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

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

Hongyuan Zhang
School of Computer Science and Engineering
ChongQing University of Technology
400054 ChongQing China
E-mail: www.464234870@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, P1, P2, ..., Pn, as the computer node, DBi(i=1,2,...,n) as local transaction database for DB stored in the Pi node, the number of transaction is Di; then

\[ DB = \bigcup_{i=1}^{n} DB_i, \quad D = \sum_{i=1}^{n} D_i \]

Global frequent itemsets mining is through many nodes cooperation and finally dig out the global frequent item[10,11] EDB of DB and the maximum frequent itemsets FDB.

B. Description of Global Maximum Frequent Itemsets

Global transaction database as db, number of transaction as d, Dbi(i=1,2,...,n) as local transaction database for db stored in the Pi node, the number of transaction is di; then

\[ db = \bigcup_{i=1}^{n} db_i, \quad d = \sum_{i=1}^{n} d_i \]

Global maximal frequent itemsets used EDB and FDB which have been mined, and digging out the whole transaction database’s global frequent item DB \( \cup \) db and global maximal frequent itemsets FDB \( \cup \) db.

C. Relevant Definition

Definition 1: To a set X, Local database DBi(i=1,2,...,n) includes X’s transaction number, called local frequency of X in dbi was Xsi DB.

Definition 2: To a set X, Global transaction database DB includes X’s transaction number, called global frequency of X in DB, use XsDB as the symbol. The global frequency of X in db was Xsdb.

Definition 3: To a set X, if XsDB \( \geq \) minsup \( \times \) D, called X is a local frequent itemsets of DB, all local frequent itemsets compose to FDB, where minsup is the minimum support threshold. All local frequent itemsets in db compose to Fdb.

Definition 4: To a set X, if XsDB \( \geq \) minsup \( \times \) D, called X is a global frequent itemsets of DB, all global frequent itemsets compose to FDB. All global frequent itemsets in db compose to Fdb.

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 FDB. All global frequent itemsets in db compose to Fdb.

Definition 7: x is a item of DB, set X={x1}, if XsDB \( \geq \) minsup \( \times \) D, called x is a global frequent item of DB, all global frequent itemsets compose to EDB. All global frequent itemsets in db compose to Edb.
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 \( X.s_{DB} \geq (D_1 + D_2 + ... + D_n) \times \text{min sup} \). According to the Pigeonhole principle, there is at least one local database \( DB_i \), make \( X.s_{DB_i} \geq \text{min sup} \times D_i \), so \( X \) is a local frequent itemsets of DB, 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 \( Y \in F_{DB \cup db} \), 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 item of DB, but also a local frequent items of DB, if also a global frequent item of db, \( X=(\{x\}) \), that \( X.s_{DB} \geq \text{min sup} \times D \) and \( X.s_{db} \geq \text{min sup} \times d \), therefore \( X.s_{DB \cup db} = X.s_{DB} + X.s_{db} \geq \text{min sup} \times (D + d) \), 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.

Algorithm MAFIM:

Input: The local transaction database \( DB_i \), which has \( M_i \) tuples and \( M = \sum M_i \), \( n \) nodes \( P_i (i=1,2,...,n) \), the center node \( P_0 \), the minimum support threshold \( \text{min sup} \).

Output: The global frequent itemsets \( F \).

Methods: According to the following steps.

step1./* the data were distributed by mapreduce*/

\[ for(i=1;i<=n;i++) \]

\[ P_0 \text{ transmits } DB_i \text{ to } P_i; \]

Step2./*local frequent itemsets were computed by FP-tree*/

\[ for(i=1;i<=n;i++) \]

\[ \{ \text{ creating the FP-tree}; \]

\[ F_i = \text{ FP-growth}(\text{FP-tree}), \text{ null}; \]

\[ \} \]

Step3./* the mining results were combined by the center node*/

\[ for(i=1;i<=n;i++) \]

\[ P_i \text{ sends } F_i \text{ to } P_0; \]

\[ P_0 \text{ combines } F_i \text{ and produces } F'; \]

\[ F' = \bigcup_{i=1}^{n} F_i; \]

Step4./*global frequent itemsets were computed*/

for each items \( d \) in the remain of \( F' \)

\[ d.s = \sum_{i=1}^{n} d.s_i; \]

step5./*global frequent itemsets were got by mapreduce and the search strategy*/

for each items \( d \) in the remain of \( F' \)

if \( (d.s=\text{min sup} \times M) \)

\[ F = F \cup d; \]

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 \( \text{min sup}=0.42 \).

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>LOCAL DATABASE DB1, DB2, DB3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local database</td>
<td>ID</td>
</tr>
<tr>
<td>DB1</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>102</td>
</tr>
<tr>
<td>DB2</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>201</td>
</tr>
<tr>
<td></td>
<td>202</td>
</tr>
<tr>
<td>DB3</td>
<td>300</td>
</tr>
<tr>
<td></td>
<td>301</td>
</tr>
<tr>
<td></td>
<td>302</td>
</tr>
</tbody>
</table>

According to table 1 and \( \text{min sup}=0.42 \), can draw the global frequent items. in accordance with the degree of support in descending order, as shown in Table II.
The global frequent itemset composed of \(E = \{c, b, f, q, a, m, k\}\).

The search strategy implementation process as shown in table III. \(\text{Min}_\text{sup} = 0.42\).
REFERENCES


