文章编号:1001 - 9081(2006)02 - 0391 - 02

快速挖掘频繁项集的并行算法

何 波,王华秋,刘 贞,王 越
(重庆工学院 计算机科学与工程学院, 重庆 400050)
(hebo@cqit.edu.cn)

摘 要:传统的挖掘频繁项集的并行算法存在数据偏移、通信量大、同步次数较多和扫描数据库次数较多等问题。针对这些问题,提出了一种快速挖掘频繁项集的并行算法(FPMFI)。FPMFI 算法让各计算机节点独立地计算局部频繁项集,然后与中心节点交互实现数据汇总,最终获得全局频繁项集。理论分析和实验结果表明 FPMFI 算法是有效的。

关键词:数据挖掘;并行算法;频繁项集
中图分类号: TP311 **文献标识码**:A

Fast and parallel algorithm for mining frequent itemsets

HE Bo, WANG Hua-qiu, LIU Zhen, WANG Yue
(*Department of Computer Science and Engineering, Chongqing Institute of Technology, Chongqing* 400050, *China*)

Abstract: There were problems in traditional parallel algorithms for mining frequent itemsets more or less: data deviation, large scale communication, frequent synchronization and scanning database. Aiming at these problems, this paper proposed a fast and parallel algorithm for mining frequent itemsets(FPMFI). It made computer nodes compute local frequent itemsets independently, then center node exchanged data with other computer nodes and combined, finally, a global frequent itemsets resulted. Theoretical analysis and experimental results suggest that FPMFI is effctive.

Key words: data mining; parallel algorithm; frequent itemsets

0 引言

挖掘关联规则问题的核心是发现频繁项集[1]。现今已有多种发现频繁项集的串行算法,如 Apriori[4]、PARTITION 及抽样算法等。但是,现在进行频繁项集挖掘的数据库往往很大(按 GB 乃至 TB 计),而且采用传统的串行算法所需的时间和空间开销也很大,效率较低。为了提高挖掘频繁项集的效率,研究人员提出了并行挖掘算法,主要包括 CD(Count Distribution)、DD(Data Distribution)[5]、CaD(Candidate Distribution)和 FDM 等。这些并行挖掘算法各有优点,但是仍然存在数据偏移、通信量大、同步次数较多和扫描数据库次数较多等问题。针对这些问题,论文提出了一种快速挖掘频繁项集的并行算法(FPMFI)。

1 相关定义和定理

1.1 并行挖掘频繁项集的问题描述

全局事务数据库为 DB,总的事务条数为 D。设 P_1、P_2、…、P_n 为 n 台基于无共享体系结构的计算机节点(简称节点),即它们之间除了通过网络传递信息外,其他资源(如硬盘、内存等)全部都是独立的,$DB_i(i = 1,2,\cdots,n)$ 是 DB 经过分割存储于节点 P_i 上的分事务数据库,其中的事务有 D_i 条,则 $DB = \bigcup_{i=1}^{n} DB_i$,$D = \sum_{i=1}^{n} D_i$。

并行挖掘频繁项集问题就是如何通过 n 台节点同时工作,节点 $P_i(i = 1,2,\cdots,n)$ 只处理自己的私有数据 $DB_i(i = 1,2,\cdots,n)$,各台节点间仅仅通过网络传递有限的信息,最终在整个事务数据库 DB 中挖掘出频繁项集。

1.2 相关定义

定义 1 对于某一项集 X,在局部数据库 $DB_i(i = 1,2,\cdots,n)$ 中包含 X 的事物的条数,称为 X 的局部频度,用 $X.si$ 表示。

定义 2 对于某一项集 X,在全局事务数据库 DB 中包含 X 的事物的条数,称 X 为的全局频度,用 $X.s$ 表示。

定义 3 对于项集 X,若 $X.si \geqslant min\_sup \times D_i(i = 1,2,\cdots,n)$,则称 X 是相对于 D_i 的局部频繁项集 F_i,若 X 中的元素为 k 个,即 $|X| = k$,则称 X 为数据库 DB_i 的局部频繁 k - 项集 F_i^k,其中 min_sup 表示最小支持度阈值。

定义 4 对于项集 X,若 $X.s \geqslant min\_sup \times D$,则称 X 是全局频繁项集 F,简称频繁项集。若 X 中的元素为 k 个,即 $|X| = k$,则 X 称为频繁 k - 项集 F_k。

1.3 相关定理

定理 1 若项集 X 是 DB_i 的局部频繁项集,则 X 的所有非空子集 $Y(Y \subseteq X)$,也是 DB_i 的局部频繁项集。

证明 项集 X 是局部频繁项集,则 $X.si \geqslant min\_sup \times D_i$,设 $Y \subseteq X$,则 $Y.si \geqslant X.si$,故 $Y.si \geqslant X.si \geqslant min\_sup \times D_i$,则 Y 是 DB 的局部频繁项集,证毕。

推论 1 若项集 X 不是局部频繁项集,则 X 的超集一定不是局部频繁项集。

定理 2 若项集 X 为全局频繁项集,则至少存在一个局

收稿日期:2005 - 04 - 06;**修订日期**:2005 - 09 - 21 **基金项目**:重庆市教委应用基础研究项目(020612)

作者简介:何波(1978 -),男,四川华蓥人,硕士,主要研究方向:数据挖掘、智能信息推荐; 王华秋(1975 -),男,重庆人,讲师,博士研究生,主要研究方向:并行计算、数据挖掘、人工智能研究; 刘贞(1972 -),男,重庆人,讲师,硕士,主要研究方向:数据库系统,数据仓库、商业智能; 王越(1962 -),男,四川人,教授,博士,主要研究方向:嵌入式系统,数据挖掘.

部数据库 $DB_i(i=1,2,\cdots,n)$，使 X 及 X 的所有非空子集均为 DB_i 的局部频繁项集。

证明　若将 n 个局部数据库看成 n 个鸽巢，项集 X 在 DB 中出现的次数对应于鸽子的个数，则 $X.s \geqslant (D_1+D_2+\cdots+D_n)\times min\_sup$。根据鸽巢原理可知，至少存在一个局部数据库 DB_i，使得 $X.si \geqslant min\_sup \times D_i$，即 X 在局部数据库 DB_i 是局部频繁项集。由推论 1 可知，X 的所有非空子集在局部数据库 DB_i 上为局部频繁项集，证毕。

定理 3　设 $max\_X.si = \min\{Y.si \mid Y \subset X \text{ and } |Y| = |X|-1\}$，其中 $|X|$ 表示项集 X 中的项目数，则 $X.si \leqslant max\_X.si$。

证明　对于任意 Y，因为满足 $Y \subset X$ 且 $|Y| = |X|-1$，根据集合与子集的关系，必有 $X.si \leqslant Y.si$，所以 $max\_X.si$ 是 X 在局部数据库 DB_i 的局部频度的上界，故：$X.si \leqslant \min\{Y.si \mid Y \subset X \text{ and } |Y| = |X|-1\}$，即：$X.si \leqslant max\_X.si$。

2　FPMFI 算法

2.1　FPMFI 算法设计思想

如果只是将全局事务数据库划分成大小相等的分块，提交给各节点并行挖掘，很可能会造成负载不均，一些节点进行大量的计算，另一些节点处于空闲状态，这就是数据偏移问题。FPMFI 算法采用水平等间距投影方法进行数据分配，将全局事务数据库中的 M 个元组分成 $M_1,M_2,\cdots,M_n$（$\sum_{i=1}^{n} M_i = M$），第 i 个节点上对应的 M_i 个元组集合可表示为 $\{T_i^j \mid T_i^j = O_q \text{ and } q = n\times(j-1)+i\}$，其中 T_i^j 表示第 i 个节点的第 j 个元组，O_q 表示全局事务数据库中的第 q 个元组。经过以上变换，全局事务数据库 DB 被分割为 n 个规模为 $\lfloor \frac{M}{n} \rfloor$ 的局部数据库 $DB_1,DB_2,\cdots,DB_n$；即 $DB = \bigcup_{i=1}^{n} DB_i$。因为 DB_i 等间距获取 DB 中的元组，全局事务数据库比较均衡地划分成各个局部数据库，所以 FPMFI 算法减少了数据偏移的发生。

FPMFI 算法考虑在 n 个节点中选择一个节点 P_a 作为中心节点，各节点向中心节点 P_a 发送其局部频繁项集 F_i，P_a 汇总得到所有节点局部频繁项集的并集 $F'(F' = \bigcup_{i=1}^{n} F_i)$，并向其他节点广播 F'。P_a 从各个节点收集 F' 中所有局部频繁项集 d 的局部频度 $d.si$，最终获得 d 的全局频度 $d.s$。设置中心节点汇总数据避免了一个局部频繁项集同时出现在多个节点而造成的重复计算。

大多数并行挖掘算法需要多次同步，比如，如果频繁项集的最大长度为 k，往往需要 k 次同步，这样造成各节点间的通信量较大，挖掘效率较低。FPMFI 算法让各节点在不知道其他节点信息情况下独立地计算局部频繁项集 F_i，当所有的节点都计算出 F_i 后，再与中心节点交互实现数据汇总，最终获得全局频繁项集 F。FPMFI 算法依据的是定理 2，一个全局频繁项集至少在一个局部数据库中为局部频繁项集，因此各节点局部频繁项集 F_i 的并集一定是全局频繁项集 F 的超集。FPMFI 算法中计算局部频繁项集的工作可以异步执行，只是在最后结束时才进行一次同步，减少了同步次数和各节点间的通信量。

FPMFI 算法中各节点计算局部频繁项集采用类 Apriori 算法。考虑到所有的节点计算出局部频繁项集 F_i 后需要与中心节点交互实现数据汇总，算法将节点的候选项集 L_i 存储在哈希树这种数据结构中。在局部剪枝和数据汇总中所需要的两套不同的支持合计数都可以从此哈希树中获取，这样就不需要在进行数据交换汇总时重新扫描数据库，减少了数据库的扫描次数。依据定理 3，候选项集的局部频度不超过其子集局部频度的最小值，因而当候选项集 L_i 的局部频度超过其子集局部频度的最小值，就可以停止搜索，这样通过候选项集局部剪枝，减少了数据库的扫描次数。

2.2　FPMFI 算法描述

FPMFI 算法步骤如下：

1）采用水平等间距投影方法进行数据分配，将全局事务数据库 DB 分割为 n 个规模为 $\lfloor \frac{M}{n} \rfloor$ 的局部数据库 $DB_1,DB_2,\cdots,DB_n$；

2）各节点在不知道其他节点信息情况下独立地计算局部频繁项集 F_i。计算局部频繁项集时将进行局部剪枝，各节点的候选项集 L_i 存储在哈希树中，最终可计算出该节点的局部频繁项集 F_i；

3）各节点向中心节点 P_a 发送其局部频繁项集 F_i，使得 P_a 拥有所有节点局部频繁项集的并集 $F'(F' = \bigcup_{i=1}^{n} F_i)$，$F'$ 是全局频繁项集 F 的超集，P_a 向其他节点广播 F'；

4）各节点利用哈希树计算 F' 中所有局部频繁项集 d 的局部频度 $d.si$，将其发送给中心节点 P_a，P_a 汇总后最终获得 d 的全局频度 $d.s$；

5）检查各局部频繁项集的全局频度 $d.s$ 是否满足最小支持度阈值 min_sup，得到最终的全局频繁项集 F。

FPMFI 算法描述如算法 1 所示。

算法 1　FPMFI 算法

输入：全局事务数据库为 DB，共有 M 个元组，n 台基于无共享体系结构的计算机节点 $P_i(i=1,2,\cdots,n)$，其中 P_a 作为中心节点，最小支持度阈值 min_sup；

输出：全局频繁项集 F；

1）采用水平等间距投影方法进行数据分配

```
for( q = 1;q < = M;q++)
{if ( q mod n = = i )
 {if ( i = =0)
  i = n ;
  /* 此时第 q 个元组应在第 n 个节点中 */ O_q insert to DB_i ;
  /* O_q 表示 DB 的第 q 个元组;DB_i 表示第 i 个节点的分事务数
  据库 */
  }
 }
for( i = 1;i < = n;i++)
{ DB_i transfer to P_i ;
}
```

2）各节点生成局部频繁项集

```
for( i = 1;i < = n;i++)
{
 F_i^1 = find_frequent_1 - itemsets( DB_i );
 /* F_i^1 表示局部频繁 1 - 项集 */
 for( k = 2; F_i^(k-1) ≠ ∅;k++)
 {L_i^k = apriori_gen( F_i^(k-1),min_sup );
  /* L_i^k 表示局部候选 k - 项集 */
  hash_tree( L_i^k );
  /* 用哈希树存储候选项集 L_i^k */
  for each item c ∈ L_i^k
  {if ( c.si ≤ max_c.si )&&( c.si ≥ min_sup × D_i )
   /* c.si 表示项集 c 在 DB_i 的局部频度,利用哈希树计算;D_i
   表示的 DB_i 的元组数 */
```

（下转第 402 页）

4 实验结果与讨论

实验中选取一张培养细胞灰度图像，大小为256 × 256，小波变换选取的是紧支集的二次 B 样条小波。

从图 1 中可以看出 canny 算法边缘太细，有些噪声；尺度独立算法连续性不太好，有边缘丢失；聚类算子介于两者之间，与分类数有很大关系；而采用本文算法能有效弥补三种方法的不足，又保持了它们的优点，取得了良好的折衷。边缘图像具有很好的清晰度和连续性，边缘信息比较完备，基本分割出了细胞的大致轮廓。

表 1 细胞边缘分割点的方法比较

方法	正确边缘点	误判点	正确率
Canny	14534	2346	83.8%
聚类	13589	3321	75.6%
尺度独立	13986	2312	83.4%
融合综合	15123	2190	85.5%

从表 1 中可以看出，本文方法对细胞边缘点的检测上具有良好的正确率。本文提出的算法对于组织细胞的动态检测分割具有良好的效果，为下一步的参数测定和观察打下良好的基础。同时，该方法也可以应用到其他类型图像的检测分割。

参考文献：

[1] CANNY J. A Computational Approach to Edge Detection. IEEE T-PAM I[J]. 1986, 8(6): 679 – 698.

[2] MALLAT SG, ZHONG S. Charaterization of Signals from Multiscale Edges. IEEE T-PAM I[J]. 1992, 14(7): 710 – 732.

[3] ER PIERRE K, MARC LJ, SAINT JP, *et al*. Wavelet based multifractal formalism to assist in diagnosis in digitized mammograms. Image Anal Stereo l[J]. 2001, 20 (3) : 169 – 174.

[4] IEL FERNA′NDEZ G, BERGER TH. Waveletbased system for recognition and a beling of polyhedral junctions. Optical Engineering [J]. 1998, 37 (1) : 158 – 165.

[5] 周敏，龙昭华. 基于 MAS 小波变换的数字图像轮廓提取算法[J]. 重庆邮电学院学报, 2004, 16(2): 44 – 48.

（上接第 392 页）

```
      F_i^k = F_i^k ∪ c;
    else
      delete c from L_i^k;                    /* 根据定理 3 局部剪枝 */
    }
  }
}
```

3) 各节点都获取所有节点局部频繁项集的并集

```
for( i = 1; i <= n; i++)
  P_i send F_i to P_a;
  /* F_i 表示 P_i 的局部频繁项集; P_a 表示中心节点 */
  P_a combine F_i equal F';
  /* F' = ∪_{i=1}^{n} F_i，表示所有节点局部频繁项集的并集 */
  P_a broadcast F';
```

4) 计算项集的全局频度

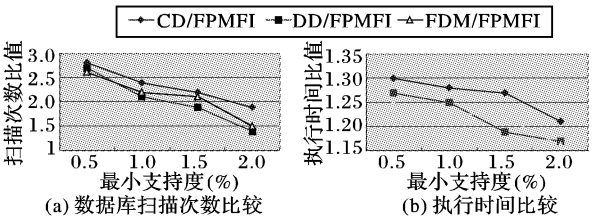
```
for( i = 1; i <= n; i++)
{ for each items d ∈ F'
  P_i send d.si to P_a;                       /* 利用哈希树计算 d.si */
}
for each items d ∈ F'
  d.s = Σ_{i=1}^{n} d.di                      /* d.s 表示项集 d 的全局频度 */
```

5) 获得全局频繁项集 F

```
for each items d ∈ F'
  if ( d.s ≥ min_sup × M )
    /* M 表示全局数据库 DB 的元组数 */
    F = F ∪ d
```

3 算法分析与性能测试

为了测试算法的性能，将 FPMFI 算法与经典的挖掘频繁项集的并行算法 CD、DD 和 FDM 算法进行性能比较。测试环境为 10M 局域网，采用 5 台联想 PC 机构成分布式计算机节点，其中 1 台作为中心节点，各 PC 机配置均为 P4 2.4G，内存 512M，Windows2000 professional 操作系统，SQL Server 2000 数据库管理系统。实验数据来自某大型商业连锁店的销售数据。在测试中，采用水平等间距投影方法分别在 5 台 PC 生成测试数据库，每台 PC 为 5000 条数据。测试程序的编程语言为 VC++ 6.0，消息传递库为标准 MPI，测试结果如图 1 所示。



(a) 数据库扫描次数比较　(b) 执行时间比较

图 1 测试结果

由测试的结果可知，在相同支持度下，与 CD、DD 和 FDM 等并行挖掘算法相比，数据库扫描次数和执行时间都降低了，而且随着支持度的下降，FPMFI 算法性能优势更加明显。

4 结语

FPMFI 算法采用水平等间距投影方法进行数据分配，减少了数据偏移现象。算法让各计算机节点独立地计算局部频繁项集，再与中心节点交互实现数据交换及汇总，最终获得全局频繁项集，减少了同步次数、各节点间的通信量以及数据库的扫描次数。

参考文献：

[1] 杨明，孙志挥，吉根林. 快速挖掘全局频繁项目集[J]. 计算机研究与发展, 2003, 40(4): 620 – 626.

[2] PARTHASARATHY S, ZAKI MJ, OGIHARA M. Parallel data mining for association rules on shared – memory systems[J]. Knowledge and Information Systems, 2001, 3(1): 1129.

[3] ZAKI MJ. Parallel and distributed association mining: A survey[J]. IEEE Concurrency, Special Issue on Parallel Mechanisms for Data Mining, 1999, 7(4): 14 – 25.

[4] HAN J, KAMBER M. Data Mining: Concepts and Techniques[M]. Beijing: High Educattion Press, 2001.

[5] 李航，刘宗田，陈惠琼. 挖掘关联规则的并行算法[J]. 小型微型计算机系统, 2002, 23(10): 1231 – 1234.