

リソース指向近似計算に基づくオンライン頻出圧縮アイテム集合マイニング

山本 泰生† 岩沼 宏治†

†山梨大学大学院総合研究部

1 はじめに

ストリームデータを対象とするデータストリームマイニング (data stream mining, 以降, DSM と略す) の研究が活発に進められている。ストリームデータは、金融や流通分野の取引データから医療や気象観測、交通情報のセンシングデータにいたる幅広い用途の中で、近年、大量に生成されている。DSM はこのような大規模データの新しい知的価値を創出する重要な技術であり、その応用分野は多岐にわたる。本研究ではこの中でも、トランザクションストリームデータ中の頻出アイテム集合を抽出する問題に焦点をあてる。

トランザクションはストリームとして常に到着し続けたため、すべてのデータをインメモリで保持することが難しい。よって、この問題には、出力となるアイテム集合の頻度に誤差を許しながら 1 度のデータスキャンで実行する 1 パス近似アルゴリズムがよく適用される。トランザクションが複数のアイテム (個数を L とし、 L をトランザクション長と呼ぶ) を含む一般の場合、最悪 2^L 個のアイテム集合を生成する必要がある、大規模データに適用することはこれまで原理的に難しいとされてきた。この問題に対し、著者らは、処理速度やメモリ消費といった計算資源 (リソース) の制約のもと、頻出アイテム集合を効率的に求めるリソース指向型の近似計算法: Skip LC-SS を提案している [2]。

本稿では、Skip LC-SS 法を概説した後、その重要な性質として近似精度とメモリ消費の間に成り立つトレードオフ関係を示す。次に、近似精度を保証するためのメモリ使用量について新しい知見を示す。最後に、上記のトレードオフを緩和する手法として飽和性 (closeness) [1] に基づく圧縮技法を導入するアイデアを説明する。

2 準備

本稿で用いる用語の定義を行う。 $I = \{x_1, x_2, \dots, x_u\}$ をアイテムの全体集合とすると、 I の空でない部分集合をアイテム集合と呼ぶ。データストリーム S は、到着するトランザクションの列 $\langle t_1, t_2, \dots, t_n \rangle$ として定義される。ただしトランザクション t_i は時刻 i に到着す

るアイテム集合であり、 n は到着予定の (未知の) トランザクション数である。アイテム集合 α を含む S 中のトランザクションの個数を頻度と呼び、 $\text{sup}(\alpha)$ と表す。最小サポート σ ($0 \leq \sigma \leq 1$) に対し、 $\text{sup}(\alpha) \geq \sigma n$ を満たすとき、 α を頻出アイテム集合と呼ぶ。 α をアイテム集合とする。任意のアイテム $i \in I$ に対して、 $\text{sup}(\alpha \cup \{i\}) \neq \text{sup}(\alpha)$ のときまたそのときに限り、 α を飽和アイテム集合と呼ぶ。

頻出アイテム集合をオンラインで抽出する決定性アルゴリズムは、一般に、候補のアイテム集合 α と α の見込み頻度 $c(\alpha)$ の組 $\langle \alpha, c(\alpha) \rangle$ を一つのエントリとして管理する。ここでエントリ全体を頻度表と呼び、時刻 i における頻度表サイズを $k(i)$ とする。特に $k(i)$ が一定の場合、 k と略す。 ϵ ($0 \leq \epsilon \leq 1$) を誤差パラメータとする。任意の出力 α に対して、(1) $0 \leq c(\alpha) \leq n$ かつ (2) $\text{sup}(\alpha) - \epsilon n \leq c(\alpha) \leq \text{sup}(\alpha) + \epsilon n$ を満たすとき、そのアルゴリズムは ϵ 近似であるという。

3 Skip LC-SS 法

アルゴリズムによって頻度表の管理法は異なるが、 ϵ 近似を満たす場合、メモリ消費は一般に $O(2^L)$ と L に応じて指数的に増加する。図 1 は、 ϵ 近似を満たす手法の一つ Stream mining 法を、日本の地震発生データ [2] に適用したときのメモリ消費を示している。図中の縦軸と横軸はそれぞれトランザクション ID と頻度表サイズに相当している。トランザクション ID が 15,000 付近で頻度表サイズが指数的に増加しているが、これは東日本大震災に関連する多数の地震発生イベントを持つトランザクションが到着しているためである。

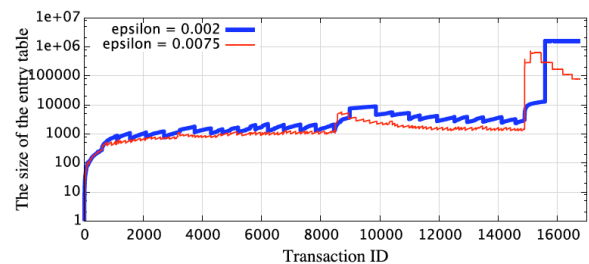


図 1: Stream Mining 法におけるメモリ消費の推移

一方、Skip LC-SS 法は、頻度表サイズ k を固定した上でトランザクション処理を線形時間 $O(kL)$ で行うこ

Resource-oriented approximation for frequent compressed itemset mining from big transaction streams

†Department of Computer Science, University of Yamanashi
Email: yyamamoto@yamanashi.ac.jp

とができる。Lossy-Counting (LC) 法の Δ 誤差の概念と Space-Saving (SS) 法の交換操作を用いて頻度表の管理を行う。あるトランザクション t_i について任意の部分アイテム集合が候補となるが、そのすべてを登録する空きが表になかった場合、見積り頻度が最小のエントリに対してのみ交換操作を行う。時刻 i においてアイテム集合 β (最小エントリに相当) が新規アイテム集合 α と交換する場合、3 項組 $(\alpha, 1, \Delta(i))$ がエントリとして登録される。ただし $\Delta(i)$ は、時刻 i における最大誤差頻度を表している。すなわち α の見積り頻度 $c(\alpha)$ は、その時点での最大誤差頻度に 1 を加えた値となる。

例 1 $S = \{\{a\}, \{a, b\}, \{b, c\}, \{a, b, c, d, e\}\}$, 最小サポート $\sigma = 0.6$, 頻度表サイズ $k = 4$ とすると, Skip LC-SS 法では下図のように頻度表を管理していく。 $i = 4$ のとき $2^5 - 1$ 個の候補集合が更新・登録可能だが, 実際に交換操作されるのは黒枠の 2 つの最小エントリのみで良い。

$i = 1$

α	f	Δ
$\{a\}$	1	0

$\Delta(1) = 0$

$i = 2$

α	f	Δ
$\{a\}$	2	0
$\{b\}$	1	0
$\{a,b\}$	1	0

$\Delta(2) = 0$

$i = 3$

α	f	Δ
$\{a\}$	2	0
$\{b\}$	2	0
$\{b,c\}$	1	0
$\{c\}$	1	0

$\Delta(3) = 0$

$i = 4$

α	f	Δ
$\{a\}$	3	0
$\{b\}$	3	0
$\{b,c\}$	1	1
$\{c\}$	1	1

replacement

$\Delta(4) = 1$

Skip LC-SS 法は、与えられたメモリ消費の範囲中で、任意のトランザクションを線形時間 $O(kL)$ で処理することができる。しかし一方で、上記の例にある通り、最小エントリ数を超える候補集合が出現する場合、最大頻度誤差 Δ が一つ増えてしまう。言い換えれば、高コストなバーストトランザクションの処理をスキップする代わりに Δ を増やす処理を行っている。図 2 は、図 1 と同じデータセットを用いて得た Skip LC-SS 法の Δ 誤差の推移である。バーストトランザクションが出現したとき Δ の値が急激に上昇することがわかる。

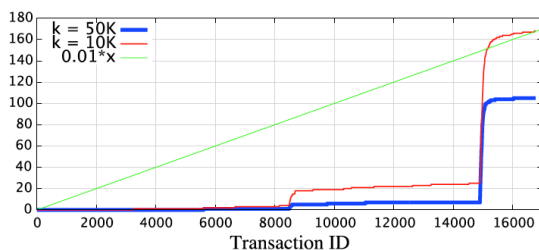


図 2: Skip LC-SS 法における Δ 誤差の推移

n 個のトランザクションを処理した後、もし $\epsilon \geq \frac{\Delta(n)}{n}$ が成り立つならば、Skip LC-SS 法は ϵ 近似を満たす。しかし、 ϵ が事前に定まっていた場合、頻度表サイズ

をどのように与えれば Skip LC-SS 法が ϵ 近似を満たすか明らかになっていなかった。本稿ではこの問題にアドレスする新しい性質として以下の定理を示す。

定理 1 任意の i において、 $k(i) = \frac{\epsilon \times \text{ave}(i)}{\epsilon(i)}$ とする。ただし、各時刻 j ($1 \leq j \leq i$) において $k(j) < |t_j|$ となるトランザクションを $t_{a_1}, t_{a_2}, \dots, t_{a_s}$ と書くと、 $\text{ave}(i) = \frac{\sum_{r \neq a_k} (2^{|t_r|} - 1)}{n - s}$, $\epsilon(i) = \frac{\epsilon \times i - s}{i}$ である。このとき Skip LC-SS 法は ϵ 近似を満たす。

4 圧縮技術を用いた性能改善

近似精度とメモリ消費量のトレードオフを緩和するため、飽和性に基づく圧縮技法 [1] を導入する。文献 [1] では、2 つのアイテム集合 α と β が飽和アイテム集合のとき、その共通部分である $\alpha \cap \beta$ が飽和アイテム集合となる性質が示されている。時刻 i までの全ての飽和アイテム集合を $\mathcal{F}_i$ とすると、 $\mathcal{F}_{i+1}$ は、この性質から、 t_{i+1} と $\mathcal{F}_i$ 中の各要素との共通部分のみに着目すれば構成することができる。頻度表サイズが制限される Skip LC-SS 法においては、 $\mathcal{F}_i$ の中で $\Delta(i)$ より大きな頻度を持つ飽和アイテム集合を保持することとなる。登録対象を飽和アイテム集合に限定することで、最大頻度誤差の急激な増加を抑制することが見込まれる。

5 まとめと今後の課題

本稿では、ストリームデータから頻出アイテム集合を抽出するリソース指向型計算法として Skip LC-SS 法について概説するとともに、その領域複雑性に関して最近得られた知見を示した。また ϵ 近似とメモリ消費とのトレードオフを緩和する方法として飽和性に基づく圧縮技術を Skip LC-SS 法の導入するアイデアを紹介した。飽和性を拡張した概念として代表パターンと呼ばれるものもある。圧縮効果の実証実験とあわせ代表パターンを絞りこむ検討を今後行う予定である。

参考文献

- [1] S.-J. Yen, C.-W. Wu, Y.-S. Lee, V. S. Tseng, and C.-H. Hsieh: A fast algorithm for mining frequent closed itemsets over stream sliding window, *IEEE Int. Conf. on Fuzzy Systems*, 27-30 (2011)
- [2] Y. Yamamoto, K. Iwanuma and S. Fukuda: Resource-oriented approximation for frequent item-set mining from bursty data streams, *ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'14)*, 165-179 (2014)