Update Mining Algorithm of Global Maximum Frequent Itemsets in Big Data Environment

Bo He, Xiao Wang

School of Computer Science and Engineering, ChongQing University of Technology, 400054 ChongQing, China.

29659807@qq.com, 52980551@qq.com

Abstract

Update mining algorithm of global maximum frequent itemsets in big data environment was proposed, namely, IMAGMFI algorithm. Firstly, the global frequent items were gained. Secondly, the FP-tree was reconstructed. Thirdly, the global maximum frequent itemsets were mined in big data environment. The experimental results show that IMAGMFI algorithm is fast and effective.

Keywords

FP-tree; Global Maximum Frequent Itemset; Update Mining Algorithm.

1. Introduction

Data mining [1] is used to find a novel, effective, useful and understandable knowledge from the data set. The main research directions of data mining includes association rules, classification, clustering and so on. The key step of association rules is to get frequent itemsets from data set, and all frequent itemsets are subsets of maximal frequent itemsets. Therefore, all frequent itemsets can be found by mining maximal frequent itemsets.

At present, the maximum frequent itemsets mining algorithm of single machine database have Discover Maximum Frequent Itemsets algorithm (DMFIA algorithm) [2] and Mining Algorithm for Constrained Maximum Frequent Itemsets (CMFIMA algorithm) [3] and so on.

The maximum frequent itemsets updating algorithm of single machine database have Fast Updating Maximum Frequent Itemsets Algorithm (FUMFIA algorithm) [4] and Updating Maximum Frequent Itemsets Algorithm (UMFIA algorithm) and so on.

Mining algorithm of global maximal frequent itemsets in distributed database have Fast Mining global maximum frequent itemsets (FMGMFI algorithm) [5] and Mining Global Maximum Frequent Itemsets (MGMF algorithm) [6] and so on.

Update algorithm of global frequent itemsets in distributed database have Fast Updating Algorithm for Globally Frequent Itemsets (FUAGFI algorithm) [7] and Updating Algorithm of Global Frequent Itemsets (UGF algorithm) [8] and so on.

With the increasing of distributed database records, some new global maximum frequent itemsets will be generated, Some old global maximum frequent items will be eliminated. Therefore, it is necessary to proceed incremental update of mining global maximum frequent item sets. Up to now, the research results of incremental updating of the global maximum frequent itemsets are very few. Therefore, on the basis of Fast Mining Global Maximum Frequent Itemsets (FMMFI algorithm) [9] which I have been put forward, further put forward Update mining algorithm of global maximum frequent itemsets in big data environment (IMAGMFI algorithm).

2. Related description

2.1 Relevant Definition

Definition 1: For an item set X, Local database DBi(i=1,2,…,n) includes X's transaction number, called local frequency of X in DBi, use X.siDB as the symbol. The local frequency of X in dbi was X.sidb.

Definition 2: For an item set X, Global transaction database DB includes X's transaction number, called global frequency of X in DB, use X.sDB as the symbol. The global frequency of X in db was X.sdb.

2.2 Relevant Theorem

Theorem 1: The global maximum frequent item sets of global transaction database DB and global increment transaction database A are respectively DB and B

The global maximum frequent item sets of global transaction database DB and global increment transaction database db are FMDB and FMdb respectively, the global maximum frequent item set of DB ∪db is FMDB U db, for any set of X∈FMDB U db, both have item set Y∈FMDB∪FMdb, promote X ⊆Y.

Theorem 2: EDB is the global frequent item of DB which according to the support component in descending order, Edb is the global frequent item of db which according to the support component in descending order, all items in EDB∩Edb are global frequent items in DB∪db.

3. An incremental mining algorithm for global maximum frequent itemsets

3.1 IMAGMFI Algorithm Thought

Global transaction database is DB, global incremental transaction database is db. DB's transctions number D is much greater than the number of transctions d of db. An incremental mining algorithm for global maximum frequent itemsets IMAGMFI need to use EDB and FMDB which FMMFI algorithm have been mined. Mined out EDB U db and FMDB U db from whole transction datebase DB ∪db. IMAGMFI algorithm need to mined global frequent item EDB U db from DB∪db. Firstly, mined out global frequent item Edb from db, accroding to support descending to sort Edb; Secondly, Edb was compared with B which have been mined out and used support descending to sort, if Edb is the same as EDB, according to theorem 2, that EDB U db is the same as EDB; Finally, if Edb and EDB are not the same, according to theorem 2, that Edb∩EDB must be global frequent item of DB∪ db, and collected all items support from db which x∈EDB and x∉Edb and collected all items support from DB which y∈Edb and y∉EDB, according to the calculation results, it is obtained EDB U db.

If EDB U db is the same as EDB, then each code Pi needn't to adjust FP-treeDB_i, the original FMDB unchanged. If EDB U db and EDB are not the same, each code Pi need to ajust FP-treeDB_i based on EDB U db, because DB and the support are both not change, that the original FMDB don't changed.

Construct FP-treedb_i based on EDB U db, mined out the global maximum frequent itemsets FMdb from db. According to theorem 1, it taked FMDB∪FMdb as candidate itemsets FMHDB U db, and used top-down purning strategy to mined out FMDB U db.

The top-down pruning strategy is described as follows:

Find out the maximum item k of all itemsets from candidate itemsets FMHDB U db, then turning to (2);

Collect all global frequency of k- itemsets from each code, then turning to (3);

Judge all k-itemsets in FMHDB U db, if k-itemsets Q was global frequent itemsets, taking Q joined into DB ∪db of the global maximum frequent itemsets, and deleted Q and its all nonempty subset from FMDB U db; Otherwise delete Q from FMHDB U db, and take all k-item subsets of Q joined into FMHDB U db, then turning to (1).

IMAGMFI algorithm used EDB and FMDB which have already mined out, it just need to mined Edb and FMdb to mine FMDB U db. Because DB's transctions D is far greater than db's transactions d, which was cost-effective to mined Edb and FMdb, IMAGMFI algorithm have higher mining efficient.

3.2 IMAGMFI Algorithm Description

IMAGMFI algorithm steps are as follows:

Mine global frequent itemsets Edb of db, and according to EDB which have already mined out, mined out global frequent itemsets EDB U db of DB ∪db.

Compare EDB U db with EDB. If they are not the same, that each code Pi need to adjust FP-treeDB_i based on EDB U db; If they are the same, that needn't to adjust FP-treeDB_i.

Construct FP-treedb_i based on EDB U db, mined out the global maximum frequent itemsets FMdb from db.

Take FMDB∪FMdb as candidate itemsets, use top-down pruning strategy to mined out FMDB U db. Then merge FP-treeDB_i and FP-treedb_i.

IMAGMFI as shown in algorithm 1.

3.3 Algorithm 1: IMAGMFI algorithm

Input: Global transactions database DB, transactions D. Global increment transaction database db, transaction d. dbi (i=1,2,…,n) as local increment transaction which db stored in Pi , transaction di, then $db = \bigcup_{i=1}^{n} db_i, d = \sum_{i=1}^{n} d_i$. P0 as center code, Minimum support threshold minsup. DB's all global frequent item EDB, DB's all global maximum frequent itemsets FMDB.

Output: DB ∪db's all global frequent item EDB U db and all global maximum frequent itemsets FMDB U db.

Step1. Mine out the global frequent item $E_{DB\ U\ db}$ of DB ∪db .

for(i=1;i<=n;i++)

{Scanning db_i once;

computing local frequency of local items E_{db_i} ;

P_i sends $E_{db_i}$and local frequency of E_{db_i} to P_0;

}

P_0 collects global frequent items E_{db} from E_{db_i};

$E_{db}$is sorted in the order of descending support count; /*According to the global support of frequent items in descending order*/

P_0 sends E_{db} to other nodes P_i; /*Transfer the global frequent items to other nodesP_i */

$E_{DB\ U\ db}=\varnothing$;

if (E_{db} equal E_{DB})

$E_{DB\ U\ db}= E_{DB}$;

else

{ $E_{DB\ U\ db}=E_{db}\cap E_{DB}$; /* $E_{db}\cap E_{DB}$must be the global frequent item of DB ∪db*/

for each item $x\in\{x \mid (x\in E_{DB})\&\& (x\notin E_{db})\}$

{ P_0 collects global frequency $x.s_{db}$ from db;

$x.s_{DB\ U\ db}= x.s_{DB} +x.s_{db}$;

if ($x.s_{DB\ U\ db}$>= minsup*(D+d))

$E_{DB\ U\ db}= E_{DB\ U\ db}\cup x$;

}

for each item $y\in\{y \mid (y\in E_{db})\&\& (y\notin E_{DB})\}$

{ P_0 collects global frequency $y.s_{DB}$ from DB;

$y.s_{DB\ U\ db}= y.s_{DB} +y.s_{db}$;

if ($y.s_{DB\ U\ db}$>= minsup*(D+d))

$E_{DB\ U\ db}= E_{DB\ U\ db}\cup y$;

}

}

Step2. Compare $E_{DB\ U\ db}$ with E_{DB},and decide whether to adjust FP-tree$_{DB_i}$。

if ($E_{DB\ U\ db}$ unequal E_{DB})
 for(i=1;i<=n;i++)
   P_i adjusts FP-tree$_{DB_i}$ based on $E_{DB\ U\ db}$;

Step3. Mining db global maximum frequent itemsetsFM_{db}。

for(i=1;i<=n;i++)
  creating the FP-tree$_{db_i}$; /*Create each node's FP-tree$_{db_i}$*/
for(i=1;i<=n;i++)
  FM_{db_i}=DMFIA(FP-tree$_{db_i}$, minsup);   /*Using DMFIA algorithm for mining global maximum frequent itemsets of each node*/
for(i=1;i<=n;i++)
  P_i sends FM_{db_i} to P_0;
P_0 combines FM_{db_i} and produces FMH_{db} ;
$FM_{db}=\varnothing$; /* FM_{DB} for the global maximum frequent itemsets*/
while ($FMH_{db}\neq\varnothing$)
{P_0 confirms the largest size k of itemsets in FMH_{db}; /*Confirm k is the maximum length of FMH_{db} */
   for all itemsets $Q\in$ k-itemsets in FMH_{db}
  if (Q are not the subset of any itemsets in FM_{db})   /* Any subset of Q is not from FM_{db}*/
    { P_0broadcasts Q;
      for(i=1;i<=n;i++)
      {P_i sends $Q.si_{db}$ to P_0; /* P_i use FP-tree$_{db_i}$ to calculate the local frequency*/
          $Q.s_{db}=\sum_{i=1}^{n} Q.si_{db}$;
    }
      if ($Q.s_{db}$>=minsup*d) /*Q are the global maximum frequent itemsets*/
    { $FM_{db} = FM_{db}\cup\{Q\}$;
P_0 deletes Q and any nonempty subset of Q from FMH_{db};
    }
     else /*Q arenot the global maximum frequent itemsets*/
      { P_0 deletes Q from FMH_{db};
      for all item $x\in Q$
      if ($Q-\{x\}$are not the subset of any itemsets in FM_{db})
            $FMH_{db} = FMH_{db}\cup\{Q-\{x\}\}$;
      }
      }
}

Step4. Using top-down pruning strategy to mine $FM_{DB\ U\ db}$。

$FMH_{DB\ U\ db} = FM_{DB}\cup FM_{db}$;
$FM_{DB\ U\ db}=\varnothing$; /*$FM_{DB\ U\ db}$ deposit the global maximum frequent itemsets of DB U db*/
while ($FMH_{DB\ U\ db}\neq\varnothing$)
{P_0 confirms the largest size k of itemsets in $FMH_{DB\ U\ db}$; /*confirm the maximum length of $FMH_{DB\ U\ db}$ is k*/
   for all itemsets $Q\in$ k-itemsets in $FMH_{DB\ U\ db}$
  if (Q are not the subset of any itemsets in $FM_{DB\ U\ db}$)   /* Q are not any items sets in $FM_{DB\ U\ db}$*/

{ P_0broadcasts Q;
for(i=1;i<=n;i++)
{P_i sends $Q.si_{DB\ U\ db}$ to P_0; /* P_i use FP-tree$_{DB_i}$ and FP-tree$_{db_i}$ to calculate local frequency */
$$Q.s_{DB\ U\ db}=\sum_{i=1}^{n} Q.si_{DB\,U\,db};$$
}
if ($Q.s_{DB\ U\ db}$>=minsup*(D+d)) /*Q are the global maximum frequent itemsets*/
{ $FM_{DB\ U\ db}=FM_{DB\ U\ db}\cup \{Q\}$;
P_0 deletes Q and any nonempty subset of Q from $FMH_{DB\ U\ db}$;
}
else /*Q arenot the global maximum frequent itemsets*/
{ P_0 deletes Q from $FMH_{DB\ U\ db}$;
for all item $x \in Q$
if ($Q-\{x\}$are not the subset of any itemsets in $FM_{DB\ U\ db}$)
$FMH_{DB\ U\ db} = FMH_{DB\ U\ db}\cup \{Q-\{x\}\}$;
}
}
}
for(i=1;i<=n;i++)
P_i combines FP-tree$_{DB_i}$ and FP-tree$_{db_i}$; /*merge FP-tree$_{DB_i}$ and FP-tree$_{db_i}$ */

4. Algorithm related experiment

In order to test the performance of the algorithm, compare IMAGMFI algorithm with FMMFI and FUAGFI algorithm.

FMMFI [9] algorithm is to mine global maximum frequent itemsets, it can update DB ∪db's global frequent itemsets by re-mining. The algorithm constructs local FP-treeDB U db_i for each distributed node Pi, and use DMFIA algorithm to fast mine global maximum frequent itemsets FMDB U db_i. Then, the data aggregation is realized with the center node interaction. Finally get the global maximum frequent itemsets FMDB U db. FMMFI algorithm have a shortcoming which is not using EDB and FMDB that have already mined out and each update is completely mining DB ∪db's global frequent itemsets FMDB U db.

FUAGFI [7] algorithm is to update global frequent itemsets, it can update global maximum frequent itemsets by update global frequent itemsets. The algorithm use constructed each local FP-tree and mined global frequent itemsets, it can reduce the amount of network data communication effectively and improve the global frequent itemsets's update efficiency. FUAGFI algorithm have a shortcoming which is updating to all global frequent itemsets.

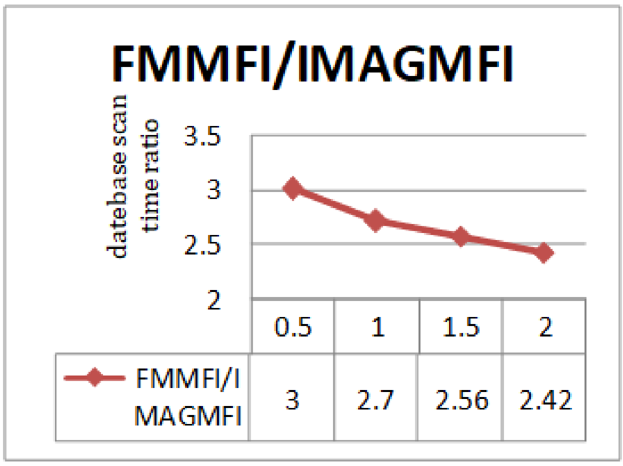

Figure 1. Comparison of database scanning times

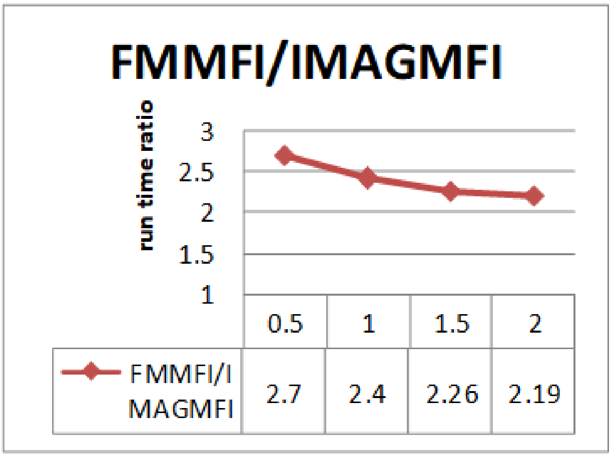

Figure 2. Comparison of runtime

Test environment for a server as the central node and six PC for distributed nodes. Test data taken from a commercial chain store sales data in September 2015. By changing the minimum support, pointing at scanning times and running time for database. compared IMAGMFI and FMMFI algorithm. As shown in figure 1 and 2.

5. Conclusion

The experimental results show that the IMAGMFI algorithm has a great advantage compared to the FMMFI algorithm with the same minimum support, and has certain advantages compared with the FUAGFI algorithm. And mined out global maximum frequent itemsets FMDB U db from DB ∪db by using top-down pruning strategy. The experimental results show that the IMAGMFI algorithm is more efficient.

Acknowledgments

This research is supported by the research fund for humanities and social sciences of the ministry of education under grant No. 19XJA910001.

References

[1] Chen Zhibo, Han Hui, Wang Jianxin. Data Warehouse and Data Mining[M]. Beijing: Tsinghua University Press, 2009.

[2] Song Yuqing, Zhu Yuquan, Sun Zhihui, Chen Geng. An Algorithm and Its Updating Algorithm Based on FP-Tree for Mining Maximum Frequent Itemsets [J]. Journal of Software, 2003, 14(9): 1586-1592.

[3] Song Yuqing, Zhu Yuquan, Sun Zhihui, Yang He biao. An Algorithm and Its Updating Algorithm Based on Frequent Pattern Tree for Mining Constrained Maximum Frequent Itemsets [J]. Journal of Computer Research and Development, 2005,42(5):777-782.

[4] Gi Genlin, Yang Ming, Song Yuqing, Sun Zhihui. Fast Updating Maximum Frequent Itemsets[J]. Journal of Computers, 2005, 28(1):128-135.

[5] Lu Jieping, Yang Ming, Sun Zhihui, Ju Shiguang. Fast Mining of Global Maximum Frequent Itemsets [J]. Journal of Software, 2005, 16(4):553-560.

[6] Wang Liming, Zhao Hui. Algorithms of Mining Global Maximum Frequent Itemsets Based on FP-Tree [J]. Journal of Computer Research and Development, 2007,44(3): 445-451

[7] Yang Ming, Sun Zhihui, Song Yuqing. Fast Updating of Globally Frequent Itemsets [J]. Journal of Software, 2004, 15(8):1189-1197.

[8] Gi Genlin, Yang Ming, Zhao Bin, Sun Zhihui. Updating Technique for Frequent Itemsets in Distributed Database Systems Based on DDMINER[J]. Journal of Computers, 2003, 26(10): 1387-1392.

[9] He Bo. Fast Mining of Global Maximum Frequent Itemsets in Distributed Database [J]. Controland Decision, 2011, 26(8):1214-1218.