相関ルールの2部グラフを用いた重要アイテムの発掘

八木　一光†　　岡野　慎吾††　　森本　康彦††

† 広島大学工学研究科　〒 739–8521 広島県東広島市鏡山 1–7–1
†† 広島大学総合科学部　〒 739–8521 広島県東広島市鏡山 1–7–1
E-mail: †m054228@hiroshima-u.ac.jp, ††morimoto@mis.hiroshima-u.ac.jp

あらまし　小売店で販売される多数のアイテムの中から，併買関係に注目した重要アイテムの発掘を行った．まず，蓄積された大量の顧客データから併買に関して統計的に有意な相関ルールを χ^2 値を基に計算した．計算された相関ルールの集合は，各アイテム集合をノード，アイテム集合間の各ルールを有向辺，ルールの χ^2 値を辺の重みとする重みつき 2 部グラフとして考えることができる．この重みつき 2 部グラフから HITS やページランクなどのアルゴリズムを利用して各アイテムの重要度を分析し，その結果からアイテムのランキングを行った．本論文では，併買を導くアイテムおよび併買されやすいアイテムを重要アイテムとした．発掘されたこれらの重要アイテムを中心に販売戦略を考えることは有効であると考える．

キーワード　相関ルール，アイテムランキング，HITS アルゴリズム，ページランク

Mining of important items from bipartite-graph of association rules

Kazumitsu YAGI†, Shingo OKANO††, and Yasuhiko MORIMOTO††

† Gradiate School of Engineering, Hiroshima University　Kagamiyama 1–7–1, Higashi-Hiroshima, 739–8521 Japan
†† Faculty of Integrated Arts and Sciences, Hiroshima University　Kagamiyama 1–7–1, Higashi-Hiroshima, 739–8521 Japan
E-mail: †m054228@hiroshima-u.ac.jp, ††morimoto@mis.hiroshima-u.ac.jp

Abstract　We consider a problem of mining of important items from bipartite-graph of association rules. First of all, we calculate association rules based on the statistical strength, which is χ^2 value, from large POS data. A set of the mined association rules can be represented as weighted bipartite-graph, whose nodes are itemsets, directed edges are rules, and weight of edges are χ^2, respectively. We calculate importance of each itemset from the graph by using HITS and PageRank algorithms. Then, we find important cause items and effect items based on the score of the link analysis.

Key words　Association Rules, Item Ranking, HITS, PageRank

1. はじめに

百貨店には多種多様な商品があり，毎日多くの顧客が様々な商品を買っていく．それに従って膨大な売上記録が蓄積されている．近年はこのような情報を詳細に分析することで得られる知識の重要性が広く認識され，利用されている．一般的に多くの小売業では POS システムを利用して売上データを記録している．この POS データを分析し，例えば「どの商品が他のどんな商品と同時に買われることが多い」とか「この商品は他の商品と比べて買われやすい」などの情報が分かれば，経営の戦略を立てる上で役に立つと考えられる．

蓄積された大量のデータからビジネスに活用できる有用な情報を取り出す技術としてデータマイニング技術が注目を集めてきた．このデータマイニング技術の代表的なものとして「相関ルール」がある [3]．POS データからは，例えば，「ビールを買う人はおつまみを同時に買うことが多い」などの併買パターンが相関ルールを用いて発掘できる．しかし，通常の相関ルールマイニングの手法を使って分析を行うと，このようなルールが数多く現れる．そのため注目すべきルールや商品が，数多くのルールの中に埋没してしまう．このような問題を解決するために本論文では様々な解析手法を用いて，多くのルールの中から注目に値するルールを絞り込む手法，および，多くの商品からどの商品が注目に値するかを探す手法を開発した．

今回，分析対象のデータには「ある日時に，ある顧客が，あ

る店舗で，ある商品群を購入した」といった情報が記録されている．このデータのトランザクション (=1 回の取引) 内で取引される各商品をアイテムという．1 つ以上のアイテムの集まりをアイテム集合といい，特に k 個のアイテムの集まりを k-アイテム集合と呼ぶ.「アイテム集合 A の全てのアイテムを含むトランザクションの割合」を $Sup(A)$ と表し，これをサポート (support) と呼ぶ.

あるアイテム集合 A と B に対して「A の全てのアイテムを買った顧客は B の全てのアイテムも購入する」ということを「$A \Rightarrow B$」と表わし，これを相関ルールと呼ぶ．このとき，A をルールの前提，B をルールの結論という．顧客の併買パタンは，この相関ルールによって表現される．また，相関ルールの評価指標の代表的なものとして確信度 (confidence) が挙げられる．確信度は，ルールの前提を含むトランザクションの中に，どれだけルールの結論を含むトランザクションがあるかを示すものである．つまり前提のアイテムを買う顧客で，結論のアイテムも購入する人の割合を表わす.「アイテム集合 A の全てのアイテムを含むトランザクションの数」を $n(A)$ としたとき，$A \Rightarrow B$ の確信度 $conf(A \Rightarrow B)$ は，

$$conf(A \Rightarrow B) = \frac{n(A \cup B)}{n(A)} = \frac{sup(A \cup B)}{sup(A)} \qquad (1)$$

と定義される．サポートは相関ルールの適用範囲の広さを，確信度は相関ルールの強さを示唆する指標と見ることができる．

アプリオリなどの一般的な相関ルール発掘アルゴリズム [2] は，ユーザの与える閾値以上となるサポート，および確信度の相関ルールの全てを効率に求めることができる．しかし，一般的に使われている確信度は必ずしもルールの評価指標として適しているとはいえない [6]．例えば，$conf(A \Rightarrow B) = 0.9$ であるなら，直感的には A と B に強い関係があるように思える．しかし，$sup(B) = 0.9$ であるなら，つまり B が単独で 90%ものトランザクションに含まれているなら，A との有意性が少なくてもルールの確信度は高く評価されてしまう．そこで，相関ルールを統計的な有意性を表す χ^2 値で評価することを考える．相関ルール $A \Rightarrow B$ に対して χ^2 値は

$$chi(A \Rightarrow B) = \frac{\left(\frac{Sup(A \cup B)}{Sup(A)} - Sup(B)\right)^2}{Sup(B)} = \frac{\left(\frac{Sup(B) - Sup(A \cup B)}{1 - Sup(A)} - Sup(B)\right)^2}{Sup(B)} \qquad (2)$$

と定義される．

これは，前提と結論が独立であるという仮定の下で予想される結論のサポートと，ルールの前提を満たすか否かで変化する実際の結論のサポートとのズレの大きさを評価するものである．従って χ^2 値が大きいほど前提と結論が独立であるとは考えにくいことを意味する．式の定義より，前提と結論の相関の有意性が強いほど χ^2 値は大きくなる．(2) 式の χ^2 値は結論部が期待値よりも高い方向に大きくズレる場合も，低い方向にズレる場合も大きくなるが，売上に関してはその高低には異なる意味があるので (2) の式を，

$$\chi^2_{AB} = chi(a,c,i) = \frac{\sigma(Ni - ac)^2}{Nc}\left\{\frac{1}{a^2} + \frac{1}{(N-a)^2}\right\}$$
$$,\sigma = \begin{cases} 1 & Ni - ac \geqq 0 \\ -1 & Ni - ac < 0 \end{cases} \qquad (3)$$

とする．ルール $A \Rightarrow B$ に対し，$a = n(A)$，$c = n(B)$，$i = n(A \cup B)$ である．この $chi(a,c,i)$ 値は，売上増を導くときほど正の大きな値をとり，売上減を導くときほど負の小さな値をとる．

これらのことから，本論文では相関ルールを，

$$A \Rightarrow B, (\chi^2_{AB})$$

の形式で表すとする．これは A を買うことにより B に及ぼす影響が χ^2_{AB} であることを意味している．

2. 効率的な相関ルールの発掘

2.1 最適化問題

一般的に利用されているアプリオリアルゴリズムでは最小サポート，最小確信度を使った枝刈りが有効であったが，χ^2 値によるルール評価では，ルールの χ^2 値の上限・下限を予測することができない [2]．しかし,「ルールの結論を固定し，その結論に対して最適の χ^2 値を持つルールを求める」というように問題を限定すれば，効率のよい計算方法が存在する [1,7]．このような問題を最適化問題という．この問題では，結論を固定し，前提を変化させるときの χ^2 値の上限値・下限値を利用する．

χ^2 値を求める式 (3) で，結論 (つまり c) が固定されていることに注意して式を変形すると，

$$chi(a,i) = \frac{(Ni - ac)^2}{Nc}\left\{\frac{1}{a^2} + \frac{1}{(N-a)^2}\right\} \qquad (4)$$

となる．chi は a と i の関数と考えられる．サポートの単調性よりどんな相関ルールでも $a \geqq i$ であるため，以下の図 1 の影の部分で相関ルールの χ^2 値が求まる．

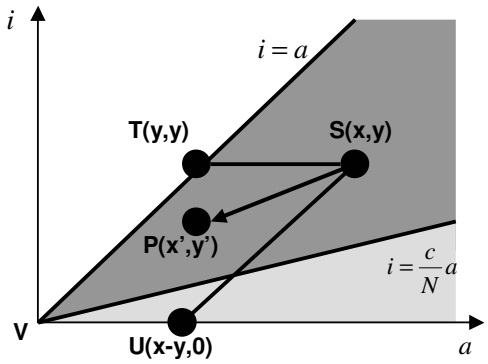

図 1　相関ルールが存在する平面状の領域

今，ルール $A \Rightarrow B$ の a と i が点 $S(x,y)$ にあるとする．さらに結論 B はそのままで，前提を $A'(\supseteq A)$ に変えると，$(a,i) = P(x',y')$ になるとする．A が A' になるということは，サポートの単調性より $x' \leqq x$ かつ $y' \leqq y$ である．また，この

とき i の値の減少よりも，a の値の減り方の方が大きいはずである (なぜなら，ルールのアイテム集合は，完全にルールの前提の集合に含まれる) から，P 点は図の点 S,T,U,V を結んだ平行四辺形内で移動することになる．式 (4) を a と i でそれぞれ偏微分すると，

$$\frac{\partial}{\partial a}chi(a,i) = \frac{2(Ni-ac)^2}{c}\left\{\frac{(i-c)^3}{(N-a)^3} - \frac{i}{a^3}\right\} \quad (5)$$

$$\frac{\partial}{\partial i}chi(a,i) = \frac{2(Ni-ac)}{Nc}\left\{\frac{1}{a^2} + \frac{1}{(N-a)^2}\right\} \quad (6)$$

となる．

式 (5) に注目すると，$Ni-ac \geqq 0$ のとき式 (5) は 0 より小さくなり，$Ni-ac < 0$ であるならば式 (5) は 0 より大きくなる．つまり，i が一定であるとき，図 2 のように χ^2 値は直線 $i = \frac{c}{N}a$ から離れるほど大きくなり，近づくほど小さくなる．

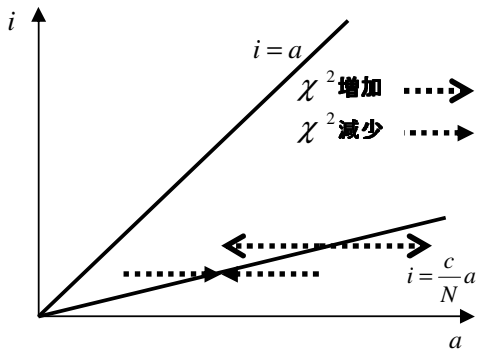

図 2　i 固定時の χ^2 値の増減

同様に，式 (6) に関して a が一定であるとき，$Ni-ac \geqq 0$ であるならば式 (6) は 0 以上となり，$Ni-ac < 0$ であるならば式 (6) は 0 より小さくなる．従って図 3 のように χ^2 値は直線 $i = \frac{c}{N}a$ から離れるほど大きくなり，近づくほど小さくなる．

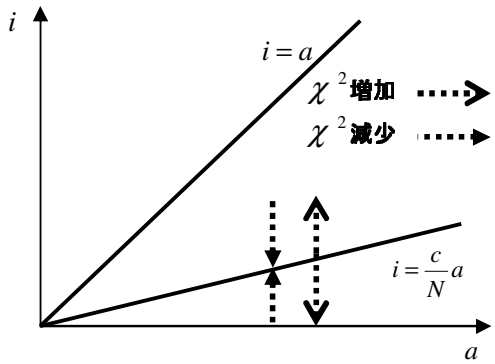

図 3　a 固定時の χ^2 値の増減

さらに $i = a$ と $i = a-(x-y)$，つまり直線 TV と SU 上を移動したときの χ^2 値の変化を調べると，図 4 のようになる．

以上のことから図 1 の平行四辺形内で χ^2 値が最大になる点は，T または U のどちらかであることがわかる．固定した結論

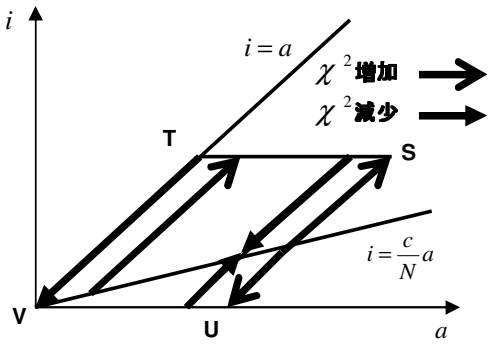

図 4　直線 TV と SU 上の χ^2 値の増減

B のもと，$A' \Rightarrow B(A' \supseteq A)$ の強さの上限を知ることはできる．

最適化問題として，ルールの強さが最大になる前提を選び出すのであれば $A \Rightarrow B$ の χ^2_{AB} 値の方が，$A' \Rightarrow B$ の $\chi^2_{A'B}$ 値の上限よりも大きい場合は，A を含むアイテム集合はこれ以上のルールを構成することはないとして枝刈りすることができる．例では結論を固定して最適な前提を選び出したが，逆に前提を固定し，最適な結論を選び出す場合にもこのような枝狩りは有効である．

2.2　ベストパートナーマイン

上述の枝狩りを利用し，重要相関ルールを見つけ出す以下のような発見的アルゴリズムを実装した．

このアルゴリズムでは，あるアイテム集合をルールの前提として固定し，その前提との χ^2 値を最大にする結論を見つけ出す．次に見つかった結論を固定しその結論との χ^2 値を最大にする前提を見ける．さらにその前提との χ^2 値を最大にする結論を見つけ出す，ということを次々と繰り返していく．これを**ベストパートナーマイン** (BPM) と呼ぶことにする．前提と結論を交互に固定して最適なパートナーを見つけていくので，前述の最適化問題の効率的枝狩りを利用することができる．

前提 (または結論) を固定したときに，相関ルールの χ^2 値を最大にする相手のことを**ベストパートナー** (best partner) と呼ぶ．このときベストパートナーを指定したアイテム集合 (固定した前提や結論) は**デジグネーター** (designator)，また，あるアイテム集合をベストパートナーとして指名しているデジグネーターを，指名されたアイテム集合の**サポーター** (supporter) と呼ぶ．

図 5 はベストパートナーの指名の様子を示したものである．

{a} をデジグネーターとしてみれば，ベストパートナーは {b,c} であり，サポーターは {b} と {c} である．{b} をデジグネーターとするなら {a} がそのベストパートナーとなり，{b} を指す矢印 (ルール) がないためサポーターはなし，ということになる．図中の矢印はベストパートナーの指名を指すもので，相関ルールの矢印とは異なる．例えば，{b} から {a} へ矢印が出ているが，相関ルールとしては {a}⇒{b} を意味する．

ベストパートナーを調べ終えたデジグネーターの集まりを，そのデジグネーターで前提であったか結論であったかで区別して**チェックトアンテシデント** (checkedAntecedent) または**チェッ**

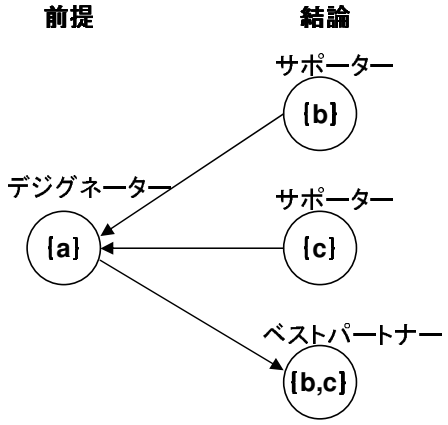

図 5　{a} を前提としたパートナー

クトコンシクエント (checkedConsequent) と呼ぶ．ベストパートナーを調べ終えていないデジグネーターの集まりは，それぞれ**ニューアンテシデント** (newAntecedent) または**ニューコンシクエント** (newConsequent) と呼ぶことにする．図 5 の例では，前提のデジグネーターとして矢印を出し終えた {a} がチェックトアンテシデント，同じく結論として {b} と {c} がチェックトコンシクエントになる．まだ結論としてベストパートナーを指名していない {b,c} はニューコンシクエントである．

{a},{b},{c} がニューアンテシデントとニューコンシクエントとして初期化された場合の BPM の例を見る．まず {a},{b},{c} がそれぞれ前提であるときの最適ルールを作る結論が調べられる．デジグネーターとしてベストパートナーを指名したアイテムセットはチェックトアンテシデントになる (図 6 では○で囲む)．ベストパートナーは，図中の数字の順に探される．

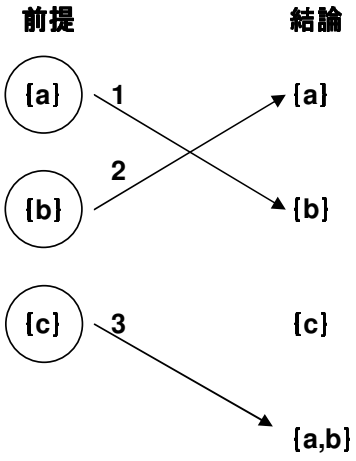

図 6　BPM の例

次に {a},{b},{c},{a,b} がニューコンシクエントになっているから，これらのベストパートナーを見つけ，それが終わったデジグネーターをチェックトコンシクエントにする (図 7)．こうしてニューアンテシデントや，ニューコンシクエントがなくなるまで (○のついていないアイテムセットがなくなるまで) ベストパートナー探しが交互に行われる．

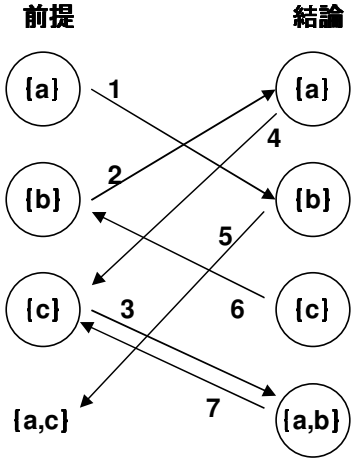

図 7　BPM の例 (その 2)

2.3　ルールの吸収

アイテム数が多いときは相関ルールの数が爆発的に増えるため，BPM のような方法でベストパートナーとならないようなルールを減らすことでルール全体の見通しがよくなる．しかし，これでもなお冗長なルールが残るため，さらにアイテム数を少なくする方法を考える．

例えば，{a}⇒{b} と {a}⇒{c} と {a}⇒{b,c} というルールが見つかった場合について考える．本論文では，このようなルール間の関係について，以下のように解釈する．もし，{a}⇒{b} と {a}⇒{c} の両方が {a}⇒{b,c} より強いルールなら，{b} と {c} を個別に見て，{b,c} の結論のルールを無視する．また，{a}⇒{b,c} の方が，{a}⇒{b} と {a}⇒{c} よりも強いルールなら，2 つのルールを無視して {b,c} が結論であるルールだけを採用する．つまり {b},{c} を個別に扱うよりも，一まとめに {a} との関係を捉えれれば十分であると考える．さらに，{a}⇒{b} と {a}⇒{c} のどちらか片方だけが {a}⇒{b,c} より強いときには，強い方のルールと {b,c} を結論にしたルールを採用する．以上は,「結論が共通で、前提が包含関係にあるルール」でも同様にあてはまる．

さらに両方のケースを組み合わせ，{a}⇒{c} よりも {a,b}⇒{c,d} の方が強いのであれば，そちらに注目するようにする．つまり「前提と結論のどちらか，あるいは両方が包含関係にあるような一連のルールでは，最も χ^2 が大きいルールだけを採用する」ということである．他のルールをより強いルールで表すため，これを**ルールの吸収**と呼ぶ．

BPM とルールの吸収を組み合わせることでさらに効率的にルールを削ることができる．例えば BPM で作られたルールの指名関係の一部が図 8 のようになったとする．

{a} をデジグネーターとしてみると，サポーターによるルールとして {a}⇒{b} と {a}⇒{c} があり，ベストパートナーによるルールとして {a}⇒{b,c} がある．当然 {a} が前提であるときのルールでは {a}⇒{b,c} が最高であるため，図中の 1 で指された二つのルールは吸収される．

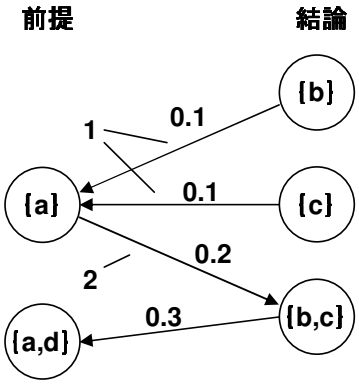

図 8　BPM と組み合わせたルールの吸収

次に {b,c} をデジグネーターとしてみると，{a}⇒{b,c} と {a,d}⇒{b,c} の二つのルールがある．やはり {b,c} 結論に対して {a,d}⇒{b,c} が最高のルールであるため，図中の 2 で指されたルールは吸収される．この例では，ルールの吸収によって 4 つのルールが 1 つに減ることになる．

BPM と組み合わせたルール吸収では,「各デジグネーターに注目し，そのベストパートナーがサポーターを含むアイテムセットであった場合，サポーターによるルールを吸収する」ということになる．BPM による計算が収束したあとに，すべてのチェックトアンテシデントとチェックトコンシクエントに注目してルールの吸収を行うことで，不要なルールを取り除くことができる．

3. リンク分析による重要アイテムの分析

前章までに述べた方法を用い，POS データから併買に関して関連の強い相関ルールを算出した．この相関ルールは各アイテムを頂点，各ルールを重みつき有向枝とする巨大な有向グラフとして考えることができる．この有向グラフを**相関ルールグラフ**と呼ぶ．本章では相関ルールグラフの枝のつながり方を解析して，注目に値する重要アイテムを発掘する．そこで情報検索の分野で HITS [4] として知られるアルゴリズムを用いて，アイテムのリンク関係からそのスコアを求め，スコアが高いものを上位から順に列挙する．

3.1　リンク解析

検索エンジン等に利用されている HITS は，大量の情報から有益な情報を得るために開発されたアルゴリズムである．検索エンジンの多くは検索語を入力して，その語に関連したページを列挙するが，一般的に，検索語を含むページは膨大な量になることが多い．そのため，各検索エンジンでは検索語を含むページの中で，いかにして重要なページを上位に表示するかを工夫している．そこで用いられるテクニックの一つが HITS のランキング手法で，多くの頂点から入ってくる有向枝を持つオーソリティ(authority) と呼ばれる頂点と，多くのオーソリティ頂点への有向枝を持つハブ (hub) 頂点を定義し，それらの相互関係を利用して各頂点のオーソリティ値とハブ値を求め，どのページが重要であるか判断するものである．

HITS の計算方法について述べる．各ページ p のオーソリティとハブのスコアは，以下の 2 式を反復することにより計算される．ページ p のオーソリティのスコアを y_p，ハブのスコアを x_p とする．

$$y_p = \sum_{q,q\to p} x_q \tag{7}$$

$$x_p = \sum_{q,p\to q} y_q \tag{8}$$

ここで，$p \to q$ は Web ページ p が q へのリンクをもっていることを示す.

式 (7),(8) より x_p の値は y_q の合計になり，y_p の値は x_q の合計になる．検索エンジンでは一般的に，特定のトピックに関する優良サイトはオーソリティのスコアが高くなり，優良リンク集はハブのスコアが高くなる．

この手法を相関ルールグラフに取り入れ，本論文において，ハブ頂点は併買を導く「**原因アイテム**」，オーソリティ頂点は併買される「**結果アイテム**」とする．

実際の相関ルールに基づいた計算法についてみていく．例えば，図 9 のような相関ルールグラフについて考える．このグラフは図 10 のような 2 部グラフで表すこともできる．

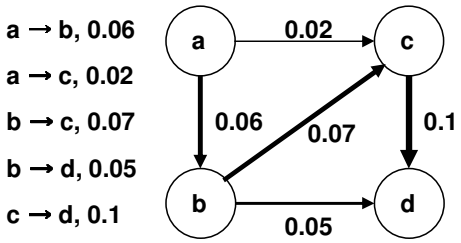

図 9　相関ルールグラフの例

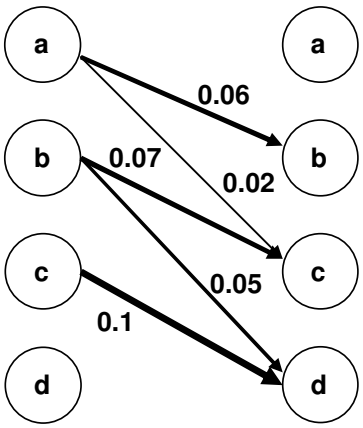

図 10　図 9 の 2 部グラフ表現

この 2 部グラフを以下の隣接行列 A で表す.

$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 0 & 0.06 & 0.02 & 0 \\ 0 & 0 & 0.07 & 0.05 \\ 0 & 0 & 0 & 0.1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

この隣接行列の行はリンクしているアイテムを指し，列はリンクされているアイテムを指す．つまり，a から b へのリンクの重みは a 行 b 列の要素となる．リンクのない要素は 0 とする．行列 A の転置行列は A^T となる．この隣接行列 A とその転置行列 A^T を用いて原因アイテムや結果アイテムのスコアを求める．

アイテム a,b,c,d は原因アイテムとしてのスコアを要素にもつベクトルを $\mathbf{x}$ と，結果アイテムとしてのスコアを要素にもつベクトルを $\mathbf{y}$ とする．まず，全てのアイテムに対するオーソリティ値をすべて 1 に初期化 ($\mathbf{y}_0$) し，隣接行列 A をもとに原因アイテム値 $\mathbf{x}_1 = A\mathbf{y}_0$ を以下のように求める．

$$\begin{pmatrix} 0.08 \\ 0.12 \\ 0.1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 & 0.06 & 0.02 & 0 \\ 0 & 0 & 0.07 & 0.05 \\ 0 & 0 & 0 & 0.1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

次に，隣接行列の転置行列 A^T を用いて結果アイテム値ベクトル $\mathbf{y}_1 = A^T\mathbf{x}_1$ を以下のように求める．

$$\begin{pmatrix} 0 \\ 0.0048 \\ 0.01 \\ 0.016 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0.06 & 0 & 0 & 0 \\ 0.02 & 0.07 & 0 & 0 \\ 0 & 0.05 & 0.1 & 0 \end{pmatrix} \begin{pmatrix} 0.08 \\ 0.12 \\ 0.1 \\ 0 \end{pmatrix}$$

同様に，この $\mathbf{y}_1$ から $\mathbf{x}_2$ を，$\mathbf{x}_2$ から $\mathbf{y}_2$ を，と計算を進めていく．i 回目のベクトル値を $\mathbf{x}_i$,$\mathbf{y}_i$ とすると，$\mathbf{x}_i = AA^T\mathbf{x}_{i-1}$,$\mathbf{y}_i = A^TA\mathbf{y}_{i-1}$ となる．このように毎回，原因アイテムのベクトルは AA^T 倍に，結果アイテムのベクトルは A^TA 倍に増えていくことになる．

原因アイテムのベクトル $\mathbf{x}_i$ と結果アイテムのベクトル $\mathbf{y}_i$ は，$i \to \infty$ に近づけていくと $\mathbf{x}_i$,$\mathbf{y}_i$ はそれぞれ $\mathbf{x}_{i-1}$,$\mathbf{y}_{i-1}$ の単なる定数倍となる．このときの $\mathbf{x}_i$,$\mathbf{y}_i$ は，それぞれ AA^T,A^TA の固有ベクトルとなり，定数は AA^T,A^TA の第一固有値 ($= \lambda_x, \lambda_y$) となる．第一固有値は Power Method を利用して簡単に求めることができる．これらを考慮すると最終的には，式は $\mathbf{x}_i = \lambda_x\mathbf{x}_{i-1}$，$\mathbf{y}_i = \lambda_y\mathbf{y}_{i-1}$ となる．また，実際の計算過程ではアイテムのスコアが無意味に減少しないように，毎回正規化を行う．アイテムスコアは正規化することにより一定の値に収束していく．

この計算で求めた固有ベクトル $\mathbf{x}$ でその要素の値が高いものほど併買を導く重要な原因アイテムであると考え，同様に固有ベクトル $\mathbf{y}$ でその要素の値が高いものほど他のアイテムから併買されやすい重要な結果アイテムであると考える．

以下で図 10 の例での各スコアを示す．

$$\mathbf{x}: \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \to \begin{pmatrix} 0.63 \\ 1.61 \\ 1.75 \\ 0 \end{pmatrix} \to \begin{pmatrix} 0.37 \\ 1.66 \\ 1.97 \\ 0 \end{pmatrix} \to \cdots \to \begin{pmatrix} 0.25 \\ 1.66 \\ 2.08 \\ 0 \end{pmatrix}$$

$$\mathbf{y}: \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \to \begin{pmatrix} 0 \\ 0.80 \\ 1.20 \\ 2.05 \end{pmatrix} \to \begin{pmatrix} 0 \\ 0.36 \\ 1.20 \\ 2.40 \end{pmatrix} \to \cdots \to \begin{pmatrix} 0 \\ 0.17 \\ 1.15 \\ 2.70 \end{pmatrix}$$

この結果から，原因アイテムとしてスコアの高いアイテムは c で，結果アイテムとしてスコアの高いアイテムは d となる．

3.2 ページランク

相関ルールグラフに適用できる他の解析手法として google の中核解析要素の一つであるページランクが挙げられる．これは相関ルールグラフのようなリンク関係があった場合に，どのページが最も重要であるかを定量的に知るためのランキング手法である [5]．ページランクのアルゴリズムはページのリンク先にリンク関係の強さに応じたスコアを分配して，分配された和をスコアとして更新し，それを繰り返す．

ページランクの計算方法について述べる．ここで，$Rank(v)$ をページ v のページランク，N をグラフ G のノード総数，ω_u をページ u からの外向きのリンク数とし，ι_v は v を指しているページの集合とする．そのとき v の i 回目のページランク $Rank_i(u)$ は，

$$Rank_i(v) = \sum_{u \in \iota_v} \frac{Rank_{i-1}(u)}{\omega_u} \tag{9}$$

となる．

しかし，式 (9) では，リンクの状態によってスコアが正常に評価されない場合が存在する．そのため，このままでは相関ルールグラフに適用することはできない．そこで，ダンプニングファクターと呼ばれる値を計算に用いて式を変化させる．

$$Rank_i(v) = c + (1-c) \sum_{u \in \iota_v} \frac{Rank_{i-1}(u)}{\omega_u} \tag{10}$$

ダンプニングファクターは c とし，0〜1 までの定数となる．通常，ページランクは式 (10) を繰り返し適用することにより計算される．

この手法を相関ルールグラフに取り入れて，アイテムのランキングを求める．ページランクにおいてよいアイテムとは，ランキングの高いアイテムから多くリンクされているアイテムをいい，計算方法としては，アイテムのリンク先にリンクの強さに応じたスコアを分配し，その和をアイテムの新たなスコアとして更新する．例えば，図 11 のような相関ルールグラフを考える．

この相関ルールグラフのリンク関係を以下の隣接行列 A で表す．

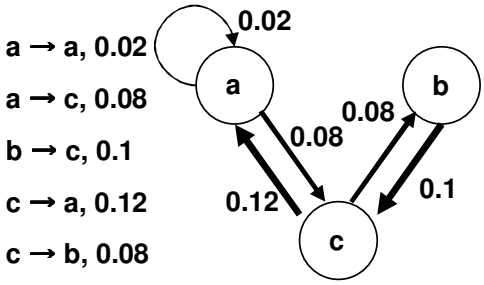

図 11　相関ルールグラフの例 (その 2)

$$A = \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{pmatrix} 0.02 & 0 & 0.12 \\ 0 & 0 & 0.08 \\ 0.08 & 0.1 & 0 \end{pmatrix} \end{array}$$

ページランクは「どれだけリンクしているか」ではなく「どれだけリンクされているか」を重視しているため，隣接行列の行はリンクしている方を指し，列はリンクされている方を指している．また，アイテムのスコアがリンク先に χ^2 の強さに応じて分配されるように隣接行列 A を以下のように修正する．

$$A = \begin{pmatrix} \frac{0.02}{0.1} & 0 & \frac{0.12}{0.2} \\ 0 & 0 & \frac{0.08}{0.2} \\ \frac{0.08}{0.1} & \frac{0.1}{0.1} & 0 \end{pmatrix} = \begin{pmatrix} 0.2 & 0 & 0.6 \\ 0 & 0 & 0.4 \\ 0.8 & 1 & 0 \end{pmatrix}$$

ページランクの計算方法と同じように，アイテム a,b,c のアイテムとしてのスコアを 3 次元ベクトル $\mathbf{r}$ で表わす．すべてのアイテムに対するスコアをすべて 1 に初期化 ($\mathbf{r}_0$) し，隣接行列を用いて以下のように計算する．ダンプニングファクターは（情報検索分野で適当とされる）0.2 に設定する．

$$\begin{pmatrix} 1.0 \\ 0.52 \\ 1.64 \end{pmatrix} = 0.2 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + (1-0.2) \begin{pmatrix} 0.2 & 0 & 0.6 \\ 0 & 0 & 0.4 \\ 0.8 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

このときのアイテム値ベクトルは，$\mathbf{r}_1 = c + (1+c)A\mathbf{r}_0$ と表される．この i 回目の計算の繰り返しによって，アイテムのスコアをベクトル $\mathbf{r}_i$ とすると $\mathbf{r}_i = c + (1-c)A\mathbf{r}_{i-1}$ となり，i を 1 ずつ増加させながら繰り返し，収束するまで計算する．また，計算の過程において毎回正規化を行っていく．この計算で収束したベクトルの要素値がアイテムのスコアを表わし，このスコアが高いアイテムほど，重要なアイテムであると考える．

以下で図 11 の例でのスコアを示す．

$$\mathbf{r}: \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1.00 \\ 0.52 \\ 1.64 \end{pmatrix} \rightarrow \begin{pmatrix} 1.24 \\ 0.72 \\ 1.04 \end{pmatrix} \rightarrow \cdots \rightarrow \begin{pmatrix} 1.00 \\ 0.68 \\ 1.32 \end{pmatrix}$$

この結果からアイテムのランキングが一番高いのは c となる．

3.3 実　　験

ある実際の百貨店のデータから，χ^2 値を重みとする有向枝をもつ相関ルールグラフを計算し，重要アイテムを発掘する上述の手法を適用した．データには約 14,000 点のアイテムがあり，450,000 ものトランザクションからなるデータであった．

このデータから χ^2 値による相関ルールを抽出した．最小サポートを 0.01%の設定で全ルール発掘と BPM の比較を行った．その結果を表 1 に示す．

表 1　BPM と全ルール発掘の比較

	実行時間 (s)	発掘ルール数
BPM	0.796	755
全ルール発掘	0.625	15318

発掘ルールの数を比較すると，BPM のルール数が大きく減っていることがわかる．ここで発掘されたルールが不要なルールを除いた最適相関ルールとなる．実行時間に関しては全ルール発掘よりも計算時間はかかっている．これは最適化問題としてベストパートナーの指名を行う際の枝刈りの効果より，ルールの上限値の計算コストの方が大きくなったことが主な原因と考えられる．このように、データによって BPM の計算量の時間は変わり，場合によっては全ルール発掘の方が短い場合もある．しかし，不要なルールを取り除くことでルール全体の見通しがよくなった点において全ルール発掘より優れているため，BPM は有効であると考えられる．

また，ルール吸収の効果について比較した結果が表 2 である．ルールの吸収によって，発掘ルール数を減らすことができた．

表 2　ルール吸収の効果

	発掘ルール数
BPM +「ルール吸収」	757
BPM	870

さらに BPM の有効性を確認するため，擬似マーケットバスケットデータを作成し，BPM と従来のアプリオリによる全ルール発掘とを比較した．使用したデータは，k アイテムによるアイテム集合が全て最小サポートを満たしていると仮定したものである．アイテム集合のトランザクション数は，サポートの単調性に違反することがないような範囲で乱数を使用して設定した．

「5 アイテム」,「10 アイテム」,「15 アイテム」の 3 つのデータ集合を作成したが，例えば「5 アイテム」の場合は，5-アイテム集合のトランザクション数を 0 から 999 の間の値に，4-アイテムセットを 1000 から 2000 の間の値に，というように 1-アイテム集合まで行い，合計した全トランザクション数が 10000 となるように作成した．同じように,「10 アイテム」なら 10-アイテム集合のトランザクション数を 0 から 999 の間の値に，9-アイテムセットを 1000 から 2000 の間の値に，というように 1-アイテム集合まで行い，合計した全トランザクション数が 20000 となるように作成した．

作成した 3 つのデータセットに対する BPM と全ルール発掘

表 3　アイテムセット数と実行時間 (s) の関係

アイテム数	5	10	15
頻出アイテムセット数	31	1023	32767
BPM	0.047	0.093	620.797
全ルール発掘	0.016	0.578	4907.188

の比較が表 3 である．

アイテム数が増えると，得られる頻出アイテムセットは爆発的に増え，ルール発掘の実行時間も急激に増えていくことがわかる．特に全ルール発掘の計算時間の増加が著しい．一方，BPM は全ルール発掘ほど時間がかからなかった．これは BPM が頻出アイテムセットの数に比例するという予測されるためで，この結果はこれを裏付けるものと考える．

表 4　アイテムセット数と発掘ルール数の関係

アイテム数	5	10	15
アイテムセット数	31	1023	32767
BPM+ルール吸収あり	8	23	28
BPM+ルール吸収なし	8	24	28
全ルール発掘	180	57002	14283372

表 4 は，発掘ルール数についての比較である．実行時間以上に BPM と全ルール発掘の違いが際立っている．

次に最小サポートが 0.001%の設定で（ベストパートナー関係以外のルールを含む）最適相関ルールを求め，その結果 5,302 個のルールを発掘した．これらのルールからなるグラフは 850 個の頂点と 5,302 本の有向枝を持っていた．このグラフから重要な原因アイテム，結果アイテムの抽出を行った．原因アイテムでは婦人用の衣料品がスコア上位に現れた．結果アイテムでもやはり婦人衣料品が強く傾向に現れていた．結果アイテムは上位数点のスコアが非常に高く，リストの下位にいくに従って，急激にその値が減少した．また，ページランクによるアイテムのランキングにおいても，スコアの上位の多くは婦人衣料品であった．(注1)

以上から，この百貨店では婦人衣料品系のアイテムが重要であると判断できる．婦人衣料品を買うことを目的とした顧客はいろいろなものを併買しやすいし，されやすいことがわかることは経営戦略上も役に立つと考える．

4. おわりに

今回，POS データを対象としたアイテム間のリンク分析を行った．まず，アイテム間の強い相関を見るために，相関ルールを χ^2 値を基に求め，併買の強さを調べた．その相関ルールはアイテムの併買パターンとしてわかりやすい法則ではあるが，発掘されるルールの数が膨大になるということと，列挙されたルールからだけではすぐに経営戦略に応用することができないという問題があった．そこで，ルールの数を制限し，併買に関して相関の高いルールだけを取り出す最適化問題に着目し，この最適化問題を効率化し，重要相関ルールだけを見つけ出す BPM のアルゴリズムを実装した．これにより，効率よく有意な重要ルールだけを取り出すことができる．

次に，発掘された重要な相関ルールを有向グラフとして考え，情報検索の分野で用いられている HITS やページランクと呼ばれるデータのランキング手法を適用し，重要なアイテムを計算した．これにより，併買を導きやすい重要アイテム，または併買されやすい重要アイテムを把握することができた．

このようにして得られた結果は経営戦略を立てる上で有効であると考えられる．今後はこれらの知見を使ったさらなる解析手法の確立を目指したい．

今回の実験では，実データにおけるルール発掘の計算時間面での BPM の有効性が確認できなかったが，擬似データの結果は BPM が有効に働くことを支持している．今後はその他の実際の POS データで有効性を確認する必要がある．また，BPM ではルール数が大きく減少したが，そのルールだけで本当に十分であるか，ベストパートナー以外のルールの中に重要なものが含まれていなかったかを詳しく検討する必要もある．

一つの改善案として「デジグネーターから複数指名する」ということが考えられる．今回はベストパートナー以外のルールは切り捨てたが，例えば「2 番目のルールでもベストパートナーとほとんど変わらない強いルール」も存在するだろう．指名の際，ある程度 χ^2 が強ければ複数の相手を指名できるということにすれば，より多くの重要ルールを見つけることができると予測される．

これらのことを踏まえた上で，今後は実データに対して効率よくルールを発見できる，BPM とは別のヒューリスティックな手法を考案することが課題となる．

(注1)：詳細で具体的なランキング結果は契約上開示できないので本論文内では，その全体的傾向をまとめるにとどめた．

文　献

[1] 瀬々潤, 森下真一, “最適なアソシエーションルールの効率的探索, “ソフトウェア科学会データマイニングワークショップ, pp.38-47, 2000.

[2] Rakesh Agrawal, Tomasz Imielinski, and Arum Swami. Mining association rules betweensets of items in large databases. In *Proceedings of ACM SIGMOD Conference*, pp.207-216, May 1993.

[3] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proceedings of VLDB Conference*, pp.487-499, 1994.

[4] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5): pp.604-632, 1999.

[5] S.Brin and L.Page, The anatomy of a large scale hypertextual web search engine, In *Proceeding of WWW Conference*, pp.107-117, 1998.

[6] S.Brin, R.Motwani, and C.Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *SIGMOD 1997, Proceeding ACM SIGMOD International Conference on Management of Data*, pages 265-276, Tucson, Arizona, USA, May 13-15, 1997.

[7] S.Morishita and J.Sese. Traversing Lattice Itemset with Statistical Metric Pruning. In *Proceeding the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 125-134, San Diego, CA 1999.