第10回 データマイニング——巨大データからの知識発見(1)

福田剛志，森本康彦，松澤裕史

はじめに

情報を収集し分析することは，政治，ビジネス，科学，果ては芸術に至るまで人間の知的活動を成功させる鍵である．「情報」といえばなにかすぐに役立つ形で与えられるもののようだが，実際突き詰めれば最終的にはビット列で表されるデータの集まりに過ぎず，そこからどのような意味を汲み取るかは，人によってあるいは文脈によって大きく異なる．

コンピュータにデータを入力するだけでその意味を解釈し，まるで人間のように複雑な意思決定を行うシステム——すなわち人工頭脳は，計算機にかかわる科学者すべての夢である．しかしながら，人間のような知性を持つコンピュータを作ることは，しばらく実現しそうにない．当面最終的な意思決定は賢い人間様に任せるのが実用的である．

そこで情報を有効に利用するためにはまずデータを収集し，それを人間が解釈しやすい表現に変換する必要がある．コンピュータは昔からこのデータの収集とその整理・集約のために用いられてきた．データベース，スプレッドシート，統計処理ソフトウェア，視覚化ツールなどは皆，このための道具である．

しかし昨今のデータ入力技術の進歩，各種業務のオンライン化，インターネットの普及，記憶装置の大容量化と劇的な低価格化などによって，取り扱わなければならないデータの量が爆発的に増えた．このため従来のデータ収集，整理，集約の技術では追いつかなくなってきてしまった．そこで脚光を浴びたのがデータウェアハウジング，OLAP，データマイニングといった大量のデータを蓄えてそこから「情報」あるいは「知識」を抽出するための一連の技術である．本稿の主題であるデータマイニングの“マイニング”とは採鉱するという意味で，文字通り山のようなデータから価値ある情報を掘り出そうというわけである．

データマイニングの推進要因

データマイニングが注目を集めたのは，技術的な要因からばかりではない．現代のビジネス環境の大きな変化がデータマイニングの強力な推進役となっている．

これまで厳しい競争に曝されてきた企業は，競争力をつけるため内部のリストラ・業務改革を続けてきた．しかしその改革も限界に近くなり，これ以上簡単には大きな効果を望めなくなってしまった．そこで企業が競争に勝って利益を上げるためにはその内部だけでなく，外部との関係まで含めた効率化が求められている．近ごろ CRM (Customer Relationship Management) とか SCM (Supply Chain Management) がもてはやされているのはこのためである．

市場全体が今後大きく成長することの見込めない業界では，この傾向はひときわ顕著で，シェアを拡大するためには競争相手から顧客を“盗み取る”必要がある．このためには顧客と競争相手の分析が不可欠である．最近の顧客は十分な情報を持っており，要求が次第に厳しくなってきていることも見逃せない．

約 100 年前に G. Bell によって発明された電話は，半世紀以上の歳月をかけて市場に浸透した．同じころに T. Edison によって発明された電灯もやはり同様であった．その後のテレビや冷蔵庫のような家電製品も，数十年という時間をかけてゆっくりと家庭に入ってきた．これに対して現代は，ポケットベル，携帯電話，ゲーム機，パソコン，インターネットを通じたサービスなど見ればわかるように，次々と新

〈表1〉販売データ（ファクトテーブル）

日時	店舗	商品	顧客	単価	数量
1999/07/16	商店1	自転車A	顧客X	20,000	2
1999/07/16	商店2	一輪車B	顧客Y	8,000	4
1999/07/19	商店3	スクータC	顧客Z	12,0000	1
⋮	⋮	⋮	⋮	⋮	⋮

〈表2〉店舗データ（次元テーブル）

店舗	店舗形態	地域	都道府県
商店1	個人商店	東北	岩手
商店2	スーパー	関東	埼玉
商店3	量販店	九州	宮崎
⋮	⋮	⋮	⋮

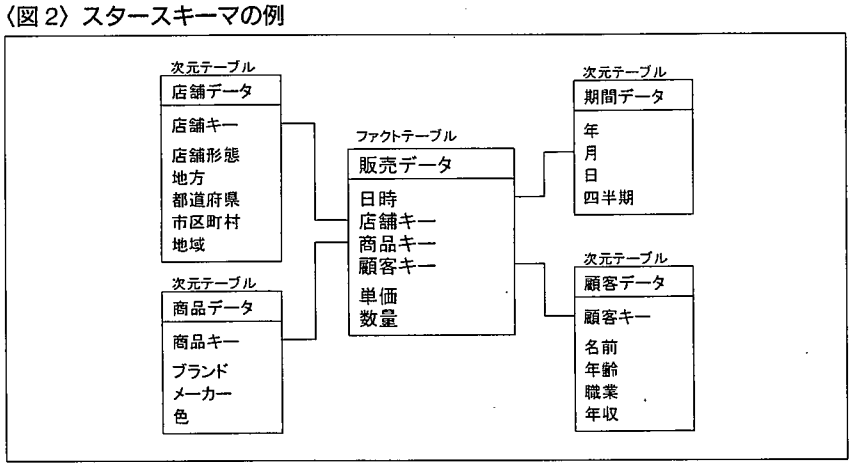
〈図1〉ある期間の地域ごとの売上高を集計する問い合わせ

```
SELECT 地域, SUM(単価 * 数量) AS 売上
FROM 販売データ, 店舗データ
WHERE 販売データ.店舗 = 店舗データ.店舗 AND
      日時 BETWEEN 1999/04/01 AND 1999/06/30
GROUP BY 地域.
```

〈表3〉集計結果

地域	売上
東北	1,234,000
関東	2,468,000
九州	1,357,000
⋮	⋮

〈図2〉スタースキーマの例



しい商品やサービスが現れ，短い期間で急速に普及し，場合によってはあっという間に市場から消えていく．このような環境では一瞬の判断の遅れが致命傷となる．

また，従来は価値が低いと思われて未開発であったニッチ市場でも，IT技術と巧妙なビジネスモデルで大きな利益が上がる可能性がある．対象とする顧客がごく小さなコミュニティであっても，そのニーズをうまく捕らえた商品を提供できれば，非常にロイヤリティの高い優良な顧客として囲い込むことができるからである．

データマイニングの技術はこうしたビジネス環境の下で生まれ，過剰ともいえる期待を集めて実際の意思決定の現場に急速に広がった．

データウェアハウスとOLAP

例えばある企業の売り上げデータとして，「いつどの店舗でどの商品を誰にいくらでどれだけの量販売した」という事実がデータベースの表1のようなテーブルに登録されているとする．この会社の販売したものをすべてこのテーブルに記録しておけば，それを集計することによって，この会社がどの商品でどの顧客からいくら収益を上げているかといったことをいつでも調べることができる．このような事実を集めた表をファクトテーブル（fact table）と呼ぶ．

このファクトテーブルをユーザの興味に従っていろいろな角度から集計するために，店舗，商品，顧客の分類や時間の階層（週，月，四半期，年度など）を表す別のテーブルを用意する．例えば，表2のような店舗を分類するリレーションがあれば，表1のファクトテーブルとジョインすることにより，店舗形態ごと，地域ごと，県ごとの売り上げを集計することができる．図1と表3は地域ごとの第2四半期の売上高を集計する問い合わせとその結果の例である．

この店舗データのように集計の軸となる値を保持するリレーションを次元テーブル（dimension table）と呼ぶ．このようにファクトテーブルの周りに次元テーブルをいくつか持つようなスキーマをスタースキーマ（star schema）という（図2）．いくつもの次元を自由に組み合わせて（例えば地域，時間，商品），その組み合わせの任意の位置（例えば関東，先月，自転車）に集約演算の結果が存在するのでこの構造を多次元の立方体になぞらえて，データキューブ（data cube）[1]と呼ぶ．

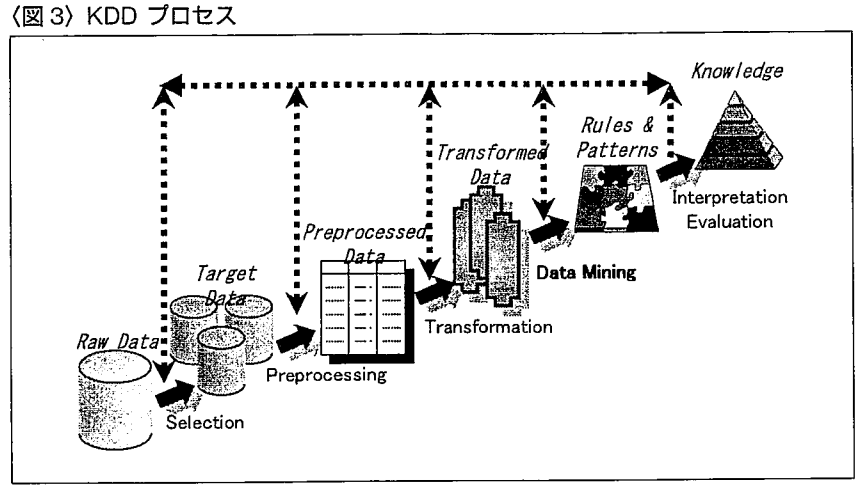
例えば店舗が持つ商品の在庫や注文を管理するためのデータベースのように日常業務に用いられるデータベース

(オペレーショナルデータベース) は頻繁に書き換えられるものの，(新しい店舗ができたりしなければ) 全体のデータ量は基本的にあまり変わらないことが多い．これに対して，このファクトテーブルに格納されるデータは時間を追って追加されていく一方なので，その量は増え続け，その結果オペレーショナルデータベースに比べてずっと大きくなる．しかし，ファクトテーブルも次元テーブルも基本的に読み出し専用であまり書き換えられることがない．このような性質を持つ大量のデータを格納するデータベースあるいはそのための仕組みを，データウェアハウス (データの倉庫；data warehouse) という．スタースキーマはデータウェアハウスの典型的な構造である．

データウェアハウスは先に挙げた例のように，さまざまな軸 (次元) に沿って集計され，企業の経営戦略を立案するための情報源となる．このような人間の意思決定をサポートするためのデータベースは，ユーザの気まぐれな興味に従ったアドホックな問い合わせを受け付け，実時間で結果を返さなければならない．このような分析的な処理を行うことを OLAP (On-Line Analytical Processing)[2] と呼ぶ．

従来のオペレーショナルデータベースが行う OLTP (On-Line Transaction Processing) では，例えば銀行口座間の振り替えのように定型的で簡単な問い合わせを1秒間に何回こなせるかというような効率が追求されてきた．これに対して OLAP では，事前に定義されない複雑な (主に集約演算を伴う) 分析的問い合わせに，できるだけ短い応答時間で答えることが必要となる．このための技術として，さまざまな問い合わせに答えるために共通に役立つ計算を事前に行っておく手法[3][4]や，専用の記憶モデルを持つ多次元データベース (multi-dimensional database) の研究・製品化が盛んに行われた．

〈図3〉KDD プロセス



KDDプロセス

OLAP が人手による仮説検証をサポートするための技術であるのに対して，データマイニングはデータ分析を一歩自動化の方向へ進めようとする技術である．

データマイニングのプロセスはまずマイニングするデータの準備 (データ源の選択，データの形式の変換，標本抽出，集約，欠損値や無効値のコード化，数値の離散化など) を行い，続いてその結果にデータマイニングアルゴリズムを適用して，パターンやルールを発見する．そしてその結果は，視覚化されてユーザに提示されるか，販売キャンペーンを管理するシステムのような，マイニング結果を入力とする下流のアプリケーションで用いられる．この一連のプロセスを KDD (Knowledge Discovery and Data mining)[5] プロセスと呼び，データマイニングはその中のルールやパターンを導く1ステップと位置付けることができる (図3参照)．

市販のデータマイニングツールはデータマイニングの中心的な役割を果たすが，どのようなツールを使っても，問題領域の十分な知識や技術知識がなければ，効率的な知識発見は困難である．経験のないユーザがデータマイニング技術を有効に利用できるように，コンサルティングサービスや，教育サービス，データ分析の請負サービスなどを提供しているベンダも少なくない．

統計学とのかかわり

データマイニングは統計学の世界では古くからある言葉で，「適切な仮説を前もって持たずに，しらみつぶし，デタラメにパターンを探すこと」というような，否定的な意味で用いられてきたそうだ[6]．鉱業はやみくもに地面を掘っていると考えられていたのだろうか．

従来，統計学者が取り扱うデータは，特定の問題を念頭において，それに答えるために収集されることが多い．しかしすでに述べたように，コンピュータの記憶装置には大量のデータが，日常の業務の結果として蓄えられている．このようなデータに対しては，適切な仮説を前もって立てることは容易ではない．それで以前は否定されてきたデータマイニングが脚光を浴びているのである．

データマイニングの定義はいろいろありえるが，「大規模なデータから思いがけない（unsuspected）パターンを発見すること」といってよいだろう．仮説検証型のアプローチでは，はじめに仮説を立てる以上，事前に予想できる結論しか導くことができず，「思いがけない」パターンを発見するには至らない．また，統計学ではデータ全体を説明するような大域的なモデルを構成する方法をとることが多いが，データマイニングではすぐあとで例示するように，データの細かな一部でしか成り立たないようなパターンにしばしば注目する．

このように，データマイニングは探索的データ解析（exploratory data analysis）（例えば文献[7]）にごく近い目的をもっているが，あくまで実用技術として，(1)非常に大規模なデータを対象としていること，(2)データの収集法に対してコントロールがきかないことが多いこと，(3)新しい種類のデータやパターンに注目していること，(4)人間とコンピュータがいかに役割を分担できるかに注目していること，などの点が強調されているといった差があるように思う．

タスクと手法

データマイニングは大きく分けると次の4種類の異なるタスクに分類される．

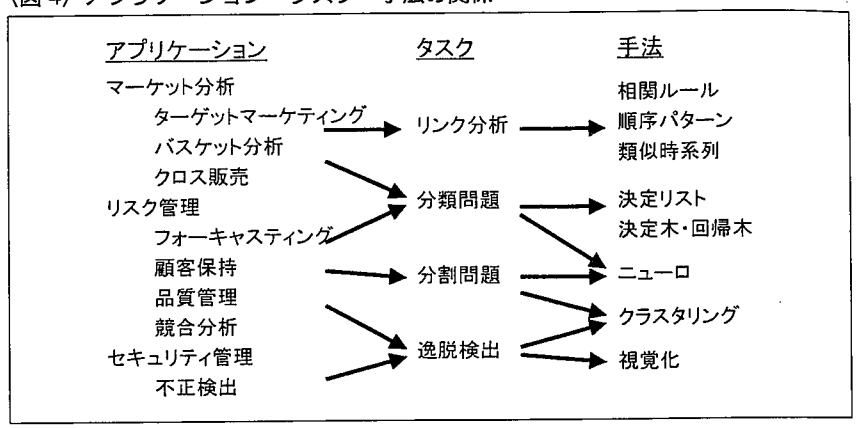
1. リンク分析（link analysis）：商品，イベント，数値などの間の関係（共起，時間，因果）を発見する問題．
2. 分類（クラス分け；classification）：あらかじめ分類の分かっているデータを分析して，分類の規則を発見する問題．
3. 分割（セグメンテーション；segmentation）：似かよったデータをまとめるようなグループ（クラスタあるいはセグメントと呼ばれる）を発見する問題．
4. 逸脱検出（deviation detection）：はずれ値を発見する問題．

1つの目的を持ったアプリケーションでも，いろいろな側面から異なるタスクに定式化して解くことがある．例えば顧客の分類をしたいと思ったとき，分類問題に定式化するのがもっとも素直だが，あらかじめどのような分類が良いか明らかでない場合はクラスタリングしてみるべきだし，分類の要因を分析するためには要因間のリンク分析を行うこともある．

また，それぞれのタスクを解くためにいろいろな手法が考案されている．図4にこの関係を大雑把にまとめた．

データマイニングと知識発見に関する情報は文献[8]に，データマイニングのツール，ソリューション，企業，ウェブ上の情報へのリンク集，求人情報，教育コース，出版物，データセットなどに分類されて良くまとめられている．今回はデータマイニングの最も基本的な手法である相関ルールについて説明する．

〈図4〉アプリケーション・タスク・手法の関係



相関ルール

例としてスーパーマーケットのキャッシュレジスタで収集されるデータに注目しよう．顧客が買い物かごに入れた商品のラベルを，店員がバーコードリーダを使って次々に読み込んでいく．このように収集されたデータは，どのような商品が同時に買われたかという事実の集まりであると考えることができる*1．そこから例えば「目玉商品Aと日用品Bを購入した顧客は，同時に高級品Cも高い確度で購入することが多い」という事実がわかれば，

- A，B，Cのセット商品を発売する．
- AやBの特売を行う際にはCの在庫を増やしておく．
- 顧客の利便性を考えて，商品の配置を近づける．
- 逆に顧客に店内を長く歩き回ってもらうため，商品の配置を遠ざける．

などといった販売戦略を立てることができそうである．さらに目玉商品Aと日用品Bを購入した顧客のうち，何パーセントが高級品Cを購入するか（後で定義する確信度）が判明すれば，目玉商品Aと日用品Bの売り上げ傾向から，高級品Cの売り上げを

*1 「いつ」「どんな人が」「いくらで」といった情報も収集されているが，ここでは無視する．

ある程度予測することができるだろう．この事実は，

{目玉商品 A，日用品 B}⇒{高級品 C}

という式で表現できる．

一般に X，Y を集合として

$$X \Rightarrow Y$$

と記述される事実を相関ルール（association rule）と呼ぶ．

対象とする問題は小売店で売られる商品の併買関係に限る必要はない．顧客の特徴（年齢，性別，職業，趣味）の間の関係を調べるために用いてもよいし，テキストマイニングでは，文書データを対象として，キーワードの間の相関ルールを求めることもある[9]．時間的に引き続いて起こる事柄の間の関係を順序パターン（sequential pattern）として発見する問題にも類似の考え方が応用できる[10]．

「相関」は本来“correlation”に対する訳語として用いられており，筆者らは当初“association rule”に対して「関連ルール」とか「結合ルール」という訳語をあててきた．実際，correlation と association は区別すべき概念だが，どうやら相関ルールという語が一般に定着してしまったようなので，ここでもそれを採用する．

商品の集合を好き勝手自由に組み合わせれば，形式的には相関ルールを作ることができる．こうして作られる相関ルールの総数は，商品がたった10種類ある場合で約57,000個，100種類の場合は 5.15×10^{47} 個以上というとんでもない値になる*2．こんなに多くては一つひとつしらみつぶしに評価していくわけには行かないし，その中で役に立つものはほんのわずかであるに違いない．ではどういう相関ルールに価値があるのだろうか．

価値ある相関ルールはある程度以上確からしいことが必要なのは当然だろう．相関ルール $X\Rightarrow Y$ の確からしさは，$Pr[Y|X]$ の推定値すなわち，商品の集合 X を購入した顧客数 a のうち，Y をも購入した顧客数 b の割合 b/a で計り，この値を相関ルールの確信度（confidence）と呼ぶ．さらに，相関ルールが適用できるデータの量がある程度以上大きいことが重要である．なぜなら，あまりにわずかなデータにしか適用できない相関ルールは出番が少なく，役に立つ機会が少ないからである．適用できるデータ量は相関ルールの X と Y を同時に購入した顧客数 b の全顧客数 N に対する割合 b/N で図るのが普通で，この値を相関ルールのサポートと呼ぶ．そこで確信度とサポートがある程度以上大きい相関ルールが有効で，価値があると考えることにする*3．

価値の高い相関ルールを作るアイテム集合があらかじめわかっていれば，データベースに対する簡単な問い合わせで，相関ルールの確信度とサポート（あるいはその他の指標による価値）を知ることができる．商品集合ごとに購入した顧客の数を数えて集計すればよいのだから，これは OLAP の得意とする演算である．しかし通常は，どのアイテムを組み合わせれば価値のある相関ルールができるかは事前にはわからない．もしわかっていれば，そもそもマイニングの必要などないのであるから，ユーザが価値のありそうな相関ルールの候補を与えることを仮定するわけにはいかない．すべての相関ルールを調べて重要なものを選ぶという方

*2 m 個のアイテムから相関ルールに用いる k 個のアイテムを選ぶ $\binom{m}{k}$ 通りのそれぞれについて，相関ルールの前提部と結論部の分け方が 2^k-2 通りある．

*3 次号では別の指標を導入して相関ルールを評価する．

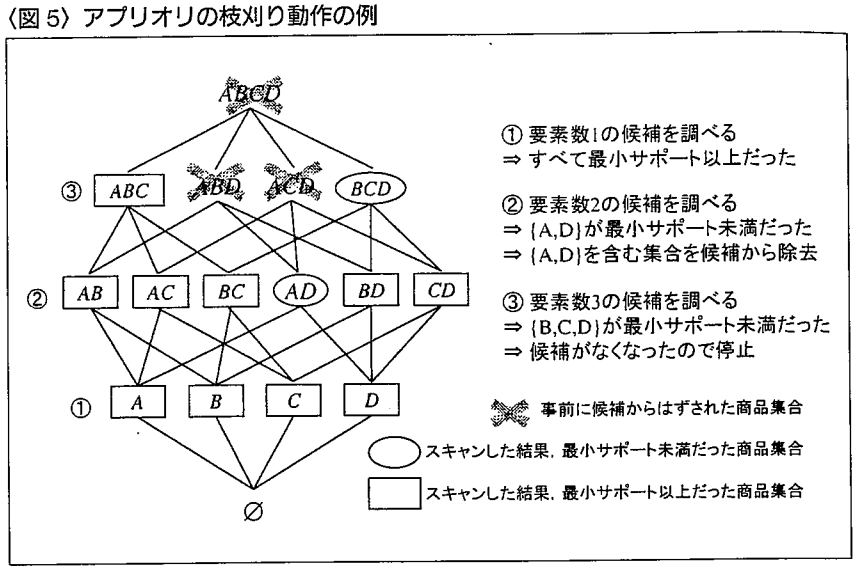
法は，潜在的な相関ルールの数があまりにも多いため，役に立たない．そこで，自動的にデータベースから価値のある相関ルールを効率的に，しかも漏れなく発見する方法が必要となる．

アプリオリアルゴリズム

IBM アルマデン研究所の Rakesh Agrawal らは 1994 年，ユーザが最小確信度と最小サポートを与えて，それ以上の確信度，サポートを持つ相関ルールをすべて発見する効率的なアルゴリズム“アプリオリ”(apriori) を発表した[11]．これは後に数多くの派生研究を生むこととなり，データマイニングの研究に火をつけたともいえる重要なアルゴリズムである．同研究所で開発された Quest システム[12][13]はアプリオリを含む数々のデータマイニング手法を実装した世界初の本格的なデータマイニングシステムである．

アプリオリアルゴリズムは次のような単純な観察に基づいて，探索すべき相関ルールの枝刈りを行う．商品の集合 I を購入した顧客の全体に対する割合（I のサポートという）を support(I) とすると，I を含む商品集合 $J \supseteq I$ のサポート support(J) は support(I) を超えることはない．したがって，もし I のサポートがユーザの与えた最小サポートよりも小さければ，I を含むようなどんな商品集合も最小サポート以上のサポートを持つことはありえない．相関ルール $X \Rightarrow Y$ のサポートと確信度を求めるために必要な support(X) と support($X \cup Y$) は X と $X \cup Y$ を購入した顧客をそれぞれ数えれば求めることができるが，support($X \cup Y$) は相関ルールのサポートそのものなので，この値が最小サポートより小さい場合には，$X \cup Y$ とそれを含む商品集合は相関ルールを作るために用いる商品集合の候補からはずすことができる．

〈図 5〉アプリオリの枝刈り動作の例



注意しなければならないのは，データベースは膨大で主記憶には普通入らず，2 次記憶に置かなければならないことである．一方，相関ルールの数は通常はたかだか数千程度で，相関ルールだけは上手に主記憶内に保持したい．そこで候補となる商品集合を主記憶内に置き，顧客 1 人ひとりが購入した商品のリストを 2 次記憶から逐次的に読み出して，候補となっている商品集合がそこに含まれていれば購入顧客数を 1 増やすということを繰り返す．データベースのスキャンはコストがかかるので，その回数を減らすため 1 回のスキャンで複数の商品集合の候補を同時に処理する．オリジナルのアルゴリズムは，k 回目のスキャンで k 個の商品からなる候補をまとめて処理している*4．動作の概略を図 5 に図示する．

ここで処理上のボトルネックとなるのは，候補となっているたくさんの商品集合の中から，各顧客が購入しているものを見つける部分である．この高速化には，ハッシュ木を使う方法による実装方法が用いられている．実装方法の詳細は，文献[16]にある解説や文献[17]に公開されている実装例を参照するとよいだろう．

またデータを並列計算機上に均等に分配して置けば，入出力を並列化できることに加えて，条件を満たすデータ数を集計する操作も自然に並列化でき，質の高い台数効果が確認されている[11][18]．実際，データマイニングが扱うデータ数は数百万から数千万件に達することが普通であり，このような大量データの場合は並列計算機の使用が欠かせない．

以上のようにして求めた最小サポート以上のサポートを持つ商品集合の中から $X \cup Y$ と X に相当する 2 つの商品集合を選んで $X \Rightarrow Y$ を作れば，1 つの相関ルールの候補ができる．このなかから確信度，すなわち support($X \cup Y$)/support(X) が最小確信度以上のものを選べば，アルゴリズムの目的は達成される．

さて，1 つの相関ルールを取り出してみると，あまりにプリミティブな情報に見える．しかし人間が持っている知識や先入観と対比させたり，複数の相関ルールを比較することにより，相関ルールは役立つ知識となる．このた

*4 ここは工夫のしどころで，さまざまな改良が提案されている．例えば[14][15]．

め，一度にたくさんの相関ルールを見比べるて解釈することができるような視覚化ツール（例えば文献[19]）は，相関ルールを活用する上で非常に役に立つ．

次号では，より高度な知識を発見する方法として，最適な相関ルールを発見する方法や，それを複数組み合わせて分類モデルを構成する手法，さらに新しいデータマイニングの応用について説明する予定である．

参考文献

[1] Jim Gray, Adam Bosworth, Andrew Layman, and Hamid Pirahesh : Data cube : A relational aggregation operator generalizing group-by, crosstab, and sub-totals. Technical report, Microsoft, November 1995.

[2] E. F. Codd, S. B. Codd, and C. T. Salley : Beyond decision support, *Computerworld*, Vol. 27, No. 30, July 1993.

[3] A. Gupta, V. Harinarayan, and D. Quass : Aggregate-query processing in data warehousing environments, *Proc. of VLDB Conference*, pp. 358-369, 1995.

[4] Venky Harinarayan, Anand Rajaraman, and Jeffrey D. Ullman : Implementing data cubes efficiently, *Proc. of ACM SIGMOD Conference*, pp. 205-216, June 1996.

[5] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy : *Advances in Knowledge Discovery and Data Mining*, MIT Press, Cambridge, MA., 1996.

[6] David J. Hand : Data mining : Statistics and more ?, *The American Statistician*, Vol. 52, No. 2, May 1998.

[7] John W. Tukey : *Exploralory Data Analysis*, Addison-Wesley, Reading, 1977.

[8] http://www. kdnuggets. com/

[9] 川原稔，河野浩之，長谷川利治：文献データベース情報検索に対するデータマイニング技術の適用，情報処理学会論文誌，Vol. 39, No. 4, pp. 878-887, 1998.

[10] Ramakrishnan Srikant and Rakesh Agrawal : Mining sequential patterns : Generalizations and performance improvements, *Proc. of EDBT*, 1996.

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

[12] Rakesh Agrawal, Andreas Arning, Toni Bollinger, Manish Mehta, John Shafer, and Ramakrishnan Srikant : The quest data mining system, *Proc. of KDD*, August 1996.

[13] http://www. almaden. ibm. com/cs/quest/

[14] Serrgay, Brin, Rajeev Motowani, Jeffery Ullman, and Shalom Tsur : Dynamic itemset counting and implication rules for market basket data, *Proc. of ACM SIGMOD Conferebce*, pp. 255-264, 1997.

[15] Roberto J. Bayardo Jr. : Efficiently mining long patterns from databases, *Proc. of ACM SIGMOD Conferebce*, May 1998.

[16] 福田剛志，森本康彦，徳山豪：『データマイニング』，データサイエンスシリーズ，共立出版，印刷中．

[17] http://fuzzy. cs. uni-magdeburg. de/~borglt/

[18] Takayki Shintani and Masaru Kitsuregawa : Parallel mining algorithms for generalized association rules with classification hierarchy, *Proc. of ACM SIGMOD Conferebce*, pp. 25-36, 1998.

[19] 福田剛志，森下真一：相関ルールの可視化について，電子情報通信学会技術研究報告 95-81, pp. 41-48, May 1995.

●福田剛志
（ふくだ・たけし）
日本アイ・ビー・エム(株)
東京基礎研究所
主任研究員．博士（情報科学）．ソフトウェアのリバースエンジニアリング，オブジェクト指向データベース，データマイニングなどの研究に従事．オープンウォータスイミングが趣味で，毎年初島熱海間12km の遠泳に参加している．97年にはリレーでドーバー海峡を完泳．現在，研究所の運営管理の仕事をしながら充電中．

●森本康彦
（もりもと・やすひこ）
日本アイ・ビー・エム(株)
東京基礎研究所
データマイニング，機械学習などの研究に従事．投資信託，外貨預金から競馬まで自ら体を張ってデータマイニングの実験中．この研究はまだ発展する余地が大きい事を実感している．ラーメンとビールをおいしくいただくためにランニングを始めたら，いつのまにかフルマラソンに参加するようになりました．4時間台が目標ののんびりランナー．好きな言葉は「自由」と「まあ，ええやないか」

●松澤裕史
（まつざわ・ひろふみ）
日本アイ・ビー・エム(株)
東京基礎研究所
副主任研究員．ソフトウェアの部品化・再利用，3次元ソリッドモデリング，データマイニングなどの研究に従事．スキーとドライブが趣味で，冬の週末は頻繁に志賀高原に足を運んでいる．現在は，テキストマイニングの研究に従事．