高技术通讯 2009 年 第 19 卷 第 8 期:817 ~ 824

doi:10.3772/j.issn.1002-0470.2009.08.009

基于周期采样的数据流频繁项集挖掘算法研究①

侯 伟② 杨炳儒 吴晨生* 周 谆

(北京科技大学信息工程学院 北京 100083)

(*北京市科学技术情报研究所 北京 100037)

摘 要 针对用于数据流频繁项集挖掘的现有方法存在引入过多次频繁项集以及时空性能与输出精度较低的问题,利用 Chebyshev 不等式,构造了项集频度周期采样的概率误差边界,给出了动态检测项集支持度变化方法。提出了一种基于周期采样的数据流频繁项集挖掘算法 FI-PS,该算法通过跟踪项集支持度变化确定项集支持度的稳定性,并以此作为调整窗口大小以及采样周期的依据,从而以一个较大的概率保证项集支持度误差有上界。理论分析及实验证明该算法有效,在保证挖掘结果准确度相对较好的条件下,可获得较优执行性能。

关键词 数据挖掘,数据流,频繁项(FI)集,周期采样(PS)

0 引 言

随着信息技术的发展,数据库中知识发现(knowledge discovery in databases,KDD)技术应运而生。作为 KDD 中的重要过程,数据挖掘(data mining,DM)采用智能算法从数据中提取有益的数据模式[1]。事实上,大量数据以数据流的形式产生[2],如日志数据、交易数据等。数据流与传统的静态数据不同,它具有连续性、无序性、无界性及实时性的特点[3],要求挖掘算法能够实时地反映模式的变化。面向数据流的频繁项集挖掘已成为研究热点之一。与静态环境不同,数据流自身特点通常迫使挖掘算法对数据最多扫描一遍,且不能保存全部历史数据,同时对其处理速度的要求也较为严格,而且项集组合爆炸现象会进一步加大算法设计的难度。现有算法所采取的策略一般是在输出结果的准确性与处理速度之间取得平衡。

相关算法基本可以分为两类,即以 Lossy Counting[4]、FP-stream[5]为代表的具有确定误差边界的算法,以及以 FDPM[6]、LSFI[7]为代表的利用概率边界(如 Chernoff 边界),在一定概率下具有误差边界的算法。这些算法在一定程度上缓解了时空复杂性,然而前一类算法通常会引入过多的次频繁项集,从而会降低性能及整体精度,后一类算法的设计基于项集支持度稳定的假设条件,该假设过于严格,在概念迁移(concept drift)较频繁时误差较大。本文在研究分析现有方法的基础上,提出了一种基于周期采样的数据流频繁项集挖掘算法——FI-PS (FI 表示频繁项集(frequent itemset), PS 表示周期采样(period sampling)),作为一种新的具有概率误差边界的挖掘方法,该算法跟踪项集支持度变化,它以项集支持度的稳定性作为调整采样窗口大小和采样周期的依据,从而以一个较大的概率保证项集支持度误差有上界。

1 问题背景与定义

设 $IU = \{x_1, x_2, \cdots, x_n\}$ 为由全体项目 $x_i (1 < i < n)$ 构成的集合,其任意非空子集 $I \subseteq IU$ 称为项集。具有 k 个项目的项集称为 k-项集。事务为一二元组 (tid, Y),其中 tid 为事务索引,Y 为一项集。若事务 T 的项集 Y 为项集 X 的超集,即 $Y \supseteq X$,则称事务 T 支持项集 X。

数据流由不断到达的事务构成,一般假设它是无限的。一事务流可视为一个 landmark 窗口 $W = [T_1, T_2, \cdots, T_\tau]$,其中 τ 为当前时刻。一个项集 X 在窗口 W 内的频度 $Freq(X)$ 是指 W 内支持 X 的事

① 国家自然科学基金(60675030),国家科技成果重点推广计划(2003EC000001)和北京市自然科学基金(4022008)资助项目。

② 男,1981 年生,博士生;研究方向:数据挖掘;联系人,E-mail: dr.houwei@gmail.com

(收稿日期:2008-05-12)

务数。项集 X 在窗口 W 内的支持度定义为 $Sup(X) = Freq(X)/N$，其中 N 为 W 中的事务总数。若 $Sup(X) \geqslant s$，则 X 称为 W 中的频繁项集，其中 $s(0 \leqslant s \leqslant 1)$ 为用户预设的最小支持度阈值。

给定事务流与最小支持度阈值 s，数据流频繁项集挖掘可以归结为尽可能准确地找出窗口 W 中的所有频繁项集及其支持度。大量理论分析与实践证明，在数据流中无法设计高效且精确的频繁项集挖掘方法。因此一般采取输出近似结果的变通方式，即算法仅给出项集支持度的估计值 $\overline{Sup(X)}$，而该项集的真实支持度是 $Sup(X)$，两者一般不同。目前有两类方法：(1)估计支持度满足 $Sup(X) - \overline{Sup(X)} \leqslant \varepsilon$，即其具有确定边界 ε；(2)估计支持度满足 $\Pr\{|Sup(X) - \overline{Sup(X)}| \leqslant \varepsilon\} \geqslant 1 - \delta$，即误差超过边界 ε 的概率具有确定下界 $1 - \delta$。由于项集频繁性不稳定，算法不仅需要维护当前频繁项集的支持度，还要维护支持度大于 $s - \varepsilon$ 的项集，即次频繁项集，其很可能变为频繁。

Lossy Counting[4] 是第一类方法中最具代表性的。它将事务流划分为一系列桶，每个桶包含 $B = \lceil 1/\varepsilon \rceil$ 个事务，ε 为误差上界，结构 D 维护项集信息，项集 X 的信息由三元组 $(X, \overline{Freq(X)}, err(X))$ 表达，其中 $\overline{Freq(X)}$ 为 X 插入 D 后的频度，$err(X)$ 为 X 插入 D 前频度的上界。当 β 个桶到达时，算法更新 D，删除满足 $\overline{Freq(X)} + err(X) < bid$ 的项集，并插入次频繁项集。可证明，D 包含当前所有频繁项集，且估计支持度误差小于 ε。与其类似，FP-stream 算法[5] 的估计支持度同样具有 εN 的确定误差边界。不同之处在于，它采用 pattern-tree 结构维护项集及其频度，并且采用 tilted-time window 策略，对历史数据进行多时间粒度的维护，因此支持项集历史支持度查询。在此类方法中，较难选择合适的 ε，过大则降低输出频度的精度，反之则会产生过多的次频繁项集，从而降低性能，该现象随事务的不断到达会逐渐明显。

在第二类方法中，FDPM[6] 是最具代表性的算法之一。它以 Chernoff 边界为基础，挖掘过程中不断删减估计支持度小于 $(s - \varepsilon_B)$ 的次频繁项集，其中 $\varepsilon_B = \sqrt{(2s\ln(2/\delta))/B}$，$\delta$、$s$ 为预设的失败概率与支持度阈值。对于任一频繁项集，算法以 $(1 - \delta)$ 为概率下界将其输出。FDPM 算法不保证找到所有频繁项集，而力求不输出任何非频繁项集。LSFI 算法[7] 利用 Heoffding 边界，通过随机采样以达系统降载效果。其以 $n_0 = (1/2\varepsilon^2)\ln(2/\delta)$ 作为采样集大小，当负荷过载时算法根据过载情况确定采样率，以采样集代替事务流，直到负载下降。这类方法试图以部分样本代表数据流整体，然而项集支持度一般是不稳定的，因此难以设计合理的采样方法，从而输出结果精度较低。

2 算法理论基础

2.1 周期采样概率误差边界

为应对数据流环境中高速、持续产生的事务，一些算法采取随机采样等策略，并利用 Chernoff，Hoeffding 等概率边界，以部分事务近似数据流整体。为方便讨论，在此给出下列引理使用的部分假设：设 $x_i(i = 1,2,\cdots,n)$ 为一系列独立的服从 0 - 1 分布的随机变量，概率 $\Pr\{x_i = 1\} = p$，随机变量 $X = \sum x_i$ 服从二项分布。实数 ε, δ 是常量，这里分别表示支持度误差上界与失败概率上届，X/n 为估计支持度。

引理 1 (Chernoff 边界[8]) 对任意 $\varepsilon \in (0,1)$，下列不等式成立：

$$\Pr\{X \leqslant (1 - \varepsilon)np\} \leqslant e^{-\varepsilon^2 np/2} \quad (1)$$

$$\Pr\{X \geqslant (1 + \varepsilon)np\} \leqslant e^{-\varepsilon^2 np/3} \quad (2)$$

依据 Chernoff 边界，可知在一定 ε 下，随着采样集的大小 n 趋于正无穷，X/n 依概率 1 存在于区间 $[p - \varepsilon, p + \varepsilon]$ 内。在数据流频繁项集挖掘中，将上面两式的右端视为失败概率上届 δ，从而依式(1)可得 $\delta = e^{-\varepsilon^2 np/2}$，由此得 $n_0 = (-2\ln(\delta))/p\varepsilon^2$，当采样集的大小超过 n_0 时，则有 $\Pr\{p - X/n \leqslant \varepsilon p\} \geqslant 1 - \delta$，进而达到以部分数据估计数据流整体的效果。

引理 2 (Hoeffding 边界[9]) 对任意 $\varepsilon \in (0,1)$，下列不等式成立：

$$\Pr\{|X - np| \geqslant n\varepsilon\} \leqslant 2e^{-2n\varepsilon^2} \quad (3)$$

Hoeffding 边界可以获得与 Chernoff 边界类似的结论，其中依式(3)右端 $\delta = 2e^{-2n\varepsilon^2}$ 可得 $n_0 = (-\ln(2/\delta))/2\varepsilon^2$，其与 p 无关，因此相对简洁且易于计算，然而由其获得的边界较为松散。

根据以上引理，可知其需要满足采样样本独立同分布的假设，否则就必须保证采样是均匀的，即一个时间段的采样数量与该时间段内样本数量成正比。举例说明。设存在时刻 t 将事务流分为前后两部分，即 $W_1 = [T_1,\cdots,T_t]$ 与 $W_2 = [T_{t+1},\cdots,T_\tau]$，其

中 τ 为当前时刻，W_1 与 W_2 对项集 I 的支持度分别为 p_1 和 $p_2(p_1 \neq p_2)$，此时有 $p = |W_1| p_1 + |W_2| p_2$。当且仅当 $n_1 |W_2| = n_2 |W_1|$ 成立，$p' = n_1 p_1 + n_2 p_2$ 为 p 的无偏估计。

由此，理想的基于采样的方法应能准确检测事务流中项集支持度的变化，同时应使采样尽可能均匀。周期采样作为一类均匀采样技术，广泛应用于信号分析与数据压缩等领域。在 FI-PS 算法中，周期采样保证了采样的均匀性，根据项集支持度的稳定情况，算法可使用不同的采样周期 r，在保证估值准确的同时使执行效率大为提高。

定理 1　(周期采样概率误差边界)。设样本总体 U 大小为 N，由一系列独立同分布，服从 0－1 分布的随机变量 $x_i(i = 1,2,\cdots,N)$ 构成，其 $\Pr\{x_i = 1\} = p = \sum_{x_i \in U} x_i / N$，依采样周期 r 对总体采样构成采样集 S，记估计值 $\hat{p} = \sum_{x_i \in S} x_i / |S|$，对任意 $0 < \varepsilon < 1$，不等式

$$\Pr\{|p - \hat{p}| \leqslant \varepsilon\} \geqslant 1 - \delta \tag{4}$$

成立，其中

$$\delta = \frac{r - 1}{4N\varepsilon^2} \tag{5}$$

证明：首先构造随机变量 $Err = p - \hat{p}$，由期望 $\mathrm{E}(\hat{p}) = \mathrm{E}\left(\sum_{x_i \in S} x_i / |S|\right) = p$ 可知 $\mathrm{E}(Err) = \mathrm{E}(p) - \mathrm{E}(\hat{p})$，从而 $\mathrm{E}(Err) = 0$。由周期采样性质可知 $r = N/|S|$，从而有

$$\begin{aligned}
\mathrm{D}(Err) &= \mathrm{D}\left(\frac{\sum_{x_i \in U} x_i}{N} - \frac{\sum_{x_i \in S} x_i}{|S|}\right) = \mathrm{D}\left(\frac{\sum_{x_i \in U} x_i}{N} - \frac{r\sum_{x_i \in S} x_i}{N}\right) \\
&= \frac{1}{N^2}\mathrm{D}\left(\sum_{x_i \in U} x_i - r\sum_{x_i \in S} x_i\right) \\
&= \frac{1}{N^2}\mathrm{D}\left(\sum_{x_i \in S} x_i - \sum_{x_i \in S} x_i + \sum_{x_i \in U} x_i - r\sum_{x_i \in S} x_i\right) \\
&= \frac{1}{N^2}\mathrm{D}\left((1 - r)\sum_{x_i \in S} x_i + \sum_{x_i \in U} x_i - \sum_{x_i \in S} x_i\right) \\
&= \frac{1}{N^2}\left[\mathrm{D}\left((1 - r)\sum_{x_i \in S} x_i\right) + \mathrm{D}\left(\sum_{x_i \in U} x_i - \sum_{x_i \in S} x_i\right)\right] \qquad (6) \\
&= \frac{1}{N^2}\left[(1 - r)^2\left(\frac{N(p - p^2)}{r}\right) + \frac{N(r - 1)(p - p^2)}{r}\right] \qquad (7) \\
&= \frac{p(1 - p)(1 - r)}{N}
\end{aligned}$$

其中式(6)利用了随机变量 x_i 独立性假设，式(7)利用了 0－1 分布方差 $\mathrm{D}(x_i) = p(1 - p)$。将以上结果代入 Chebyshev 不等式，得

$$\begin{aligned}
\Pr\{|Err - \mathrm{E}(Err)| \leqslant \varepsilon\} &\geqslant 1 - \frac{\mathrm{D}(Err)}{\varepsilon^2} \\
&= 1 - \frac{p(1 - p)(r - 1)}{N\varepsilon^2} \geqslant 1 - \frac{r - 1}{4N\varepsilon^2}
\end{aligned} \tag{8}$$

式(8)利用了不等式 $p(1 - p) \leqslant 1/4$，即得式(4)，定理得证。

2.2　项集支持度分布变化检测

根据定理 1 以及上述引理可知，概率边界的正确使用是以支持度分布稳定为假设前提的，在数据流频繁项集挖掘中，0－1 分布的概率 p 应尽可能平稳。一个直观的方法是，随着事务的不断到达，通过检测历史支持度 p' 与最近支持度 p 之间的差异来判断是否发生了变化。概念迁移的研究已取得了一些成果[10]，然而这些方法一般基于针对分类精度的检测，不能直接引入到数据流频繁项集挖掘问题中。同时，如果仅以 $|p' - p|$ 是否超过阈值来判断改变的发生，一则阈值难以合理预设，同时过于刚性。FI-PS 算法利用已获得的周期采样概率边界，估计 $|p' - p|$ 超过阈值的概率，并以此概率作为判断支持度变化的依据，该方法使人更易理解且更为易用。

定理 2　设 $p' \in (0,1)$ 为一概率值，其他假设条件及讨论环境同定理 1，则下列不等式成立：

$$\Pr\{|p - p'| \leqslant \varepsilon\} \geqslant \beta \tag{9}$$

其中 $\beta = (1 - \delta)\left(1 - \frac{|p' - \hat{p}|}{2\varepsilon}\right)$。即 $p' \in [p - \varepsilon, p + \varepsilon]$ 事件发生的概率下界为 β，β 称为平稳概率。

证明：由 $x_i(i = 1,2,\cdots,N)$ 独立同分布，且具有期望与方差，考虑到采样量很大，根据中心极限定理，可以认为 $\hat{p} = \sum_{x_i \in S} x_i / |S|$ 服从正态分布，进而随机变量 $Err = p - \hat{p}$ 近似服从正态分布，且均值为 0。设 Err 的概率密度为 $f(x)$，则根据定理 1 $\int_{-\varepsilon}^{\varepsilon} f(x)\mathrm{d}x \geqslant 1 - \delta$，设 $b = p' - \hat{p}$，且有随机变量 $R = p - \hat{p} - b = p - p'$，因此 R 服从正态分布，其方差与 Err 方差相同，均值为 $-b$，概率密度为 $f(x + b)$。从而有

$$\begin{aligned}
\Pr\{|R| \leqslant \varepsilon\} &= \int_{-\varepsilon}^{\varepsilon} f(x + b)\mathrm{d}x \\
&= \int_{-\varepsilon+b}^{\varepsilon+b} f(x)\mathrm{d}x \\
&= \int_{-\varepsilon+b}^{\varepsilon} f(x)\mathrm{d}x + \int_{\varepsilon}^{\varepsilon+b} f(x)\mathrm{d}x \\
&\geqslant \int_{-\varepsilon+b}^{\varepsilon} f(x)\mathrm{d}x
\end{aligned} \tag{10}$$

$$= \int_{-\varepsilon}^{\varepsilon} f(x)\mathrm{d}x - \int_{-\varepsilon}^{-\varepsilon+b} f(x)\mathrm{d}x \tag{11}$$

若将函数 $f(x)$ 以 $x = 0$ 为界分为两段，可知此两端函数均为单调函数，且在均值处达到最大，因此有 $\int_{-\varepsilon}^{-\varepsilon+b} f(x)\mathrm{d}x \leqslant q \mid b \mid$，其中 $q = \frac{1}{2\varepsilon}\int_{-\varepsilon}^{\varepsilon} f(x)\mathrm{d}x$。根据式(11)，有 $\Pr\{\mid R \mid \leqslant \varepsilon\} \geqslant \int_{-\varepsilon}^{\varepsilon} f(x)\mathrm{d}x - q \mid b \mid = \int_{-\varepsilon}^{\varepsilon} f(x)\mathrm{d}x\left[1 - \frac{\mid b \mid}{2\varepsilon}\right] \geqslant (1-\delta)\left(1 - \frac{\mid p' - \hat{p} \mid}{2\varepsilon}\right)$，定理得证。

依据定理 2，可以通过概率值评估历史支持度与当前支持度的差别，该方法相对单一阈值方法更加易理解、灵活。值得注意的是，式(9)在 $\mid p' - \hat{p} \mid$ 较大时，概率边界 β 一般较松。

3 FI-PS 算法

3.1 数据结构 FI-tree

在频繁项集挖掘问题中，一般采用树形结构维护频繁项集及其支持度，本文对 P-tree 结构[11]加以扩展，提出 FI-tree 结构，该结构继承了 P-tree 结构空间效率高的优点，并依项集支持度的平稳性建立了索引列表，在 FI-PS 算法中将利用这些索引对项集高效检索，其结构有利于负边界[12]的计算与对项集遍历。FI-tree 的基本结构是一个二叉树，如图 1 所示。

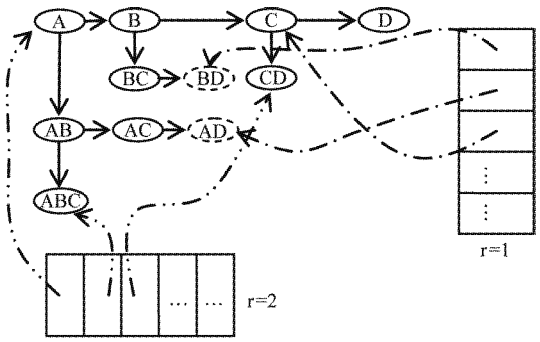

图 1 数据结构 FI-tree

定义 1(负边界) 设项集集合 S，其负边界 $Bd^-(S)$ 也是一个项集集合，其中每一个元素(项集)不属于 S，且该元素的任何真子集均属于 S。即，$Bd^-(S) = \{p \mid \forall p \in U \setminus S \wedge (\forall m \subset p, m \in S)\}$，其中 U 为项集全集。在本文中，S 是由全体次频繁项集构成的集合。

在 FI-tree 结构中，任一节点是一四元组 $\langle I, \hat{p}, r, b\rangle$，其中 I 为项集，$\hat{p}$ 为估计支持度，r 为该项集的采样周期，b 是布尔值，它标识了该项集是否为负边界中的项集。结构中节点代表一个次频繁项集，即 $\hat{p}$ 大于 $s - \varepsilon$，其中 s 和 ε 为预设的支持度与误差阈值，或一负边界项集(如图 1 中虚线圈中的项集)。根据定义 1 可知，在非次频繁项集中，负边界项集的支持度最接近 $s - \varepsilon$，即负边界中的项集最有可能成为次频繁项集。

FI-tree 结构中，一个索引列表对应一个采样周期，其作用是维护相应采样周期的项集信息内存地址。索引列表依项集大小排序，这有益于减少不必要的项集包含验证。其中项集以二进制表达，这种方法有利于比较项集的包含关系，相当于一次整数减法运算，因此可被高效执行。

3.2 支持度变化检测

FI-PS 算法利用周期采样估计项集支持度，以降低算法时间复杂度，然而其理论基础(定理 1)是建立在支持度分布稳定的假设基础上，因此算法应保证在以一定采样周期对一段时间内的事务进行采样时，目标项集的支持度在该段时间内的各个部分是基本一致的。同时，如果目标项集的支持度，在事务流中发生变化时，算法应能够及时反应，并采取更小的采样周期。

定义 2(采样窗口) 针对一目标项集，FI-PS 算法将事务流分为连续且不相交的基本窗口，基本窗口长度 $w = 1/(4\delta\varepsilon^2)$。设 r 为目标项集采取的采样周期，当 $r = 1$ 时基本窗口即为采样窗口，当 $r \geqslant 2$ 时一个采样窗口由 $r - 1$ 个基本窗口构成。

定义 3(指标窗口) 针对一目标项集，指标窗口是一个历史采样窗口，设对该窗口采样统计所得的估计支持度为 p'，对该采样窗口之后的任一采样窗口进行周期采样所得的估计支持度 $\hat{p}$，则一定有

$$(1-\delta)\left(1 - \frac{\mid p' - \hat{p} \mid}{2\varepsilon}\right) \geqslant \beta \tag{12}$$

其中 β 为定理 2 中的平稳概率，ε 与 δ 定义与 2.1 节一致。即在指标窗口与当前采样窗口之间的所有采样窗口中，目标项集的支持度基本稳定。

采样窗口的长度对应于定理 1 中的 N，依定义 2可知参数 ε 与 δ 及变量 N 与 r 满足等式(5)，由此保证不等式(4)成立。采取固定的指标窗口而不是滑动的，是因为支持度的变化可能是缓慢的，固定指标窗口不仅对快速变化较为敏感，同时可以有效检测支持度的缓慢变化。

为降低时间复杂性,FI-PS 算法针对不同的项集采取不同的采样周期,对于支持度稳定的项集采用较低的采样周期,反之选取较高的采样周期,以 2 的倍数递阶,即 $r = 2^n(0 \leqslant n)$。当完成对一个采样窗口的统计处理时,算法以从该窗口所得到的目标项集估计支持度 $\hat{p}$ 和指标窗口的 p',以验证式(12),如果不等式成立则提高采样周期和采样窗口长度,否则以采样周期 $r = 1$ 对最新采样窗口重新统计计数,并将该采样窗口作为后续窗口的指标窗口。

值得注意的是,算法为保证估值准确,需要维护一个缓冲区,大小由最大的采样窗口决定。如果数据流中项集支持度较为稳定,则最大采样窗口的长度会不断加大,对于一般的应用环境,以空间代价换取高精度与高速度是不必要的,因此 FI-PS 算法引入参数 M,限定最大采样窗口长度,以此平衡空间代价与执行效率与精度。

3.3　FI-PS 算法流程

由于无法预判一个项集将来是否为频繁项集,为了确保每一个频繁项集都被找到,FI-PS 算法需要对 FI-tree 结构中维护的项集进行周期更新。

在 FI-tree 结构中,只维护次频繁项集以及次频繁项集的负边界项集。当一个基本窗口结束时,算法遍历 FI-tree 结构,将 $\hat{p} < s - \varepsilon$ 的项集从结构中删除,同时将负边界中 $\hat{p} \geqslant s - \varepsilon$ 的项集标识为次频繁项集,最后重新计算负边界。对于新的负边界项集,以其在当前基本窗口中的支持度作为其估计支持度。依此思想,可构造算法 1,如图 2 所示。

在第 4 行 Bootstrap 函数中, FI-PS 算法利用 Fp-growth 算法挖掘第一个基本窗口中的次频繁项集,在所得结果的基础上,第 5 行算法构建 FI-tree 结构。第 10 行,对支持度的更新只限于部分项集,这些项集的采样周期可以整除当前事务的索引号。当一个基本窗口结束时,如果一个项集的采样窗口同时结束,将在第 16 行考查其支持度是否仍然平稳。若不平稳则调整其采样周期为 1,且于第 19 行针对其当前采样窗口做重新统计,并调整其指标窗口为当前采样窗口;若平稳,则于第 21 行增加项集采样周期。在调整 FI-tree 结构的过程中,负边界可能发生了改变,第 28 行检查负边界中是否存在次频繁项集,若存在则将其标记为次频繁项集,并再次计算次频繁项集的负边界。对于新加入负边界的项集,以其在当前基本窗口的支持度作为其估计支持度。反复检查直至负边界内不存在次频繁项集为止。

当用户查询当前频繁项集时,算法输出 FI-tree 结构中除负边界以外的所有项集,过程实现简单,这里不再细述。

算法 1. FI-PS 算法

输入: 支持度阈值 s, 误差边界 ε, 失败概率 δ,
平稳概率 β, 最大采样窗口长度 M;
输出: FI-tree.

```
1.   FI-PS( s,ε,δ,β,M )
2.   {
3.     Initialize();      //计算基本窗口长度等
4.     Bootstrap();       //挖掘第一个基本窗口
5.     FI-tree = CreateFI-tree();
6.     when new tuple t arrived
7.     {
8.       tid = t.id; //获得事务 t 的 id
9.       for each index_list in FI-tree
10.        if tid is divisible by index_list.r
11.          Update( index_list,t );
12.        if a basic window is over
13.        {
14.          for each node in FI-tree
15.          {
16.            if node.p is changed //调整采样周期
17.            {
18.              node.r = 1;
19.              node.p' = ReCount(); //重新统计
20.            }else if 2 × node.r ⩽ M
21.             node.r = 2 × node.r;
22.            if node in negative boundary
23.              if node.p̂ ⩾ s − ε
24.                make node out of negative boundary;
25.              else if node.p̂ < s − ε
26.                delete node;
27.          }
28.          ReCreateNegativeBoundary();
29.        }
30.    }
31.  }
```

图 2　FI-PS 算法流程

3.4　讨论分析

文献[6]讨论了 False-Positive 与 False-Negative 两类数据流频繁项集挖掘问题,前者旨在结果尽可能包括所有频繁项集,同时允许包含部分次频繁项集;后者旨在结果尽可能不包括非频繁项集,同时允许一部分频繁项集被遗漏。本文上述内容在介绍 FI-PS 算法时,是围绕 False-Positive 问题展开的,然

而通过将次频繁项集定义改为支持度大于 s 的项集,即与频繁项集等价,同时调整图 2 中第 26 行节点删除条件为 $node.\hat{p} < s$,以及第 23 行次频繁项集的插入条件为 $node.\hat{p} \geqslant s$,算法即可适用于 False-Negative 的情况。

算法应用于 False-Positive 情况时,一个频繁项集被遗漏的概率小于 δ^f,f 为该项集估计误差大于 ε 的采样窗口个数,由定理 1 可知其发生误差大于 ε 的概率为 δ。若一频繁项集被遗漏,则有其总体支持度 $p_t \geqslant s$,同时总体估计支持度 $\hat{p}_t < s - \varepsilon$,因此至少在一个采样窗口中 $p_w - \hat{p}_w \geqslant \varepsilon$,因此 f 不小于 1。与此相似,算法应用于 False-Negative 情况时,一个非频繁项集被输出,等价于事件 $\hat{p}_t - p_t > 0$,其概率小于 $(\frac{1}{2})^f$,f 为目标项集满足 $\hat{p}_w - p_w > 0$ 成立的采样窗口数目,由于 $\hat{p}_t - p_t > 0$ 因此至少存在一个采样窗口,在其中 $\hat{p}_w - p_w > 0$ 成立,考虑 $\hat{p}_w$ 近似服从以 p_w 为期望的正态分布,可知 $\hat{p}_w - p_w > 0$ 成立的概率为 1/2。同理可知 False-Negative 情况下,FI-PS 算法遗漏一个频繁项集的概率也小于 $(1/2)^f$。上述说明,当支持度稳定时,算法输出的准确性随时间逐步增加。

4 实验分析

相对其他方法,FI-PS 算法的优势在于其时间性能与结果准确性较为平衡,特别在分布不稳定数据流中,FI-PS 算法可以获得相对更准确的结果。因此实验设计分别在稳定与不稳定的事务数据流(下简称稳定流与不稳定流)环境下进行。数据由 IBM 数据生成器[13]产生,其中稳定流由 10 万个事务构成,共涉及 1000 个项目,平均事务长度取 5,其他参数取缺省值;不稳定流,由 5 个独立生成的大小为 2 万的事务集拼接构成,每一部分产生的参数与生成稳定流的一致。实验环境:PC 主频 2.0GHz,内存 1GB,操作系统为 Windows 2003。算法采用标准 C++ 实现,编译器为 MinGW。实验选择 Lossy Counting[4] 与 FDPM[6]作为对比算法,前者是针对 False-Positive 问题的具有确定误差边界的算法,后者是针对 False-Negative 问题的基于概率误差边界的算法。以下实验只考虑 FI-PS 算法针对 False-Positive 问题的形式。

以下实验从时间性能及结果准确度两个方面考察算法。在时间性能方面,以每 1 万条事务处理用时为评价指标;在结果准确度方面,引入文献[6]使用的 *Recall* 与 *Precision* 两指标,前者表示结果中频繁项集占全部频繁项集的比例,后者表示结果中频繁项集数与输出项集数目的比例,它们用于衡量算法区分频繁项集与次频繁项集的能力,同时实验定义指标

$$SupportPrecision = 1 - \frac{\sum_{i \in FI \cap Re} | FISup(i) - ReSup(i) |}{\sum_{i \in FI \cap Re} FISup(i)} \quad (13)$$

其中 FI 与 Re 分别指 Apriori 算法(其获得精确结果)与待评价算法获得的结果集,$FISup(i)$ 与 $ReSup(i)$ 两函数分别返回项集 i 的真实支持度与估计支持度,该指标在一定程度上反映算法估计支持度的精度水平。下列实验中,取 $s = 0.20$,容差 $\varepsilon = 0.05$,FDPM 与 FI-PS 中 $\delta = 0.1$,$\beta = 0.8$,$M = 64$。

理论上以上三种算法都适用于稳定流环境,然而随着到达事务的增加,由于 FI-PS 算法基于周期采样,因此通过不断调整采样周期,它在时间性能方面体现出比较明显的优势,如图 3 所示。Lossy Counting 与 FDPM 算法的时间性能都较平稳,而 FDPM 性能较好,这主要是由于其维护的项集数量较少。在 *Recall/Precision* 指标方面,如表 1 所示,FI-PS 算法与 Lossy Counting 算法相对比较稳定,两算法在 *Precision* 指标的错误主要来源于次频繁项集的引入。FDPM 算法在 *Recall* 指标上依时间逐步精确,这是由于初始时到达的事务较少,其估计误差较大,导致一些频繁项集被遗漏。在 *SupportPrecision* 指标方面,FI-PS 算法在前期略低于其它算法,这是由于周期采样带来的计数偏差,随着处理事务数量的增大,这种差距逐渐缩小。

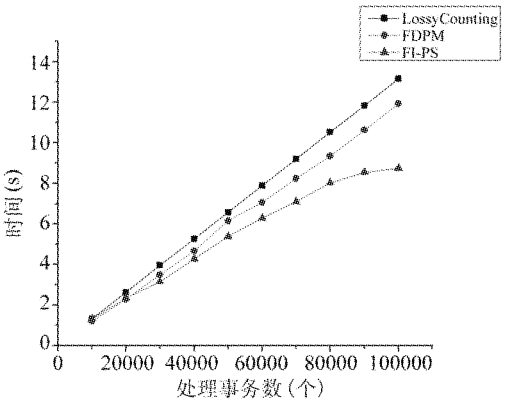

图 3 稳定流环境下各算法耗时比较

表 1　稳定流环境下各算法准确性比较

算法 / 事务数	Lossy Counting			FDPM			FI-PS		
	R	*P*	*SP*	*R*	*P*	*SP*	*R*	*P*	*SP*
10000	1.000	0.953	0.992	0.962	1.000	0.995	1.000	0.952	0.994
2000	1.000	0.961	0.999	0.981	1.000	1.000	1.000	0.961	0.996
3000	1.000	0.966	1.000	0.982	1.000	1.000	1.000	0.967	0.995
4000	1.000	0.955	1.000	0.985	1.000	1.000	1.000	0.954	0.994
5000	1.000	0.951	1.000	0.991	1.000	1.000	1.000	0.953	0.995
6000	1.000	0.956	1.000	0.989	1.000	1.000	1.000	0.952	0.997
7000	1.000	0.962	1.000	0.991	1.000	1.000	1.000	0.961	0.999
8000	1.000	0.957	1.000	0.993	1.000	1.000	1.000	0.954	1.000
9000	1.000	0.965	1.000	0.995	1.000	1.000	1.000	0.966	1.000
10000	1.000	0.959	1.000	0.993	1.000	1.000	1.000	0.960	1.000

（注：*R*，*P*，*SP* 分别代表 *Recall*，*Precision*，*SupportPrecision*）

在不稳定流环境下，依处理事务数量增大，FI-PS 算法在时间性能方面的优势仍然较为明显，如图 4所示。事务流中项集的支持度波动，对三种算法的计算耗时均产生了不同程度的影响，尤其对 Lossy Counting 算法的影响更为明显，因为波动造成该算法需要维护大量次频繁项集，从而产生较大的维护代价。在 *Recall* 指标方面，如表 2 所示，FI-PS 算法与 Lossy Counting 算法差距很小，从长期来看明显好于 FDPM 算法，这是由于一些频繁项集在某一段时间内支持度低于阈值，因此其部分频度被 FDPM 算法忽略。相反在 *Precision* 指标方面，FI-PS 算法与 FDPM 算法明显好于 Lossy Counting 算法，这种差距是由于后者引入了过多的次频繁项集所导致。最后在 *SupportPrecision* 指标方面，FI-PS 算法略好于 Lossy Counting 算法，且优于 FDPM 算法，这得益于采样周期的动态调整和对负边界中项集支持度的预估。

综上可知，FI-PS 算法在准确度方面稍好于其它两种算法，且其执行效率相对更高，特别在事务流不稳定环境下，其优势更为显著。

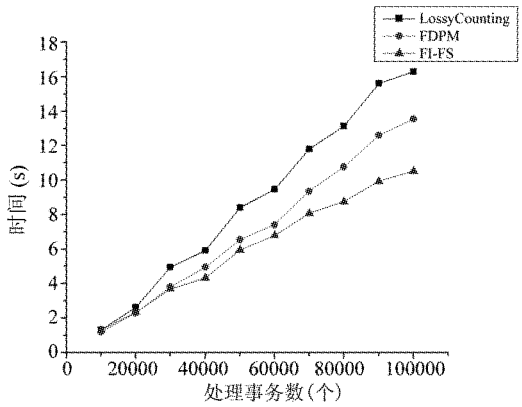

图 4　不稳定流环境下各算法耗时比较

表 2　不稳定流环境下各算法准确性比较

算法 / 事务数	Lossy Counting			FDPM			FI-PS		
	R	*P*	*SP*	*R*	*P*	*SP*	*R*	*P*	*SP*
1000	1.000	0.953	0.992	0.962	1.000	0.995	1.000	0.952	0.994
2000	1.000	0.961	0.999	0.981	1.000	1.000	1.000	0.961	0.996
3000	1.000	0.872	0.973	0.872	0.934	0.967	0.989	0.935	0.979
4000	1.000	0.898	0.986	0.927	0.952	0.989	0.999	0.948	0.989
5000	1.000	0.864	0.960	0.863	0.902	0.947	0.976	0.913	0.962
6000	1.000	0.879	0.984	0.882	0.949	0.951	0.998	0.942	0.988
7000	1.000	0.841	0.961	0.859	0.912	0.939	0.981	0.916	0.967
8000	1.000	0.863	0.983	0.868	0.945	0.945	0.996	0.942	0.986
9000	1.000	0.859	0.959	0.852	0.909	0.924	0.974	0.907	0.957
10000	1.000	0.865	0.979	0.871	0.895	0.931	0.997	0.901	0.983

（注：*R*，*P*，*SP* 分别代表 *Recall*，*Precision*，*SupportPrecision*）

5 结 论

以往的数据流频繁项集挖掘算法在一定程度上缓解了挖掘算法的时空复杂度。然而过多次频繁项集的引入和项集支持度的变化限制了现有方法的执行性能与输出精度。本文提出一种基于周期采样的数据流频繁项集挖掘算法,该算法是一种具有概率误差边界的挖掘方法,它通过跟踪项集支持度变化确定项集支持度的稳定性,并以此为依据,动态调整采样窗口大小以及采样周期,从而以一个较大的概率保证项集支持度误差有上界。理论分析及实验证明该算法有效,在保证挖掘结果准确度相对较好的条件下,可获得较优执行性能。

参考文献

[1] Han J, Kamber M. Data Mining: Concepts and Techniques. San Francisco: Morgan Kaufmann Publishers, 2000. 15-32

[2] Gaber M M, Zaslavsky A, Krishnaswamy S. Mining data streams: a review. *ACM SIGMOD Record*, 2005, 34(2):18-26

[3] Garofalakis M N, Gehrke J, Rastogi R. Querying and mining data streams: you only get one look. In: Proceedings of the 28th International Conference on Very Large Data Bases, Hong Kong, China, 2002. 635-635

[4] Manku G S, Motwani R. Approximate frequency counts over data streams. In: Proceedings of the 28th International Conference on Very Large Data Bases, Hong Kong, China, 2002. 346-357

[5] Giannella C, Han J, Pei J, et al. Mining frequent patterns in data streams at multiple time granularities. In: Proceedings of Next Generation Data Mining, Cambridge, USA, 2003. 91-212

[6] Yu J, Chong Z, Lu H, Zhou A. False positive or false negative: mining frequent itemsets from high speed transactional data streams. In: Proceedings of the 30th International Conference on Very Large Data Bases, Toronto, Canada, 2004. 204-215

[7] Dang X H, Ng W K, Ong K L. Adaptive load shedding for mining frequent patterns from data streams. In: Proceedings of the 8th Data Warehousing and Knowledge Discovery, Krakow, Poland, 2006. 342-351

[8] Hagerup T, Rüb C. A guided tour of chernoff bounds. *Information Processing Letters*, 1990, 33(6): 305-308

[9] Hoeffding W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 1963, 58(301): 13-30

[10] Klinkenberg R. Learning drifting concepts: example selection vs. example weighting. *Intelligent Data Analysis*, 2004, 8(3): 281-300

[11] Coenen F, Goulbourne G, Leng P. Tree structures for mining association rules. *Data Mining and Knowledge Discovery*, 2004, 8(1): 25-51

[12] Wang C Y, Tseng S S, Hong T P, et al. Online generation of association rules under multidimensional consideration based on negative-border. *Journal of Information Science and Engineering*, 2007, 23(1): 233-242

[13] IBM Almaden Research Center. Synthetic data generation code for associations and sequential patterns. http://www.almaden.ibm.com/software/quest/Resources/index.shtml Intelligent Information Systems: IBM, 1999

FI-PS: an algorithm for mining frequent itemsets in data streams based on period sampling

Hou Wei, Yang Bingru, Wu Chensheng*, Zhou Zhun

(School of Information Engineering, University of Science and Technology Beijing, Beijing 100083)

(* Beijing Municipal Institute of Science and Technology Information, Beijing 100037)

Abstract

Aiming at the problems in the existing algorithms for mining frequent itemsets in data streams, such as unwanted sub-frequent itemsets, poor performance and low accuracy, the paper constructs a probabilistic error bound for itemset frequency period sampling based on the Chebyshev inequality, and gives a method of dynamically detecting the change of itemset supports, and then, proposes the FI-PS, an algorithm based on period sampling for mining frequent itemsets in data streams. The algorithm can periodically measure the stabilities of supports by monitoring the changes of frequencies. The size of sampling window and sampling period are adjusted accordingly. Thereby, the error of support is upper bounded with the biggish probability. The theoretical analysis and experiments prove its effectiveness and efficiency.

Key words: data ming, data stream, frequent itemset (FI), period sampling (PS)