The Mining Algorithm of Frequent Itemsets based on Mapreduce and FP-tree

Bo He
School of Computer Science and Engineering
ChongQing University of Technology
400054 ChongQing China
E-mail: hebo@cqut.edu.cn

Hongyuan Zhang
School of Computer Science and Engineering
ChongQing University of Technology
400054 ChongQing China
E-mail: www.464234870@qq.com

Jianhui Pei
School of Computer Science and Engineering
ChongQing University of Technology
400054 ChongQing China
E-mail: 794349116@qq.com

***Abstract*—The date mining based on big data was a very important field. In order to improve the mining efficiency, the mining algorithm of frequent itemsets based on mapreduce and FP-tree was proposed, namely, MAFIM algorithm. Firstly, the data were distributed by mapreduce. Secondly, local frequent itemsets were computed by FP-tree. Thirdly, the mining results were combined by the center node. Finally, global frequent itemsets were got by mapreduce and the search strategy. Theoretical analysis and experimental results suggest that MAFIM algorithm is fast and effective.**

Keywords-FP-tree; Mapreduce; Frequent itemsets; Big data; Data mining

I. INTRODUCTION

Data mining[1] was used to find a novel, effective, useful and understandable knowledge from the dataset. The main research directions of data mining include association rules[1] , classification, clustering and so on. The date mining based on big data[2,3] was a very important field. The key step of association rules was to get frequent itemsets[4,5] from dataset, and all frequent itemsets were subsets of maximal frequent itemsets. Therefore, all frequent itemsets could be found by mining maximal frequent itemsets. In order to improve the mining efficiency[6,7], the mining algorithm of frequent itemsets based on mapreduce and FP-tree was proposed, namely, MAFIM algorithm.

II. RELATED DESCRIPTION

A. Description of Mining Global Frequent Itemsets

The global transaction database[8,9] as DB, number of transaction as D. P_1、P_2、…、P_n, as the computer node, DB_i(i=1,2,…,n) as local transaction database for DB stored in the P_i node, the number of transaction is D_i, then $DB=\bigcup_{i=1}^{n} DB_i$, $D=\sum_{i=1}^{n} D_i$. Global frequent itemsets mining is through many nodes cooperation and finally dig out the global frequent item[10,11] E_{DB} of DB and the maximum frequent itemsets F_{DB}.

B. Description of Global Maximum Frequent Itemsets

Global transaction database as db, number of transaction as d.db_i (i=1,2,…,n) as local transaction database for db stored in the P_i node, the number of transaction is d_i , then $db=\bigcup_{i=1}^{n} db_i$, $d=\sum_{i=1}^{n} d_i$. Global maximal frequent itemsets used E_{DB} and F_{DB} which have been mined, and digging out the whole transaction database's global frequent item $DB \cup db$ and global maximal frequent itemsets $F_{DB \cup db}$.

C. Relevant Definition

Definition 1: To a set X, Local database $DB_i(i=1,2,\ldots,n)$ includes X's transaction number, called local frequency of X in DB_i, use $X.si_{DB}$ as the symbol. The local frequency of X in db_i was $X.si_{db}$.

Definition 2: To a set X, Global transaction database DB includes X's transaction number, called global frequency of X in DB, use $X.s_{DB}$ as the symbol. The global frequency of X in db was $X.s_{db}$.

Definition 3: To a set X, if $X.si_{DB} \geq minsup \times D_i(i=1,2,\ldots,n)$, called X is a local frequent itemsets of DB_i, all local frequent itemsets compose to F_{DB_i} , where minsup is the minimum support threshold. All local frequent itemsets in db_i compose to F_{db_i}.

Definition 4: To a set X, if $X.s_{DB} \geq minsup \times D$, called X is a global frequent itemsets of DB, all global frequent itemsets compose to F_{DB} . All global frequent itemsets in db compose to F_{db}.

Definition 5: To sets X and Y, if $X \subseteq Y$, called X is a subset of Y, Y is a superset of X.

Definition 6: DB's global frequent itemsets X, if X is not a superset of all global frequent itemsets, called X is a global frequent itemsets of DB, all global frequent itemsets compose to F_{DB}. All global frequent itemsets in db compose to F_{DB}.

Definition 7: x_i is a item of DB, set X={ x_i }, if $X.s_{DB} \geq minsup \times D$, called x_i is a global frequent item of DB, all global frequent itemsets compose to E_{DB}. All global frequent itemsets in db compose to E_{db}.

D. Relevant Theorem

Theorem 1: If the itemsets X is a global frequent itemsets for DB, then X is a local frequent itemsets in a local database DB_i (i=1,2,…,n).

Prove: X is a global frequent itemsets for DB,satisfy $X.s_{DB} \geq (D_1 + D_2 + ... + D_n) \times \min\sup$. According to the Pigeonhole principle, there is at least one local database DB_i, make $X.si_{DB} \geq minsup \times D_i$,so X is a local frequent itemsets of DB_i theorem 1 established.

Theorem 2: If the itemsets X is a global maximum frequent itemsets for DB, then X is a subset of a local maximal frequent itemsets in a local database DB_i (i=1,2,…,n).

Prove: itemset X is the global maximum frequent itemsets of DB. The itemset X is global frequent itemsets . According to theorem 1, X is a local frequent itemsets of DB_i for a local database. According to the definition of 6, X is a subset of a local maximal frequent itemsets on the DB_i, theorem 2 established.

Theorem 3: The global maximum frequent itemsets of global transaction database DB and global increment transaction database A are respectively DB and B

The global maximum frequent itemsets of global transaction database DB and global increment transaction database db are F_{DB} and F_{db} respectively, the global maximum frequent itemset of DB $\cup$ db is $F_{DB \cup db}$, for any set of $X \in F_{DB \cup db}$, both have itemset $Y \in F_{DB} \cup F_{db}$, promote $X \subseteq Y$.

Prove: If itemset X is any of any global maximum frequent itemsets, according to theorem 2, X may be a subset of the global maximal frequent itemsets in DB, and may be a subset of the global maximal frequent itemsets in db, theorem 3 established.

Theorem 4: E_{DB} is the global frequent item of DB which according to the support component in descending order, E_{db} is the global frequent item of db which according to the support component in descending order, all items in $E_{DB} \cap E_{db}$ are global frequent items in DB $\cup$ db.

Prove: If item X is any one of $E_{DB} \cap E_{db}$, then x is not only a global frequent items of DB, but also a global frequent items of db, X={x}, that $X.s_{DB} \geq minsup \times D$ and $X.s_{db} \geq minsup \times d$,therefore,$X.s_{DBUdb} = X.s_{DB} + X.s_{db} \geq minsup \times (D+d)$, theorem 4 established.

III. MAFIM ALGORITHM

MAFIM algorithm was proposed. Firstly, the data were distributed by mapreduce. Secondly, local frequent itemsets were computed by FP-tree. Thirdly, the mining results were combined by the center node. Finally, global frequent itemsets were got by mapreduce and the search strategy.

The pseudocode of MAFIM is described as follows.

Alogrithm MAFIM

Input: The local transaction database DB_i which has M_i tuples and $M = \sum_{i=1}^{n} M_i$, *n* nodes P_i(*i*=1,2,…*n*), the center node P_0, the minimum support threshold *min_sup*.

Output: The global frequent itemsets *F*.

Methods: According to the following steps.

```
step1. /* the data were distributed by mapreduce*/
        for(i=1;i<=n;i++)
            P0 transmits DBi to Pi ;
Step2. /*local frequent itemsets were computed by FP-tree*/
        for(i=1;i<=n;i++)
        {   creating the FP-tree^i;
            Fi =FP-growth(FP-tree^i, null);
        }
step3./* the mining results were combined by the center node*/
        for(i=1;i<=n;i++)
            Pi sends Fi to P0;
        P0 combines Fi and produces F' ; /* F' = ∪_{i=1}^{n} Fi */
Step4./*global frequency of itemsets were computed*/
        for each items d ∈ the remain of F'
            d.s = Σ_{i=1}^{n} d.si ;
step5./*global frequent itemsets were got by mapreduce and the search strategy*/
        for each items d ∈ the remain of F'
            if (d.s>=min_sup*M)
                F=F ∪ d;
```

IV. EXAMPLE OF MAFIM ALGORITHM

With 3 stations P1, P2 and P3, corresponding to a local database DB1, DB2 and DB3. Each database as shown in table I. Minimum support threshold min_sup=0.42.

TABLE I. LOCAL DATABASE DB1, DB2, DB3

Local database	ID	Transaction
DB1	100	a，b，c，k，m，f，e，l，p
	101	c，k，b，m，o，q
	102	a，b，c，d，e
DB2	200	f，h，j，q
	201	a，b，c，m，l，f，k
	202	c，r，s，t，q
DB3	300	a，b，c，d，e，f
	301	b，c，d，k，q
	302	f，s，m，q

According to table 1 and min_sup=0.42, can draw the global frequent items, in accordance with the degree of support in descending order, as shown in Table II.

TABLE II. GLOBAL FREQUENT ITEMS AND SUPPORT COUNT

Global frequent Items	Support count(Global frequency)
c	7
b	6
f	5
q	5
a	4
m	4
k	4

The global frequent itemset composed of E={c, b, f, q, a, m, k}.

The search strategy implementation process as shown in table III. Min_sup=0.42.

TABLE III. THE SEARCH STRATEGY

F'	Set the length of K	The search strategy	F
{{ c, b, m, k}, {c, b, a}, {c, b}}	*4*	*{ c, b, m, k}*	
{{c, b, a}, {c, b, m}, {c, b, k}, {b, m, k}, {c, b}}	*3*	*{c, b, a} ✓*	*{{c, b, a}}*
{{c, b, m}, {c, b, k}, {b, m, k}}	*3*	*{c, b, m}*	*{{c, b, a}}*
{{c, b, k}, {b, m, k}, {b, m}, {c, m}}	*3*	*{c, b, k} ✓*	*{{c, b, a}, {c, b, k}}*
{{b, m, k}, {b, m}, {c, m}}	*3*	*{b, m, k}*	*{{c, b, a}, {c, b, k}}*
{{b, m}, {c, m}, { m, k}}	*2*	*{b, m}*	*{{c, b, a}, {c, b, k}}*
{{c, m}, { m, k}}	*2*	*{c, m}*	*{{c, b, a}, {c, b, k}}*
{{ m, k}}	*2*	*{ m, k}*	*{{c, b, a}, {c, b, k}}*

V. EXPERIMENTS OF MAFIM

MAFIM compares with CD and FDM in terms of communication traffic and runtime. The experimental data comes from the sales data in July 2015 of a supermarket. The results are reported in Fig .1 and Fig .2.

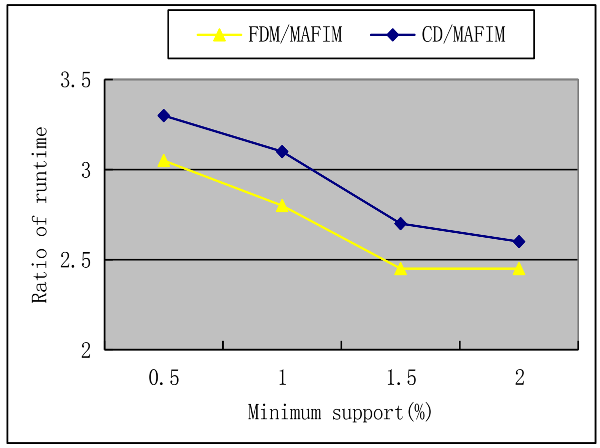

Figure 1. Comparison of communication traffic

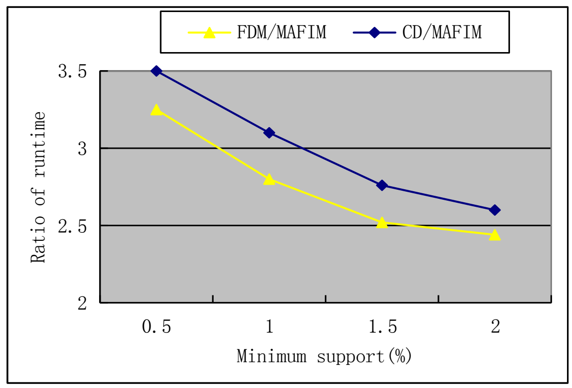

Figure 2. Comparison of runtime

The comparison experiment results indicate that under the same minimum support threshold, the communication traffic and runtime of MAFIM decreases while comparing with CD and FDM.

VI. CONCLUSION

The mining algorithm of frequent itemsets based on mapreduce and FP-tree was proposed. Firstly, the data were distributed by mapreduce. Secondly, local frequent itemsets were computed by FP-tree. Thirdly, the mining results were combined by the center node. Finally, global frequent itemsets were got by mapreduce. It can promote highly the efficiency of data mining.

ACKNOWLEDGMENT

This research is supported by scientific and technological research program of chongqing municipal education commission under grant No.KJ1709214. This research is supported by the national statistical science research project under grant No.2015LZ22 and the humanities and social science research project of chongqing municipal education commission under grant No.15SKG131.

REFERENCES

[1] Han JW, Kamber M, Pei J. Data Mining: Concepts and Techniques Third Edition [M]. San Francisco: Morgan Kaufmann, 2011.

[2] Big Data Across the Federal Government [EB/OL]. http://www.whitehouse.gov/sites/default/files/microsites/ostp/big_data_fact_sheet_final_1.pdf, 2012.

[3] Science. Special Online Collection: Dealing with Data [EB]. http://www.sciencemag.org/site/special/data/, 2011.

[4] Marconi K, Lehmann H. Big Data and Health Analytics[M]. Boca Raton:CRC Press, 2014.

[5] He B, Yan H. Incremental Updating Algorithm of Global Maximum Frequent Itemsets in Distributed Database[J]. Journal of Sichuan University(Engineering Science Edition), 2012,44(3):112~117. (in Chinese with English abstract)

[6] McKinsey&Company. The big-data revolution in US health care: Accelerating value and innovation [R]. http://www.mckinsey.com/industries/healthcare-systems-and-services/our-insights/the-big-data-revolution-in-us-health-care, 2013.

[7] He B. Fast Mining of Global Maximum Frequent Itemsets in Distributed Database [J]. Control and Decision, 2011,26(8):1214~1218. (in Chinese with English abstract)

[8] Muin J. Khoury and John P. A. Ioannidis. Big data meets public health[J]. Science, 2014, 346(6213) : 1054-1055.

[9] Chen ZB, Han H, Wang JX. Data Warehouse and Data Mining[M].Beijing: Tsinghua University Press, 2009.

[10] Song YQ, Zhu ZH, Chen G. An algorithm and its updating algorithm based on FP-tree for mining maximum frequent itemsets[J]. Journal of software, 2003,14(9):1586~1592(in Chinese with English abstract)

[11] Bayardo RJ. Efficiently mining long patterns form databases[C]. In: Haas LM, Tiwary A, eds. Proc. Of the ACM SIGMOD International Conference on Management of Data. Dallas:ACM Press, 2000. 1~12.