相関ルールとその周辺

岡田　孝，元田　浩

1. はじめに

最近のデータマイニングの発展を要素技術の観点から振り返ると，1993年にAgrawalらが提起した相関ルールが大きな要因となっている[1]．元来の相関ルールは，スーパーマーケットでの買い物篭の内容を調べて，販売促進や店舗レイアウトに役立てようというバスケット分析を指向して提起された方法論である．しかし，その枠組みが一般的なデータ解析に適用できる柔軟なものであることが評価され，現在でも非常に活発な研究が行われている．今後もデータマイニング主要技術の一つとして位置づけられていくであろう[2]．すでに邦文でも，人工知能学会誌の特集号[3]や福田らの書籍[4]で解説されているが，本稿では改めて相関ルールの紹介を行うとともに，その問題点と関連する最近の代表的な成果を取り上げて解説し，今後の課題を明らかにしたい．

2. バスケット分析と相関ルール

2.1 相関ルールとは

マーケットで売られている個々の商品をアイテム，一人の顧客が購買したアイテムのリストをトランザクションと呼ぶ．データベース中の全トランザクションを解析すると，例えば，「バターを買った顧客は，その80%がパンと牛乳も買っており，この3種の商品すべてを買った人は全顧客の4%である」というような知見が得られるであろう．これを次のように表したものが相関ルールである．

[バター]⇒[パン，牛乳]；$sup=4\%$, $conf=80\%$

ここで，ルールの条件部，帰結部ともに複数のアイテムを含んでよい．また，ルール中のすべてのアイテムが現れるようなトランザクションの割合を支持度(sup)，条件部のアイテムを購買した顧客中で帰結部のアイテムを買った人の割合を確信度($conf$)と呼ぶ．最低支持度($minsup$)と最低確信度($minconf$)を指定して，データベースからすべての相関ルールを求めることが，Agrawalらの提起した問題である[1]．

この問題を次のように定式化することができる．全アイテムの集合を$I=\{i_1, i_2, \cdots, i_m\}, I\neq\emptyset$とし，その部分集合をアイテムセットと呼ぶ．$D$を全トランザクションの集合とする．ここで各トランザクションTはIの部分集合である．$\emptyset\neq X, Y\subset I$でかつ$X\cap Y=\emptyset$を満たすものを，相関ルール$X\Rightarrow Y$と呼ぶ．アイテムセット$X$の支持度$sup(X)$とは，$D$中の$X$を含むトランザクションの割合であり，ルール$X\Rightarrow Y$の支持度$sup(X\Rightarrow Y)$は$sup(X\cup Y)$で，また確信度$conf(X\Rightarrow Y)$は$sup(X\cup Y)/sup(X)$で定義される．

ルールの確信度は通常の条件付き確率にすぎないが，スーパーマーケットでこのようなルールを調査すれば，(1)右辺に利益率の高い商品が現れるルールを調べ，その左辺から目玉商品を選定する，(2)よく併売される商品群を近くに配置する，(3)多数のルールで条件部に現れる商品をチラシに載せる，など多くの応用が考えられる．

属性とその値の対をアイテムとすれば，表形式を含む一般的なデータに相関ルールの枠組みを活用できる．例えば，トランザクション以外に性別，年齢が表形式データとして利用できれば，次のようなルールの検出が可能となる．

[性別：男，年齢：40代，ワイン]⇒[チーズ]

また，帰結部をクラス属性に固定すれば，クラス識別の要因を説明するルールのみを取り出せる．ただし，本来相関ルールは特徴を説明するためのもの(Characterization rule)であって，識別するためのもの(Disctrimination rule)ではないことに注意しよう．

数値属性はカテゴリー化する必要があるが，トラン

おかだ　たかし
関西学院大学 情報メディア教育センター
〒662-8501 西宮市上ヶ原1-1-155
もとだ　ひろし
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

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

ザクション形式と表形式のデータを統一的に扱えるため，受講科目解析による履修指導やカルテの分析など，伝統的なデータ解析においては手がつけにくかった領域でも素直な分析が可能である．

2.2 アプリオリアルゴリズム

大量のデータを対象とした時，すべての相関ルールを計算することは実際には困難であった．この課題を現実的な時間で処理することに成功し，しかも以降の研究の立脚点となったのがアプリオリ（apriori）アルゴリズムである[5]．このアルゴリズムの第1段階では，$F=\{X\subset I|sup(X)>minsup\}$ で定義される頻出アイテムセットを網羅的に計算し，第2段階は *minconf* 以上の確信度を持つルールを，これらのアイテムセット間から見出す．後段は簡単に行えるため，以下前段の内容を図1に示す例に沿って説明する．

アイテム群 $I=\{a, b, c, d, e\}$ とし，図1左に示すトランザクションから，$minsup=3/8$ として頻出アイテムセットのラティスを構築しよう．図の右側には，アイテムセットとその支持度数が示されており，下線で示されたものが頻出アイテムセットである．計算は以下のように進める．(1) 1アイテムのみからなるすべての候補アイテムセットC1を準備し，データベースを読んでこれらの支持度を求める．支持度が *minsup* 以上のアイテムセットのみをF1として残す（この場合はF1=C1）．(2) F1内のすべてのアイテムセット対から長さ2の候補アイテムセットC2を生成し，データベースを読んで頻出アイテムセットF2を決定する．(3) F2のすべての対からC3を生成する．C3中のアイテムセットに対し，1アイテムを削除した長さ2のアイテムセットのすべてがF2の中に存在するか否かを調べ，もし存在しなければC3から削除する．データベースを読んで，残ったC3の支持度からF3を決定する．(4)以下F4，F5を同様に求め，頻出アイテムセットがなくなったところで，計算を止める．

このアルゴリズムで計算コストが高いのは，トランザクションを読み候補アイテムセットの支持度を更新する所である．候補アイテムセットはハッシュ木に格納するが，*minsup* を満たすアイテムの種類を1,000としても，すべての組み合わせを数えれば，C2で50万，C3では1億近い数の候補が存在する．図1の例では，F2の組み合わせでC3を作る．ここで，C3中の(a b c)は(a b)と(b c)から生成されるが，(a c)はF2の中に存在しない．したがって実際に数えるまでもなく，(a b c)の支持度が *minsup* を超えることはなく，C3からこのようなアイテムセットをあらかじめ除去できる．

ID	購買アイテム
1	a b c
2	a b d
3	a e
4	b c
5	a b c d
6	e
7	a b d e
8	d

[C1] (a)5 (b)5 (c)3 (d)4 (e)3

[C2] (a b)4 (a c)2 (a d)3 (a e)2 (b c)3 (b d)3 (b e)1 (c d)1 (c e)0 (d e)1

[C3] (a b d)3 (a b c)? (b c d)?

図1 頻出アイテムセットのラティス

ラティス中でアイテムセットの支持度は，下部に進むほど単調に減少する．アプリオリアルゴリズムは，この単調性[1]を候補アイテムセットの枝狩りに利用することで，効率的な計算を可能にしたといえる．

2.3 相関ルールの問題点

多くの研究者がこの方法に注目するとともに，その問題点も明らかにされてきた．主要な論点は以下の4種と考えられる．

(1) 相関ルールの英語はassociation ruleであって，correlationではない．すなわち，バターを買う人の80%が牛乳を買うとしても，もし全顧客の80%が牛乳を買っているならば，これらの間に統計的な相関はなくルールは無意味である．

(2) アイテムが密な状況（例えば多変量解析で扱う表形式データ）では，ラティスの第3層以下においても頻出アイテムセットが多数現れ，ラティスサイズの組み合わせ爆発により計算が不能となる．

(3) データベースのサイズが大きく主記憶に常駐できない時，その読み込みに時間がかかる．またサイズが小さくとも，ラティスの各層ごとに候補アイテムの支持度を数えるにはコストがかかる．

(4) 出力されるルール数が莫大な数に上り，ルールの視察が実質的に困難である．*minsup, minconf* 値を上げてルール数を減らすと，その内容は既知のことばかりとなり，解析自体が無意味となる．

以下の各節では上記問題点に関連する事項に絞って，最近の注目すべき成果を取り上げて解説する．

3. データベースの圧縮格納

計算高速化のため多くのアイデアが試されたが，もっとも有効とされたのは最初にデータベースを読みこみ，後の計算で必要となるアイテムセットの支持度を

[1] 頻出アイテムセットの部分集合は頻出アイテムセットでなければならない．すなわち，非頻出アイテムセットを部分集合として含むアイテムセットは頻出ではない．

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

主記憶中に保持する戦略であった．FP-treeアルゴリズム[6]が有名であるが，ここではより簡明で効果も高いとされる部分和の方法[7]を解説する．

$I=\{a, b, c, d\}$とし，2^4種のすべての可能なアイテムセットで表されるトランザクションが各一つ存在するデータベースを想定しよう．この方法では，すべてのアイテムセットを図2に示すset enumeration treeの形で表現する．例えば，(a)の節点下には，(a)を含む長さ2のアイテムセット中から辞書順で最初の(a b)を置く．(a b)の弟の位置に，bに続くc, dが付加された(a c)，(a d)の節点を配置する．また，子の節点として，(a b c)を，さらにその兄弟として(a b d)を置く．このようにすれば，木にはすべてのアイテムセットが正確に一度だけ出現する．

各トランザクションでそのアイテムセットが整列済みなら，それを容易に図2の木で辿ることができる．各節点から見れば，トランザクションが自分の上を通って子への途を辿るか，または自分が終点となる場合に，その回数を数える．これにより，図2の節点に付した数が得られる．このようにして得られた木をP-tree，節点iに付された数を部分和Q_iと呼ぶ．

節点jと正確に一致するトランザクション数をP_jとすると，$Q_i=\sum P_j(\forall j, j\supseteq i, j\ follows\ i$ 辞書順) となる．この値を使えば，アイテムセットiの支持は，$sup(i)=Q_i+\sum P_j(\forall j, j\supset i, j\ precedes\ i$ 辞書順) で表される．例えば，$sup(\mathrm{bc})=Q(\mathrm{bc})+P(\mathrm{abc})+P(\mathrm{abcd})=Q(\mathrm{bc})+Q(\mathrm{abc})=4$となる．

辞書順で先行するアイテムセットをもれなく見つけて，支持度を計算するには図3のT-treeを利用する．

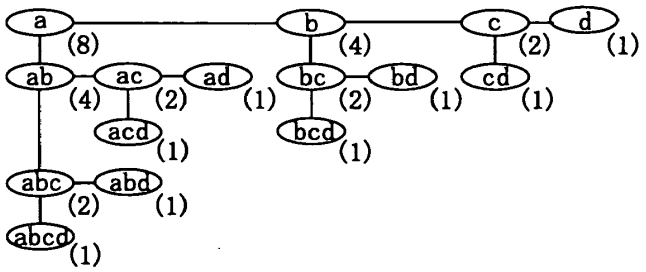


図2　P-treeによる部分和の表現

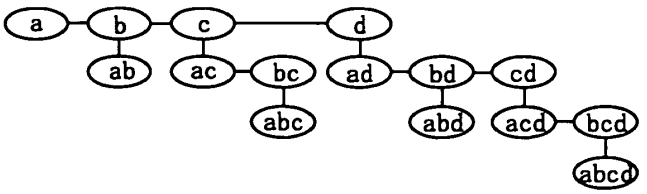


図3　T-treeによる支持度の計算

この木では，まず1アイテムセットを第1層に辞書順に配置する．第2層には，第1層のアイテムを最後に持ち，辞書順で親よりも先行する2アイテムセットを，やはり辞書順に配置する．以下同様に，P-tree中のすべての節点を第3層以下にも配置する．各節点には支持度を加算するためのカウンターを付しておく．

ここで，例えば$Q(\mathrm{acd})$による支持度への寄与を考えてみよう．この寄与は，T-tree中の$Q(\mathrm{a})$, $Q(\mathrm{ac})$にはすでに取り入れられている．したがって，d, ad, cd, acdの支持度を計算する際のみ，これを取り込む必要がある．この場合，T-tree中でacdの最後のアイテムdの節点から始めて節点acdに至るまでの道筋で，acdの部分集合となっている節点に$Q(\mathrm{acd})$を加算する．この操作をすべてのアイテムセットについて行えば，結果としてT-treeの各節点に支持度が計算される．

この方法を実際に適用する場合，可能なすべてのアイテムセットを用意するとP-treeのサイズが爆発するので，トランザクションの読み込み過程で必要な節点のみを動的に生成し，tree構造を構成していく操作が必要である[8]．また，相関ルールを求める際には，T-tree全体を生成しておく必要はない．アプリオリアルゴリズムと組み合わせ，ラティスの各レベル毎にT-treeの対応するレベルを生成すればよい．最低支持度に満たない節点を枝狩りすれば，効率的に頻出アイテムセットの支持度を求められる．

典型的なバスケット分析に適用した結果では，P-treeのサイズがほぼデータベースのサイズに比例して増加し，アイテムの種類が増加しても組み合わせ爆発を起こさないことが示されている．ただし，FP-treeでも同じであるが，いわゆるアイテムが密な表形式データに適用した場合，実際にどの程度の属性数まで対応できるかは明らかでない．しかしこの方法は，データベースの圧縮・再構築と見なすべきものであり，P-treeを主記憶中に置ける限りは，高速に各種の頻度を計算することができる．相関ルールのマイニングに限らず，データベースの対話的解析一般に活用できると思われる．

4．Correlationを求めて

出力ルールに，correlationの意味での相関を表していないルールが多数混じっているならば，ルール群全体が利用者にとっては無意味に等しい．そこで，条件部を空とした場合との確信度の比$conf(X\Rightarrow Y)/$

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

$conf(\emptyset \Rightarrow Y)$をリフト値と呼び，これによって興味深いルールのみを選択することが考えられた．一般的には，X, Yだけでなく$\bar{X}, \bar{Y}$をも考慮した分割表をもとに，例えばχ^2値による評価をルールに与えることが考えられる．しかし，頻出アイテムセットだけが数えられているため，生成されたラティスから分割表のすべてのセルの数値を求めることはできない．

Brinらは，相関の高い属性の集合が一度見つけられれば，そのすべての上位集合でも相関が高いことを指摘し，χ^2値が高い値を示す最小の属性の組と分割表中で特徴的なセルを求める方法を提案した[9]．しかし，マイニングの立場からすれば，同じ属性対間で高い相関が見い出されるにせよ，多くの属性集合で指定されるより限定された事例群には，一段と興味深いルールが隠されている可能性が残る．

この点で興味深いのは，森下らによるχ^2値にもとづく枝狩り法である[10]．ルールの右辺となるべき目的属性Cを固定し，図4左上に示す分割表を想定する．nと$m=sup(C)$が固定されているので，χ^2値は$x(I)=sup(I)$と$y(I)=sup(I \cup C)$の関数となる．

ここで$J \supset I$なる新たな条件Jに対応する分割表を考えると，点$(x(J), y(J))$の値域は図4右上で点を打った平行四辺形内に限られる．森下らはχ^2の凸関数性を使い，どのように条件Jを選んでも，2点$(y(I), y(I)), (x(I)-y(I), 0)$における$\chi^2$値の大きい方の値が上限となることを証明した．

例えば条件Iでの分割表が図4左下の分割表(a)で表されるとき，Iに何らかのアイテムを付加した条件Jのχ^2値は図の右下に示す分割表(b)，(c)から計算される43，25のうち，大きい方を上限値とすることになる．あらかじめ最低のχ^2値を与えるなら，アプリオリアルゴリズム同様ラティスの各レベルで，上限値に満たない節点から下のサブラティスを枝狩りすることができる．実際にバスケットデータに適用したところ，3アイテムのレベルではたった一つの候補を調べればよいほどであり，最低支持度による枝狩りと比べてその効率ははるかに高い．密なアイテムのデータによる評価が待たれる．

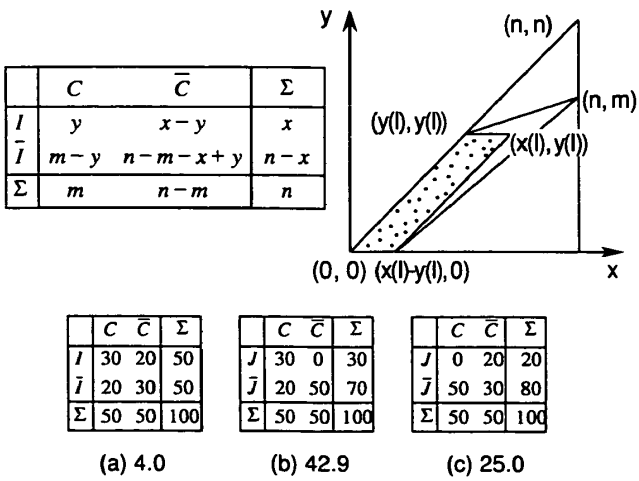


図4　凸関数性によるχ^2値の上限

5. カスケードモデル

筆者の一人により提案された本モデルも，相関ルールの1種の発展であると見なせる．例えば図5左の表で，属性A, Bの値からYの値p, nを説明する問題を考える．A, Bの値をアイテムとして構築されたラティスを，このモデルでは図右側のように描く．ここで，それぞれの湖が節点を，その間の滝がリンクを表す．湖の広さと滝の幅は事例数と大まかに対応し，また湖の高さが目的属性Yの純度を表すと考えよう．ここで発電能力の大きな滝を選びルールとして表現するのが，カスケードモデルである[11]．

滝の発電能力を表現するため，Giniによる平方和を用いる[12]．数値変数の平方和定義は(1)式のように変形できるが，ここでカテゴリー値の場合も事例i, j間での$x_i - x_j$の値を，$x_i = x_j$の時に0，他は1とすれば，(2)式の平方和定義が得られる．ただし，nは全事例数を表し，$p(a)$はその属性が値aを取る確率である．

$$SS = \frac{1}{2n} \sum_i \sum_j (x_i - x_j)^2 \tag{1}$$

$$SS = \frac{n}{2}\left(1 - \sum_a p(a)^2\right) \tag{2}$$

一群の事例をある属性の値でG個の群に分割した時，元の全平方和（TSS：Total sum of squares）は(3)式のようにそれぞれの群内平方和（WSS：Within

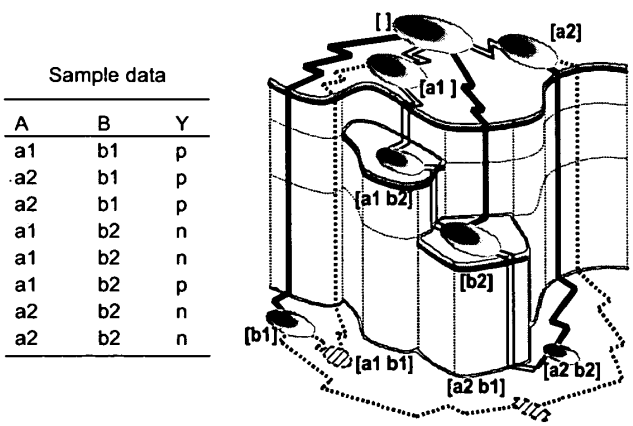


図5　ラティスのカスケード表現

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

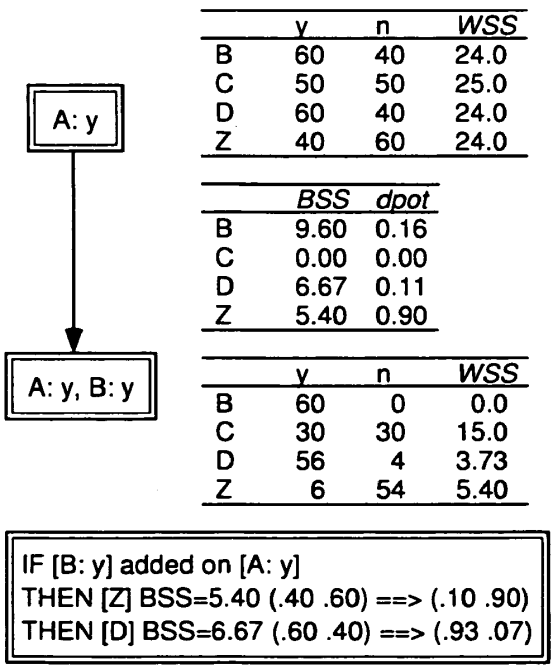
-group sum of squares）および群間平方和（*BSS*：Between groups sum of squares）に分割できる．なお，*BSS* は(4)式で定義し，添字 U, L は分割前と後を指示する．

$$TSS=\sum_{g=1}^{G}(WSS_g+BSS_g) \tag{3}$$

$$BSS_g=\frac{n^L}{2}\sum_a(p_a^L(g)-p_a^U(g))^2 \tag{4}$$

U, L で指定される事例群を滝の上側と下側に対応するラティス内の節点と見なせるので，この BSS_g を滝の発電能力と解釈できる．したがって，ラティス中で *BSS* 値の大きなリンクを選択して，それをルールとして提示すればよい．

図6はラティス中のリンクとそれから導かれるルールの一例を示す．ここで，問題は説明属性A-Dの値から，目的属性Zの値を説明することであり，またすべての属性はy, nいずれかの値を取るとする．図では，上側節点がアイテムセット[A: y]で表され，[B: y]のアイテムが滝に沿って付加されている．ここで，両節点の右に示す表には，属性A以外の各アイテムの支持度数を示す．これらの度数から中央の表に示すように各属性毎に *BSS* 値を計算できる．*BSS* 値が大きいD, Zの属性では，下側節点での支持度数が[D: y]，[Z: n]に偏っており，付加されたアイテム[B: y]とこれらの属性が高い相関を持つことがわかる．反対に，属性Cの分布は上下節点でまったく変化せず，*BSS* 値も0となる．



	y	n	*WSS*
B	60	40	24.0
C	50	50	25.0
D	60	40	24.0
Z	40	60	24.0

	BSS	*dpot*
B	9.60	0.16
C	0.00	0.00
D	6.67	0.11
Z	5.40	0.90

	y	n	*WSS*
B	60	0	0.0
C	30	30	15.0
D	56	4	3.73
Z	6	54	5.40

図6　リンクからのルール表現

目的属性Zに対する *BSS* 値が大きなリンクを選択し，これを図6下側に示すルールとして表す．ここで，ルール左辺は主条件部と前提条件部に分かれている．この場合，リンクに沿って付加されたアイテム[B: y]が主条件を表し，上側節点のアイテムセット[A: y]が前提条件となる．ルール右辺には，*BSS* 値とともに，目的属性Zの分布が主条件の付加によりどのように変化したかを示す．図の属性Dのように主条件と相関の高い属性が存在する場合は，たとえそれが説明変数であっても，付加的な右辺情報としてルール中に表示する．この情報は説明変数間の高い相関を示すため，実際の問題に適用してルールの解釈を行う際には，非常に有効な情報を与える．

カスケードモデルの計算でルールを検知するには，ラティス中でその上下節点だけを生成すればよい．すなわち図6のルールの場合，[A: y　B: y　D: y　Z: n]のようなアイテムセットを生成する必要がない．したがって，密にアイテムが分布する場合でも，ラティス上層部の節点を調べるだけで強い相関を検知し，しかも他の説明変数との関連まで含んだ有効なルールを生成することができる[13]．

図6のルールは，分割表で表現すれば図7で表される．χ^2 値が分割表全体を対象とした相関の有無を問題とするのに対し，カスケードモデルではこの表のB行でのZ値の分布を Σ 行に示されるZ値の分布と比較し，相関の有無を *BSS* 値として表していることになる．

ところで，図6の上側節点から[B: y]を付加した下側節点を次の層に生成するとき，*BSS*(B)の値はあらかじめ計算できる．他方 *BSS*(Z)の値は，この場合 *BSS*(B)を上限値とすることが証明されている[13]．したがって，下側節点におけるZ属性の各アイテム支持度を計算するまでもなく，このリンクからは *BSS*(Z)値が10よりも大きなルールを導けない．

この上限値はこれより下部に存在するすべてのリンクに適用できるものではないが，より下層のラティスの近似的な枝狩りに用いることができる．ただし，現実に表データを扱う場合，この上限値による枝刈りで

	Z	$\bar{Z}$	Σ
B	6	54	60
$\bar{B}$	34	6	40
Σ	40	60	100

図7　分割表で見たカスケードモデル

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

は組み合わせ爆発を防げない．そこで，カスケードモデルの適用に際しては別に枝刈り用の BSS 値を与え，これよりも BSS(B) 値が小さいリンクの展開を抑止してラティスサイズを制御している．

6. ラティス意味論の拡張

これまでのすべての説明で，アイテムセットラティスにおける節点間のリンクには，上下両側アイテムセット間に部分集合関係の存在が前提とされてきた．半順序関係を前提とした範囲内でも，他の意味をアイテムセット間のリンクに与えることができる．このような例として，離散時系列とグラフのマイニングを簡単に紹介する．

相関ルール研究の初期から，各トランザクションに購入者 ID とタイムスタンプを付した形式のデータが取り上げられてきた[14]．顧客毎に購入履歴を時系列順にまとめて，例えば[小泉 ((b) (b) (a d))]のような上位レベルのレコードを作成する．他方，ラティス内の節点には((b) (c))や((b) (b) (d))のような購買アイテムの時系列順のリストを割り当てる．アイテム間の相対的順序を保った部分列関係がリスト間に存在する時にのみ，節点間にリンクを張る．このようにすれば，((b) (b))⇒((b) (b) (d))のようなルールを導くことができる．これにより，顧客の購買履歴から次にどのような商品が売れるであろうかという，時系列的な分析が可能となる．沼尾の解説には，表形式データと組み合わせた要因結果分析も解説されているので参照されたい[15]．

トランザクションの形式を，さらに複雑なグラフ構造に拡張した例が，Apriori-based graph mining である[16]．ここでトランザクションとしては，化学構造式のグラフ表現に発ガン性などの生理活性を付加したものを想定しよう．図 8 には，この方法で生成されるラティスの一部を示す．ラティス中の i 番目の層には，頂点数 i のグラフが格納され，グラフ間に部分同型関係が存在する場合にリンクが張られる．頻出グラフからなるラティスの生成は，基本的にアプリオリアルゴリズムと同様に進行する．ただし，$(i+1)$ 層の候補グラフは，i 層の頻出グラフの対（自分自身と対になってもよい）から $(i-1)$ 個の頂点を重ね合わせて生成する．また，生成される候補グラフ群に同型のものが重複して現れないように，格別の注意が必要である．

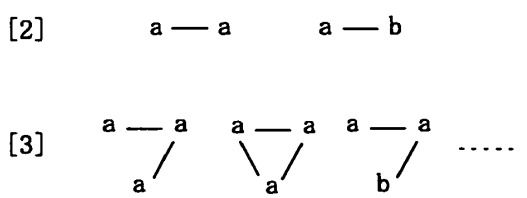


図 8　グラフへの拡張

ここで，特定の頻出グラフで，例えば発ガン性の有無がデータセット全体と比べて大きく有に偏っているならば，そのグラフに対応する部分構造が化学発ガン性の原因となっているのではないか，という仮説を立てることができる．グラフの種類を化学構造式のような色つき無向グラフから，サイクルを持たない有向グラフへと換えることにより，購買履歴よりもより一般的なネットワークフロー型時系列データのマイニングを行うことができる．問題領域毎に異なった意味をラティス中の節点とその間の半順序関係に与えれば，無限に豊富なマイニングが可能となる．

7. おわりに

相関ルールは，「何でもアイテム化してバスケットに放り込めば分析が可能」という非常に柔軟な方法論であり，カルテや雑多な社会事象なども取り扱える可能性があることから，今後が大いに期待されている．反面，まだまだ若い方法論であり，理論面から実装技術，応用に至るまで多くの課題が山積している．今後一層の研究の発展が期待されている．

最後に，問題点としてあげながら触れることのなかった，多すぎるルール数の問題を考えてみたい．出力されるルール数を削減する必要は広く認識され，すでに多くの研究がなされている．相関ルールを定義通りに生成すると，(a)⇒(b c) のルールと並んで (a b)⇒(c) のような冗長なルールが現れる．形式的な側面から不要と判断できるルールを削減する研究が多く行われており，実際にルール数を減らすことができる．他方，節 4 で述べた correlation の意味で有効なルールのみを出力することもルール数の減少に寄与する．さらに χ^2 や BSS のように単一の数値でルールの強さに全順序をつけることも，多数の候補からのルール選択を容易にする．また，ルール群を可視化することにより，データの全体的な傾向を把握させる試みも多い．

しかし，筆者らの独断ではあるが，ルール数の多さはもっと本質的な問題点に根ざしているかのように思える．例えば，二つのルール (a b)⇒(d) と (a c)⇒(d) が出現し，しかも b と c の間には非常に高い相関が

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

存在する状況が考えられる．この場合，形式的にはこれらのルールは独立に扱われるべきものであろう．しかし，実際はこれらのルールは同じ山を違った方向から見ているにすぎない．

重回帰分析でdをa, b, cにより説明しようとするならば，相関の高い説明変数を除くために変数選択の過程が必要となる．単純に計算を進めると，数値的な不安定性などの問題を引き起こす．それに引き替え，ルールの導出では見かけ上何の問題も起こらず，すべては解析者による視察に押しつけた形となっている．多変量解析が長年月をかけて取り組んできた共線性の問題が，ルール数の多さというまったく違った形で現れてきたのが本質である，と見るべきであろう．このことは問題解決の困難さを予想させるものではあるが，反面ルール表現を使えば，同一の事象を複数の異なった側面から浮き彫りにできる可能性をも示している．

現在，解析者の積極的なレスポンスをマイニング過程に取り入れることを重視したアクティブマイニングが注目されている[17]．ルール数の多さを欠点としてではなく，更なる飛躍への踏み台として考える中から，解析者との積極的な相互作用が可能になるものと期待される．

参考文献

[1] Agrawal R., Imielinski T. and Swami A. N.: "Mining association rules between sets of items in large databases", Proc. SIGMOD, pp. 207-216, ACM (1993).

[2] 元田, 鷲尾："データマイニング展望", システム/制御/情報, Vol. 46, pp. 169-176 (2002).

[3] 沼尾編："大規模データベースからの知識獲得", 人工知能学会誌, Vol. 12, No. 4, pp. 496-549 (1997).

[4] 福田, 森本, 徳山："データマイニング", データサイエンスシリーズ3, 共立 (2001).

[5] Agrawal R. and Srikant R.: "Fast algorithms for mining association rules in large databases", Proc. VLDB, pp. 487-499, Morgan Kaufmann (1994).

[6] Han J., Pei J. and Yin Y.: "Mining frequent patterns without candidate generation", Proc. SIGMOD, pp. 1-12, ACM (2000).

[7] Coenen F., Goulbourne G. and Leng P. H.: "Computing association rules using partial totals", Proc. PKDD, pp. 54-66, Springer (2001).

[8] Goulbourne G., Coenen F. and Leng P. H.: "Algorithms for computing association rules using a partial-support tree", *J. Knowledge-Based Systems*, Vol. 13, pp. 141-149 (2000).

[9] Brin S., Motwani R. and Silverstein C.: "Beyond market baskets: generalizing association rules to correlations", Proc. SIGMOD, pp. 265-276, ACM (1997).

[10] Morishita S. and Sese J.: "Traversing itemset lattice with statistical metric pruning", Proc. PODS, pp. 226-236, ACM (2000).

[11] Okada T.: "Rule induction in cascade model based on sum of squares decomposition", Proc. PKDD, pp. 468-474, Springer (1999).

[12] Gini C. W.: "Variability and mutability, contribution to the study of statistical distributions and relations", *Studi Economico-Giuridici della R. Universita de Cagliari*, (1912). Reviewed in Light R. J. and Margolin B. H.: "An analysis of variance for categorical data", *J. Amer. Stat. Assoc.*, Vol. 66, pp. 534-544 (1971).

[13] Okada T.: "Efficient detection of local interactions in the cascade model", Proc. PAKDD, pp. 193-203, Springer (2000).

[14] Agrawal R. and Srikant R.: "Mining sequential patterns", Proc. ICDE, pp. 3-14, IEEE (1995).

[15] 沼尾, 清水："流通業におけるマイニング", 文献[3], pp. 528-535.

[16] Inokuchi A., Washio T. and Motoda H.: "An apriori-based algorithm for mining frequent substructures from graph data", Proc. PKDD, pp. 13-23, Springer (2000).

[17] Motoda H. (ed.): "Active mining", IOS press (2002).

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．