数理解析研究所講究録
第1799巻 2012年 159-162

2011年度冬のLAシンポジウム [S3]

ストリーム中の頻出アイテム発見に対する O(loglog$N$) 領域乱択アルゴリズム

緒方正虎* 山内由紀子* 来嶋秀治* 山下雅史*

1 概要

頻出アイテム発見は与えられたアイテム集合から，ある割合を越える頻度で出現するアイテムを発見する問題である．ユーザが与える閾値 $\theta \in (0,1)$ に対して，データ長 N のストリームから $\theta \cdot N$ 回を越える頻度で出現するアイテムを発見する．2003年にKarpらはこの問題に対して偽陽性を許す $O(\theta^{-1}\log N)$ ビットの決定性オンラインアルゴリズムと，決定性アルゴリズムによる頻出アイテム発見の下界を示した．本論文ではこの問題の記憶領域に着目し，$O(\theta^{-2}\log^2\theta^{-1} + \log\log N)$ ビットの記憶領域で動作する，シンプルな乱択アルゴリズムを提案する．提案手法は他の $O(\log N)$ ビットで動作する乱択アルゴリズム，決定性アルゴリズムと比較してメモリのオーバーフローに対して頑健である，

はじめに

近年，スーパーのPOSシステムやインターネットのパケットモニタリングなど，大規模なデータのストリームを出力する計算機システムが増加している．頻出アイテム発見はこのような大規模データ処理の基本的な問題として注目されている．頻出アイテム発見問題は入力されるストリームを $\boldsymbol{x} = (x_1, x_2, ...x_N)$ とした時，ユーザが与える閾値 $\theta \in (0,1)$ に対して，$\theta \cdot N$ 回以上出現するすべてのアイテムを発見する問題である．ストリームデータ処理では，アルゴリズムは未知の入力サイズに対して正常に動作することが求められる．仮にメモリオーバーフローが生じると重篤な問題を引き起こしかねないため，動作に要する記憶領域を慎重に設定する必要がある．

単一ストリーム上の頻出アイテム発見問題について，これまでさまざまな研究がなされてきた．Boyerら [1] は majority algorithm として知られる $\theta = 1/2$ に対する $O(\log N)$ アルゴリズムを与えた．これとは独立にFischer[2] らも本質的に同様のアルゴリズムを与えた．2003年にはKarpら [4]，Demainら [3] が決定性アルゴリズムを提案した．この手法は偽陽性を含む頻出アイテム集合を出力するもので，$O(\frac{1}{\theta}\log N)$ ビットの記憶領域で実行されることが知られている．また，Karpらは同時に厳密な頻出アイテム検知に必要な記憶領域について，アイテムのシンボル数を $|\Sigma|$ とした時の下界，$\Omega(|\Sigma|\log(N/|\Sigma|))$ ビットを示した．Mankuら [5] は lossy counting と呼ばれる別の決定性アルゴリズムを提案した．この手法は $O(\varepsilon^{-1}\log\delta^{-1}\log N)$ ビットの記憶領域で動作する近似数え上げアルゴリズムを基にしており，任意の $\varepsilon \in (0,1)$ について $(\theta - \varepsilon)$ の頻度を越えるアイテムを出力する．

単一ストリーム上の確率的手法についてはToivonenら [6] が $O(\varepsilon^{-2}\log\delta^{-1}\log N)$ ビットの記憶領域で動作するアルゴリズムを提案した．この手法はデータの一様サンプリングを基にしており，確率 $1-\delta$ ですべての頻出アイテムを乱択する．Manku[5] らも sticky sampling と呼ばれる別のアルゴリズムを提案している．このアルゴリズムは $O(\varepsilon^{-1}\log\delta^{-1}\log N)$ ビットで動作し，確率 $1-\delta$ ですべての頻出アイテムを出力する．両アルゴリズムは到着したアイテムに応じて乱択確率を変更することで頻出アイテムを発見する．しかし，ここで触れたすべてのアルゴリズムは頻出アイテム検知に $O(\log N)$ ビットの記憶領域を要する．

本論文では未知のデータ長 N に対し $O(\theta^{-2}\log^2\theta^{-1} + \log\log N)$ ビットの記憶領域で動作する頻出アイテム発見アルゴリズムを提案する．提案手法は $\log\log N$ ビットの記憶領域で到着アイテムの近似数え上げを行う．これは Morris ら [9]，Flajolet[10] らと同様のアイデアに基づいている．提案アルゴリズムは指数近似による頻出アイテムの数え上げと，入力に対して平均して定数時間での動作が可能である．また，他の乱択アルゴリズムと比較してメモリオーバーフローに対してロバストである．

本論文の構成は次の通りである．第2章で問題設定，第3章でアルゴリズムのアイデアを示す．第4章でアルゴリズムの動作の解説と証明を行い，第5章で結論を述べる．

2 問題

Σ をアイテムの全集合とする．簡単の為，Σ は有限集合で，集合の各要素は $\sigma := \log|\Sigma|$ ビットで識別できるとする．Σ の要素から成る系列 $\boldsymbol{x} = (x_1, ..., x_N)$ に対して，$f(s) \in \{0, ..., N\}$ を $\boldsymbol{x}$ 中のアイテム $s \in \Sigma$ の出現回数とすれば，与えられた定数 $\theta \in (0,1)$ に対して，

$$f(s; \boldsymbol{x}) \geq \theta N$$

*九州大学

を満たす時，s を頻出であるという．

3 提案手法のアイデア

仮に，ストリーム x のデータ長 N が既知であるならば，頻出アイテムの発見は容易である．c を近似パラメータとして，それぞれのアイテム x_i が到着するたびに確率 c/N で乱択し，すべてのアイテムが入力されたのちに閾値を越えるアイテムを出力すればよい．しかし，実際のアプリケーションを考えれば，データ長 N が既知であるというのは不自然な仮定である．

未知のデータ長 N に対するアルゴリズム設計のアイデアの一つに，ビットシフトがある．ビットシフトアルゴリズムは入力されるデータ長を数え上げ，到着アイテム数に応じて乱択確率を指数的に減少させる．この時，一度乱択したアイテムも確率的に捨てられていく．固定サイズの記憶領域で乱択したアイテムを保持できるため，未知の入力長 N に対して有効なアイデアだが，解析が困難である．

解析の難しさを除けばビットシフトは良い戦略だと思われるが，ストリームのデータ長の数え上げに $O(\log N)$ ビットの記憶領域が必要となる．読み込んだデータ長に応じて記憶領域を確保するインクリメンタルなプログラムを組むことは可能であるが，$O(\log N)$ ビットの記憶領域が必要であり，より少ない領域で動くアルゴリズムの方が好ましい．

提案手法の基本的なアイデアはストリームのデータ長 N の指数近似である．データ長の指数部のみを近似することにより，$O(\log\log N)$ ビットの記憶領域でのアルゴリズム設計が可能となった．提案手法は指数近似のデータ長を用いて，ビットシフトのアイデアを導入する．アイテム x_i を読み込む時，すでに到着したアイテム総数 N_i の指数近似により，乱択確率を決める．一度乱択され，頻出アイテム集合 K に保持されたアイテムは N_i が増加するに従い，再度乱択される．この操作は **"flush"** と呼ばれ，提案手法では N_i が 2 倍になるごとに，保持されている全アイテムを確率 $1/2$ で再度乱択する．乱択確率の減少と保持するアイテムの更新によって，提案手法はストリームの全アイテムをあたかも同一確率で乱択したかのように動作する．また **"flush"** によりアイテム保持に用いる記憶領域を定数にとどめ，アルゴリズムのオーバーフローを防いでいる．アルゴリズムの記憶領域はデータ長 N の指数近似のみに依存し，$O(\log\log N)$ ビットを必要とする．

次の章でアルゴリズムの擬似コードを示し，その証明を行う．アルゴリズムは $O(\theta^{-2}\log^2(\theta^{-1}) + \log\log N)$ ビットで動作し，その記憶領域は $N \gg |\Sigma| \gg 1/\theta$ の条件下でアイテム集合 $|\Sigma|$ に対して独立である．

4 提案アルゴリズム

$\theta \in (0,1)$，$\gamma \in (0,1)$，$\delta \in (0,1)$ をユーザの定めるパラメータとする．与えられた θ，γ，δ に対し，提案アルゴリズムが $\{s \mid f(s) \geq N\theta(1-\gamma)\}$ を満たす全てのアイテムを出力し，$\{s \mid f(s) < N\theta\}$ を満たすアイテムを頻出アイテム集合に含まない確率は $1-\delta$ 以上である．以下に $x = (x_1, x_2, ..., x_N)$ の文字列に対する提案アルゴリズムを示す．提案アルゴリズムは頻

Algorithm 1

0. **Initialize** K consisting of 2^b pairs of an item, and a number $(s, K[s])$,
   where s uses σ bits, and $K[s]$ uses b bits.
   Set: "exponent" $h = 0$.
1. **Read** x_i. Suppose $x_i = s^*$ for convenience.
   If no more input, then **goto** 5.
2. With probability 2^{-h}, **"record"** s^* in K:
   add s^* in K unless s^* in K, and $K[s^*]++$.
   If $\sum_{s\in K} K[s] = 2^b$, then **"flush"**:
   3-(i) **Set** $h++$.
   3-(ii) **For** each $s \in K$,
   Choose k with probability $\binom{K[s]}{k}/2^{K[s]}$, and
   Set $K[s] := k$.
   3-(iii) **Delete** symbol $s \in K$ if $K[s] = 0$
4. **Goto** 1
5. **Output** every s satisfying that $K(s) \geq (1-\gamma/2)\theta\sum_{s'\in K} K[s']$.

出アイテム候補の記憶に $(b+\sigma)2^b + \lceil \lg h \rceil$ ビット，その他の作業領域に $O(\lg h)$ ビットを用いる，Step 2 で用いられる確率 2^{-h} は，計算機上で確率 $1/2$ のベルヌーイ試行を h 回繰り返すことにより $O(\lg h)$ 領域で実現できる．つまり到着したアイテム x_i は h 回投げたコインがすべて表だった場合に乱択される．アルゴリズム中の Step3-(ii) の"**flush**"についても同様のアイデアを用いている．これらの試行はすべて到着アイテム数を近似した $\lceil \log h \rceil$ の h に従って実行される．ここで $h+b$ はストリーム長 $N \geq 2^b$ の近似であるので，$2^h K[s]$ はアイテムの出現数 $f(s)$ を近似している．アルゴリズムが出力するアイテムは最大で $1/(\theta(1-\gamma/2))$ 個である．

定理 4.1. 任意の定数 $\theta \in (0,1)$，$\gamma \in (0,1)$，$\delta \in (0,1)$ を与える．パラメータ b を $b \geq \lceil \lg((\theta\gamma/2)^{-2}\ln(3((1-\gamma/2)\theta\delta)^{-1}))\rceil + 3$ とした時，任意のストリームに対して提案アルゴリズムがすべての頻出アイテムを出力し，かつ，$f(s) < (1-\gamma)\cdot\theta N$ を満たす頻出でないアイテム $s \in \Sigma$ を出力しない確率は $1-\delta$ である．

定理 4.1 はアルゴリズムが以下の記憶領域を必要とすることを示している．

$$
\begin{aligned}
& 2^b(b+\sigma)+\lg\lg N \\
\simeq\ & \frac{32\ln\left(\frac{3}{(1-\gamma/2)\theta\delta}\right)}{(\gamma\theta)^2}\lg\left(\frac{\ln\left(\frac{3}{(1-\gamma/2)\theta\delta}\right)}{(\gamma\theta)^2}\right)+\lg\lg N \\
=\ & O\left(\frac{\log^2(\theta^{-1})}{\theta^2}+\log\log N\right)\text{ビット}.
\end{aligned}
$$

証明の簡便さのため b は最適化されていないことを断っておく.

考察 4.1. ストリーム $\boldsymbol{x}=(x_1,\ldots,x_N)$ を最後まで読み込んだ時，それぞれのアイテム x_i がアルゴリズムに乱択されている確率は等しく 2^{-h} である．ここで，h はアルゴリズム 1 の終了時における *exponent* である.

証明. あるアイテム x_i が到着した時の exponent の値を h' とする．この時 x_i は $2^{-h'}$ の確率で K に保存される．またアルゴリズムの Step3-(ii) で h' が更新された時，記憶した x_i を確率 $1/2$ で K 中に保持し，$1/2$ の確率で消去する．するとアルゴリズム終了時の任意のアイテム $s\in\Sigma$ の出現数は幾何分布 $\binom{K[s]}{k}/2^{K[s]}$ に従う．従って任意のアイテム x_i がアルゴリズムの最後まで K に保持されている確率は 2^{-h} である. □

補題 4.1. パラメータ b を $b\geq\lceil\lg((\theta\gamma/2)^{-2}\ln(3((1-\gamma/2)\theta\delta)^{-1}))\rceil+3$ とする．$h\geq 1$ が成り立つ時，アルゴリズム 1 について以下の確率が成り立つ.

$$
\Pr\left[\sum_{s'\in K}K[s']\leq 2^{b-2}\right]\geq 1-\frac{\delta}{3}
$$

証明. $\sum_{s\in K'}K[s']$ は Step3-(ii) の処理を除き単調増加である．従って $\sum_{s'\in K}K[s']$ が k となる確率は二項分布 $\binom{2^b}{k}/2^{2^b}$ に従う．従って以下の式を得る.

$$
\Pr\left[\sum_{s'\in K}K[s']\leq 2^{b-2}\right]\leq\sum_{j=0}^{2^{b-2}}\binom{n}{j}\frac{1}{2^{2^b}}
$$

ここで，二項分布に対する次の不等式を導入する.

$$
\sum_{j=0}^{(p-t)n}\binom{n}{j}p^j(1-p)^{n-j}\leq e^{-2t^2n}\qquad(1)
$$

パラメータをそれぞれ $t\in(0,p)$. $n=2^b$, $p=1/2$, $t=1/4$ とすれば次の不等式を得る.

$$
\sum_{j=0}^{2^{b-2}}\binom{n}{j}\frac{1}{2^{2^b}}\leq e^{-2(1/4)^2 2^b}=e^{-\ln(\delta/3)}=\frac{\delta}{3}
$$

□

考察 4.2. $n=\sum_{s'\in K}K[s']$ とした時，K に含まれている任意のアイテム $s\in\Sigma$ の出現数は次の超幾何分布に従う.

$$
\Pr[K[s]=j]=\frac{\binom{f(s)}{j}\cdot\binom{N-f(s)}{n-j}}{\binom{N}{n}}
$$

上の考察 4.2 を用いることによって次の補題を導く.

補題 4.2. パラメータ b を $b\geq\lceil\lg((\theta\gamma/2)^{-2}\ln(3((1-\gamma/2)\theta\delta)^{-1}))\rceil+3$ とする．$\sum_{s'\in K}K[s']\geq 2^{b-2}$ が成り立つ時，アルゴリズムがすべての頻出アイテム $\{s\in\Sigma\mid f(s)\geq N\theta\}$ を発見する確率は $1-\delta/3$ 以上である.

証明. $s\in\Sigma$ を $f(s)\geq\theta N$ を満たす頻出アイテムとする．超幾何分布に関する Chvatal[11] による以下の不等式を用いる.

$$
\sum_{j=0}^{k}\frac{\binom{M}{j}\cdot\binom{N-M}{n-j}}{\binom{N}{n}}\leq e^{-2(\frac{k}{n}-\frac{M}{N})^2 n}\qquad(2)
$$

上の不等式のパラメータをそれぞれ，$N=|x|$，$M=f(s)$，$n=\sum_{s'\in K}K[s']\geq 2^{b-2}$，$k=(1-\gamma/2)\theta n$ とする．この時 $k/n\leq(1-\gamma/2)\theta$，$M/N\geq\theta$ が成り立つことより，次の不等式をえる.

$$
\begin{aligned}
& \Pr\left[K[s]<(1-\gamma/2)\theta\sum_{s'\in K}K[s']\right] \\
\leq\ & \sum_{j=0}^{(1-\gamma/2)\theta n}\frac{\binom{M}{j}\cdot\binom{N-M}{n-j}}{\binom{N}{n}} \\
\leq\ & e^{-2(\theta-(1-\gamma/2)\theta)^2 2^{b-2}}\leq e^{-2(\theta\gamma/2)^2 2^{b-2}}\leq\frac{(1-\gamma/2)\theta\delta}{3}
\end{aligned}
$$

頻出アイテムは高々 $1/\theta$ 個しかないので，偽陰性の確率は $\lfloor 1/\theta\rfloor(1-\gamma/2)\theta\delta/3\leq\delta/3$ 以下である. □

補題 4.3. パラメータ b を $b\geq\lceil\lg((\theta\gamma/2)^{-2}\ln(3((1-\gamma/2)\theta\delta)^{-1}))\rceil+3$ とする．$\sum_{s'\in K}K[s']\geq 2^{b-2}$ がアルゴリズム終了まで成立していた時，アルゴリズムが任意の頻出でないアイテム $\{s\mid f(s)<(1-\gamma)\theta N\}$ を出力する確率は $\delta/3$ 以下である.

証明. s を不等式 $f(s)<(1-\gamma)\theta N$ を満たす，任意の頻出でないアイテムとする．ここで先の不等式をもう一度用いる．$N=|x|$，$M=N-f(s)$，$n=\sum_{s'\in K}K[s']\geq 2^{b-2}$，$k=(1-(1-\gamma/2)\theta)n$ とすれば，$k/n<1-(1-\gamma)\theta$，$M/N\geq 1-(1-\gamma/2)\theta$ が成り立つ．すると次の不等式を得る

$$
\begin{aligned}
& \Pr\ [K[s]\geq(1-\gamma/2)\theta n] \\
=\ & \Pr[n-K[s]<(1-(1-\gamma/2)\theta)n] \\
\leq\ & \sum_{j=0}^{(1-(1-\gamma/2)\theta)n}\frac{\binom{M}{j}\cdot\binom{N-M}{n-j}}{\binom{N}{n}} \\
\leq\ & e^{-2((1-(1-\gamma)\theta)-(1-(1-\gamma/2)\theta))^2 2^{b-2}} \\
\leq\ & e^{-2(\theta\gamma/2)^2 2^{b-2}}\leq\frac{(1-\gamma/2)\theta\delta}{3}
\end{aligned}
$$

アルゴリズムは最大で $1/(\theta(1-\gamma/2))$ 個の要素を出力する．したがって，偽陽性の確率は

$$
\frac{1}{\theta(1-\gamma/2)}\cdot\frac{(1-\gamma/2)\theta\delta}{3}\leq\frac{\delta}{3}.
$$

□

これらの補題4.1, 4.2, 4.3より定理4.1を得る．最後に $(h+b)$ による $\lg N$ の近似の精度を示す．

補題 4.4. パラメータ b を $b \geq \lceil \lg((\theta\gamma/2)^{-2}\ln(3((1-\gamma/2)\theta\delta)^{-1}))\rceil + 3$ とする．アルゴリズムにアイテム x_n が入力された時，次の不等式が成り立つ．

$$\Pr[\lfloor \lg n \rfloor - 1 \leq h + b \leq \lceil \lg n \rceil + 1] \leq 1 - \delta$$

証明. 上の不等式は以下を意味していることに注意する．

$$\begin{aligned} &\Pr \quad [\lfloor \lg n \rfloor - 1 \leq h + b \leq \lceil \lg n \rceil + 1] \\ &= \quad 1 - (\Pr[h + b < \lg n - 1] + \Pr[h + b > \lg n + 1]) \end{aligned}$$

$X(n;\alpha)$ を $1/2^\alpha$ の確率で表が出るコインを用いた n 回のベルヌーイ試行による幾何分布に従う確率変数とする．観察4.2より，$X(n;\lfloor \lg(n) \rfloor - b - 1) < 2^b$ の時，またその時に限り $h + b < \lfloor \lg(n) \rfloor - 1$ である．ここで Chernoff 上界を用いると次の不等式を得る．

$$\begin{aligned} \Pr \quad & [h + b < \lfloor \lg(n) \rfloor - 1] \leq \Pr[X(n;\lg(n) - b - 1) < 2^b] \\ = \quad & \Pr\Big[X(n;\lg(n) - b - 1) < \frac{2^b \cdot E[X(n;\lg(n) - b - 1)]}{E[X(n;\lg(n) - b - 1)]}\Big] \\ = \quad & \Pr\Big[X(n;\lg(n) - b - 1) < \frac{2^b \cdot E[X(n;\lg(n) - b - 1)]}{n2^{-\lg(n)+b+1}}\Big] \\ = \quad & \Pr\Big[X(n;\lg(n) - b - 1) < \frac{2^b}{2^{b+1}} E[X(n;\lg(n) - b - 1)]\Big] \\ = \quad & \Pr\Big[X(n;\lg(n) - b - 1) < \big(1 - \frac{1}{2}\big) E[X(n;\lg(n) - b - 1)]\Big] \\ \leq \quad & e^{-\frac{1}{2}\frac{1}{2^2}E[X(n;\lg(n)-b-1)]} \leq e^{-\frac{1}{2^3}n2^{-\lg(n)-b-1}} \leq e^{-2^{b-2}} \leq \frac{\delta}{2} \end{aligned}$$

$X(n;\lfloor \lg(n) \rfloor - b + 1) \geq 2^b$ の時，またその時に限り $h + b > \lceil \lg(n) \rceil + 1$ が成り立つことも同様にして証明が可能である．Chernoff 上界を用いれば次の不等式を得られる．

$$\begin{aligned} \Pr \quad & [h + b \geq \lceil \lg(n) \rceil + 1] \leq \Pr[X(n;\lg(n) - b + 1) \geq 2^b] \\ = \quad & \Pr\Big[X(n;\lg(n) - b + 1) \geq \frac{2^b \cdot E[X(n;\lg(n) - b + 1)]}{E[X(n;\lg(n) - b + 1)]}\Big] \\ = \quad & \Pr\Big[X(n;\lg(n) - b + 1) \geq \frac{2^b \cdot E[X(n;\lg(n) - b + 1)]}{n2^{-\lg(n)+b-1}}\Big] \\ = \quad & \Pr\Big[X(n;\lg(n) - b + 1) \geq \frac{2^b}{2^{b-1}} E[X(n;\lg(n) - b + 1)]\Big] \\ = \quad & \Pr\big[X(n;\lg(n) - b + 1) \geq 2E[X(n;\lg(n) - b + 1)]\big] \\ \leq \quad & e^{-\frac{1}{3}E[X(n;\lg(n)-b+1)]} \leq e^{-\frac{1}{3}n2^{-\lg(n)+b-1}} \leq e^{-2^{b-3}} \leq \frac{\delta}{2} \end{aligned}$$

□

5 結論

本研究ではシンプルな頻出アイテム発見乱択アルゴリズムを提案した．アルゴリズムは $O(\theta^{-2}\log(\theta^{-1}) + \lg\lg N)$ ビットの記憶領域で動作し，メモリのオーバーフローに対してロバストである．今後は計算機実験を行い，実データを用いた他のアルゴリズムとの性能比較，各パラメータの最適化を行う．

参考文献

[1] R.S. Boyer and J.S. Moore, A fast majority vote algorithm, Technical Report ICSCA-CMP-32, Institute for Computer Science, University of Texas, 1981.

[2] M. Fischer, and S. Salzburg, Finding a majority among n votes: solution to problem 81-5, J. Algorithms, 3 (1982), 376–379.

[3] E.D. Demaine, A. López-Ortiz, and J.I. Munro, Frequency estimation of internet packet streams with limited space, In Proc. of 10th Annual European Symposium on Algorithms (2002), 348–360.

[4] R.M. Karp, S. Shenker, and C. Papadimitriou, A simple algorithm for finding frequent elements in streams and bags, ACM Transactions on Database Systems, 28 (2003), 51–55.

[5] G.S. Manku, and R. Motwani, Approximate frequency counts over data streams, In Proc. of 28th Intl. Conf. on Very Large Data Bases (2002), 346–357.

[6] H. Toivonen, Sampling large database for association rules, In Proc. of 22nd Intl. Conf. on Very Data Bases (1966), 134–145.

[7] G. Cormode, and M. Hadjieleftheriou, Methods for nding frequent items in datastreams, The VLDB Journal, 19 (2010), 3–20.

[8] H. Liu, Y. Yuan, and J. Han, Methods for mining frequent items in data streams, Knowl. and Inf. Syst, 26 (2011), 1–30.

[9] R. Morris, Counting large numbers of events in small registers, Communications of the ACM, 21 (1978), 840–842.

[10] P. Flajolet, Approximate counting: a detailed analysis, BIT, 25 (1985), 113–134.

[11] V. Chvatal, The tail of the hypergeometric distribution, Discrete Mathematics, 25 (1979), 285–287.