An Optimized Parallel Confidence Measures Algorithm On Web Log Data

Khushboo Sharma
M.tech(c.s.e.)
DR. C. V. Raman University

Mr. S. R. Tandan
Asst. proff. C.S.E. dept
Dr. C. V. Raman University

Abstract

The enormous content of information on the World Wide Web makes it obvious candidate for data mining research. Web usage mining is the type of Web mining activity that involves the automatic discovery of user access patterns or association rules from one or more Web servers. Most of the existing algorithms of association rule mining require multiple passes over the database for discovering frequent patterns resulting in a large number of disk reads and placing a huge burden on the input/output subsystem. However, the work so far has been concentrated on designing serial algorithms. Since the databases to be mined are often very large (measured in gigabytes and even terabytes), parallel algorithms are required. Here, Web Usage Mining, approach has been combining with the Association Rules, parallel confidence measure Algorithm to optimize the content of the World Wide Web.

1. Introduction

Data mining is a technique used to deduce useful and relevant information to guide professional decisions and other scientific research [9]. It is a cost-effective way of analyzing large amounts of data, especially when a human could not analyze such datasets.

With the explosive growth of data available on the World Wide Web, discovery and analysis of useful information from the World Wide Web becomes a practical necessity. Information provided are interested in techniques that could learn Web users' information needs and preferences. This can improve the effectiveness of their Web sites by adapting the information structure of the sites to the users' behavior.

Web mining [10] is the application of data mining technologies to huge Web data repositories. Basically, there are two domains that pertain to Web mining: Web content mining and Web usage mining. The former is the process of extracting knowledge from the content of Web sites, whereas the latter, also known as Web log mining, is the process of applying data mining techniques to the discovery of usage patterns from Web data and is targeted towards applications. Web Usage Mining mines the secondary data (Web server access logs, browser logs, user profiles, registration data, user sessions or transactions, cookies, user queries, mouse clicks and any other data as the result of interaction with the Web) derived from the interactions of the users during certain period of Web sessions.

There are many efforts towards mining various patterns from Web logs, e.g. [4, 11, 15]. Web access patterns mined from Web logs are interesting and useful knowledge in practice. Examples of applications of such knowledge include improving designs of web sites, analyzing system performance as well as network communications, understanding user reaction and motivation, and building adaptive Web sites [5, 10, 13, 14].

Sequential pattern mining, which discovers frequent patterns in a sequence database, was first introduced by Agrawal and Srikant [2] as follows: given a sequence database where each sequence is a list of transactions ordered by transaction time and each transaction consists of a set of items, and all sequential patterns with a user-specified minimum support, where the support is the number of data sequences that contain the pattern. Since its introduction, there have been many studies on efficient mining techniques and extensions of sequential pattern mining method to mining other time- related frequent patterns [2, 12, 8, 7, 3, 9, 6].

All of the above studies on time-related (sequential or periodic) frequent pattern mining adopt an Apriori like paradigm, which promotes a generate and test method: first generate a set of candidate patterns and then test whether each candidate may have sufficient support in the database (i.e., passing the minimum support threshold test).

However, as these algorithms are level-wise, Apriori -like in nature, they encounter the same difficulty when the length of the pattern grows long, which is exactly the case in Web access pattern mining. In Web log mining, the length of Web log pieces can be

pretty long, while the number of such pieces can be quite huge in practice.

However, the work so far has been concentrated on designing serial algorithms. Since the databases to be mined are often very large (measured in gigabytes and even terabytes), parallel algorithms[16] are required. We present in this paper parallel algorithm for mining association rules on web log data, an important data mining problem. The algorithm has been designed to investigate and understand the performance implications of a spectrum of trade-offs between computation, communication, memory usage, synchronization, and the use of problem-specific information in parallel data mining. Specifically, The focus of the algorithm is on minimizing communication. It does so even at the expense of carrying out redundant duplicate computations in parallel.

The organization of the paper is as follows: Section 2 covers literature review. Section 3 reports on the proposed approach to address the problem, Section 4 reports on the initial analysis of experimental data, and finally Section 5 gives some final remarks and indications for the continuation of the work.

2. LITERATURE REVIEW

Web Mining and Web Usage Mining

Data mining efforts associated with the Web, called Web mining, can be broadly categorized into three areas of interest based on which part of the Web to mine; Web Content mining, Web Structure mining, and Web Usage Mining [12]. In Web mining, data can be collected at the server-side, client-side, proxy servers or a consolidated Web/business database [11]. The information provided by the data sources described above can be used to construct several data abstractions, namely users, page-views, click-streams and server sessions.

Web Usage Mining is defined as the process of applying data mining techniques to the discovery of usage patterns from Web logs data which to identify Web user's behavior [11]. Web Usage Mining is the type of Web mining activity that involves an automatic discovery of user access patterns from one or more Web servers.

As shown in Fig. 1, three main tasks are performed in Web Usage Mining; Pre-processing, Pattern Discovery and Pattern Analysis. Fig.1 represents a brief description about the main task of Web Usage Mining process

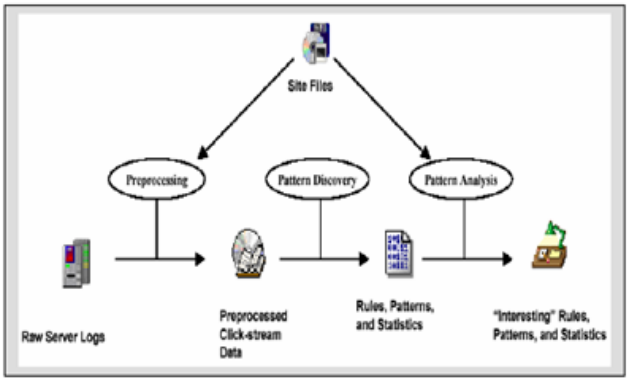


Figure 1: A High Level Web Mining Process

Association Rules and Apriori Algorithm

The problem of deriving Association Rules from data was first formulated in [1] and is called the "market-basket problem". The problem is that we are given a set of items and a large collection of transactions which are sets (baskets) of items. The task is to find relationships between the containments of various items within those baskets.

Apart from the supermarket scenario there are many other examples where Association Rules have been used, for example users' visits of WWW pages which the structure and its content can be optimized.[14] use page accesses from a Web server log as events for discovering frequent episodes.[15] introduce the concept of using the maximal forward references in order to break down user sessions into transactions for the mining of traversal patterns. [5] perform mining process for online newspaper Web access logs by using Apriori algorithm.

The task in Association Rules mining involves finding all rules that satisfy user defined constraints on minimum support and confidence with respect to a given dataset. Most commonly used Association Rule discovery algorithm that utilizes the frequent itemset strategy is exemplified by the Apriori algorithm [1].

Apriori was the first scalable algorithm designed for association-rule mining algorithm. Apriori is an improvement over the AIS and SETM algorithms [2]. The Apriori algorithm searches for large itemsets during its initial database pass and uses its result as the seed for discovering other large datasets during subsequent passes. Rules having a support level above the minimum are called large or frequent itemsets and those below are called small itemsets [15]. The algorithm is based on the large itemset property which states: Any subset of a large itemset is large and if an itemset is not large and then none of its supersets are large [2].

Apriori Algorithm
L_1 := {frequent 1-itemsets};
k := 2;
while (L_{k-1} ¹ 0) **do**
C_k:=new candidates of size *k* generated from L_{k-1};
forall transactions *t* Î D do
Increment the count of all candidates in C_k that are contained in *t*;
L_k:=All candidates in C_k with minimum support;
k:=k+1;
end
Answer:= È*k* L_k;

3. PROPOSED METHODOLOGY

The algorithms assume a shared-nothing architecture [16], where each of N processors has a private memory and a private disk. The processors are connected by a communication network and can communicate only by passing messages. The communication primitives used by our algorithms are part of the MPI (Message Passing Interface) communication library supported on the SP2 and are candidates for a message-passing communication standard currently under discussion. Data is evenly distributed on the disks attached to the processors, i.e. each processor's disk has roughly an equal number of transactions. We do not require transactions to be placed on the disks in any special way.

Algorithm

This algorithm uses a simple principle of allowing redundant computations in parallel on other- wise idle processors to avoid communication".

Algorithm:

Pass 1:

1. Each processor P^i generates its local candidate itemset C^i_1 depending on the items present in its local data partition D^i
2. Develop and Exchange local counts C^i_1
3. Develop global support counts C_1

Pass *k>1*:

(1) P^i generates the complete C_k using the complete L_{k-1} created at the end of pass (k-1).
Each processor has the identical L_{k-1} thus generates identical C_k and puts its count values in a common order into a count array

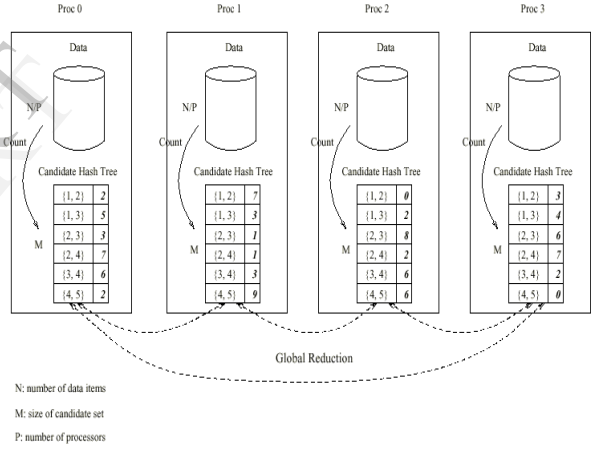
(2) P^i makes a pass over data partition D^i and develop local support counts for candidates in C_k

(3) P^i exchanges local C_k counts with all other processors to develop global C_k counts. All processors must synchronize.

(4) P^i computes L_k from C_k

(5) P^i independently decide to terminate or continue to the next pass

In the first pass, each processor P^i dynamically generates its local candidate itemset C_i^1 depending on the items actually present in its local data partition D_i. Hence, the candidates counted by different processors may not be identical and care must be taken in exchanging the local counts to determine global C_1. Thus, in every pass, processors can scan the local data asynchronously in parallel. However, they must synchronize at the end of each pass to develop global counts.



4. EXPERIMENTAL EVALUATION

The experimental results and analysis of frequent sequences discovered from web log data is described in this section. The experimental analysis is conducted on the Click Stream Data set which is a server side web log data containing 12,000 records. Each record contains the following fields – a shop identifier, time stamp, IP address, unique session identifier, page visited, and referrer. The performance is tested on a computer with a 1.41GHz processor. The program is developed using JDK 1.6.

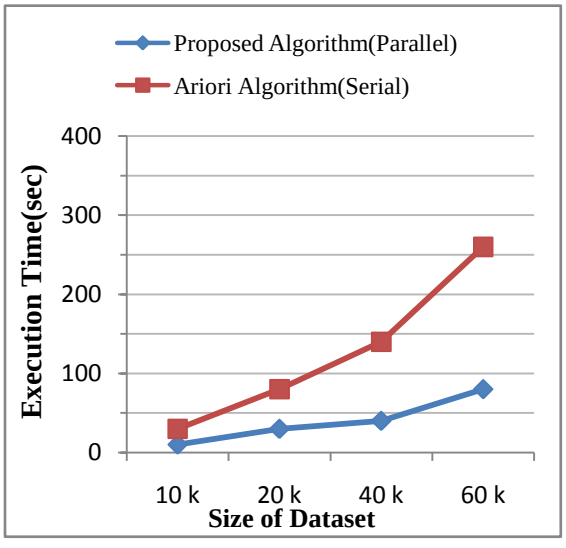


Fig 2: Execution time variation

Figure 2 compares the time taken by the apriori algorithm and proposed algorithm for different size of dataset. It is evident from the graph that proposed algorithm takes less time and space to generate rules. Most of the algorithms require multiple passes over the database for discovering frequent patterns resulting in a large number of disk reads and placing a huge burden on the I/O subsystem It is obvious form the figure that as the size of dataset increases time taken by the algorithm decreases.

5. CONCLUSION

This paper deals with the problem of discovering association rules from large amount of Web log data collected by web servers. The contribution of the paper is to show how frequent pattern discovery tasks can be applied on the web log data in order to obtain useful information about the user's navigation behavior. We have proposed parallel Confidence Measure Algorithm for parallel mining of association rules on web log data. The algorithm was designed to minimize communication. No data tuples are exchanged between processors only counts are exchanged. Processors can operate independently and asynchronously during the pass over the data, but need synchronization at the end of every pass. Although we focused on parallelizing the mining of association rules, the results and experience from this study have wider applicability.

6. REFERENCES

[1] Agrawal, R., Imielinski, T., and Swami, A. N. Mining Association Rules Between Sets of Items in Large Databases. Proceedings of the ACM SIGMOD, International Conference on Management of Data, pp.207- 216, 1993.

[2] Agrawal. R., and Srikant. R., Fast Algorithms for Mining Association Rules, Proceedings of 20th International Conference of Very Large Data Bases. pp.487-499,1994.

[3] Kosala and Blockeel, "Web mining research: A survey," SIGKDD: SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery and Data Mining, ACM, Vol. 2, 2000

[4] J. Pei, J. Han, B. Mortazavi-Asl, and H. Zhu, "Mining access patterns efficiently from web logs," in PADKK '00: Proceedings of the 4th PacificAsia Conference on Knowledge Discovery and Data Mining,

[5] P. Batista, Mario, and J. Silva, "Mining web access logs of an on-line newspaper," 2002.

[6] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan, "Web usage mining: Discovery and applications of usage patterns from web data," SIGKDD Explorations, Vol. 1, No. 2, pp. 12-23, 2000.

[7] F. Masseglia, P. Poncelet, and M. Teisseire, "Using data mining techniques on web access logs to dynamically improve hypertext structure". In ACM SigWeb Letters, 8(3): 13-19, 1999.

[8] F.M. Facca, P.L. Lanzi "Mining interesting knowledge from Weblogs: a survey", Data and Knowledge Engineering Vol. 53, No. 3, June 2005, pp 225-241.

[9] Chen, M.-S., Jan, J., Yu, P.S. (1996). Data Mining: An Overview from a Database Perspective. IEEE Transactions on Knowledge and Data Engineering, (8:6). pp 866.883.

[10] Jiawei Han, Micheline Kamber. Data Mining Concepts and Techniques [M]. Beijing: China Machine Press, 2001, p290-297.

[11] Srivasta, J., Cooley, R., Deshpande, M., and Tan P. N. (2000). Web Usage Mining: Discovery and Application of Web Usage Pattern from Web Data. Department of Computer Science and Engineering, University of Minnesota.

[12] Kosala, R., Blockeel, H. (2000). Web Mining Research: A Survey. ACM SIGKDD (Special Interest Group on Knowledge Discovery and Data Mining) Explorations. June, (2:1). Pp 1-10.

[13] Suneetha K R, Krishnamoorti R Web Log Mining using Improved Version of Apriori Algorithm International Journal of Computer Applications (0975 – 8887) Volume 29– No.6, September 2011.

[14] Renáta Iváncsy, István Vajk Frequent Pattern Mining in Web Log Data Acta Polytechnica Hungarica Vol. 3, No. 1, 2006
[15] Chen, M.-S., Jan, J., Yu, P.S. (1996). Data Mining: An Overview from a Database Perspective. IEEE Transactions on Knowledge and Data Engineering, (8:6). Pp 866.883.
[16] Rakesh Agrawal , John C. Shafer "Parallel Mining of Association Rules" IEEE transactions on Knowledge and Data Engineering 1996.