International Journal of Database Theory and Application
Vol.8, No.4 (2015), pp.1-10
http://dx.doi.org/10.14257/ijdta.2015.8.4.01

The Opportunistic Projection Mining Algorithm in Massive Data

WenwuLian[1]LinglingFu*[1] and Chao Huang[1,2]

Yulin Normal University, No.299Jiaoyu Road (minddle),Yulin,Guangxi,China.
Guilin University of Electronic Technology, NO.1
JinjiRoad,Guilin,Guangxi,China.
wwll_lf@163.com

Abstract

At present, many mining algorithms have been proposed only small data set mining and sparse data sets, encountered massive data sets and intensive data sets tend to collapse. In this paper, a kind of support for the massive data mining opportunistic projection set by frequent pattern tree model to construct a new algorithm, called OP. OP is completely different from the previous algorithm , using a new model supports two sets projection methods : projection -based virtual tree representation , based on non- filtered projection array representation , not only time efficiency is particularly high, and in particular to save memory space. Finally, the article by the Apriori, FP-Gorwth and H-Mnie comparative experiments confirm the size and characteristics of the various OP database mining efficiency and scalability are the best.

Keywords: *massive data; opportunistic projection; data mining; op algorithm*

1. Introduction

With the rapid development of computer and storage technology in all aspects of business, social and engineering, medicine, etc. will produce large-scale data, and how to deal with these data is a major problem in front of people. Although the pro- deposit huge amounts of data, but failed to find important information hidden in them, leading to lack of knowledge. Therefore urgently needed an effective technology to discover hidden knowledge in the data ring think, this appeared in the mountains of data mining technology, which can effectively utilize the data collected and to fully exploit the value of these data. Efficiency of traditional serial processing of massive data mining algorithms [1] on a single processor is too low , cannot meet people's needs. Restricted processor itself hardware conditions, the improvement of the law and the serial character cannot get good results. Research on high-performance parallel mining algorithm is particularly important. In addition, incremental mining knowledge has been used when using the updated data set for mining, updates have been excavated in the basics, including the acquisition of new knowledge, removing expired knowledge, get updated on the data set all useful knowledge, but not all re-excavation, which can improve mining efficiency . An effective way to deal with massive data is incremental mining . Commonly used data mining algorithms association rules, clustering and classification of three. Association rule is mainly used to find links between the different services, already widely used in e-commerce personalized recommendation, medical diagnostics and other industries, has been a hot research data mining, association rules algorithm Apriori algorithm is the most typical algorithm many algorithms are improved on Apriori algorithm[2] , while many of the incremental mining algorithm is reasonable in its expansion , which is the most typical FUP algorithm DWCheng *et al* . FUP algorithm like Apriori algorithm in mining process requires multiple scans of the data set, multiple scans will seriously affect the efficiency of the algorithm, especially when dealing with massive amounts of data . Many scientific

ISSN: 2005-4270 IJDTA
Copyright ⓒ 2015 SERSC

name - are these two algorithms have been improved, but also made some improvements parallel algorithms. But in parallel algorithm design needs to consider the problem of load balancing, parallel strategies and other aspects. MapReduce

shielding the underlying complexity of distributed parallel processing details , so users only need to specify the Map and Reduce function , When the MapReduce program running on a cluster , do not have to consider how to divide the input data , allocation and scheduling , the system can automatically complete these working while processing cluster node failure , greatly reducing the complexity of parallel programming process , so be studied MapReduce-based massive data mining technology has great significance.

With the deepening of IT applications, especially the widespread use of bar code technology, the ability of people to produce and collect data rapidly. Thousands of databases has been widely applied to various areas of government , businesses , banks, research institutions , there has been explosive data growth , but people 's ability to handle and analyze data is quite limited, exacerbated by the rise of the Internet " data explosion, lack of medical knowledge . " trend. Data mining {FU96, FPS ten 96, MBK98 " It is in this context , the rise in the late 1980s and made significant progress in the nineties a new field of study.

In recent years, domestic and international stakeholders are aware of the importance of data that the new scientific era has arrived, you can find from the massive amounts of data in scientific laws and promote the development of social enterprises [3]. However, the huge amounts of data to find support from the scientific laws require massive data processing technology, Chinese Academy of Sciences, East China Normal University Software College dean He Jifeng has said: "By participating in a number of projects, I found that the current massive collection of data on some way, but there is no way to deal with massive data base . " The current massive data processing technology is indeed the case, although a long time before the data mining technology to produce, but that's just some of the basic algorithm for the application of a small amount of data to be required for the massive data mining for data processing and basic algorithm improvements. The massive data mining, the recent, relevant experts and scholars also some mining algorithm has been improved. In addition, with the rise of the Internet of Things, also need to analyze the amount of data processing to achieve unprecedented growth in networking operating environment , the current implementation of massive data, there are still difficulties in processing and handling of these problems we need more in-depth the study[4].

Currently, due to the rapid development of social networking sites, e-commerce and other network services, network services and network information so that the scale of fission growth, this will to deal with large-scale data is a big challenge. Financial services, retail, healthcare, telecommunications, aviation and other fields also produce large amounts of data in data mining how to handle massive amounts of data, improve the quality and efficiency of mining is an urgent need to address the problem. Research on this subject also has very important significance.

2. Related Works

In this paper, data mining algorithms massive data processing technology involved are: data mining related algorithms, massive data processing techniques. In essence, this study is the massive data mining category, is a hot issue in the present study.

Attention and research from the perspective of huge amounts of data in recent years, about the massive data processing by many domestic and foreign experts and business concerns. There is a journal of data growing a cover story [5]: Magazine February 11, 2011 issue of "Science" published in the topic - "Data Processing", huge amounts of data around the current increase in the topic. In this introduction to the topic "Challenges and

 Copyright ⓒ 2015 SERSC

Opportunities" that data collection, maintenance and use has become the main direction of scientific research, for many disciplines, the massive data means more severe challenges.

For massive data growth, many domestic and foreign in massive data mining, knowledge discovery stakeholders in the field of in-depth research. Massive data storage and processing power itself to data mining or machine learning made high demands , Google has done work in this sense. Google's Map Reduce is presented on a large computer cluster of massive data model is a framework for concurrent processing. It starts by setting a Map function to transform input data into the appropriate key - value pairs, and then through a custom function come together to reduce the value of having the same key and outputs the result . Most of the real world can use this model to represent the massive data processing [6]. In addition, parallel database technology and parallel database technology combined with the product, and is considered a high-performance database system, it can greatly improve the efficiency of a relational database processing huge amounts of data.

Also, how to deal with huge amounts of data, data mining has been a bottleneck to be solved. At present, many algorithms handle massive amounts of data, such as parallel or serial algorithms are generally not solve the contradiction between speed and accuracy. But distributed computing on the data processing has obvious advantages, 2004 Tan ZhengRen Wu Yu , who managed to split the original data set into several small massive data sets , and then be distributed processing, based on this idea , to a segmentation algorithm based on Rough Set of massive data . Program combines data proved segmentation algorithm for distributed processing can quickly deal with huge amounts of data and compared with the algorithm for processing the entire data set, without losing the correctness of the algorithm. 2004 Zhao Gong, Li Jianzhong, who is currently at large for the size of the database , association rule mining algorithm running time too long problem, an effective method for solving this problem that parallel computing, so on to the massive data mining parallel algorithm of association rules were studied, using frequent item sets can ignore local node machine in less than a quarter of the produce is only proposed a new method of parallel random sampling , combined with cluster parallel machine I / O height parallel processing capabilities with its own characteristics and improve the efficiency of processing massive amounts of data and capabilities. Simulation results show that this algorithm is accelerated compared with the number of processors p is close, communication complexity is the number of the number of processors p , and has good scalability, high precision and the ability to handle massive amounts of data . Wei Ting, Wu Jun, who granular computing based data mining application were studied. Granular Computing (Granular Computing, referred GrC) is a computing paradigm of information processing and new concepts, covering research theories, techniques, tools and methods related to the particle size of all , is mainly used to deal with vague, massive uncertainty and incomplete information. Zhi- long, Matsu red and others in the search process frequent item sets, the need to save for vertical data format based on association rule mining algorithm in memory, a large number of transactions flag list, limited memory capacity will become the biggest bottleneck in such algorithm, a new hybrid compression algorithm --HC-DM algorithms. This efficient hybrid compression of data mining algorithms HC-DM algorithms and dEclat algorithm combines, plus sorting step, can effectively reduce the amount of memory usage mining frequent item sets in the process. January 2011, Zhangzhanjie and others for how to deal with massive data query efficiency, analyze and summarize the massive data processing techniques [7].

In the massive data mining and knowledge discovery in 2010, held an international seminar, Dr. IBM TJ WatsonResearch of Wei Fan made the theme of " Things in data management and analysis challenges and ideas," the report, Professor Liu Peng PLA University delivered the keynote as " cloud computing and data mining," the report, Beijing Jiaotong University, made the theme of " clustering algorithm in parallel mode selection ," the report professor Yu Jian , and so on. Their report has been praised experts

Copyright ⓒ 2015 SERSC

at home and abroad, it also illustrates the problem of massive data processing to get everyone's attention.

With huge amounts of data generated in recent years, cloud computing has emerged , Yahoo, Facebook, Baidu , China Mobile , Taobao, Tencent and other enterprises related to cloud computing research and applications. In the second session held in China Cloud Computing Conference in May 2010 cloud storage and virtualization sub-forum, Chinese Academy of Sciences Institute of Computing Technology researcher and doctoral tutor Qing He published a “cloud-based massive data mining" wonderful speech. According to Hadoop and virtualization technology , Hou , handsome Renjun *et al*. proposed a mass data storage model based on cloud computing, the massive deployment of information and data on the Hadoop platform, using the core algorithm MapReduce cloud computing to process massive amounts of data , and the data storage virtual resource pool. And through the practical application of this model can be a good way to overcome the deficiencies of existing storage, to solve the problem of mass data storage, but also to improve the storage efficiency [8].

From the above , at present, both at home and abroad for data mining massive data processing algorithms conducted in-depth research , and related algorithms to improve and make some new approach. But for massive data processing technology is not very mature, in the foreseeable future, the industry is facing massive data processing, and there are opportunities as well as challenges.

3. Proposed Scheme

This section presents a support set by opportunistic projection model to construct a new algorithm for frequent pattern tree Opportuneprojeet [LpWH02], referred to as OP. OP is completely different from the previous algorithm, using a new model supports two sets projection methods: projection -based virtual tree representation, based on non-filtered projection array representation, not only time efficiency is particularly high, and in particular to save memory space. OP can select the appropriate mode according to the data characteristics set notation support, heuristic decision to projection methods. OP depth priority as the basic strategy to construct frequent pattern tree, complemented by a breadth-first strategy if necessary. Through a variety of experiments to test the real and artificial data sets , indicating higher efficiency OP time 1-3 orders of magnitude than the Apriori, FP-Growth and H-Mine, and also significantly better than the spatial scalability of these algorithms.

3.1 Opportunistic Projection Algorithm

3.1.1 Opportunistic Projection Heuristic Principles

To maximize time efficiency and spatial scalability, mining algorithms must make FIST generation and search strategy, PTS representation, PTS project counts and methods adapted to the characteristics of data projectors in the PTS.

The actual size of different databases, data features cannot simply be classified as pure sparse or pure -intensive. The following facts and heuristic principles are put forward time and space scalability to maximize the efficiency of opportunistic strategy based on projection [9].

Fact 1: Suppose a large database must be able to be loaded into memory is unrealistic, but assuming that the upper part of FIST can remain in memory is reasonable. Support frequent pattern length k number of transactions dropped, when k> 2. Therefore, you can use the width of the priority strategies to reduce the number of transactions. Many segmentation algorithm uses a database approach , but local frequent mode of each sub-set of libraries and sets may not be retained in memory, so that the split method failure.

 Copyright ⓒ 2015 SERSC

Heuristic Principle 1: For large databases, the first use of the breadth-first strategy to build FlsT half. When support for all k layer nodes (reduced) transaction sets can be represented when using memory-based structure, use depth-first strategy to build FIST k layer below.

Fact 2: In FIST level, each node in the PTS and the transaction is often varied randomly distributed among each other have less opportunity to share the same prefix, TTF is not well supported by a subset of the compressed mode. Because TTF requires more than TVLA additional storage overhead in FIST level, TTF often than TVLA high storage overhead. On the other hand, in the lower FIST, each node PTS less frequent items locally and relatively high support rate, and the higher the relative support rate, the higher the compression ratio TTF.

Heuristic Principle 2: In FlsTlevel, using TVLA said PTS, unless TTF estimated compression ratio is large enough.

Fact 3: The FIST high or relatively sparse branching, PTS shrink very quickly, but this time PTS often used TVLA said. For further counting and projection operation, the filter type TVLA higher efficiency than non -filtered TVLA. On the other hand, in the FIST lower or denser branching, PTS contraction was slower, but this time PTS often used TTF said . Create a filter type TTF involving high CPU overhead pattern matching.

Heuristic Principle 3: When the projector father TVLA, as long as there is enough free memory on the establishment of filter -type children TVLA. When the projector father TTF, first defined virtual child TTF, TTF shrink very quickly if you make a copy of the filter type.

3.1.2 Opportunistic Projection Heuristic Principles

Given PTS, set the number of items frequently is f, the number of transactions the total number of o appear as t, frequent project. , The TVLA exact size is 3 * f + 2 * t + (o-t), of which 3 * f is FIL size, 2 * t is LQ size, (o-t) is the size of all of the array.

$$n = \sum_{i=1}^{u} C_f^i - \sum_{i=1}^{l-1} C_{f-i-1}^{l-i-1}\left(2^i - 1\right) \le 2^{f-1}$$

OP algorithm to estimate u and l based on the average length transaction. A large number of experiments show, which gives n estimated total greater than the actual number of nodes, is a conservative estimate and security. TTF compression rate: If r is less than 6-(t / n), the TTF large storage overhead than TVLA.

3.2 Opportuneprojeet

Based on the above discussion, given the depth first and breadth-first integrated strategy, based on an array of OpportuneProject algorithm based on tree representation, virtual non- filtered and filtered projection type projector , referred oP. It consists of a guided breadth and depth of process priority process components.

3.2.1 Width Priority Process

BreadthFirst to available memory for parameter control recursive process, according to the width of the first strategy in three steps to create the FIST half[10] .

First, CreateCountingVector (v). Each layer of the current node k , the support attached to count the number of local vectors to each node of the accumulated items PTS . In accordance with the provisions of order of each node v brothers after tagging project has a corresponding component in the count vector. For example, in Figure 3 a a node (a, 3) of the partial program to PTs b, c, f, m and p, is the junction point (a, 3) items marked brother node. Therefore, the count of the length of the vector 5 is attached to one of the node to locally support the accumulated items b, c, f, m , and p the number .

Copyright ⓒ 2015 SERSC

Second, ProjectAndCount (t, D '). The transaction t along the path from the root to the start of the first projection k layer node and the corresponding cumulative count vector. If a transaction can be projected onto a node in layer k and the vector component of the count value of the node , then t may also be projected onto the layer of k + 1 , so that the t is added to D '. Otherwise ,t does not require further consideration . In this case, the number of transactions will decrease step by step.

Third, Generateehildren (v). V Create a child node for each node k of the current layer. Each value of V exceeds support rate component count threshold corresponds to a local frequent item, a v accordingly with children's. If v has no children, then v can be deleted, since v following branches will no longer grow. If and is the only child of its parent node, the node v 's father also deleted, so

```
OPPortuneProjeet(Database：D)
begin
ereate a null root for frequent item set tree T;
D'=BreadthFirst(T,D);
v=thenullrootofT;
GuidedDePthFirst(v,D');
end
BreadthFirst(FIST:T,CurrentLeve]:L,Database:D)
begin
foreachnodevatlevelLofTdo
CreateCountingVector(v);
D'={}:
ForeachtransactiontinDdo
ProjectAndCount(t,D');
foreaehnodcvatlevelLofTdo
GenerateChi}dren(v);
ifD'cannotberePresentedbyTVLAandTTF
thenBreadthFirst(T,L+l,D');
eISereturn(D');
Cnd
GuidedDePthFirst(CurrentFISTNode:P,PTS:D)
begin
FILP=TraverseAndCount(D,p);
DP=RePresent(D,P);
ForeaehentryeinFILPbyaPartieularorderingdo
begin
C=GetChild(P,e);
GuidedDePthFirst(c,DP);
end
end
```

3.2.2 There is a Depth-first Process Wizard

Assuming BreadthProject k layer at the end. So, only a path of length k FIST retained in the. The following sections FIST's press k layer has a depth-first strategy GuidedDePthFirst wizard to build.

First, TraverseAndCount (D, p) scanning D, determine the support node p of transaction sets Dp, Dp and get the FILp. If D is retained on disk or in TVLA expressed, create FILp at this time. If D expressed in TTF, the FILp already exists in the father IL.

Second, Represent (D, p). If D is retained in the form of a disk or TVLA said density estimate, and np is sufficiently large compared to the expression of Dp create TTF, or the

 Copyright ⓒ 2015 SERSC

creation of TVLA expressed as Dp. If D is expressed in TTF, compared with Dp create virtual TTF, if necessary, (much less than the size Dp D), a further copy of Dp of a filter type.

Third, GetChild (p, e) e get a label with the same project for the children of the node c p. If p is greater than or equal level of the node where k, is created at this time. . Otherwise, C has been created by BreadthProject, if not retrieved c indicates. FlsT branch where the maximum length will be less than k, no longer grow.

Guided depth priority than a simple depth-first strategy is more efficient because it has been created to avoid duplication created by BreadthFirst terminates at FIST part (k layers or more) of the path.

4. The Experimental Results and Analysis

Opportuneproject, called Op, and Apriori [Borgelt], FP-Growth [HPY00] and H-Mine [PHL + 01] in order to evaluate the efficiency and effectiveness of comparative experiments carried out on large data sets , based on the experimental platform is at 800MHz Pentium IV CPU, 512MB RAM and a 20GB hard drive , running Mierosoft Windows 2000 Server operating system.

4.1 Data Set and its Properties

Comparative data sets used in the experiments shown in Table 1 characteristics.

Dataset BMS-POS, BMS-Web View-l and BMS-Web View-2 is the actual data set , and was classified as sparse data sets " ZKM01] .BMS-POS includes an electronics retailer several years of business sales data .BMS-WebView-1 and BMS-Web View-2 includes two e-commerce site click data several months. support rate threshold of 0.1% , BMS-Web View-1 has a 3,991 a frequent pattern , 0.058 there is a time % 1,177,607. support rate threshold of 0.1% , BMS-Web View-2 has 23,294 a frequent pattern , there are 1,316,614 a 0.02 %.

Datasets from UCI machine learning connect4 dataset [MLR], is very intensive data collection, the support rate threshold from 90% down to 70% and 50%, the number of frequent patterns soared from 27,127 to 4,129,839 and 88,316,367 .

IBM artificial datasets from IBM Almaden Laboratory Data Mining Research Group [QUEST] provided generator produces. T25IZoNZoL5k is between the data sets (take D100k ~ D15m) between sparse and intensive.

Table 1. The Basic Characteristics of the Data Set

	Trans.Number	Dist.Items	Max.Trans.Size	Aver.Trans.Size
BMS-POS	515,597	1,657	164	6.5
BMS-WebView-1	59,602	479	267	2.5
BMS-WebView-2	77,512	3,340	161	5.0
Connect4	67,557	150	43	43.0
IBM Artificial	100k~15m	20,000	72	28.4

4.2 The Basic Experimental Results

Comparative analysis algorithm performance of each algorithm is run at different times in different data sets support rate threshold. Here, the running time to scan the database includes only (hard to read) and the CPU time, excluding the output mode (write accesses) of the time to reduce hard disk write operation is slower impact.

Copyright ⓒ 2015 SERSC

As shown in Figure, OP performance on each of the data sets better than the other algorithms, Fig longitudinal axis (time) in logarithmic coordinates, the horizontal axis (the support rate threshold) in percentage . OP with respect to the performance of other algorithms to improve within a reasonable range of low support rate threshold is also very significant.

Time for BMS-POS, the support rate higher than 0.4% threshold (frequently less than 6,656 the number of modes), four algorithms have the same performance. When the threshold is higher than the 0.1 percent support rate (often less than the number of modes 122,449), FP-Growth performance and OP similar. When the support rate of less than 0.1% threshold, it becomes significant difference in performance. At reasonably low support rate threshold 0.04% (frequent pattern number 984,531) level, OP needs 34 seconds, while the FP-Growth requires 72 seconds, H-Mine needs 277 seconds, while APriori needs 781 seconds . In the lower support rate threshold level. OP needs 52 seconds, while the FP-Growth requires 215 seconds , H-Mine needs 727 seconds , Apriori needs 2012 seconds. Algorithm performance from highest to lowest for OP> FP-Growth> H-Mine>Apriori.

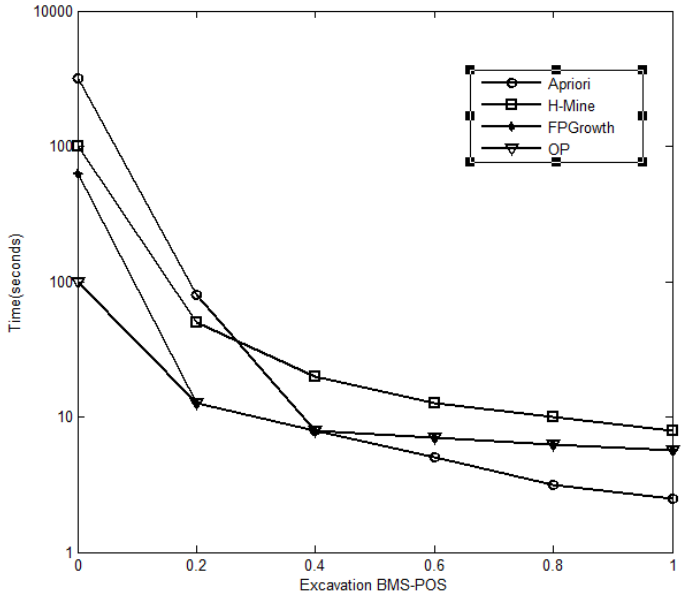


Figure 1. Support Rate 0.01% -1 %

For BMS-WebView-l and BMS-WebView-2, algorithm performance from highest to lowest for OP> H-Mine>Apriori>Fp-orowth. Support for a large threshold (BMS-WebView-1 was 0.1%, BMS-WebView- 0.3%) , the running time of the algorithm is almost the same as three , but the FP-Growth time than the other algorithms running a large magnitude . For BMS-webView-1 support rate of less than 0.06 percent threshold for BMS-WebView-2 is lower than 0.05%, OP faster than the H-Mine an order of magnitude, nearly two orders of magnitude faster than Apriori, faster than the FP-Growth two orders of magnitude or more.

For conneet4, H-Mine and Apriori running time is basically in the same order of magnitude. OP efficiency than H-Mine and Aprioi three orders of magnitude higher. Support rate threshold is large, OP and FP-Growth run time in substantially the same magnitude. However, the support rate is below the 80% threshold, the difference in performance is obvious. For example, the support rate when the threshold is 70%, OP end within five seconds , while FP a Growth run more than 27 seconds . When the support rate is below the threshold 60%, OP and FP a Growth Factor performance difference is greater than 20.

 Copyright ⓒ 2015 SERSC

4.3 Scalability Test

First, for T25I20N20kL5k datasets within 0.1% to 1% support rate threshold range, take D100k, D1m and D10m, testing and analysis. Four algorithm performance curve and DIOOk discussed earlier on D1m and D10m__ have the same trend, although the translation mode distribution occurs, such as frequent when the number of modes 60OK , D100k support rate threshold is approximately 0.195% , while D1m and D10m were 0.19 % and 0.188% .

OP for massive database scalability than other algorithms, the performance increase is more pronounced. For example , the same 600K frequent pattern discovery , when the database size from D10Ok to D1m, OP with respect to improving the performance of H-Mine factor from 7 to 20 , compared to APriori performance improvement factor from 100 to 300 . , With respect to the FP-Growth Factor performance increase from 16 to 8.

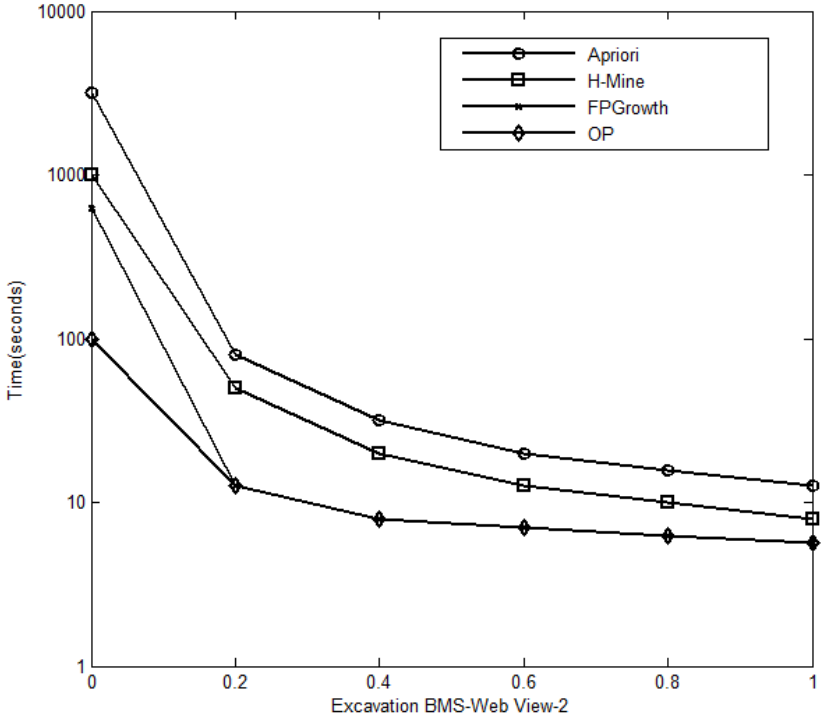


Figure 2. Support Rate 0.2%

Then, for data sets T25I20N20kL5k, database size D100k into D15m range, take 0.20,0 support rate threshold (maximum length of 13 patterns, frequently several modes of about 17K, except D200k for 45K • D100k for 99K), operating results Figure 3-14. Impressive is, OP running time is almost a linear relationship with the database size, scalability is excellent. For example, OP completed within 551 seconds D5m mining, complete D10m within 1295 seconds to complete D15m within 1961 seconds, and the memory overhead is less than 178MB. Please note that once required reading scan D15m hard time going 300 seconds. And APriori not run in D2M after, FP-Growth and H-Mine cannot run after D4M, because they ran out of memory.

5. Conclusion

This paper proposes a new algorithm can efficiently discover all the frequent patterns for various sizes of sparse or intensive database. The algorithm integrates a depth-first and breadth-first strategy, waiting for an opportunity to be able to set selection mode supports array-based or tree-based representation can apply virtual projection heuristic tree-based ,

Copyright ⓒ 2015 SERSC

non -filtered projection array-based or filter -type projectors , etc. method maximized the time efficiency and space scalability .

Acknowledgement

The work in this paper has been supported by funding from Natural Science Foundation of China (No. 61364020).

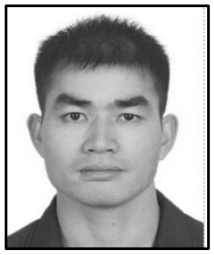
References

[1] J. Han, J. Pei and Y. Yin, Mining frequent patterns without candidate generation[C]//ACM SIGMOD Record, ACM, vol.29, no.2, **(2000)**, pp.1-12.
[2] J. Dean and S. Ghemawat, MapReduce: simplified data processing on large clusters[J]. Communications of the ACM, vol.51, no.1, **(2008)**, pp.107-113.
[3] K. Shvachko, H. Kuang and S. Radia, The hadoop distributed file system[C]//Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on. IEEE, **(2010)**.
[4] J. Deana and S. Ghemawat, "Distributed programming with Mapreduce[J] Beautiful Code", Sebastopol: O'Reilly Media, Inc, **(2007)**, pp.384.
[5] C. Aggarwal, "Towards long pattern generation in dense databases[J]", ACM SIGKDD Explorations Newsletter, vol.3, no.1, **(2001)**, pp.20-26.
[6] Y. Dong, X. Tai and J. Zhao J, "A distributed algorithm based on competitive neural network for mining frequent patterns[C]//Neural Networks and Brain, 2005. ICNN&B'05", International Conference on. IEEE, **(2005)**.
[7] R. Agrawal and J. C. Shafer, "Parallel mining of association rules[J]", IEEE Transactions on knowledge and Data Engineering, vol.8, no.6, **(1996)**, pp.962-969.
[8] R. Agrawal and R. Srikant, "Quest synthetic data generator", IBM Almaden Research Center, **(1994)**.
[9] J. Han and M. Kamber, "Data Mining, Southeast Asia Edition: Concepts and Techniques", Morgan kaufmann, **(2006)**.
[10] D. W. Cheung, S. D. Lee and B. Kao B, "A general incremental technique for maintaining discovered association rules", **(1997)**

Authors

WenwuLian, he received his B.Sc.degree in Mathematics and Applied Mathematics from Yulin Normal University and M.Sc. degree in Operations Research and Control theory from Dalian University of Technology, China in 2004 and 2009 respectively. His current research interests on Cloud computing and Data mining Mass data processing.

Lingling Fu, she received the B.Sc. degree in Mathematics and Applied Mathematics from Yulin Normal University and the M. Acc degree in Accounting Principles from Southwestern University of Finance and Economics, China in 2004 and 2009 respectively. His current research interests on Data mining and financial analysis.



Chao Huang, he received his B.S. Mgt.Sci in Yulin Normal University in 2008. He is currently pursuing M.Eng in Computer Science in Guilin University of Electronic Technology. His research interests mainly focus on Optimization of Data Mining, WEB Data Mining and the Application of Data Mining in Instructional Technology.

 Copyright ⓒ 2015 SERSC