complete set of frequent sequential patterns satisfying a minimum support in the sequence database.

In this framework, $<(t_1, t_2),(t_3),(t_4, t_5, t_1)>$ is a multidimensional sequence. All tuples in a bracket are assumed to occur at the same time. Due to t_1 and t_2 have the same value of D; both of them appear in one bracket. A sequence is an ordered list of brackets in which ordered according to their associated time in a sequence. Number of brackets show length of sequence. Therefore, above sequence is a sequence with length 3. It is important to note that, each tuple can be repeated in different bracket (time), such as tuple t_1, which appear in first and third bracket. On the other side, tuple ordering in every bracket is not important.

The Proposed Method

In this section, we present a model for multidimensional sequence and describe a parallel algorithm based on data parallelism. This model originally introduced in [12]. We assume that we want to extract all multidimensional sequence patterns from a relational table T. We also assume that T contains at least one dimension whose domain is totally ordered, corresponding to the time dimension. Other dimensions divided to tree groups: analysis, relevant and, irrelevant dimensions. Irrelevant dimensions, as the name suggests, do not any role in my approach and eliminated from table T in preprocessing phase. Tuples over analysis dimensions are those that appear in the sequential patterns to be mined. The table is partitioned according to tuple values over relevant dimensions and support computed by number of partitions that contain sequence. Note that the length of a partition is defined as the number of distinct values of time dimensions. In almost all the test cases, we found that using this length for distributing partitions among processors make a good load balancing. The well-known Apriori property holds on the algorithm, which states that any superpattern of an infrequent pattern is infrequent and any subpattern of a frequent pattern is also frequent.

To gain a better understanding of framework, let's start by looking at an example. Table T is defined over five dimensions as shown in Fig. 1. D is time dimension, A1 and A2 are analysis dimensions and R1 and R2 are relevant dimensions. According to values of relevant dimensions table T partitioned to two subsets (Fig. 2). Partition lengths in Fig. 2 are two and three, respectively.

D	A_1	A_2	R_1	R_2
1	a_1	a_2	r_1	r_2
1	a_1	a_2	r_3	r_4
1	a_5	a_6	r_3	r_4
2	a_3	a_4	r_1	r_2
2	a_5	a_6	r_3	r_4
3	a_3	a_4	r_3	r_4

Fig. 1 Example of table with five dimensions

D	A_1	A_2
1	a_1	a_2
2	a_3	a_4

D	A_1	A_2
1	a_1	a_2
1	a_5	a_6
2	a_5	a_6
3	a_3	a_4

Fig. 2 Partitioned subsets based on relevant attributes

In any partition, a sequence is called k-sequence if there exist k subsequences which be selected from different time dimension. For example $<\{(a_1,a_2),(a_5,a_6)\},\{(a_3,a_4)\}>$ is a 2-sequence in Fig. 2 right subset. Using this concept can be helpful to distribute subsets. Fig. 3 shows that how we can distribute 12 subsets among 4 processors. As you see, subsets ordered according to its length and allocate them to processors. For example, subsets with length 6, 4, and 3 allocate to the first processor and subsets with length 5, 5, and 2 allocate to the fourth processor.

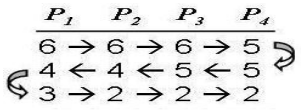

Fig. 3 Subsets distribution among 4 processors

A dynamic programming approach can be used to generate sequence with length n from sequence with length n-1.

Given a table with length m and d_i is number of tuples with time dimension value i. Number of sequences with length m computed as follows:

$$m-sequence=\prod_{i=1}^{m}(2^{d_i}-1) \quad (1)$$

As you observe, search space is quite large and the serial algorithms are not scalable. To overcome this problem, it is necessary to implement scalable parallel algorithms of sequence mining algorithms.

The main steps of our parallel algorithm are as follows:

1. One worker (processor) known as coordinator, partitions table T into P subsets according to relevant attributes.
2. Coordinator distributes P subsets among workers based on subset's length.
3. Every worker mines all k-sequence and computes its support (local support) and then sends to coordinator
4. Coordinator collects results and discards some sequences in which its support is less than min-support and sends sequence patterns to workers.
5. Repeat 3 and 4 until there is no sequence to be mined.

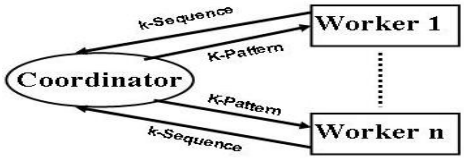

Fig. 4 Schematic diagram of method

Schematic diagram of method shows in Fig. 4.Our proposed algorithm conducts a level-by-level candidate generation and test pruning following the Apriori property in workers independently. At each level, only potentially frequent candidates are generated and tested.

IV. Experimental Results

This section describes a set of simulations used to demonstrate the benefit of our approach. We simulate many experiments with various synthetic datasets on dual core, single processor machine. For this reason, we did not obtain excellent speedup but results show good scalability of our model, run time grows almost linearly when the database size increases. Maximum length of patterns is 10 and even with corresponding value of length each examination took long time in this situation. Minimum value of support, number of analysis attributes, and size of database are variables which we have changed and obtain some plots. Note that we are looking for all of multidimensional sequential patterns. The trend is clear and consistent. Limited by space, there are only the results on some selected datasets here.

Fig. 5 shows number of sequences based on support value. Experiments were carried out on some synthetic databases. This database contains 10,000 data sequences (with an average of 47 itemsets) over 5 analysis dimensions.

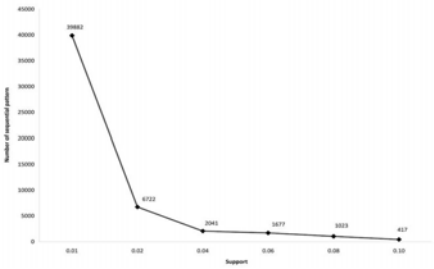

Fig. 5 number of sequences over support value

Fig. 6 and 7 show the effect of the four workers over support value and database increases, respectively. These Figures show experimental performance results we obtained by simulating our model using up to 4 workers. We can see how the system behavior is scalable. Speedup on processors is about 3.6 and run time decreases from 14605 to 4058 seconds. We illustrate how the system behavior can be scalable provided that our model implement on real parallel environment.

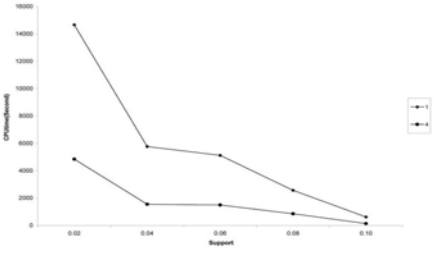

Fig. 6 Runtime over Support

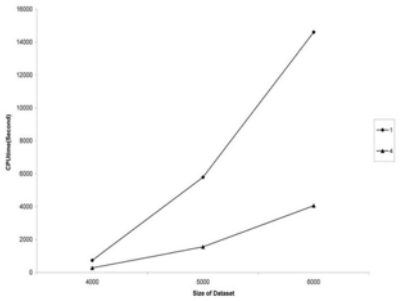

Fig. 7 Runtime over Size of Dataset

V. Conclusion

Multidimensional mining has been attracting attention in recent research into data mining. Very large search space and data volume have made many problems for serial algorithms to mine sequential patterns. In order to effectively mine, efficient parallel algorithm is necessary. In this paper, we theoretically present a multidimensional sequence model and then use SPMD (Single Program Multiple Data) strategy to parallelize multidimensional sequential pattern mining. According to this method a set of processors execute in parallel the same algorithm on different partitions of a dataset. The main goal of the algorithm is balanced workload among the processors and good scalability. We have simulated our parallel algorithm using several data sets on single processor. Simulation results on synthetic data sets show that if we implement this model on real computing environment, we may achieve good speedup.

The future research issues in multidimensional sequential pattern mining can be: multidimensional sequential pattern mining with constraints, interactive multidimensional sequential pattern mining or multidimensional sequential pattern mining integrated with taxonomies and hierarchies.

References

[1] J. Han and M. Kamber, *Data Mining: Concepts and Techniques* , 1st ed., Morgan Kaufmann, New York, August 2001.

[2] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Eleventh International Conference on Data Engineering*, P. S. Yu and A. S. P. Chen, Eds. Taipei, Taiwan: IEEE Computer Society Press, 1995, pp. 3-14.

[3] H. Pinto, J. Han, J. Pei, K. Wang, Q. Chen, and U. Dayal, "Multi-dimensional sequential pattern mining," in *Proceedings of the tenth international conference on Information and knowledge management (CIKM '01)*. New York, NY, USA: ACM, 2001, pp. 81-88.

[4] M.J. Zaki, H. Ching-Tien, " Large scale parallel data mining", *Lecture notes in artificial intelligence*, Vol 1759, Springer-Verlag 2000.

[5] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," in *Proc. 5th Int. Conf. Extending Database Technology, EDBT*, P. M. G. Apers, M. Bouzeghoub, and G. Gardarin, Eds., vol. 1057. Springer-Verlag, FebruaryMay-FebruarySeptember~ 1996, pp. 3-17.

[6] M. J. Zaki, "Spade: An efficient algorithm for mining frequent sequences," *Machine Learning*, vol. 42, no. 1/2, pp. 31-60, 2001.

[7] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "Prefixspan,: mining sequential patterns efficiently by prefix-projected pattern growth," in *Proc.17th Int'l Conf. on Data Eng*, 2001, pp. 215-224.

[8] W. Jinlin, X. Chen, Z. Kefa,W. Wei, "Parallel Research of Sequential Pattern Data Mining Algorithm", *Int'l Conference on Computer Science and Software Engineering*, vol 4, 2008, pp. 348-353.

[9] T. Shintani, M. Kitsuregawa, "Mining algorithms for sequential patterns in parallel: Hash based approach", *In Proc of the Second Pacific-Asia Conf on Knowledge Discovery and Data mining*, 1998, pp. 283–294.

[10] V. Guralnik, N. Garg, G. Karypis, "Parallel tree projection algorithm for sequence mining", In *Proc of 7th European Conf on Parallel Computing*, 2001, pp. 310–320.

[11] S. de Amo, D.A. Furtado, A. Giacometti, D. Laurent, "An apriori-based approach for first-order temporal pattern mining", in: *Proceedings of the 19th Brazilian Symposium on Databases*, Brasilia, Brazil, October 2004, pp. 48-61.

[12] M. Plantevit, Y.W. Choong, A. Laurent, D. Laurent, M. Teisseire , "M2SP: Mining Sequential Patterns Among Several Dimensions", *Principles of Knowledge Discovery in Databases*, Volume 3721, page 205-216, 2005.

[13] C.-C. Yu and Y.-L. Chen, "Mining sequential patterns from multidimensional sequence data," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 1, pp. 136-140, 2005.