1F3-04

密度基準に基づく高速な定量的相関規則導出手法

Technique for Fast Deriving Quantitative Association Rules Based on Density Standard

光永 悠紀*1 Yuki Mitsunaga　鷲尾 隆*1 Takashi Washio　藤本 敦*1 Atsushi Fujimoto　元田 浩*1 Hiroshi Motoda

*1大阪大学産業科学研究所高次推論方式
Institute of Scientific and Industrial Research,Osaka university

The application fields of basket analysis will be significantly extended, if the datasets including numeric attributes can be directly analysed by the technique. In our previous work we have developed quantitative basket analysis based on a density measure to derive the quantitative frequent item sets from the mixture of symbolic and numeric data. In this paper, an extention of the density measure to enhance the applicability and an improvement of the algorithm to save the computationnal time and memory consumption is proposed. The experimental evaluation of the improved approach shows excellent performance of both required computation time and memory consumption.

1. はじめに

データマイニングの分野においては，データ間の共起事象を相関規則として求めるバスケット分析が主流の 1 つになっている．バスケット分析で用いられる Apriori アルゴリズム [1] はアイテム間の共起相関を高速に抽出できる．しかし，アイテムで表現されるデータ構造には制限があり，数値を含むデータを Apriori アルゴリズムでは解析できない．数値を含むデータを扱うことができれば，バスケット分析を適用できる問題の範囲は飛躍的に広がるはずである．そこで，現在までに数値属性アイテムを含むデータについての相関ルール抽出を行う方法について様々な研究が行われてきたが，現実的な時間計算量で個別の相関規則に最適な数値区間分割の完全探索を行いルールを抽出する手法は存在しなかった．そこで，我々は数値を含むトランザクションデータについて現実的な時間計算量で個別の相関規則に最適な数値区間分割の完全探索を行う定量的多頻度アイテム集合導出法を提案した [2]．しかし，この手法にはまだ計算時間の削減・メモリ使用量の低減という課題が存在した．本稿ではこの課題を克服するために開発した改良アルゴリズムの提案及び，それに対して実際にデータに適用した性能評価を行う．

2. 定量的多頻度アイテム集合導出手法の課題

2.1 多頻度アイテム集合と多頻度領域

あるデータベース D において，最小支持度 (minsup) を超える頻度で各トランザクションに共起するアイテム集合を「多頻度アイテム集合 (frequent itemset)」と呼ぶ．Apriori アルゴリズムを < 年齢:32> のような数値アイテムを含めて多頻度なアイテム集合を探索できるように拡張する．まず，それぞれの数値アイテムの値を除いた部分，つまり属性を記号アイテムとして考え，最小支持度を超える多頻度アイテム集合を探索する．次に，得られた多頻度アイテム集合 f に含まれる全ての数値属性 A_i に注目し，f を含む各トランザクション t の持つ数値属性の値 $t(A_i)$ が密集している区間を抽出する．つまり，ある密集基準を満たし

$$\frac{|\{t|\bigwedge_{\forall A_i \in f} l_i \leq t(A_i) \leq u_i, f \subseteq t\}|}{|D|} \geq minsup$$

を満足する領域を抽出すればよい．ただし $l_i < u_i$，$A_i \in f$ である．これによって得られた領域を本稿では「多頻度領域 (frequent region)」と呼ぶ．

多頻度領域を探索するためには，トランザクションが密集していると判断する基準が必要である．そこで本稿では，各数値属性 A_i ごとに「それぞれのトランザクション t の持つ数値属性の値 $t(A_i)$ 同士が特定の値 Δ_i より離れていなければ近いとみなす」という基準を設定する．この Δ_i を数値属性 A_i に関する「許容距離 (permissible range)」と呼ぶ．

2.2 旧手法

多頻度アイテム集合に含まれる全ての数値属性に関して，トランザクション t から許容距離 Δ_i 以内であるトランザクションの集合を近傍集合 $E_f(t)$ とし，すべての t について求まった近傍集合 $E_f(t)$ の集合を ES_f とする．ここで，任意の $E_f(t_i)$, $E_f(t_j)$ ($\in ES_f$) の積集合が空集合でないとき，t_i と t_j を結合する．この近傍集合の集合 ES_f をもとにして得られる多頻度集合の候補 C_f は

$$C_f = \{E_f(t) \in ES_f | E_f(t) s\, are\, mutually\, connected\, in\, ES_f\}$$

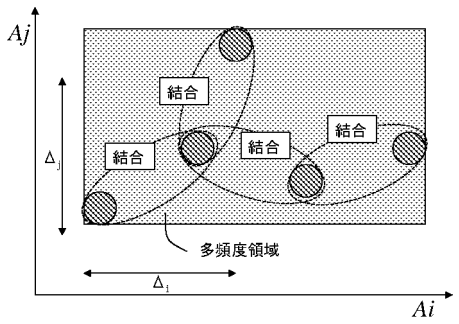
となる．このような C_f は ES_f から複数得られる．これによって生成された多頻度集合の候補 C_f に含まれる要素数が最小支持度以上であるとき，その多頻度集合の候補を多頻度集合 F_f と呼び，F_f における各属性 A_i の最大値と最小値で囲まれた範囲が多頻度領域 Id_f となる．

$$F_f = \{C_f | \frac{|C_f|}{|D|} \geq minsup\}$$

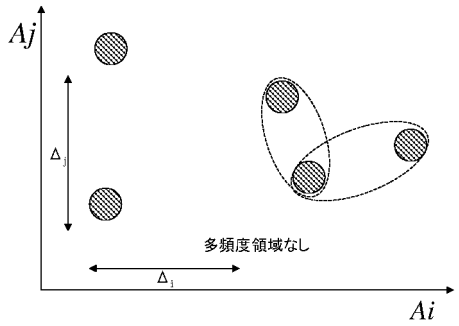
$$Id_f = \bigotimes_{A_i \in f} [\min_{t \in F_f}(t(A_i)), \max_{t \in F_f}(t(A_i))]$$

ただし，$\otimes$ は直積記号である．例えば 2 種類の数値アイテム A_i, A_j の多頻度領域を求めると図 1 のようになる．ただし，最小支持度は 4 とする．図におけるそれぞれの点をトランザクションとすると，(1) では各数値アイテムに対して許容距離 Δ_i 及び Δ_j 以下であるトランザクションを結合してゆけば多頻度集合の候補の要素数は 5 となる．よって，最小支持度を上回るので図のような長方形の多頻度領域が得られる．また，

連絡先: 大阪大学産業科学研究所
〒 567–0047 大阪府茨木市美穂ヶ丘 8-1
E-mail:mitsunaga@ar.sanken.osaka-u.ac.jp



（1）密集している場合



（2）密集していない場合

図 1: A_i, A_j における多頻度領域

（2）においては幾つかのトランザクションの結合はされるが，多頻度集合の候補の要素数が 3 であり最小支持度を下回るために多頻度領域は得られない．

3. 定量的多頻度アイテム集合導出手法の改良

本研究では定量的多頻度アイテム集合導出手法の性能向上のために, Apriori による多頻度アイテム集合候補作成手法の改良，多頻度領域導出手法の改良及び新しい密度基準の導入を行う．以下，これらの内容につき詳しく説明する

3.1 多頻度アイテム集合候補作成手法の改良

多頻度アイテム集合の候補の作成には Apriori アルゴリズムが用いられている．Apriori アルゴリズムは，要素数 1 の多頻度アイテム集合からはじめて，ボトムアップ的に要素数 k の多頻度アイテム集合から要素数 $k+1$ の多頻度アイテム集合の候補を作り出すことを行う．具体的に述べると $k-1$ 個の要素が共通な要素数 k の 2 つの多頻度アイテム集合

$$P_k = \{A_1, A_2, \cdots, A_{k-1}, A_k\}$$
$$Q_k = \{A_1, A_2, \cdots, A_{k-1}, A'_k\}$$

より要素数 k+1 の多頻度アイテム集合の候補を上記の 2 つのアイテム集合の和集合

$$\begin{aligned} R_{k+1} &= P_k \cup Q_k \\ &= \{A_1, A_2, \cdots, A_{k-1}, A_k, A'_k\} \end{aligned}$$

とする．さらに候補を絞り込むために，このようにして候補にあがった要素数 $k+1$ のアイテム集合に対して，要素数 k のアイテム集合からなる各部分集合が全て多頻度アイテム集合として存在しているかどうかをチェックする．これは多頻度アイテム集合であるためには，その部分集合は全て多頻度アイテム集合でなければならないためである．要素数 k の多頻度アイテム集合は前段階の処理で全て取り出されており既知であるので，このチェックは効率的に実行することが可能である．このようにして残った各候補についてのみ，実際の支持度をデータから検索して計算することで効率的に多頻度アイテム集合を求めることができる．

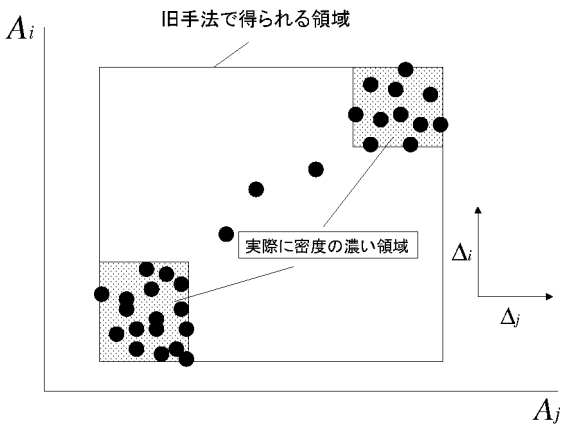


図 2: 旧手法で得られる領域の問題点

ここで，多頻度領域の性質を用いることにより、より効率的に候補生成を行う手法を提案する．通常の Apriori では，候補を作成したあとに要素数の一つ少ないアイテム集合が全て多頻度アイテム集合であるかをチェックする．しかし，数値属性アイテムを含む多頻度アイテム集合の候補を作成する場合を考えると，P_k，Q_k に共通する数値アイテムそれぞれについて，そのアイテムを属性とする数値軸上に P_k 及び Q_k の多頻度領域をそれぞれ正射影した区間がトランザクションデータを含まない場合，P_k，Q_k を結合してできる多頻度領域は存在しないという性質がある．よって，この性質を用いて多頻度アイテム集合候補の生成の枝狩りを行うことにより計算時間を短縮することができる．

このチェックの簡単化のために各多頻度領域に，FrequentID と ParentID を付加する．FrequentID は各多頻度領域に付けられたユニークな ID であり，ParentID は候補作成の際にその領域の元となった多頻度領域の FrequentID とする．P_k と Q_k の多頻度領域に ParentID が一致するするものが一つもなかった場合次のレベルの候補は生成せず，一つでもあった場合に前述のように共通データの存否のチェックを行う．

最後に，作成された多頻度集合の候補に対して，全てのアイテムを記号アイテムとした場合に P_k，Q_k を両方含むトランザクション数をカウントしこれが最小支持度を超えない場合は更なる枝狩りを行う．

3.2 多頻度領域導出手法の拡張

以前の手法は各トランザクションについて全ての数値属性アイテムにおいて許容距離 Δ_i 以内のトランザクションからなる近傍集合を求め，それをマージしていくことで多頻度領域を求めていた．この手法だと近傍集合を求める際に，各トランザクションについて近傍であるトランザクションの情報をメモリ上に全て記憶するのでメモリの使用量が非常に大きくなっていた．また，距離 Δ 以内のものを全て結合していき領域を求める方法には，実際には密度が薄い部分まで同じ領域として取り込んでしまい，図 2 に示すような分布の少ない領域が続くよう

なデータでは適切な領域を探索できないという問題があった．この問題に対し，以下の改良を施した．

密度の濃い領域を適切に探索するために，Δ に加えて新たな密度の評価の指標として最小データ点数 (MinPts) を採用した．まず，MinPts を用いるにあたって必要な定義を導入する．

定義 1:(core)

あるトランザクション t について許容距離 Δ 以内に MinPts 以上のトランザクションが存在する時トランザクション t を core と呼ぶ．

定義 2:(directly density-reachable)

トランザクション t_i が core であり，トランザクション t_i からトランザクション t_j までの距離が Δ 以内であるとき，t_j は t_i から directly density-reachable であるとする．

定義 3:(density-reachable)

トランザクションの集合 t_1,..., t_n において，任意の x について t_{x+1} が t_x から directly density-reachable であるとき，core ではないトランザクション t_y が t_1,..., t_n のうちのいずれかからの距離が Δ 以内であった時，t_y は t_1,..., t_n から density-reachable であるとする．

定義 4:(density-connected)

ある 2 つのトランザクション t_i, t_j がトランザクション t_k から density-reachable であるとき t_i と t_j を density-connected であると呼ぶ．

以上の定義の下で，多頻度領域を導出する拡張アルゴリズムを定式化した．要素数 l の多頻度アイテム集合 f_l に含まれる数値アイテム属性の集合を $f_{n(f_l)}$ とする．

1 データ D から f_l を含むトランザクションを抜き出したリストを L_{f_l} とする．

2 $A_i \in f_{n(f_l)}$ である各数値アイテム属性 A_i について，

- **2-1** A_i について L_{fl} をソートする．
- **2-2** L_{fl} を A_i について互いに density-connected である極大なトランザクション集合に分割し，各々を $L_{f_{li}}(i = 1,2,\ldots)$ とする．
- **2-3** 各 $L_{f_{li}}$ について $L_{f_l} = L_{f_{li}}$ として 2 を再帰的に実行し，$f_{n(f_l)}$ の中のどのアイテムに対して操作を行っても L_{fl} に含まれるトランザクション数に変化がなくなるまで操作を繰り返す．

この操作を行って残ったリストの長さが最小支持度を超えていた場合，その各属性について最小値と最大値を求めたものが多頻度領域となる．

トランザクション数を N とすると，以前の手法では各トランザクション毎に最悪の場合 $N-1$ 個の近傍集合を記憶するためのメモリが必要であったが，改良手法では最悪でも全てのトランザクションに共通した (全てのトランザクションがリストに含まれる場合)N 個の要素を記憶するための領域しか必要としない．このためメモリ使用量の大幅な低減が可能である．また，MinPts により距離 Δ 内に MinPts 個以上のトランザクションが密集した密集性の高い領域を見つけることができる．

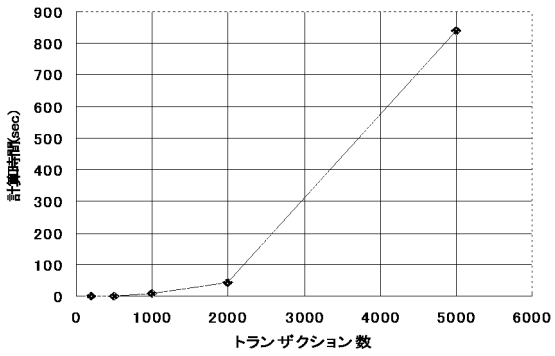


図 3: 手法改良前のトランザクション数に対する計算時間

4. 性能評価

4.1 データの概要

パラメータの変化による性能の評価を行うために人工データを用いて性能評価を行った．人工データはトランザクション数，トランザクションの平均サイズ，アイテムの種類数，人工的に埋め込む多頻度パターンの種類数を指定して生成した．今回は，トランザクション数の大きなデータに適用できるように計算時間の短縮とメモリ使用量の低減を目的として手法の改良を行ったので，トランザクション数の増加に対する計算時間及びメモリ使用量の変化に着目し性能評価を行う．今回は MinPts は 1 として評価を行った．

4.2 分析結果と考察

図 3 に改良前の手法によるトランザクション数と計算時間の関係のグラフを，図 4 に今回提案した手法のトランザクション数に対する計算時間・メモリ使用量の変化のグラフを示す．改良前の手法が $O(N^2)$ 程度の必要計算時間を示しているのに対し、今回の提案手法は $O(N)$〜$O(N\log N)$ 程度の計算量を示している．またメモリ使用量は，ほぼ $O(N)$ である．以下にこれらに関する理論的考察を示す．

前述の拡張アルゴリズム中のステップ 1 は明らかに $O(N)$ である．ステップ 2-1 はリスト L_{f_l} のソートであるが，高々 $|L_{fl}| \sim N$ であるので $O(NlogN)$ である．ステップ 2-2 では各属性 A_i の数直線上でトランザクションの分布に基づき，各トランザクションの周囲に MinPts 個以上のトランザクションが存在する多頻度領域を求める．各トランザクションについて MinPts 個の他のトランザクションのカウントが必要であるため，$f_{n(f_l)}$ に含まれる数値属性数を d とすると，このステップは最大 $O(dMinPtsN)$ の計算時間がかかり，N については $O(N)$ である．ステップ 2-3 では，新たに作成されたリストに対して多頻度領域が収束するまで，ステップ 2 が深さ優先探索で再帰的に実行される．途中 A_1 から A_k まで再帰的処理が終わっても収束しない場合には，再度 A_1 から再帰的処理が続けられる．ここで，多頻度領域が収束するまでに最悪どのくらいの計算量が必要か考察する．$f_{n(f_l)}$ に含まれる属性全てについてステップ 2-1〜2-2 を行うことを 1 サイクルとすると，収束するまでに計算量が最も多くなるのは 1 サイクルでリストから一つずつトランザクションが減っていく場合である．このとき収束するまでに N サイクルかかるため，計算時間は $O(dMinPtsN^2)$ となる．しかし，このようなことは起こり難くく，実際には各サイクルでトランザクションの一部がリストから減っていくことが殆どである．各サイクルでトランザク

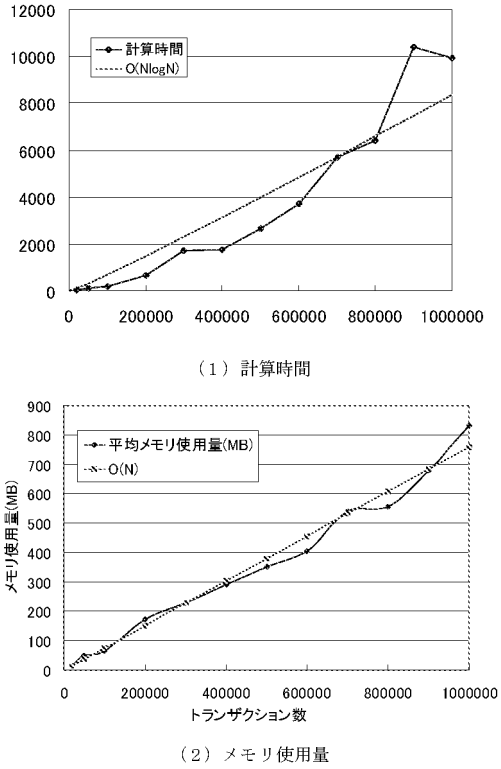


図 4: トランザクション数に対する計算時間・メモリ使用量の変化

ションが減っていく割合の期待値を α $(0 < \alpha < 1)$, 収束するまでに必要なサイクル数を m とする．探索は領域内のトランザクション数が最小支持度未満になると打ち切られるため，m は高々 $minsup \simeq \alpha^m N$ を満たす程度である．この等式を m について解くと $O(logN)$ となる．よって全体として計算時間は $O(dMinPtsNlogN)$ となり N に対しては $O(NlogN)$ である．この結果は図 4 (1) と整合する．

次に，メモリ使用量に関して考察する．メモリは主に各トランザクションに含まれるアイテムの記憶と，求められた多頻度アイテム集合及び多頻度領域の記憶に使われる．各トランザクションに含まれる平均のアイテム数を s とするとメモリ使用量は sN に比例して増えていく．一方，多頻度アイテム集合・多頻度領域の記憶に用いられるメモリ量は，トランザクション数に直接依存しない．よって，メモリ使用量は全体として $O(N)$ であるといえる．この結果は図 4 (2) と整合する．

以上のように，本稿で提案した手法を用いるとトランザクション数 N に対して，計算時間 $O(NlogN)$，メモリ使用量 $O(N)$ で定量的多頻度アイテム集合を導出することが可能である．

5. むすび

本研究では以前考案した定量的多頻度アイテム導出手法の高速化・メモリ使用量の低減を目的とした改良手法の提案を行った．テストデータを用いて提案手法の性能評価を行った結果，実行速度・メモリ使用量共に大幅に改善されることがわかった．

今回提案した手法の枠内では，定量的多頻度アイテム集合を導出することはできるが，アソシエーションルールを導出することはできない．このための拡張が課題として挙げられる。

参考文献

[1] R.Agrwal and R.Srikant, Fast algorithms for mining association rules, Proceedings of the 20th VLDB Cnference,pp. 487-499, 1994.

[2] T.Washio , A.Fujimoto and H.Motoda, Extention of Basket Analysis and Quantitative Association Rule Mining, 人工知能学会　知識ベースシステム研究会（第 67 回）SIG-KBS-A403, pp. 117-122 (2004).

[3] R.Srikant and R.Agrawal, Mining Quantitative Association Rules in Large Relational Tables, Proc. of the 1996 ACM SIGMOD International Conference on Management od Data, pp. 1-12 (1996).

[4] K.Wang , S.H.W.Tay and B.Liu, Interestingness-Based Interval Merger for Numeric Association Rules, Proc. of 4th Int. Conf. on Knowledge Discovery and Data Mining (KDD), pp. 121-128 (1998).

[5] T.Fukuda , Y.Morimoto , S.Morishita and T.Tokuyama, Mining Optimised Association Rules for Numeric Attributes, Proc. of 15th ACM SIGACT-SIGMOD-SIGART Symposium on Principle of Database System, pp. 182-191 (1996).

[6] R.Rastogi and K.Shim, Mining Optimized Association Rules with Categorical and Numeric Attributes, Proc. of the 14th International Conference on Data Engineering, pp. 503-512 (1998).

[7] T.Fukuda , Y.Morimoto , S.Morishita and T.Tokuyama, Constructing Efficient Decision Trees by Using Optimised Numeric Association Rules, Proc. of VLDB96 , VLDB Endowment., pp. 1-10 (1996).

[8] D.Z.Chen , 全 眞嬉 , 加藤直樹 , 徳山 豪, 高次元ピラミッド構築問題とデータマイニングへの応用, 情報処理学会誌 アルゴリズム Vol.88,No.11, pp. 71-78 (2003).

[9] N.Katoh, Finding an Optimal Region in One- and Two-Dimensional Arrays, IEICE TRANS. INF. & SYST.,VOL.E83-D, NO.3, pp. 438-446 (2000).

[10] J.Wijsen and R.Meersman, On the Complexity of Mining Quantitative Association Rules, Data Mining and Knowledge Discovery,Vol.2, No.3, pp. 263-281 (1998).

[11] http://www.ics.uci.edu/~mlearn/MLRepository .html, University of California Irvine Machine Learning Repository.