doi:10.12068/j.issn.1005－3026.2025.20240015

一种高效的分布式FDR假阳性控制算法

刘旭泽[1]，王慧颖[2]，褚良宇[3]，赵宇海[1]
（1.东北大学 计算机科学与工程学院，辽宁 沈阳 110819；2.国家电网辽宁省电力有限公司 信息通信分公司，辽宁 沈阳 110065；3.东北大学 医学与生物信息工程学院，辽宁 沈阳 110819）

摘 要：为了解决大数据挖掘中多重假设检验导致的假阳性问题，以及控制伪发现率(false discovery rate，FDR)理论结果计算过程极其耗时的问题，针对理论FDR值的计算效率问题，提出了一种分布式假阳性控制算法DPFDR(distributed permutation testing-based false discovery rat，DPFDR). 该算法首先基于条件频繁模式树(conditional frequent pattern tree，CFP)方法进行代表模式挖掘，利用代表模式对模式空间进行压缩. 然后，根据代表模式对相应任务的工作量进行预估，按照工作量进行数据划分，并通过负载均衡策略将任务分配到各计算结点上. 最后，通过合并、排序各结点的计算结果，获得有效的FDR假阳性控制阈值. 真实数据集上的一系列实验结果表明，提出的DPFDR算法能极大提升FDR假阳性控制阈值的计算效率.

关 键 词：假阳性；数据挖掘；分布式计算；伪发现率；显著性阈值
中图分类号：TP 311 **文献标志码**：A **文章编号**：1005-3026(2025)05-0037-09

An Efficient Distributed False Positive Control Algorithm for FDR

LIU Xu-ze[1]，*WANG Hui-ying*[2]，*CHU Liang-yu*[3]，*ZHAO Yu-hai*[1]
(1. School of Computer Science & Engineering, Northeastern University, Shenyang 110819, China; 2. Information and Communication Branch of Liaoning Electric Power Company, State Grid, Shenyang 110065, China; 3. School of Medicine & Bioinformatics Engineering, Northeastern University, Shenyang 110819, China. Corresponding author: ZHAO Yu-hai, E-mail: zhaoyuhai@mail.neu.edu.cn)

Abstract: To address the issue of false positives caused by multiple hypothesis testing in big data mining, as well as the extremely time-consuming nature of calculating theoretical results for controlling the false discovery rate (FDR). Aiming at the computational efficiency of theoretical FDR values, a distributed false-positive control algorithm based on DPFDR (distributed permutation testing-based false discovery rate) is proposed. The algorithm firstly mining the representative patterns based on the conditional frequent pattern tree (CFP) method, and using the representative patterns to compress the pattern space. Then, the workload of the corresponding task is estimated according to the representative mode, the data is divided according to the workload, and the task is allocated to each compute node through the load balancing policy. Finally, the effective FDR false-positive control threshold is obtained by merging and sorting the calculation results of each node. A series of experimental results on real data sets show that the proposed DPFDR algorithm can greatly improve the computational efficiency of FDR false positive control threshold.

Key words: false positive; data mining; distributed computing; false discovery rate; significance threshold

随着科技的进步，在当今社会的各行业中都出现了大量数据.仅使用单一的假设检验已无法满足研究人员的需求，多重假设检验越来越受到研究人员的喜爱.例如，在流行病研究领域、基因

收稿日期：2024-01-17
基金项目：国家自然科学基金资助项目(62432003).
作者简介：刘旭泽(1992—)，男，辽宁盘锦人，东北大学博士研究生；赵宇海(1975—)，男，辽宁辽阳人，东北大学教授，博士生导师.

表达数据分析以及单核苷酸多态性数据分析等领域,都涉及到大规模数据的统计分析.在分析计算的过程中往往又需要同时检验多个判断,这种情况就是典型的多重假设检验问题[1].在多重假设检验中,目前需要解决的一大难题是处理海量数据的假阳性控制问题.

在对判断进行检验的过程中不可避免的会出现两类错误.第Ⅰ类错误指的是零假设H_0正确时,经过计算后得到的结果显示H_0是错误的,就拒绝了H_0造成的错误.第Ⅱ类错误指的是在真实的零假设H_0不正确时,计算后得到的结果却显示H_0是正确的,导致研究人员错误的接受了H_0.假阳性错误是指统计假设检验中的第Ⅰ类错误[2-3],研究人员在验证判断时希望第Ⅰ类错误和第Ⅱ类错误都尽可能小[4].但当样本大小一定时,两类错误相互制约,因此想要同时使两类错误达到最小是不可能的.研究人员发现在科学研究和日常的生产生活中出现假阳性错误导致的危害非常大,因为它向相关的研究人员报告了原本就不存在的现象.基于假阳性结果进行后续的研究和应用将会造成难以估计的危害.所以将出现假阳性错误的概率控制在一个很小的阈值内的同时,使得犯第Ⅱ类错误的概率尽可能小是目前研究的焦点.

目前,多重假设检验已广泛应用于各个领域,但是现阶段提出的各种多重假设检验假阳性控制算法都是基于小数据集的单机算法.然而,随着数据量的飞速增长,使用单机进行多重假设检验假阳性控制,会出现数据不能完全存入内存或单机内存无法承担计算过程中出现的大量结果.因此,现有单机假阳性控制方法已经不足以满足人们的需求.在此背景下,涌现了很多分布式计算框架,如由Apache基金会[5]开发的分布式基础框架Hadoop,在Hadoop之上的计算模型MapReduce以及一种基于内存的大数据处理引擎Spark等[6]都是为了解决海量数据存储和海量数据分析提出的分布式计算框架.在大规模数据的情况下设计高效的分布式假阳性控制算法有利于满足当下各个领域的使用需求,并且会创造巨大的商业价值.从市场营销到医疗保健,多重假设检验假阳性控制在许多应用中都非常重要.例如,频繁项目集挖掘试图找到所有客户共同购买的产品,而显著模式挖掘试图检测老年客户比年轻客户更经常共同购买的产品,在此过程中,降低挖掘结果错误数量,提高挖掘结果准确程度是人们关注的重点之一.

本文主要针对多重假设检验假阳性控制的研究,从提高大规模数据假阳性控制计算效率的角度,提出一种面向大数据挖掘的分布式FDR假阳性控制算法.针对单机情况下大规模数据探索性实验的多重假设检验假阳性控制计算速度慢甚至无法计算等问题,采用模式压缩的方法,利用CFP树结构挖掘代表模式集进行任务量预估.根据预估的计算量,采用负载均衡策略进行数据分区,使用Spark框架并行地进行FDR假阳性控制计算.同时,使用BH过程进行FDR假阳性控制计算.

1 假阳性控制相关研究

对假阳性进行控制主要目的是对多重假设检验进行校正,以减少多重假设检验中出现错误的情况,在科研领域以及实际生产生活中有广泛的应用.假阳性控制一直是研究的热点.

对于多重假设检验及其假阳性控制的研究已经有很多年,传统的假阳性控制方法是簇错误率(family-wise-error,FWER)方法.FWER控制方法是将需要同时验证的多个检验看做一个整体的检验簇[7-8],将这个检验簇中出现一个假阳性错误的概率控制在α水平下,这样就极大降低了在多重假设检验中控制假阳性错误的难度.早在1935年,Bonferroni就提出了基于FWER的假阳性控制方法——Bonferroni校正[9].Bonferroni校正通过将在一组数据上检验的n个独立假设中,每个假设所使用的统计显著性水平设置为测试单个假设时(通常检验单个假设时所使用的统计显著水平为α)的$1/n$,来控制多重假设检验中总体出现假阳性错误的数量.但是,当提出需要同时检验的假设数量n非常大的情况下,则α/n无限接近于0.显然,根据Bonferroni校正确定的新的统计显著水平对于多重假设检验来说过于严格,实用价值相对较低.Holm对Bonferroni校正进行了改进,提出了SRB(sequentially rejective Bonferroni)算法[10].SRB算法首先将n个检验所对应的p值从小到大排序,用$p_1,p_2,\cdots,p_n$表示.接下来依次进行$p_1\leqslant\alpha/n,\cdots,p_i\leqslant\alpha/(n-i+1)$的比较.如果$p_1\leqslant\alpha/n$成立就认为$p_1$对应的检验在总体显著水平控制在$\alpha$的情况下是显著的,并拒绝与之对应的假设.直到$p_i\leqslant\alpha/(n-i+1)$不成立,则认为$p_i$及以后所对应的检验都不显著.这样就可以在连续更高的显

著水平 $\alpha/(n-i+1)$ 下对假阳性错误进行控制,该方法明显优于Bonferroni校正法,后续被各个领域的研究人员所广泛使用.但是SRB算法依旧过于保守,检验效果不是十分理想.Simes[11]提出了Simes算法,该算法在SRB算法的基础上进行改进,提出了一种新的用于控制FWER的思路,同样地,对 p 值从小到大进行排序,从 $i=1$ 开始找到最大的 i 使得 $p_i \leqslant \alpha/(n-i+1)$ 成立,则拒绝假设 $H_1,\cdots,H_i$. Simes算法较SRB算法有更大的功效,但是当出现假阳性错误对应的 p 值很小的情况下,Simes算法并不能保证FWER $\leqslant \alpha$,Simes算法是一种弱FWER控制方法.Hochberg[12]将Simes与Holm假阳性控制算法结合起来,从最不显著的 p 值 p_n 开始检验,直到找到第1个满足 $p_i \leqslant \alpha/(n-i+1)$ 的 i,拒绝所有 $p_1,p_2,\cdots,p_i$ 对应的零假设,该方法避免了Simes算法出现的问题,并且比SRB算法更简洁.Chaubey等[13]提出了一种在检验统计量相依的情况下调整 p 值的假阳性控制方法,该算法可以将总体出现假阳性的概率控制在 α 水平下,但是该算法实现过程中涉及大量的重抽样和置换,因此它的计算速度相对较慢.

上述通过传统控制FWER的方法来控制多个同时检验中假阳性错误的数量过于保守,检验功效较低.伪发现率(false discovery rate,FDR)[14-15]表示假阳性错误在所有被拒绝假设中所占比例.当所有的零假设都为真的情况下,FDR与FWER对于假阳性的控制效果相同.FDR控制[16]在提高功效的同时,没有FWER控制那么保守,但控制FDR时并不能保证FWER得到有效的控制.Benjamini等[14]在提出FDR概念的同时提出了一种基于FDR控制的算法——BH过程.BH过程是Simes算法适用于FDR的一种改进,对 p 值从小到大排序后进行计算,如果 $p_i \leqslant i\alpha/n$, $i=n,\cdots,1$ 成立,就拒绝 $H_1,\cdots,H_i$,否则不拒绝 H_i. Liu等[17]提出了基于排列的多重测试校正方法PBA(permutation-based approach)算法,Pellizzoni等[18]提出了基于Yekutieli-Benjamini重采样过程的基于置换检验的FDR控制显著模式挖掘算法FAST-YB(Yekutieli-Benjamini),算法通过随机排列类标签破坏事务与类标签之间的关联.因此重新计算的 p 值的分布是零分布的近似值,这样可以更准确找到 p 值的截断阈值(校正后的显著性阈值).

从假设检验的角度进一步进行问题分析后,本文将使用两类标签 W_1, W_0 来表示参数的"范围",由于"假设"是对于真实参数所属范围的一种虚拟认定,那么零假设 H_0 就可以看作是真实参数属于 W_1 这个标签的范围,备择假设 H_1 就可以看作真实参数属于 W_0 这个标签的范围.本文将选取事务数据集作为真实参数,显然零假设 H_0 就变为事务 T_i 属于标签 W_1. 令 S_i 是事务 T_i 中包含的项集,那么如果一个事务 T_i 中包含项集 S_i,并且该事务的标签是 W_1,就可以确定一种规则 $L:S_i \rightarrow W_1$,这就变成了一种在关联规则挖掘中多重假设检验的假阳性控制问题.

2　分布式FDR假阳性控制算法

由于通过传统控制FWER的方法来控制多个同时检验中假阳性错误的数量比较保守,检验功效较低.在面对探索性实验的多重检测问题时首选的是FDR假阳性控制算法.本文提出一种分布式假阳性控制算法(distributed permutation testing-based FDR,DPFDR),通过负载均衡的策略来提高FDR分布式假阳性控制算法计算效率.利用代表模式对模式压缩后预估工作量,按照工作量进行数据分区来避免计算过程中出现数据倾斜导致计算速度变慢的情况.

2.1　基本概念与研究思路

假设有 m 个感兴趣的需要检验的零假设 H_{0i},其中 $i=1,\cdots,m$. 若假设 m_j 是正确的零假设,但是它的数量和内容都是未知的,则存在 $m-m_j$ 个错误的零假设.设 c_i 表示一个零假设 H_{0i} 是否为真,若 $c_i=0$ 则表示零假设 H_{0i} 为真,若 $c_i=1$ 则表示零假设 H_{0i} 为假.将零假设 H_{0i}, $i=1,\cdots,m$ 通过计算求得的 p 值用 p_i 表示,令 X_i 为检验统计量则有 $p_i=1-F_{H_{0i}}(X_i)$,其中 $F_{H_{0i}}(X_i)$ 是 X_i 在 H_{0i} 下的分布函数.并将零假设对应的 p 值按由小到大的顺序进行排序可以得到排序后的 p 值为 $p(1)\leqslant p(2)\leqslant\cdots\leqslant p(m)$. 根据表1多重假设检验结果可以有如下定义.

表1　多重假设检验结果

Table 1　Results of multiple hypothesis testing

项目	不拒绝 H_0	拒绝 H_0	总计
H_0 为真	U	V	n_0
H_0 为假	T	S	$n-n_0$
总计	$n-R$	R	n

定义1 $V_m(p)$:令p是预先指定的值,$I_{[\cdot]}(x)$为关于[·]的指标函数,$I_{[0,p]}(p_i)$表示p_i在[0,p]之间的假设的数量,$V_m(p)$为设置显著级别为p时犯假阳性错误的数目,即错误地接受需要拒绝的假设的数量.$V_m(p)$如式(1)所示:

$$V_m(p)=\sum_{i=1}^{m}(1-c_i)I_{[0,p]}(p_i). \tag{1}$$

定义2 $R_m(p)$:令p是预先指定的值,$R_m(p)$为拒绝零假设H_{0i}的数量,$R_m(p)$如式(2)所示:

$$R_m(p)=I_{[0,p]}(p_i). \tag{2}$$

定义3 FDR(p):令p是预先指定的值,FDR(p)[19]是伪发现率结果,根据式(1)和式(2),FDR(p)如式(3)所示.

$$\mathrm{FDR}(p)=E\left(V_m(p)/R_m(p)\right). \tag{3}$$

对于式(3)有以下的特性:$R_m(p)\geqslant 0$并且FDR(p)=0时,$R_m(p)=0$.

使用直接调整方法和BH过程[14]来对探索性实验的假阳性进行控制.在实验中发现确定需要被检验的假设的过程中需要寻找事务数据集中包含的所有模式.由于使用BH过程进行假阳性控制需要计算出所有模式的p值,并根据求出的p值集合寻找修正显著阈值.在计算过程中涉及到的计算量十分庞大,因此可以使用一种分布式的策略来提高计算效率,算法的总体框架流程如图1所示.

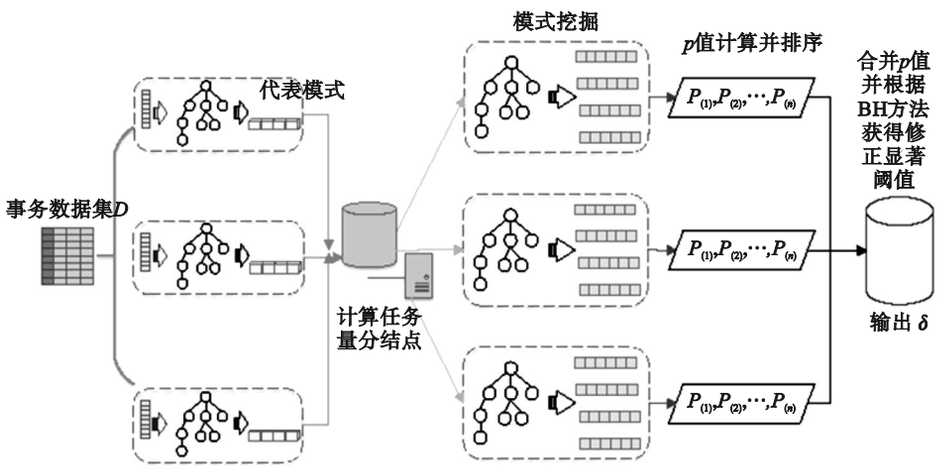

图1 FDR假阳性控制总体框架图

Fig. 1 Overall framework of FDR false positive control

算法步骤如下:

1)数据分区.通过计算频繁1项集中每项的条件模式集,将它们构成一个新的数据集,并在其中寻找具有代表性的模式集来预估算法任务量并对数据进行分区处理,将处理过的数据子集按照其计算量平均分配到集群中的各个结点上进行并计算.

2)假设确定.根据分配到各个结点的代表性模式和处理过的数据子集,构建FP树挖掘需要检验的模式,并结合模式对应的标签确定要检验的假设.

3)p值计算.找到所有要检验的假设$H_1,\cdots,H_m$后,使用Fisher精确检验[20]计算每一个假设所对应p值,并将计算后的p值按从小到大顺序进行排序.

4)计算显著性阈值.将各个结点上计算得到的p值合并,按升序进行排序后使用BH过程进行FDR假阳性控制计算求出最终的显著性阈值.

2.2 任务量预估

使用代表模式[21]来预估FDR假阳性控制算法的工作量.根据预估的分布式FDR假阳性控制算法的整体工作任务量以及集群上结点的数量进行数据的划分,以避免因为数据倾斜造成的部分结点工作量较大、计算时间较长的问题.

任务量预估如图2所示,分为以下4个步骤:

1)构建频繁1项集和FP树.使用事务数据集构建频繁1项集,并根据频繁1项集与事务数据集构建FP树.

2)构建条件模式基.将项头表(根据频繁1项集构建)的每一个结点取出结合FP树求出其条件模式基并将其构建成一个新的条件数据集.

3)挖掘代表模式集.由于使用条件模式基构建的数据集进行模式析出,会析出很多重复模式,造成计算负担.因此对条件模式基构建的数据集进行模式压缩,挖掘出其中的代表模式,构建代表模式集再进行任务量的预估.

4) 预估工作量.根据项头表构建的条件树进行代表模式挖掘可以预估出项头表中每一个结点执行FDR假阳性控制过程确定假设时需要的任务量.将这些结点工作需要的任务量相加就可以预估出算法进行假设确定时的任务量.

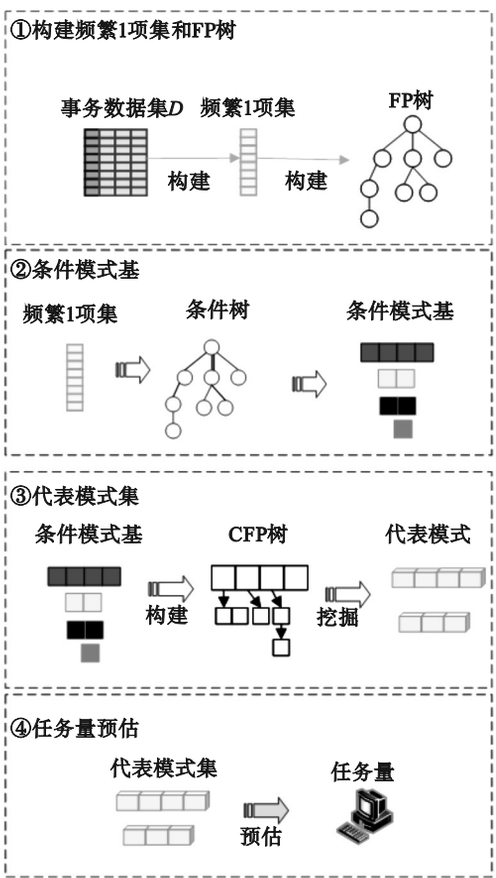

图2　任务量预估

Fig. 2　Task volume estimation

2.3　代表模式挖掘

在分布式FDR假阳性控制计算过程中使用代表模式挖掘方法预估任务量并进行模式压缩.使用基于CFP树[22]的方法进行代表模式挖掘.

代表模式[21]是指可以γ覆盖所有频繁模式的模式,γ覆盖针对的是两个频繁闭合模式.频繁闭合模式是指一个模式S,它的直接超集的支持度计数都不等于该模式S的支持度计数.根据表2和表3可知,模式$\{a,b,c\}$出现在为1,2,3的事务中,模式$\{a,b,c\}$的直接超集$\{a,b,c,d\}$的支持度计数为1,所以模式$\{a,b,c\}$为闭合模式.最小支持度阈值为1模式$\{a,b,c\}$的支持度计数为3>1,所以该模式为频繁闭合模式.对于模式$\{b,c\}$来说它出现在1,2,3事务中,但是模式$\{b,c\}$的直接超模式$\{a,b,c\}$也出现在事务1,2,3中,即模式$\{b,c\}$和模式$\{a,b,c\}$的支持度计数是相同的,所以模式$\{b,c\}$不是闭合模式.

表2　事务数据集

Table 2　Transaction data sets

TID	事务
1	*a,b,c,j*
2	*a,b,c,d,j*
3	*a,b,c*
4	*c,d,e,f*
5	*d,e,f,j*

表3　部分频繁模式集

Table 3　Partially frequent patterns sets

ID	模式	支持度
1	*a*	3
2	*b,c*	3
3	*a,c*	3
4	*a,b,c*	3
5	*a,b,c,d*	1

定义4　$D_t(S_1,S_2)$:给定两个模式S_1和S_2,可以将它们之间的距离定义如下:

$$D_t(S_1,S_2)=1-\frac{|T(S_1)\cap T(S_2)|}{|T(S_1)\cup T(S_2)|}. \tag{4}$$

定义5　γ覆盖:给定一个参数$\gamma\in[0,1)$和两个模式S_1和S_2,如果存在$S_1\subseteq S_2$并且$D_t(S_1,S_2)\leqslant\gamma$,就称$S_1$被$S_2$给$\gamma$覆盖.

引理1　对于两个模式S_1和S_2,并且存在$S_1\subseteq S_2$,$\mathrm{supp}(S_1)=\mathrm{supp}(S_2)$.如果模式$S_2$被模式$S$给$\gamma$覆盖,那么模式$S_1$被模式$S$给$\gamma$覆盖.

引理2　对于两个模式S_1和S_2,并且存在$S_1\subseteq S_2$,$\mathrm{supp}(S_1)=\mathrm{supp}(S_2)$.如果模式$S$被模式$S_1$给$\gamma$覆盖,那么模式$S$被模式$S_2$给$\gamma$覆盖.

引理3　给定两个模式S_1和S_2,如果模式S_1被模式S_2给γ覆盖,则可以使用$\mathrm{supp}(S_2)$近似$\mathrm{supp}(S_1)$,$(\mathrm{supp}(S_1)-\mathrm{supp}(S_2))/\mathrm{supp}(S_1)\leqslant\gamma$.

证明:

$$\frac{\mathrm{supp}(S_1)-\mathrm{supp}(S_2)}{\mathrm{supp}(S_1)}=1-\frac{\mathrm{supp}(S_2)}{\mathrm{supp}(S_1)}=$$

$$1-\frac{|T(S_2)|}{|T(S_1)|}\leqslant 1-\frac{|T(S_1)\cap T(S_2)|}{|T(S_1)\cup T(S_2)|}\leqslant\gamma.$$

引理4　给定两个模式S_1和S_2,如果模式S_1被模式S_2给γ覆盖,那么存在$\mathrm{supp}(S_2)\geqslant\mathrm{supp}(S_1)\times(1-\gamma)$.

如果存在一个模式S_γ,对于模式集$\{S_i\}$中的每一个模式都可以被模式S_γ给γ覆盖,就认为模式S_γ是一个代表模式,可以用来进行模式压缩.

设$C(S)$为一组可以被γ覆盖的频繁模式,在

频繁模式集中找到最小具有代表性的模式集相当于在$C(S)$中找到一个可以覆盖所有频繁模式的最小数量的集合.这是一个NP难的集覆盖问题,一般使用贪婪算法[23]来解决该问题.那么挖掘代表性模式的关键问题就变成在$C(S)$中寻找这样的模式集合,将使用CFP树来进行$C(S)$的查找.使用表2中的事务数据集来描述CFP树查找$C(S)$的过程.表4为从表2事务数据集中发现的支持度计数大于等于1的所有频繁模式的完整集合.

根据表4所示的所有频繁模式构建CFP树结构,CFP树中的每一个结点都是带有索引项的可变长度的数组,并且结点中的所有项都按照其索引大小的升序进行排序.图3显示了使用表4中频繁模式构建的CFP树,CFP树通过模式增长的方式构建,是按升序的方式构建的频繁项集.

表4 所有模式

Table 4 All patterns

n项集	所有模式(min_support=2)
1	*e*:2,*f*:2,*a*:3,*b*:3,*d*:3,*j*:3,*c*:4
2	*ef*:2,*ed*:2,*fd*:2,*ab*:3,*ac*:3,*aj*:2,*bc*:3,*bj*:2,*dc*:2,*dj*:2,*jc*:2
3	*abc*:3,*abj*:2,*acj*:2,*bcj*:2,*efd*:2
4	*abcj*:2

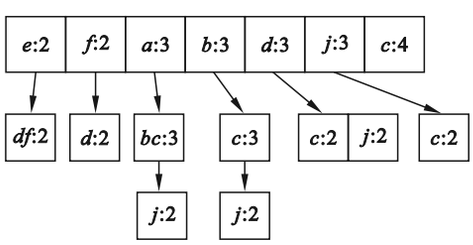

图3 CFP树结构

Fig. 3 CFP tree structure

在CFP树中的每一个结点node都包含了多个信息,首先有结点包含的项items,其次还有结点的支持度计数supp,接下来还包括结点的子结点的指针link,最后还有一个排序后的ID信息.CFP树允许不同模式共享前缀和后缀信息,共享前缀是指模式$\{e,d\}$和$\{e,f\}$在图3中它们共享前缀$\{e\}$.在结点node从包含该结点的事务子集中进行扩展时就会出现后缀共享的情况.如果存在$\mathrm{supp}(S)=\mathrm{supp}(S\cup\{i\})$,那么对于任何模式$X$都有$\mathrm{supp}(S\cup X)=\mathrm{supp}(S\cup\{i\}\cup X)$.例如项$c$是项$b$的候选扩展项,模式$\{b\}$和$\{b,c\}$具有相同的支持度计数,它们共享同一棵子树.CFP树上的e:2→df:2这条链表可以表示$\{e\}$,$\{e,d\}$,$\{e,f\}$和$\{e,d,f\}$这4种模式,那么根据CFP树就可以计算得到γ覆盖模式S的代表模式.

2.4 数据分区

基于代表模式算法对条件模式基进行压缩就可以预估出每个代表模式可以代表的模式数量.用k表示代表模式的长度,则$\mathrm{num}=2^k$为每个代表模式可以代表的模式的数量.根据BH方法的计算原理可知,计算所有模式的p值,需要挖掘出所有的模式.设代表模式集的大小为$L=|C(S)|$,根据项头表结点的条件模式基可以得到该结点的代表模式,计算出每一个结点进行模式挖掘的计算量,也就是进行p值计算的计算量.假设一个项头表结点计算得到的代表模式集的大小为$L_i=|C(S)|$,则该结点需要计算的数据量为$\mathrm{total}_i=L_i\times\mathrm{num}=|C(S)|\times 2^k$.将所有结点需要计算的数据量相加除以集群中结点的数量,可以求出在保证负载均衡的情况下每个结点需要计算的数量.将项头表中的结点按照其各自计算的工作量分配到各个结点上,执行后续的假设确定和BH假阳性控制工作,并以此来避免因计算不均衡造成的数据倾斜问题.

3 实验与性能分析

实验测试主要集中在两个方面:分布式假阳性控制算法对于假阳性控制结果与单机版假阳性控制算法控制结果的差异;分布式假阳性算法对计算速率的提升能力.使用不同的数据集来证明算法的合理性以及普遍适用性.

3.1 实验环境与数据集

本文算法使用Java语言编写,采用Spark框架来进行分布式计算.

本文提出分布式假阳性控制算法,主要实验在集群上完成.实验中使用的数据集信息如表5所示.数据集获取自以下地址:https://www.csie.ntu.edu. tw/~cjlin/libsvmtools/datasets/; http://fimi.uantwerpen. be/data/; https://github. com/VandinLab/TopKWY.在数据集描述中用(L)标记的数据集是带有二分类标签的数据集,用(U)标记的数据集是没有分类的数据集,$|D|$代表数据集中事务的数量.这些数据集里,A9a源于人口普查收入数据,Bms-Web2来自电子商务的点击流数据,Breast-Cancer源于威斯康星州乳腺癌数据集,Codrna是从libSVM站点检索到的cod-rna数据集,Ijcnn1数据集源于IJCNN2001神经网络竞赛,T10I4D100K_new是使用IBM Almaden Quest研究小组的生成器生成的.对于没有将事务划分

为两类的数据集,选择将频率更接近于0.5的单个项目从事务数据集中删除,将数据集人为划分为2组,用n/n_1表示数据集中事务数据量和标签为1的事务数的比例.

表5　实验数据集

Table 5　Experimental data sets

数据集	$\|D\|$	项目数	事务平均长度	n/n_1	事务长度最大值	事务长度最小值
A9a(L)	32,561	247	13. 9	4. 17	14	11
Bms-Web2(U)	77,158	330,285	4. 59	25	66	1
Breast-Cancer(L)	12,773	1,129	6. 7	11. 11	53	1
Codrna(L)	271,617	16	8	3. 03	8	8
Ijcnn1(L)	91,701	44	13	10	13	13
T10I4D100K_new(U)	100,000	870	10. 1	12. 5	20	1

3.2　控制效果测试

本文使用BH过程实现FDR假阳性控制.由于使用的是带有二分类标签的事务数据集,在多重假设检验假阳性控制阶段将会使用FP-Growth算法进行模式挖掘,以此来确定需要检验的假设.实验主要是验证在上述多重假设检验过程中使用BH过程进行FDR假阳性控制后的控制效果.

实验结果显示,使用BH方法进行分布式假阳性控制计算可以将FDR的值控制在选定阈值范围内,图4显示了在控制过程中挖掘的模式数量与数据集事务计数中的项目数的关系,表6所示为使用不同数据集进行FDR假阳性控制后的FDR水平.本实验用户设置的α阈值为0.05.实验结果表明,在不同数据集上使用本文提出的算法进行假阳性控制,可以得到小于并十分接近用户期望阈值的结果.

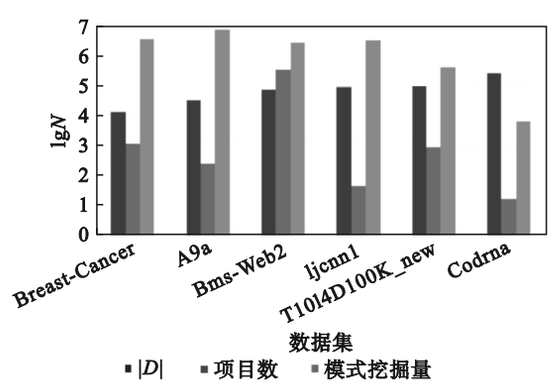

图4　事务数项目数和模式挖掘数

Fig. 4　Number of transactions number of items and number of pattern mining

3.3　准确性测试

对分布式FDR假阳性控制算法的准确率进行测试.本文使用了一种负载均衡的分布式FDR假阳性控制算法,该算法对带有二分类标签的事务数据集进行处理和分区,并使用Spark框架并行地执行假设确定和p值计算等关键步骤,最后使用BH过程来进行假阳性控制.因此本文实验主要目的是验证分布式FDR假阳性控制算法的准确性,即验证使用分布式的策略进行FDR假阳性控制计算的最终p值结果与单机情况下使用BH过程进行假阳性控制计算的最终p值结果是否一致.

表6　不同数据集的FDR控制效果

Table 6　FDR control effect of different data sets

数据集	FDR
A9a	0. 0499
Bms-Web2	0. 0496
Breast-Cancer	0. 0421
Codrna	0. 0497
Ijcnn1	0. 0497
T10I4D100K_new	0. 0491

图5展现了不同数据集上获得FDR假阳性控制的修正显著性阈值.实验结果表明,使用分布式方式并行FDR假阳性控制计算与BH过程计算结果一致.

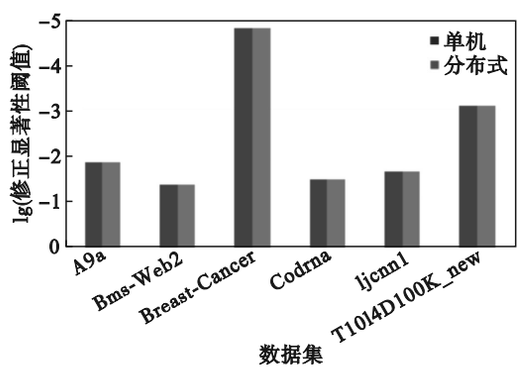

图5　FDR不同数据集的修正显著性阈值

Fig. 5　Modified significance thresholds for different data sets of FDR

3.4　运算效率

对比使用Spark框架的分布式FDR假阳性控制算法与单机使用BH过程执行假阳性控制的计算速率.本文在分布式FDR假阳性控制计算过程中,使用代表模式进行模式压缩预估工作量,并进行数据的分区.

挖掘代表模式会造成一些额外的时间开销，具体时间见图6.根据上述实验结果可知，使用项头表中的结点挖掘代表模式进行工作量预估所花费的时间与整个FDR假阳性控制算法所花费的时间相比可以忽略不计，因此利用代表模式集来预估工作量几乎不会增加FDR假阳性控制算法的计算时间.使用分布式框架进行FDR假阳性控制计算的主要目的就是提高多重假设检验中假阳性控制的计算速度，突破单机计算的局限性.

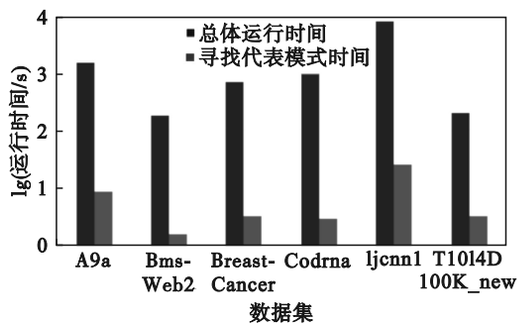

图6 寻找代表模式运行时间

Fig. 6 Find representative mode running time

图7为使用负载均衡策略的分布式算法与未使用负载均衡的分布式算法的计算速度的对比.根据实验结果可知，使用负载均衡的策略可以提高分布式代码的计算速度，降低数据倾斜带来的危害.图8为分布式FDR假阳性控制算法在5结点情况下运行时间与单机使用BH过程进行假阳性控制以及对比算法PBA[17]算法，fastYB[18]算法的运行时间，其中fastYB算法置换次数为5 000.

图9和10是分布式FDR假阳性控制算法的加速比与可伸缩性，前者是不同数量分布式系统结点下算法运行速度提升的比例，后者反映了数据量以倍数增大的情况下算法运行的情况.实验结果表明，使用本文提出的分布式FDR假阳性控制算法所花费的时间更少，达到了实验的预期效果.

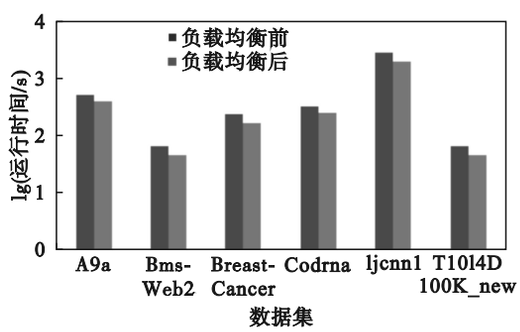

图7 负载均衡前后时间对比

Fig. 7 Time comparison before and after load balancing

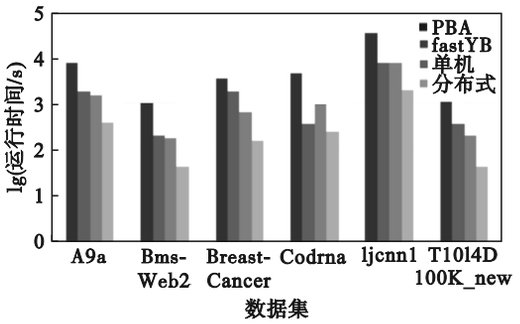

图8 分布式FDR假阳性控制算法与现有算法的运行时间

Fig. 8 Running time of distributed FDR false positive control algorithm and existing algorithms

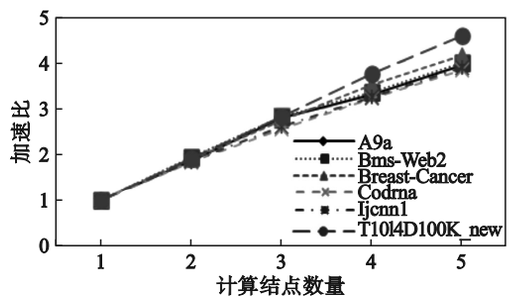

图9 分布式算法加速比

Fig. 9 Distributed algorithm speed ratio

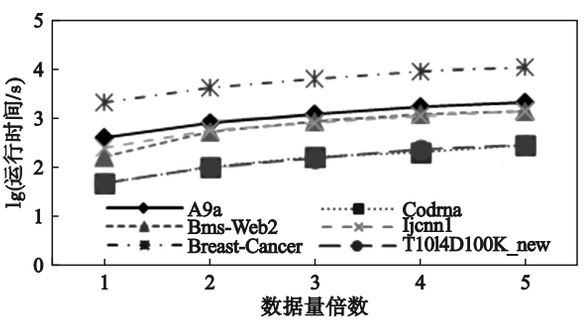

图10 分布式算法可伸缩性

Fig. 10 Distributed algorithm scalability

4 结 论

1）针对假阳性误差控制问题，本文提出一种使用CFP树进行代表模式挖掘的DPFDR分布式假阳性控制算法，有效地将犯假阳性错误的个数在所有被拒绝原假设中所占比例(FDR)控制在α水平以下，减少了多重假设检验带来的假阳性控制问题.

2）针对无并行计算工具情况下的计算效率问题，采用分布式计算，结合CFP树结构挖掘代表模式集进行任务量预估，并采用负载均衡策略进行数据分区，极大提升了计算效率；在大量实

际数据集上进行实验,验证了本文算法较传统算法在计算处理时间上有显著提高.实验结果表明,分布式算法与单机假阳性控制算法得到的最终结果一致,结果小于并十分接近用户期望阈值.因此本文提出的分布式多重假设检验假阳性控制算法具有良好的使用价值.

参考文献:

[1] Erdogmus H. Bayesian hypothesis testing illustrated: an introduction for software engineering researchers[J]. *ACM Computing Surveys*, 2022, 55(6): 1-28.

[2] Kelter R. Power analysis and type I and type II error rates of Bayesian nonparametric two-sample tests for location-shifts based on the Bayes factor under Cauchy priors [J]. *Computational Statistics & Data Analysis*, 2022, 165: 107326.

[3] de Araújo Silva A, Gouvêa M A. Study on the effect of sample size on type I error, in the first, second and first-two digits Excess tests [J]. *International Journal of Accounting Information Systems*, 2023, 48: 100599.

[4] Liu H P, Zhang J V, Wang D, et al. Extended endocrine therapy in breast cancer: a basket of length-constraint feature selection metaheuristics to balance type I against type II errors[J]. *Journal of Biomedical Informatics*, 2022, 131: 104112.

[5] Sharma V S, Afthanorhan A, Barwar N C, et al. A dynamic repository approach for small file management with fast access time on Hadoop cluster: Hash based extended Hadoop archive[J]. *IEEE Access*, 2022, 10: 36856-36867.

[6] Luo C, Cao Q, Li T R, et al. MapReduce accelerated attribute reduction based on neighborhood entropy with Apache Spark[J]. *Expert Systems with Applications*, 2023, 211: 118554.

[7] Llinares-López F, Sugiyama M, Papaxanthos L, et al. Fast and memory-efficient significant pattern mining via permutation testing [C]// Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Sydney, 2015: 725-734.

[8] Dey M, Bhandari S K. FWER goes to zero for correlated normal [J]. *Statistics & Probability Letters*, 2023, 193: 109700.

[9] Terada A, Sese J. Bonferroni correction hides significant motif combinations [C]// 13th IEEE International Conference on BioInformatics and BioEngineering. Chania, 2013: 1-4.

[10] Holm S. A simple sequentially rejective multiple test procedure[J]. *Scandinavian Journal of Statistics*, 1979, 6 (2): 65-70.

[11] Simes R J. An improved Bonferroni procedure for multiple tests of significance[J]. *Biometrika*, 1986, 73(3): 751-754.

[12] Hochberg Y. A sharper Bonferroni procedure for multiple tests of significance[J]. *Biometrika*, 1988, 75(4): 800-802.

[13] Chaubey Y P, Westfall P H, Young S S. Resampling-based multiple testing: examples and methods for p-value adjustment[J]. *Technometrics*, 1993, 35(4): 450.

[14] Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing [J]. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1995, 57(1): 289-300.

[15] Nawaz M S, Azam M, Aslam M. An efficient double exponentially weighted moving average Benjamini-Hochberg control chart to control false discovery rate[J]. *Quality and Reliability Engineering International*, 2019, 35 (8): 2677-2686.

[16] Cui J F, Wang G H, Zou C L, et al. Change-point testing for parallel data sets with FDR control[J]. *Computational Statistics & Data Analysis*, 2023, 182: 107705.

[17] Liu G M, Zhang H J, Wong L S. Controlling false positives in association rule mining[J]. *Proceedings of the VLDB Endowment*, 2011, 5(2): 145-156.

[18] Pellizzoni P, Borgwardt K. FASM and FAST-YB: significant pattern mining with false discovery rate control [C]// 2023 IEEE International Conference on Data Mining (ICDM). Shanghai, 2023: 1265-1270.

[19] Sidák Z. On multivariate normal probabilities of rectangles: their dependence on correlations [J]. *The Annals of Mathematical Statistics*, 1968, 39(5): 1425-1434.

[20] Bestgen Y. Using Fisher's exact test to evaluate association measures for N-grams[EB/OL]. (2021-04-29) [2023-12-29]. https://doi.org/10.48550/arXiv.2104.14209.

[21] Liu G M, Zhang H J, Wong L S. A flexible approach to finding representative pattern sets[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2013, 26(7): 1562-1574.

[22] Liu G M, Lu H J, Yu J X. CFP-tree: a compact disk-based structure for storing and querying frequent itemsets [J]. *Information Systems*, 2007, 32(2): 295-319.

[23] 季策,王金芝,耿蓉.基于Dice系数的弱选择回溯匹配追踪算法[J].东北大学学报(自然科学版), 2021,42(2): 189-195.
(Ji Ce, Wang Jin-zhi, Geng Rong. Weak-selection backtracking matching pursuit algorithm based on Dice coefficient[J]. *Journal of Northeastern University (Natural Science)*, 2021,42(2): 189-195.)