3C4-3

組合せ集合の meet 演算による共通因子抽出

竹内 文登 *1　安田 宜仁 *2　湊 真一 *1*2
Fumito TAKEUCHI　Norihito YASUDA　Shin-ichi MINATO

*1北海道大学工学部 情報エレクトロニクス学科
Department of Electronics and Information Engineering,School of Engineering,Hokkaido University

*2JST ERATO 湊離散構造処理系プロジェクト
JST ERATO Minato Discrete Structure Manipulation System Project

In frequent pattern mining, frequent patterns tend to be numerous so that humans cannot check the output. It is desirable that small set of high-quality patterns from numeraous frequent patterns are extracted. For such purposed, displaying closed pattern is widely used because of its good properties; it produces small number of representative patterns and it can be processed in proportion to the number of closed patterns, not that of frequent patterns. However, patterns containing many items tend to not be included in closed patterns. Thus one must reduce the minimum support to obtain patterns with many items. In this paper we proposed a new extraction method based on 'meet' operation. We use the meet operation defined against combination sets, which can be concidered as a transaction database. The method provides the following two features. (i) the output is always a subset of closed patterns and patterns containing small items are tend to be excluded. (ii) the processing time does not depend on the number of frequent patterns.

1. はじめに

頻出パターンマイニングは，データマイニングの最も主要なトピックの一つであり,「データベース中に高頻度に存在するパターンを全て列挙する」というものである．1994 年の Agrawal[Agrawal 94] 等による Apriori アルゴリズムの研究を機に盛んに研究されるようになり，様々なアルゴリズムが提案されている．

大規模なデータベースに対して，その頻出パターン集合は巨大であるため，そのうち制約を満たすものだけを抽出することが求められる．従来の手法は，データベース中に出現した回数 (サポートという) を制約とする手法と，飽和や極大なパターンを代表元とするような，パターン間の関係を考慮した制約を設ける手法を組合せた手法が提案されている [Agrawal 94], [Uno 03]．これらの手法は，サポートを超えるパターンを列挙したのち，パターン間の関係を考慮した制約を満たすものを求める手法と，2 種類の制約を満たすパターンを一気に求める手法に大別され，本稿における提案手法は，後者に分類される．以下に例とともに概略を示す．

図 1. のようなコンビニ等の購買履歴から「よく同時に購入される商品のパターンが知りたい」とする．近年では，ホットスナックのチキン専用のバンズなどが売られており，チキンとバンズは同時に購入される．中には，(珍しいが) 専用バンズだけを購入する客も存在する．このとき頻出パターン集合において,「バンズとチキン」の組合せの頻度は「バンズ」だけの頻度より小さいたため，頻出パターン集合から望まれる組合せ，つまり「バンズとチキン」の組合せが見つかりにくくなる．これは，データベース中にそのパターンが何回現れたかを指標としたためであり，パターンのアイテム数が多ければ多いほどその頻度は小さくなる．

これはアイテム数が多いパターンが知りたいときに望ましくない性質である．改善する 1 つの手法として，頻出パターン集合に現れる組合せの順位付けを変える方法が考えられる．この例では,「バンズとチキン」のパターンの方が高順位となるような順位付けが望まれる．これを実現するためのアイディアとしては，従来では「バンズとチキン」を含む組合せがあったとき,「バンズ」と「チキン」それぞれのパターンも出現数として数えたのに対し,「バンズとチキン」のパターンのみを出現数として数えるというものである．

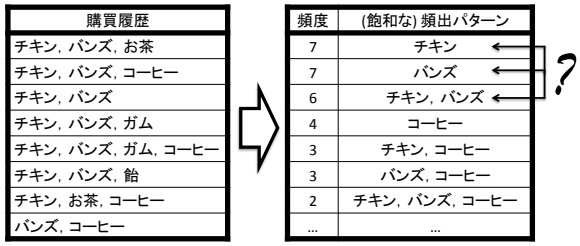

図 1: 購買履歴における頻出パターン集合の例

我々はこの順位付けを行うため，Knuth により提案された meet 演算 [Knuth 09] と呼ばれる演算に着目した．この演算は，データベース中の 2 つのデータの共通因子を全て列挙する演算であり，上記のアイディアを実現すると思われる．本稿では，データベースに対しこの meet 演算を用いて共通因子を抽出することで，データベース中の頻出パターンの候補を列挙することを目的とする．

一方で，このような組合せの集合 (以下，組合せ集合) は，ゼロサプレス型二分決定グラフ (*ZDD : Zero-Suppressed BDD*)[Minato 93] と呼ばれるデータ構造を用いて効率よく扱うことができる．ZDD は組合せ集合を圧縮して表現するだけでなく，meet 演算などの組合せ集合演算は ZDD 間の演算で行えるため，高速かつ小メモリで組合せ集合を処理できることが知られている．

本稿では，データベースを ZDD を用いて処理し，meet 演算を用いて共通因子の抽出を行う手法を提案する．また，提案手法と従来の頻出パターン集合との異なる順位付けの違いについてと，ZDD 上における meet 演算の性能について実験と考察を行う．

連絡先: 竹内 文登，北海道大学工学部 情報エレクトロニクス学科，fumito@alg.ist.hokudai.ac.jp

2. 準備

2.1 組合せ多重集合と頻出パターンマイニング

アイテム集合 $I = \{1, \ldots, n\}$ が与えられたとき，その部分集合 $C \subseteq I$ を「組合せ」という．複数の組合せからなる集合を「組合せ集合」という．さらに同じ組合せを重複して複数もつ場合,「組合せ多重集合」という．以下では，組合せは括弧を用いずに表すことにする．例えば，a と b と c からなる組合せと，c と d からなる組合せを要素とする組合せ多重集合 F は $F = \{abc, cd\}$ と表現する．このとき，組合せの個数を $|F|$ で表す．

本稿では，トランザクションデータベースを各トランザクションを組合せとした組合せ多重集合として扱う．組合せ多重集合における頻出パターンマイニングに対して一般的な定式化を行う．組合せ $C \subseteq I$ に対して，C を含む組合せ多重集合 F の組合せを出現と呼び，C の出現の集合を $Occ(C) = \{K \mid C \subseteq K, K \in F\}$ とする．定数 α (サポートという) に対して，$|Occ(C)| \geq \alpha$ を満たす C 中を頻出組合せという．このとき，頻出な組合せ C が他の頻出な組合せに含まれないならば，C は極大頻出組合せという．また，C が $Occ(C') = Occ(C)$ となる C' を全て含むとき，C は飽和であるという．

頻出パターン集合を求めることは，組合せ多重集合 F とサポート α が与えられたとき，F 中に少なくとも α 回以上現れる部分組合せを列挙することである．その際，飽和な組合せだけ，または極大な組合せだけを列挙する手法も考案されている [Uno 03].

2.2 組合せ多重集合と meet 演算

組合せ多重集合には，和集合や共通集合などの一般の集合代数の演算に加えて，本稿で扱う meet 演算 [Knuth 09] を定義することができる [*1]．meet 演算は組合せ多重集合の二項演算として定義される．本稿では演算子として "$\sqcap$" を用いる．組合せ多重集合 F,G に対して，F と G の meet 演算の結果は次式で定義される．

定義 1 *meet* 演算

$$F \sqcap G = \{\alpha \cap \beta \mid \alpha \in F, \beta \in G\}$$

これは，定義より組合せ多重集合 F と G の任意の組合せのペアに対する共通部分を求める演算である．例えば，$F = \{abcd, bcde\}, G = \{abc, bce\}$ に対して，

$$\begin{aligned} F \sqcap G &= \{abcd \cap abc, abcd \cap bce, bcde \cap abc, bcde \cap bce\} \\ &= \{abc, bc, bc, bce\} = \{abc, 2bc, bce\} \end{aligned}$$

となる．

また，F と G の meet 演算の結果 $F \sqcap G$ には，次のような特徴がある．

1. $|F \sqcap G| = |F| \times |G|$ である．
2. $F \sqcap G$ のうち係数の大きい組合せは F と G に共通して多く現れている組合せである．

1. これは F のそれぞれの組合せに対して，G のすべての組合せとの共通部分を計算しており，定義より明らかである．

2. $F \sqcap G$ には同一の組合せが複数現れる場合がある．このとき，$F \sqcap G$ 中におけるその組合せの係数は，どれだけのペアの共通部分がその組合せとなるかを示している．つまり，$F \sqcap G$ のうち係数の大きい組合せは F と G に共通して多く現れているということができる．このとき，ペアワイズで共通組合せを求めるとき，その部分組合せの係数は増えないという性質がある．先ほどの例において，$F \sqcap G = \{abc, 2bc, bce\}$ であったように，bc の係数が 2 であるのに対し，b や c の係数は 0 である．一方，F と G が共通パターンを持たないとき，$F \sqcap G$ は空集合の組合せだけからなる集合となる．

組合せ多重集合は，ZDD を拡張した「ZDD ベクトル」(ZDDV)[湊 06] と呼ばれるデータ構造を用いて表現することができる．また組合せ多重集合の演算も，ZDDV の演算で効率よく計算することができる．

3. meet 演算を用いた共通因子抽出

本節では，meet 演算を用いて組合せ多重集合から共通因子を抽出する方法について説明する．この手法では，頻出パターンを全て列挙する代わりに，以下で説明する組合せ多重集合を求め，有用なパターンを見つけ出すものである．

meet 演算の性質を利用することで，組合せ多重集合 F から F 自身に共通して現れるパターン，つまり共通因子を取り出すことができる．具体的には，F と自分自身との meet 演算，つまり $F \sqcap F$ を求めるのである．この集合を求めることで，F に複数回現れる共通因子を抽出することができる．本稿では組合せ多重集合 F がデータベースとして与えられたとき，その共通因子を求める手法として，$F \sqcap F$ を求めることで，組合せ多重集合 F の共通因子を抽出する手法を提案する．

組合せ多重集合 $F \sqcap F$ には，次の特徴がある．

1. $|F \sqcap F| = |F|^2$ である．
2. $F \sqcap F$ に多く現れる組合せは，F に多く共通して現れる組合せである．
3. $F \sqcap F$ に属する組合せは全て飽和である．
4. サポート 1,2 の極大な組合せは $F \sqcap F$ に属する．

以下で各特徴について述べる．

1. 従来法においては，頻出パターン集合の要素数はアイテム数に対して最悪指数的であった．しかし，組合せ多重集合 $F \sqcap F$ は，組合せの総数 N に対して $O(N^2)$ の要素数であり，従来法よりも比較的小さな解集合を求めている．

2. この性質から，F に多く含まれる共通パターンは，$F \sqcap F$ においてその係数が大きい傾向があるといえる．さらに前述した通り，F 中の組合せの任意のペアの共通組合せを求める際に，その部分組合せは $F \sqcap F$ には追加されない．この性質は，1. 節で述べた「チキンとバンズ」のような例に対して有効であると考えられる．

3. これは，meet 演算の定義から明らかであり，組合せ $C \in F \sqcap F$ に対して，その出現 $Occ(C)$ のうち 2 つの組合せの共通組合せが C となる組合せが存在する．このとき，$Occ(C)$ 中で頻出な組合せは C の部分集合のみであるので，C は飽和である．このことから，従来の頻出パターン集合において，サポートが 1 で，かつ飽和な組合せの組合せ多重集合を $Freq_{clo}(F, 1)$ とすると，これは $F \sqcap F$ を包含することが示される．つまり，

$$F \sqcap F \subseteq Freq_{clo}(F, 1)$$

である．しかし，一般に逆は成立しない．例えば反例として，組合せ多重集合 $F = \{abc, abd, acd\}$ 対して，$F \sqcap F =$

*1 Knuth は meet 演算を組合せ集合における演算として定義したが，組合せ多重集合の演算に拡張することができる．

$\{ab, ac, ad, abc, abd, acd\}$ であるが，a も飽和な組合せであるので，$a \in Freq_{clo}(F, 1)$ であり，

$$Freq_{clo}(F, 1) \not\subseteq F \sqcap F$$

が導かれる．

4. 組合せ多重集合 F に対し，サポートが 2 の極大な組合せ多重集合を $Freq_{max}(F, 2)$ とする．サポートが 2 の極大な組合せとは，F に 2 回以上現れるパターンであり，他の 2 回以上現れるパターンに含まれないものである．つまり，F 中の 2 つの組合せの共通部分集合になっている．$F \sqcap F$ は，F 中の任意の 2 つの組合せの共通部分集合になっているので，極大な組合せは $F \sqcap F$ に属する．つまり， 次の式が成り立つ．

$$Freq_{max}(F, 2) \subseteq F \sqcap F$$

また，$Freq_{max}(F, 1) \subseteq F \sqcap F$ も明らかである．

特徴 3,4 をまとめると以下の式が成り立つ．

$$Freq_{max}(F, 2) \subseteq F \sqcap F \subseteq Freq_{clo}(F, 1)$$
$$Freq_{max}(F, 1) \subseteq F \sqcap F \subseteq Freq_{clo}(F, 1)$$

また，$F \sqcap F$ には必ず F 自身の組合せが含まれ，つまり 1 度しか現れない組合せも含まれている．$F \sqcap F$ と F の差集合 $(F \sqcap F) \setminus F$ を計算することで,「少なくとも 2 回以上 F に現れる組合せ」という性質をもつ組合せ多重集合を求めることができ，以下の式が成り立つ．

$$Freq_{max}(F, 2) \subseteq (F \sqcap F) \setminus F \subseteq Freq_{clo}(F, 2)$$

以上の特徴から，提案手法は従来の頻出パターン集合のうち，飽和なものを幾つかを抽出する方法といえる．提案手法で求める集合に属する共通パターンの係数は，従来の頻度とは異なるものであり，それは組合せとしての頻度をより重視したものである．また，求める集合の組合せの個数は元の組合せ多重集合の要素数の 2 乗で抑えられる．

4. 実験

本節では，提案手法である組合せ多重集合 F から共通因子を抽出する方法について，以下の 2 つの実験 (以下，実験 1, 実験 2) を行った結果について説明し，考察を行う．またいずれの実験においても，組合せ多重集合を処理するためのデータ構造として ZDDV を用い，ZDDV が実装された VSOP[湊 05] と呼ばれる組合せ多重集合処理ソフトを使用した．VSOP は，組合せ多重集合を ZDDV で圧縮して表現することができ，かつ提案手法における一連の操作を効率よく ZDDV 上で行うことができる．

それぞれの実験の目的は以下のとおりである．

- 実験 1 では，従来法と提案手法におけるそれぞれの頻出パターン集合中の，組合せの順位付けの違いを調べる．
- 実験 2 では，ZDDV 上における meet 演算の計算速度を調べる．

4.1 実験準備

実験には，Mac OS X, 3.5GHz Intel Xeon E5, 主記憶 64 GB のマシンを用いた．

実験 1 では，人工データに対して実験を行う．具体的には，組合せ多重集合 F に対して $(F \sqcap F) \setminus F$ を求め，この集合における頻度による順位付けと従来手法の順位付けを比較する．作成したデータは，1000 個の組合せからなり，そのうち，800 個の組合せはアイテム a, b, c を含み，100 個の組合せは, アイテム a, b, c, d を含み，残りの 100 個はアイテム a, b を含む組合せであり，それぞれ全ての組合せが他の組合せと重複しないように，異なる a, b, c, d 以外のアイテムを 1 つもつ．

表 1: 提案法と従来法の順位付けの比較

従来手法 (飽和)		提案手法	
F 中の頻度	組合せ	$F \sqcap F$ 中の頻度	組合せ
1000	ab	799200	abc
800	abc	189900	ab
100	abcd	9900	abcd

実験 2 に用いたデータは，国際会議 FIMI-2003 のベンチマークデータ [Goethals 03] から抜粋した，チェスのデータ (chess) で，各組合せが 1 つのチェスの盤面に対応している．アイテムの種類数は 95 で，組合せ総数は 3196 である．従来法の実装として LCM (Linear time Closed itemset Miner) over ZDD[Minato 08] と Eclat (Equivalence CLAss Transformation)[Yu 14] と呼ばれる手法を用いて実験を行う．これは，どちらも飽和または極大頻出パターン集合を求める手法である．LCM over ZDD は飽和頻出パターン集合を計算し，それを ZDDV で表現し頻出パターン集合を処理する方法である．この方法を用いて実験 2 では，組合せ多重集合から同数の共通パターンを取り出すまでの時間をそれぞれ計測し比較を行う．

4.2 実験結果と考察

実験 1 の実験結果を表 1 に，実験 2 の結果を図 2 と図 3 に示す．

表 1 に実験 1 の結果を示す．実験 1 に使用したデータでは abc の組合せが 800 回と多く現れており，ab や $abcd$ が 100 回現れている．従来の方法ではこれらのうち ab は 1000 個のすべての組合せに現れるので，頻度が一番高くなっており，順に abc，$abcd$ の頻度が高い．これは，ある頻出パターンの部分集合も頻出パターン集合に現れる場合は，部分集合の頻度の方が必ず大きくなる例である．一方提案手法は，meet 演算が組合せのペアワイズの共通部分の列挙であったことから，アイテム数が多い組合せが上位になる場合があり，このデータにおいても，ab よりも abc の方が上位にきている．これは，ペアワイズの共通部分として ab より abc の方が多く現れているからであり，既存手法では得られない順位付けをすることに成功しているといえる．このデータを購買履歴と思うと，商品 a, b, c の組がよく同時に購入されることが分かる．

図 2 は，横軸を抽出する頻出パターンの総数 (対数スケール)，縦軸を計算時間としたグラフである．LCM の場合では，求める頻出パターンの個数に依存した計算時間がかかるのに対して，提案手法では求める組合せの個数に依存しない結果となった．これは，meet 演算により頻出パターンを列挙したのち，そこからサポート数を指定して組合せを絞っているためである．一方，従来法では抽出する組合せの個数に依存した計算時間がかかり，多くの組合せを出力したい場合に，提案法が有利な場合がある．また，この実験結果より提案手法は，比較的大きな組合せ多重集合に対してもそれほど遅くない計算時間で共通因子を抽出できることがわかった．

図 3 は，飽和な頻出パターン、meet 演算それぞれについて

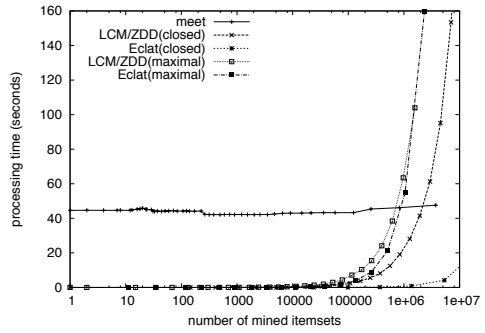

図 2: 抽出する頻出パターンに対する計算時間

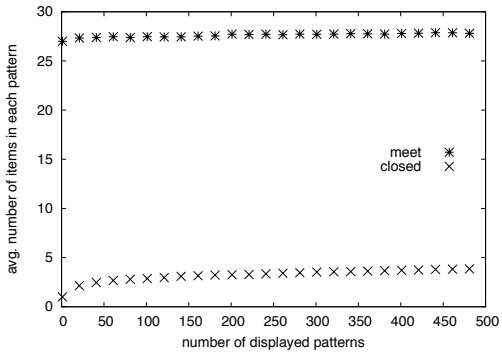

図 3: 抽出する頻出パターンの累計平均アイテム数

上位 500 件の結果について、横軸を抽出する頻出パターンの総数，縦軸をそのときの平均アイテム数としたグラフである．図 3 の結果より，提案手法は従来手法よりも，平均的に抽出する頻出パターンのアイテム数が多いことが分かる．これは，$F \sqcap F \subseteq Freq_{clo}(F, 1)$ であることを考慮すると，提案手法は従来の頻出パターン集合のうち，アイテム数の多い頻出パターンを抽出しているからである．

5. おわりに

本稿では，トランザクションデータベースから共通因子の抽出をするため，meet 演算を用いた手法を提案した．実験結果より，従来法である頻出パターン集合では得られない，組合せを考慮した異なる順序付けを行うことができた．

また，meet 演算を用いて得られる集合は，従来の飽和頻出パターン集合よりも小さい集合であり，比較的大きなデータに対しても現実的な時間内に実行することができる．

また，提案手法では，入力を ZDDV で表現し ZDDV の演算を通して出力も ZDDV の形で得ることができる．これは，入力の組合せ多重集合を ZDDV で圧縮して表現することができれば，効率よく meet 演算を計算することができるという点と，出力が様々な演算を圧縮したまま計算できる ZDDV として得られる点で，提案手法の利点であるといえる．

参考文献

[Agrawal 94] Agrawal, R. and Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases, in *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*, pp. 487–499 (1994)

[Goethals 03] Goethals, B. and (eds), M. J. Z.: Frequent itemset mining dataset repository, Frequent Itemset Mining Implementations (FIMI'03), `http://fimi.cs.helsinki.fi/data/` (2003)

[Knuth 09] Knuth, D. E.: *The Art of Computer Programming, Volume 4, Fascicle 1: Bitwise Tricks & Techniques; Binary Decision Diagrams*, Addison-Wesley Professional, 12th edition (2009)

[Minato 93] Minato, S.: Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems, in *DAC*, pp. 272–277 (1993)

[Minato 08] Minato, S., Uno, T., and Arimura, H.: LCM over ZBDDs: Fast Generation of Very Large-Scale Frequent Itemsets Using a Compact Graph-Based Representation, in *Advances in Knowledge Discovery and Data Mining, 12th Pacific-Asia Conference, PAKDD 2008, Osaka, Japan, May 20-23, 2008 Proceedings*, pp. 234–246 (2008)

[Uno 03] Uno, T., Asai, T., Uchida, Y., and Arimura, H.: LCM: An Efficient Algorithm for Enumerating Frequent Closed Item Sets, in *FIMI '03, Frequent Itemset Mining Implementations, Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations, 19 December 2003, Melbourne, Florida, USA* (2003)

[Yu 14] Yu, X. and Wang, H.: Improvement of Eclat Algorithm Based on Support in Frequent Itemset Mining, *JCP*, Vol. 9, No. 9, pp. 2116–2123 (2014)

[湊 05] 湊 真一：VSOP: ゼロサプレス型 BDD に基づく「重み付き積和集合」計算プログラム, 電子情報通信学会技術研究報告 COMP, Vol. 105, No. 72, pp. 31–38 (2005)

[湊 06] 湊 真一, 有村 博紀：ゼロサプレス型二分決定グラフを用いたトランザクションデータベースの効率的解析手法 (データマイニング, <特集>データ工学論文), 電子情報通信学会論文誌. D, 情報・システム, Vol. 89, No. 2, pp. 172–182 (2006)