2G3-3

多重データストリーム中のバースト出現に対応したオンライン型頻出系列マイニング

An Efficient On-line Sequential Mining
for Dealing with Bursty Arrivals in Multiple Data Stream

伊藤秀志[*1] 岩沼宏治[*2] 山本泰生[*2]
Syuji Ito Koji Iwanuma Yoshitaka Yamamoto

[*1]山梨大学大学院コンピュータ・メディア工学専攻
Computer Science and Media Engineering, University of Yamanashi

[*2]山梨大学大学院医学工学総合研究部
Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi

We propose an efficient on-line algorithm for extracting frequent subsequences from a multiple-data stream. This algorithm solves the problem that a large amount of memories are suddenly consumed when bursty arrivals occurs. For an on-line algorithm, suppressing memory consumption is very important, thus, an on-line algorithm often takes a form of an approximation algorithm, where an error ratio should be guaranteed to be lower than a user-specified threshold value. Our algorithm is based on an extended version [1] of LOSSY_ COUNTING Algorithm [3]. It limits the available memory space and if it keeps not enough memory, then it expires lowest frequency candidates of frequent sequences from the memory, and stored instead new candidates arriving in a data stream. The proposed algorithm can be guaranteed to have important properties such as completeness and robustness under some conditions.

1. はじめに

本論文では, 一定量のメモリしか利用できない仮定の下, 誤差や損失を一部許しながらも頻出部分系列をオンラインで抽出する，高速かつ高効率なアルゴリズムを提案する．オンライン型頻出系列マイニングとは，データストリーム等の動的なデータから，頻出系列をユーザから要求された時点で即時抽出する技術を言う．データストリームとは絶え間なく到着するデータ系列の呼称で，特に複数データが同時に到着するものを多重データストリームと呼ぶ．オンライン型アルゴリズムは無限長のデータを扱うことを仮定するため，いかにメモリ使用量を抑えて処理を行うかが最大の課題である．しかし，多重データストリームでは複数のアイテムがバースト的に出現する場合があり，一時的に大量のメモリが必要となる．これは，メモリ使用量を抑えたいオンライン型アルゴリズムにとって致命的である．そこで，既存のオンライン型アルゴリズムを改良し，前述の問題を解決する手法を新たに提案する．実験を通して従来手法よりもメモリ使用量が大幅に削減できることを示す.

2. 準備

本論文で用いる表記法と用語の定義を以下に示す.

定義 1 すべてのアイテムの集合を $I = (i_1, i_2, \cdots, i_n)$ とする．**アイテム集合系列** (以下，系列と呼ぶ) とは，アイテム集合の並びであり，$S =< s_1 s_2 \cdots s_n >$ と表記する．各 $s_i (s_i \subseteq I, 1 \leq i \leq n)$ を S の**要素**と呼び，$(a_1, a_2, \cdots, a_m)$ と略記する．S の**系列長**を $|S|$ で表す．系列 $\alpha =< s_1 \cdots s_m >$ が系列 $\beta =< t_1 \cdots t_n >$ の**部分系列**であるとは，$s_1 \subseteq t_{j_1}$，$s_2 \subseteq t_{j_2}$，$\cdots$，$s_m \subseteq t_{j_m}$ を満たす整数 $1 \leq j_1 < j_2 < \cdots < j_m < n$ が存在する場合をいい，$\alpha \subseteq \beta$ と表す.

連絡先: 山梨大学大学院コンピュータ・メディア工学専攻 岩沼・山本研究室,
〒 400-0016 山梨県甲府市武田 4 丁目 4-37,
E-mail：g11mk006@yamanashi.ac.jp

定義 2 単一系列データベースとは，アイテム集合系列であり，マイニングの対象となるデータベースである．以下，系列データベースと呼ぶ.

以下，ストリームデータを系列データベースと呼ぶ.

定義 3 アイテム系列とは，アイテムの単一要素集合の系列 $< (a_1)(a_2) \cdots (a_n) >$ であり，アイテムの系列 $< a_1 a_2 \cdots a_n >$ と同一視する．本研究で抽出する部分系列は，すべてアイテム系列である.

定義 4 $S =< s_1 s_2 \cdots s_n >$ を系列データベース，$\alpha =< t_1 t_2 \cdots t_m >$ をアイテム系列とする．S 上の α の出現頻度関数 $F(S, \alpha)$ とは，整数値 $k (0 \leq k \leq |S|)$ を返す関数である．また，このとき α の**相対頻度** $R(S, \alpha)$ を以下のように定義する.

$$R(S, \alpha) = \frac{F(S, \alpha)}{|S|}$$

相対頻度 $R(S, \alpha)$ は，α が系列データベース S 中にどの程度の割合で出現するかを表し，1 以下の非負数を取る.

定義 5 N 個の要素からなる系列データベース $S =< s_1 s_2 \cdots s_N >$ を考える．このとき，部分系列 α が S 中に**頻出**であるとは，与えられた最小相対頻度 $\sigma (0 < \sigma < 1)$ に対し，$R(S, \alpha) \geq \sigma$ が成り立つことをいう.

アイテムの頻度を計算する場合，出現頻度関数はそのアイテムがデータベース中に単純に何回出現しているかを返せばよいだけだが，部分系列の場合，そのような単純な計算方法がない．そこで本論文では，出現頻度関数として**系列先頭頻度** [3] を導入する．系列先頭頻度は右逆単調性 [3] を持ち，重複数え上げが生じない頻度尺度である.

定義 6 系列データベース $S =< s_1 s_2 \cdots s_n >$ があるとき，S の i 番目の要素から始まる長さ k の部分系列を**ウィンドウ**と

呼び，win(S,i,w) と表記し，以下で定義する．また，このときの w を**ウィンドウ幅**という．

$$\mathrm{win}(S,i,w)=\begin{cases}(s_i\cdots s_{i+(w-1)}) & \text{if } i+(w-1)\le n,\\ (s_i\cdots s_n) & \text{otherwise.}\end{cases}$$

ウィンドウは，データベース中の部分系列を探すときの注目範囲にあたる．

定義 7 $\alpha =< a_1\cdots a_m >$ をアイテム系列，$\beta =< b_1\cdots b_n >$ をアイテム集合系列とするとき，$\alpha \lhd \beta$ を $a_1 \in b_1$ かつ $\alpha \subseteq \beta$ が成り立つ場合と定める．

定義 8 系列データベース $S =< s_1 s_2\cdots s_n >$，アイテム系列 $\alpha =< t_1 t_2\cdots t_m >$，ウィンドウ幅 w $(1 \le m \le w < n)$ に対し，S における α の**系列先頭頻度**H-freq(S,α,w) を以下で定義する．

$$\text{H-freq}(S,\alpha,w)=\sum_{i=1}^{n}\lambda(\mathrm{win}(S,i,w),\alpha)$$

ここで，λ は以下の関数と定める．

$$\lambda(< s_i\cdots s_n >,< t_1\cdots t_m >)=\begin{cases}1 & \text{if } < t_1\cdots t_m > \lhd < s_1\cdots s_n >\\ 0 & \text{otherwise}\end{cases}$$

オンライン型データマイニングでは，通常よく利用される逆単調性を利用した枝刈りを行うことはできないことに注意していただきたい．以後出現頻度関数 F は，系列先頭頻度を表すものと約束する．

3. 先行研究

本論文で提案するアルゴリズムは，先行研究である Lossy Counting 法の系列拡張法 [1] を基礎としている．

本章では先行研究の手法の概要を示す．Lossy Couting 法 [2](以下 LC 法) は，データストリームから頻出アイテム，もしくは頻出アイテム集合を近似抽出する手法である．これを拡張し，頻出系列を近似抽出するようにしたのが LC 法の系列拡張法である．

3.1 アルゴリズム

最小相対頻度 $\sigma(0<\sigma<1)$，許容誤差 $\epsilon(0<\epsilon<\sigma)$，ウィンドウ幅 $w(1<w)$ を受け取り，マイニング対象であるストリームデータ中から相対頻度 σ 以上で長さ w 以下の部分系列（頻出系列）を全てを抽出することを考える．先行手法 [1] では幾つかの非頻出な部分系列も抽出するが，読み込んだストリームデータの長さが N である場合，抽出される部分系列の出現頻度は少なくとも $(\sigma-\epsilon)N$ 以上であることが保証される．また，LC 法と同様，相対頻度が ϵ 以下であるといえる部分系列を保持しないことで，メモリ使用量の増加を抑制できる．

この手法ではウィンドウを単位時刻ごとスライドさせながら，ストリームデータ中に出現する部分系列の出現頻度を数える．アルゴリズムが保持する頻度表 D は，三つ組 $(\alpha,C(\alpha),T(\alpha))$ の集合である．α はウィンドウから抽出された系列，$C(\alpha)$ は α が D に登録されてからの出現頻度，$T(\alpha)$ は D に登録された時刻である．また，部分系列 α に対応する三つ組は高々一つである．

アルゴリズムはユーザからの出力要請があるまで，以下のステップを繰り返す．

1. 現時刻を t とする．ウィンドウ w を読み，$\alpha \lhd w$ となる α すべてに，以下の 2,3 の処理を行う．
2. α に対応する三つ組が頻度表 D にあれば，その $C(\alpha)$ を一つ増やす．
3. α に対応する三つ組が無ければ，$(\alpha,1,t)$ を D に登録．
4. D をチェックし，$C(\alpha)+\epsilon(T(\alpha)-1)\le\epsilon t$ となる三つ組を削除する．
5. 時刻を 1 だけ進め，ウィンドウをスライドする．

出力要請を受けた場合は，$C(\alpha)+\epsilon(T(\alpha)-1)\ge\sigma t$ となる部分系列 α を頻出であるとし，出力する．

このアルゴリズムでは以下が成り立つ．

定理 1 [1] 読み込んだデータ列の長さを N，最小相対頻度を σ，許容誤差を ϵ とするとき，出力された部分系列 α の真の出現頻度 $F(S,\alpha)$ は，$F(S,\alpha)\ge(\sigma-\epsilon)N$ を満たす．

定理 2 [1] 読み込んだデータ列の長さを N，許容誤差を ϵ，各ウィンドウで得られる部分系列の最大数を M とするとき，このアルゴリズムのメモリ空間使用量は，$O(\frac{M}{\epsilon}\ logN)$ である．

4. バースト出現時の問題

バースト出現とは，系列データベース中に，膨大な量のアイテムが一時的に集中して出現した状況を指す．バースト出現に遭遇すると，ウィンドウから膨大な量の系列が抽出されるため，短期間に大量のメモリが消費される．従来手法のメモリ消費抑制の仕組みでは，短期間におこる爆発的なメモリ消費に対応できない．

先行手法を用い，バースト出現を持つ系列データベースから頻出系列を全て抽出する実験を行った．対象データは，山梨大学ウェブページの約 2 週間分 (2005 年 9 月 18 日〜2005 年 10 月 2 日) のアクセスログである．閲覧者が要求したページデータをアイテムとし，要求時間順に並べたアイテム集合系列である．1 分間に要求された複数のアイテムをまとめて一つのアイテム集合の要素としている．系列長 19,466，アイテム延べ数 356,421，アイテムの種類数 9,961 である．また，系列の各アイテム集合の大きさの平均は約 18 である．この系列データベースを以後 S_{log} と表記する．

図 1 は，先行研究の手法によって S_{log} をマイニングしたときの頻度表サイズの時間推移である．パラメータはウィンドウ幅 w=3，最小相対頻度率 σ=0.1，許容誤差 ϵ=0.01 である．S_{log} を正確に計算すると，出現系列の種類は約 9 千万であるが，そのうちで頻出系列はわずか 5,976 個である．図 1 では，バースト出現のために爆発的に頻度表サイズが増えている部分が何か所かあり，瞬間的には最大で約 800 万を超えている．先行手法では 9 千万個の系列全てを記憶することはせず，その 10 分の 1 以下の 800 万個に抑制することに成功している．しかし，最終的には約 6000 個の頻出系列を抽出すれば良いことを考えれば，まだまだ無駄が多いと考えられる．頻度表の瞬間最大値を 800 万からどれだけ 6 千に近づけられるかが本論文での課題である

本論文では，バースト出現にしてもメモリを大量消費せず，抽出結果の完全性も保証できる，新しいオンライン型アルゴリズムを提案する．

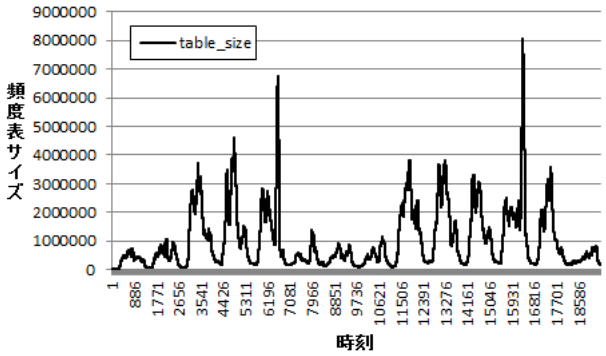

図 1: 先行研究手法における頻度表サイズの時間推移

5. 提案手法

本アルゴリズムは最小相対頻度 $\sigma(0 < \sigma < 1)$，許容誤差 $\epsilon(0 < \epsilon < \sigma)$，ウィンドウ幅 $w(1 < w)$ を受け取り，マイニング対象であるストリームデータ中から，ある条件の下で，全ての頻出部分系列（と幾らかの非頻出系列）を抽出する．

バースト出現に遭遇すると，短期間でメモリを大量消費してしまう．よって，システムの停止を防ぐためには使用できるメモリ空間を制限せざる負えないので，新たに頻度表サイズ (登録系列数) に上限値 k を設ける．しかし，上限値を設定すると頻度表サイズが上限値に達したときには，ウィンドウから抽出される新たな部分系列を登録できない．そのため新たな部分系列が頻出系列であった場合には，頻度表に残っていないため，抽出結果が不完全となる問題が発生する．

そこで，提案手法では頻度表サイズが上限値に達したときに， 表中から頻出系列となる可能性が最も低い低頻度な部分系列を削除し，代わりに未登録の系列を登録する．このとき，現在時刻 t までに表から交換により削除された系列中で，最大の出現頻度を**交換頻度** $\delta(t)$ として記憶しておく．ここで，関数 Δ を以下のように定義する．

定義 9 $\Delta(t) = \max(\epsilon(t-1), \delta(t))$

関数 max は引数のうち最大値を返す関数である．$\Delta(t)$ は，時刻 t までに頻度表から削除された系列中の最大の頻度を表す．

5.1 アルゴリズム

提案手法では LC 法の系列拡張法と同じく，ウィンドウを単位時刻ごとにスライドさせながら，ストリームデータ中に出現する部分系列の出現頻度を数える．アルゴリズムが保持する頻度表 D は，三つ組 $(\alpha, C(\alpha), \Delta(T(\alpha)))$ の集合である．α はウィンドウから得られた系列，$C(\alpha)$ は α が D に登録されてからの頻度，$T(\alpha)$ は D に登録された時刻，$\Delta(T(\alpha))$ は α が D に登録される前の考えうる最大の出現数を表す．以後，$C(\alpha)$ と $\Delta(T(\alpha))$ を足し合わせた値を**見積り頻度**と呼ぶ．なお，D 中の部分系列 α に対応する三つ組は高々一つである．なお，D が保持している三つ組の数を $|D|$ とする．

第 3.1 節のアルゴリズムと変わるのは処理 3, 4 と出力条件である．以下にそれらを示す．

3. (a) α に対応する三つ組が D に無く，$|D| < k$ ならば，D に $(\alpha, C(\alpha), \Delta(t))$ を登録する．
   (b) α に対応する三つ組が D に無く，$|D| \geq k$ ならば，以下の処理 i, ii, iii を行う．
   i. D 内から見積り頻度が最小である部分系列 β(複数ありえる) を探索する．
   ii. D から全ての β の三つ組を削除し，代わりに $(\alpha, C(\alpha), \Delta(t))$ を登録する．
   iii. $C(\beta)+\Delta(T(\beta)) > \delta(t)$ ならば，$\delta(t)$ を $C(\beta)+\Delta(T(\beta))$ で更新する．
4. D をチェックし $C(\alpha) + \Delta((T(\alpha))) \leq \Delta(t)$ を満たす三つ組を削除する．

   出力要請を受けた場合は，$C(\alpha) + \Delta(T(\alpha)) \geq \sigma t$ となる部分系列 α を頻出であるとし，出力する．

見積り頻度 $C(\alpha) + \Delta(T(\alpha))$ は，部分系列 α の考えうる最大の出現回数を表す．したがって α の真の頻度 $F(S, \alpha)$ に対して $F(S, \alpha) \leq C(\alpha) + \Delta(T(\alpha))$ の関係が常に成り立つ．よって，上記の出力条件により D に存在する $F(S, \alpha) \geq \sigma t$ を満たす α は全て出力される．

処理 4 では，見積もり頻度が $\Delta(t)$ 以下の系列の三つ組を削除することで，メモリ空間の消費を抑制している．逆に言えば，見積り頻度が $\Delta(t)$ より大きい部分系列の三つ組は D にすべて保持されているので，$\Delta(t) < \sigma t$ である限り D には頻出系列がすべて保持されていることがいえる．

本アルゴリズムは $\Delta(t) \geq \sigma t$ となった場合には，D から頻出系列の三つ組が削除される可能性があるため，完全性を保証できない．しかし，交換が起きなければ $\Delta(t)$ は一定数のままであり，σt は時間の経過により大きくなっていくため，時刻で再び $\Delta(t) < \sigma t$ となる可能性がある．理論上は，一旦 $\Delta(t) \geq \sigma t$ となって完全性が失われても，時刻経過で再び $\Delta(t) < \sigma t$ の状態になり，その状態が平均 $\frac{1}{\sigma}$ の間維持できれば，完全性が自律的に回復できる．

本アルゴリズムでは以下の定理が成り立つ．証明は紙面都合で省略する．

定理 3 現在時刻を t，最小相対頻度を σ，時刻 t の交換頻度を $\delta(t)$ とするとき，$\sigma t > \delta(t)$ の関係が時刻 1 から t の間維持されていれば，頻出系列を全て抽出できるという抽出結果の**完全性**が保証される．また，このとき抽出される部分系列の出現頻度は少なくとも $\sigma t - \Delta(t)$ 以上であるという**誤差保証**がされる．

定理 4 現在時刻を t，最小相対頻度を σ，時刻 t の交換頻度を $\delta(t)$ とするとき，$\sigma t \leq \delta(t)$ となった場合，真の頻度が $\delta(t)$ を超える頻出系列は全て抽出できるという**準完全性**が保証される．

6. 評価実験

提案手法をプログラムで実装し，系列データベースから頻出系列を全て抽出する実験を行った．対象データは 4 章で説明した S_{log} である．実験時の各パラメータは先行研究と比較するため，同じくウィンドウ幅 w=3，最小相対頻度率 σ=0.1，許容誤差 ϵ=0.01 とした．頻度表サイズの制限値 k を小さくしていき，抽出結果が完全となる k の下限値を調べた．

6.1 実験結果

実験結果を表 1 に示す．最左列は頻度表サイズ制限値 k の値である．サイズ制限値ごとに，頻出系列として抽出された部分系列の個数，処理終了時の頻度表サイズと処理中の瞬間最大

表 1: 提案手法における頻度表サイズ制限値と再現率の変化

頻度表サイズ制限値	抽出系列数	頻度表サイズ (瞬間最大値)	交換頻度	再現率	適合率
なし	5,976	171,341 (8,038,515)	0	100 %	99.76 %
180,000	5,981	159,687 (180,000)	1,290.37	100 %	99.68 %
150,000	5,983	148,206 (150,000)	1,565.13	100 %	99.64 %
120,000	118,866	118,866 (120,000)	1,967.77	100 %	5.01 %
110,000	100,255	100,255 (110,000)	2,168.09	99.9 %	5.94 %

注) 処理終了時の最小頻度 (σt) は 1946.6 である

値，処理終了時の交換頻度，そして抽出結果から求めた頻出系列の再現率と適合率がまとめられている．

結論として，S_{log} に対して再現率が 100 % となり，抽出結果が完全となるサイズ制限値の下限は 12 万程度であることがわかった．頻度表サイズの瞬間最大値は 12 万であるから，先行研究の手法の 800 万にくらべ，頻度表が使用するメモリ空間を約 $\frac{1}{67}$ へ大幅に抑制することができた．このときの交換頻度は 1968 であるが，これは最少頻度 1946.6 を上回っていることに注意されたい．この状況では実際の頻度が最小相対頻度以上かつ交換頻度以下である頻出系列が存在した場合，その三つ組が頻度表から削除される可能性がある．したがって制限 12 万では理論的には抽出結果に完全性を保証できない．しかし，興味深いことに制限 12 万のときの再現率は 100% であり抽出結果は完全であった．制限値 11 万で再現率が 100 %とならず抽出が不完全となったのは，処理終了時点で交換頻度 2,168.09 が最小頻度 1,946.6 を大きく上回り，頻出系列の一部が頻度表から削除されてしまったためである．

適合率について，制限 18 万,15 万において適合率は高い値を示しているが，制限 12 万以下では極めて低い．これは，制限 12 万以下での交換頻度が処理終了時の最小頻度 1,946.6 を超えているためである．アルゴリズムは頻度表から交換頻度以下の見積もり頻度を持つ系列を削除している．すなわち，表に保持されている系列の見積もり頻度は少なくとも交換頻度より大きい．よって，交換頻度が最小頻度を超えている状況では，頻度表に保持されている全ての系列が頻出系列と見なされるため，適合率が著しく低くなる．

6.2 考察

図 2 に提案手法が S_{log} を，頻度表サイズ制限 12 万でマイニングを行ったときの最小頻度，許容誤差，交換頻度の時間推移を表したグラフを示す．時刻を t とすれば，最小頻度は σt，許容誤差は ϵt，交換頻度は $\delta(t)$ を表している．縦軸は頻度値，横軸は時刻経過を表す．

交換頻度が一時的に最小頻度を上回り，完全性保証ができなくなる時点があるが，その後の時刻経過によって大小関係が逆転し完全性が自律的に回復している．また，交換頻度が最小頻度を上回った状態で処理が終了しているため，この抽出結果の完全性は理論的には保証することができない．しかし，実際には再現率は 100 %であり，抽出結果は完全である．以上のことから提案手法は理論値以上の頑健性を持つことが分かる．

提案手法では交換頻度が最小頻度以下であるならば完全性が保証できる．先行研究でのアルゴリズムの動作は，本アルゴリズムでの交換頻度が常に許容誤差 ϵt 以下である場合に相当している．すなわち，本アルゴリズムでは完全性を保証するための条件を，交換頻度が「許容誤差 ϵt 以下」から「最小頻度 σt 以下」に大幅に弛めていることになる．許される計算の誤差範囲が大幅に増え，完全性の保証範囲が拡大していると言える．

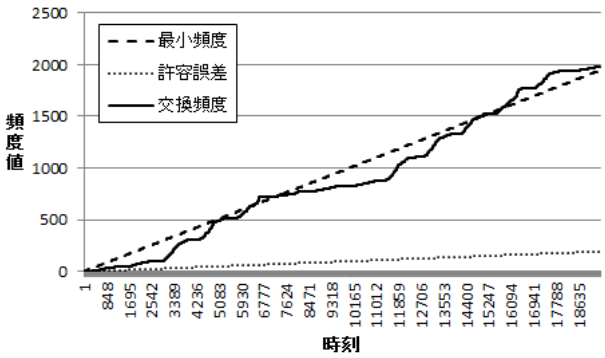

図 2: S_{log} におけるサイズ制限 12 万時の各値の時間推移

7. まとめ

本論文では多重データストリーム中のバースト出現に対応したオンライン型頻出系列マイニングを提案した．提案手法はある条件の下で抽出結果の完全性を保証できる．本手法の完全性は先行研究に比べ大幅に成り立つ範囲が広く，極めて有用である．評価実験の結果から，従来手法に比べてメモリ使用量を大幅に削減できることを示した．また実験等の結果から，アイテムのバースト出現によって抽出結果が一時的に不完全となっても，その後の時間経過によって完全性が自律的回復するなどの頑健性を持つことが分かった．

謝辞

本研究は一部，文科省科学研究費補助金（基盤 C：No.22500127）の援助を受けている．

参考文献

[1] 村田順平，岩沼宏治，石原龍一，鍋島英知: 精度保証付きオンライン型高速近似系列マイニング. 第 8 回情報科学技術フォーラム (FIT2009) 講演論文集，F-043， 2009.

[2] G.S Manku and R. Motwani: Approximate frequency counts over data streams. Proc. VLDM'02, pp346-357, 2002

[3] K. Iwanuma, R Ishihara, Y. Takano and H.Nabeshima: Extracting Frequent Subsequences from a Single Long Data Sequence: A Novel Anti Monotonic Measure and a Simple On-line Algorithm. Proc. of IEEE Inter. Conf. on Data Mining (ICDM 2005), pp.186-193,(2005)