

特別論説


 情報処理最前線
 情報処理最前線

 データマイニングの最新動向[†]
 —巨大データからの知識発見技術—

 福田 剛 志^{††} 森 本 康 彦^{††}
 森 下 真 一^{††} 徳 山 豪^{††}

In a canyon, in a cavern,
 Excavating for a mine
 Lived a miner, forty-niner
 and his daughter Clementine.*

1. はじめに

近年、巨大データベースから知識獲得を高速に行う技術として、データマイニングがデータベースと人工知能の境界分野で注目されつつある⁷⁾⁹⁾。この論説では、データマイニングの理解を深めることを目的として、「結合ルール」の生成と利用を中心に、データマイニングの実例を紹介する。

2. データマイニングとは?

データマイニングはデータベースの研究の中で比較的新しい分野である⁹⁾。マイニング (mining, 採鉱) という言葉には何ともいえない怪しげな魅力がある。現在ではギャンブラー的な投機家、ひいては詐欺師のような意味で使う「山師」は、もともとは鉱山経営者や鉱山採掘調査技師 (mining engineer) のことである。広大な大地の中から金や銀、石炭の鉱脈や水脈を発見する事業は、1つにはギャンブルの要素があり、また、鉱山採掘師の理論は常人には詐欺か超能力としか思えない不思議なものだったに違いない。伝承によれば、名高い弘法大師は多くの温泉や水脈を発見した優れた鉱山採掘師だったが、その行為は「弘法の奇跡」として言い伝えられるのがほとんどで

ある。

バーコードやクレジットカードなど最近のデータ収集技術の大幅な進歩と、記憶装置の劇的な低価格化により、情報収集はたやすい作業になり、山にたとえられるほど巨大なデータがすでに存在する。この山のようなデータベースから属性やデータ間に成り立つ規則を絞りだし、営業戦略を立案する上で役に立つ規則・法則のたぐいを得たいという自然な欲求が生まれる。しかし現行のデータベースシステムは検索・集計の簡易化と効率化を目標に作られており、残念ながら規則生成を目的には作られたわけではない。

たとえば、巨大な顧客データベースを持つ商店が、商品のダイレクトメールを顧客に出す場合、商品ごとに、どのような顧客たちのグループに焦点を合わせると利益を最大にできるか調べたいでしょう。これはほとんどの場合、既存のデータベースの情報を十分に解析できれば答が出るはずである。もちろん、成人式や七五三の着物のように年齢などではっきり限定される特定商品では、現在でも顧客の絞り込みを行っている。しかし、一般の商品ではデータベース解析の技術がともなわないために、すべての顧客、または非常に漠然とした基準を満たす顧客にダイレクトメールを出し、多大なコストを無駄にすることが日常茶飯事である。

データマイニングの目的を鉱山採掘になぞらえて説明してみよう。何も予備知識を用いずに巨大データから有効な法則を一発で取り出すとしたら、これは、「ここを掘りなさい」と杖で叩くと温泉が湧き出る「弘法の奇跡」以外のなにもものもないし、また、おそらく非常に単純な法則しか取り出せない。弘法大師の時代の鉱山学の道具はかなり文学的な知識 (格言) であった。「流れ出る川で砂金が取れば金脈がある」「大きな火山の

[†] Recent Progress on Data Mining Technology Knowledge Discovery from Giant Databases by Takeshi FUKUDA, Yasuhiko MORIMOTO, Shinichi MORISHITA and Takeshi TOKUYAMA (IBM Japan Ltd., Tokyo Research Laboratory).

^{††} 日本アイ・ビー・エム(株)東京基礎研究所

* Oh my darling, oh my darling, oh my darling Clementine!
 Thou art lost and gone forever. Dreadful sorry,
 Clementine.

「雪山讃歌」(雪よ岩よわれらが宿り…) 元歌

麓には温泉が出る」などは経験から出てきた知識と言えよう。経験はすなわち個人の持つデータベースである。これに対応するのは、ユーザが「この辺に法則があるかな?」と予測して(経験を元にした勘)、データベース問合せ(岩石の採取や地形の観察)を用いてその法則の有無を判定し、確認することである。データマイニングの第1のステップは、データベースの属性間にこれらの「格言的」ルールを多数作り、それらを使ってデータの解析を行うことである。

さらにもう少し進んでみよう。少し古い鉱山採掘の基本は「足」であった。つまり、経験を元にした知識を元にして、有望そうな山に入って、岩石を採取し、地形を見たりして鉱脈があるかどうか判断し、最終的に調査採掘するかどうか決めるのである。もう少し新しい鉱山採掘では「足」を使う前に格言的な知識のみでなく、航空写真や人工衛星からの写真や数値データを用いて地上探索の範囲を限定する。データマイニングの2番目のステップは、ユーザの予測の部分(航空写真を用いるように)自動的に限定してやり、効率の良い予測を行い、少ない数の問合せで済むようにすることである。

さらに航空写真以外にも、海中や地中での音波ソナーや人工地震などを用いた3次元的調査など、先進技術を駆使すれば、調査探索費用に対して採掘の予想利益が大きい場所を自動的に絞り定める。データマイニングの3番目のステップは、ユーザが曖昧な指定をするだけで、最もそれらしい法則を自動的に導出してくれるシステムの構築である。

この論説では、上記のシナリオにおける1番目のステップの一例として、関連データベースの非数値属性間の相関関係を調べる Agrawal たちの仕事^{1),2)}、2番目、3番目のステップに対する初期の試みとして、著者たちの DataBase SONAR (System for Optimized Numeric Association Rules)^{4),5)}と、それを用いた決定木の構成を紹介する。

3. 離散値属性間結合ルール

スーパーマーケットで買物をしたとしよう。商品にはバーコードのついたシールが貼ってあり、レジにはその読みとり装置がついている。この装置

は、店員がバーコードを読ませるだけで、素早く支払金額を計算してくれる。この便利な仕組みは、店員の仕事を楽にしたり、客がレジに列ぶ時間を短くしただけではない。金額の計算と同時に、どの商品とどの商品が、いつ、どのような人に(店員があなたの風貌を見ている)、どれだけ売れたかがコンピュータに入力されたのである。このデータは特別な仕掛けを新たに作らなくても収集できる上に、顧客を知る重要な情報源となる。

収集したデータからたとえば「目玉商品 A を購入した顧客は高い確度で日用品 B を購入する」という規則が得られたとすれば、目玉商品 A が他のどの日用品 B の売上に貢献するかが明瞭になる。さらに目玉商品 A を購入した顧客の何%が日用品 B を購入するかが判れば、売上の予測にもある程度つながる。他の例では、銀行における顧客データベースから「普通口座を保有する顧客の1%は過去にカードローン支払いを延滞したが、普通口座を保有しかつサービス A を利用する顧客の場合は0.05%しか支払延滞者がいない」という規則が得られたとする。この規則を発見できれば、支払延滞率を下げるための有効な対策として、サービス A を顧客に勧めることを検討することができる。

3.1 結合ルール

以後、「目玉商品 A を購入した顧客は高い確度で日用品 B を購入する」という規則を

$$(目玉商品 A = yes) \Rightarrow (日用品 B = yes)$$

と表現する。一般に、A, B, C, ... をあるデータベースの離散値属性とするとき

$$(A = a) \wedge (B = b) \Rightarrow (C = c)$$

という形をした規則を結合ルール (association rule) と呼ぶ¹⁾。

では結合ルールの価値はどのように測るか? 条件部を満たすデータが結論部も満たす割合を確信度 (confidence) と呼び、条件部と結論部を同時に満たすデータの全データに対する割合をルールのサポート (support) と呼ぶ。条件部を満たすデータの割合は条件のサポートと呼ぶ。この節ではルールのサポートを単にサポートと呼ぶ。規則の価値を測る上で確信度が高いことが重要であることは当然であるが、サポートもある程度高いことが望ましい。たとえばサポートが0.01%の

ルールは0.01%の顧客にしか当てはまらない出番の少ないルールであり、あまり価値がないからである。技術的な制約から言えば、非常に小さいサポートを持つルールまですべて探そうとすると、その数は容易に莫大になる。ユーザにとっても、解析する気になるルールの数はせいぜい数千のオーダーであろう。極端に小さなサポート値のルールを排除し、かつルールの数を自然に制限する方法としては、出力するルールのサポートの最小値（最小サポートと呼ぶ）を設定するのが有効である。

結合ルールを出力する本格的なシステムであるIBM アルマデン研究所のQuest^{1),2)}は、ユーザが最小サポートと最小確信度を与えると、すべての結合ルールを出力する。

3.2 実装方法

ユーザが最小サポートを与えたとき、効率的に結合ルールを生成するにはどうすればよいか？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到底入らないので、2次記憶に置くしかないことである。一方ルールの数は通常はたかだか数千程度である。そこでルールだけは上手に主記憶内に保持したい。いまルール

$(A=a) \wedge (B=b) \Rightarrow (C=c)$ のサポートと確信度は、

条件 $(A=a) \wedge (B=b)$ と

条件 $(A=a) \wedge (B=b) \wedge (C=c)$

をそれぞれ満たす顧客数から計算できることに注目する。後者の条件を満たす顧客の割合はルールのサポートを与えるので、この割合が最小サポートより小さい場合には、条件として考察の対象から外せる。したがって、これら条件を主記憶内に置き、条件を満たすデータ数を集計すれば、有効なルールをすべて抽出できる。集計作業は、各データは2次記憶から逐次的にスキャンし、データが満たす条件の各々について顧客数を1増やすことを繰り返せば、最終的に各条件を満たす顧客数を集計できる。

この手法はSrikantとAgrawal²⁾により提案されたものであるが、処理上のボトルネックになるのは、各データが満たす条件をすべて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。またデータを並列計算機上に均等分散しておけば、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、すでに質の高い台数効果も確認されている²⁾。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。したがって、データマイニングは並列計算技術の良い応用例になっており、学会でも注目されつつある。

4. 数値属性間結合ルール

前章で紹介した結合ルールは離散値を持つ属性を扱ったものである。現実のデータベースでは通常、年齢や預金残高のような数値属性が存在する。これら数値属性を離散値を持つ属性として考えてもよいが、数値間の順序も考慮した結合ルールがあれば有用である。たとえば「預金残高が v_1 以上 v_2 以下の顧客は高い確信度でサービスAを利用する」というルールから、サービスAを勧めるべき顧客層が浮かび上がってくる。このルールを前章の形式に習って、

$(\text{預金残高} \in [v_1, v_2]) \Rightarrow (\text{サービス } A = \text{yes})$

のように記述する。

4.1 最適結合ルール

いま区間 $[v_1, v_2]$ があらかじめ与えられていれば、条件 $(\text{預金残高} \in [v_1, v_2])$ を満たす顧客が何%の確信度で条件 $(\text{サービス } A = \text{yes})$ も満たすかは容易に計算できる。しかし現実には区間そのものを見つけることに興味がある。しかも高い確信度を与える区間が分かれば、サービスAを利用する顧客をうまくモデル化したことになる。ここで問題になるのは、区間の取りかたは自由度が大きいことである。極端な例としては、ごく僅かなデータしか含まないきわめて短い区間で、高い確信度を与える区間がいくつもあり得る。いくら確信度が高くても、あまりに短い区間では意味のあるルールとは言えない。

いま条件 $(\text{預金残高} \in [v_1, v_2])$ を満たすデータの数を (預金残高に関する) 区間 $[v_1, v_2]$ のサポートと呼ぶことにする。ユーザは考慮する区間が持つべきサポートの最低限の値を与える。

すると、そのしきい値以上のサポートを持つ区間の中でルールの確信度を最も高くする区間 $[v_1, v_2]$ が、サービスAを利用する顧客を一番特徴づ

けていると言える。このような区間を確信度最大化区間と呼び、ルールを最適確信度結合ルールと呼ぶことにする。これと双対的な概念として、ルールの確信度の最低限の値を与え、サポートが最大になる区間を求める問題が考えられる。このような区間をサポート最大化区間、ルールを最適サポート結合ルールと呼ぶ。

実際システムを作る際には、ユーザは

$$(X \in [v_1, v_2]) \Rightarrow (\text{サービス } A = \text{yes})$$

のように結論部だけを指定すると、システム側が自動的に $X \in [v_1, v_2]$ の具体例 (預金残高 $\in [52$ 万, 67 万], 取引月数 $\in [13, 34]$ など) を確信度の高い順に出力するようになっていけば便利である。

4.2 最適結合ルールの計算

確信度最大化区間、およびサポート最大化区間を計算する初歩的な方法は、すべての区間を生成し、最大の区間を選択することである。この方法の計算時間はデータ数の2乗に比例しそうだが、実は、データがソートされていれば、計算幾何学的方法を使うと線形時間で解けることが知られている⁵⁾。

この方法の計算時間上のボトルネックは、データを対象とする数値属性でソートしておく必要のある点にある。2次記憶上にある数百万件のデータを各数値属性ごとにそのままソートするのは、通常膨大な時間を必要とする。この計算時間を短縮しない限りは、計算幾何学的手法を用いた線形時間アルゴリズムも活かせない。

1つの解決方法として、文献5)では、ランダムアルゴリズムを用いて、数値属性の値を細かい区間(バケット)に分割する。ただし、各区間に入るデータ数がほぼ均等になるように。たとえば図-1で、横軸が預金残高、縦軸が各数値でのデータ数を表現しているとき、横軸の下に表示した各バケットには、ほぼ均等な数のデータが入っている。このような均等なバケットを多数生成した後、いくつか連続するバケットを繋ぎ合わせて確信度/サポート最大化区間を作る。バケット数としては数千程度用意すれば、最大化区間が非常に高い精度で得られる。

4.3 2次元版最適結合ルール

次に、1つの数値属性だけでなく、複数の数値属性を条件部に含む結合ルールの生成について述

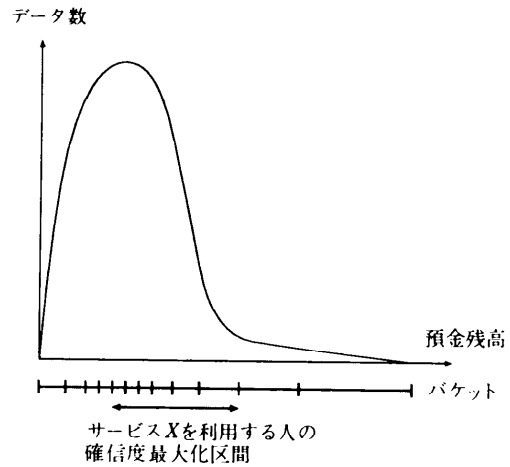


図-1 バケット分割の例

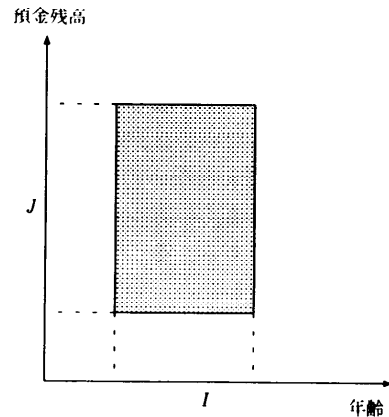


図-2 矩形領域の結合ルール

べる。たとえば、年齢と預金残高を使ってサービスAを特徴づける結合ルールとしては、次のようなルールが頭に浮かぶ。

$(\text{年齢} \in I) \wedge (\text{預金残高} \in J) \Rightarrow (\text{サービス } A)$
条件部の範囲を年齢と預金残高の張る2次元平面で図示すると、図-2の矩形領域になっている。このような矩形領域は、サービスAを利用する顧客のいる範囲を大まかに教えてくれる。また、領域に入る顧客数の最低値をユーザが与えると、最もサービスAを利用する顧客の確信度の大きい領域を見つけることが興味の対象になる。つまり1次元の場合に導入した最適確信度結合ルールの発見問題は2次元以上の場合にも自然に拡張できる。

さらに、図-3に示す一般的な形状をした連結領域Rを使えば、より柔軟にサービスAを利用

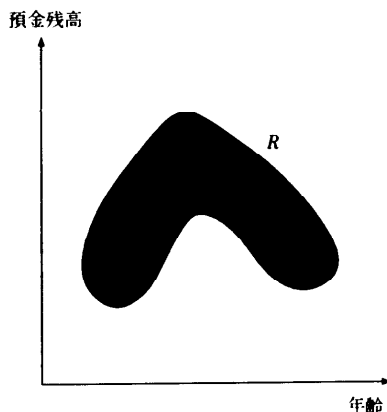


図-3 x-単調連結領域の結合ルール

する顧客を高い確信度で特徴づけられる。ルールとしては「年齢と預金残高が図-3の領域 R に入る顧客は高い確信度でサービス A を利用する」ことを意図した次の形のルールを求める。

$$(\text{年齢}, \text{預金残高}) \in R \Rightarrow (\text{サービス } A)$$

この場合にも最適確信度/サポート結合ルールを自然に定義できる。

このような2次元結合ルールの中で、ユーザが

$$(X_1, X_2) \in R \Rightarrow (\text{サービス } A)$$

のように結論部だけの指定をすると、属性のペアと領域 R が確信度順に出てくるシステムを作るのである。

4.4 2次元版最適結合ルールの計算

まず、2つの属性が張る平面を格子状に分割し、ピクセルを生成する。通常は 20×20 からたかだか 1000×1000 のピクセルへと分解する。

次にピクセルを繋ぎ合わせて矩形領域や連結領域を作る。最適確信度/サポート結合ルールを生み出す領域を求める計算量は、矩形領域や連結領域などのどのタイプの領域を求めようとするかによって大きく異なる。

一般の連結領域を対象にする場合、問題がNP困難であることが知られている^{3),4)}。そこで、少し制限をつけたx-単調領域(y軸に平行な線との交わりが必ず連続か空となる領域)を考える。図-3の R はその例である。濃淡度の最も濃いx-単調領域を取り出す問題はイメージ処理の分野でよく調べられている³⁾。文献3)の方法を使うと、x-単調領域を対象にした最適確信度/サポート結合ルール発見問題を $O(n \ln m)$ (m はデータ数, n はピクセル数)で解くことができる。

数百万件のデータで2次元結合ルールを生成する場合、単一プロセッサでは計算時間のうちで格子分解に90%以上の時間がかかることが実験により確かめられている⁴⁾。並列計算機を用いれば、格子分解を並列化してこのボトルネックを解消し、巨大データを高速に扱うことができる。

5. 結合ルールの利用と拡張

データマイニングにおいては、結合ルールを抽出して示すだけではデータの解析として十分でないことが多い。データの特徴がより簡単に分かり、使いやすい出力を求めることが非常に重要になり、そのためには様々なデータ解析の手法や、最適化技術との融合が必要になる。しかしながら、結合ルールを求めておくことはそのための基礎になるサブルーチンであると思われる。ここでは例として属性クラス分けと決定木の構成を考えてみよう。

5.1 属性のクラス分け

結合ルールをもちいて、様々なデータ解析や資源最適化が行える。スーパーマーケットで商品の配置を考えると、3.1節で述べた結合度の高い商品を近い場所に配置することが顧客の満足度向上のためには望ましい。

結合ルールのリストから、属性(商品)を頂点とし、

$$(A = \text{yes}) \Rightarrow (C = \text{yes})$$

というルールに対して、属性 A に対応する頂点から属性 C に対応する頂点に有向辺を引くと、グラフを作ることができる。有向辺にはルールのサポートを「重み」として置くことができる。

すると、顧客の満足度を高める商品の配置を見出す問題は、(数学的な定式化はいろいろあるが)このグラフ上での最適化問題、典型的には頂点のクラス分け(クラスタリング)問題に対応する。このとき、結合ルールの候補をできるだけ絞っておくことが効率に非常に効いてくる。

現在実用化されているデータマイニング用のクラスタリングのシステムは多数あるが、数理計画などの最新技術を用いたより良いシステムの開発は今後の課題となる。

さらには、何を目玉商品にすると売上が向上するかなどという意思決定問題も、各々の商品に関しての売上のモデルがある場合には、先ほどのグ

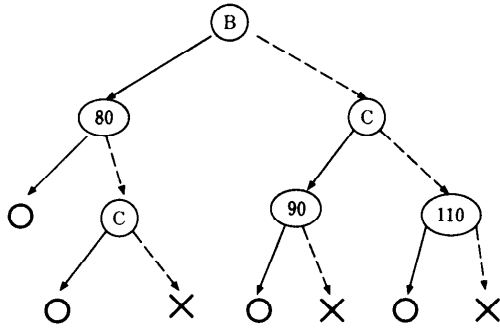


図-4 決定木

ラフを使って大域的な解析をある程度行える。

5.2 決定木

病院において患者の過去の病歴がデータベース化されているとしよう。この過去のデータの蓄積から、どのような症状や健康状態の人に病気 A の病歴があるか否かを経験的に判定する知識を生成すれば、新たな患者が A を患っているか否かを判定するための助けになり便利である。またたとえば、保険会社に蓄積された、利用者の過去の事故歴のデータベースから、特定の交通事故を起こしやすいか否かを経験的に判定する知識などを取り出したいというニーズがある。

決定木とはこのような判定問題にしばしば利用される図-4のような2進木構造である。木の各ノードはデータに対するテストであり、テストで Yes と出たデータは左の枝に進み、No と出たら右の枝へ進む。決定木の各頂点でのテスト文には結合ルールに対応するものが用いられる。図-4の決定木では、条件部の属性として最低血圧（数値）、血糖値（B）、コレステロール値（C）を考えて、血圧がある値以上か、もしくは他の値が+か-かでテストを行い、最終的に、結論部として成人病症候群である（○）かない（×）かを判定する。このとき、深さが小さく、頂点数も小さい決定木で良い判定ができれば理想的である。

最小の決定木の構成問題は NP 困難と言われており、近似的なアプローチとして有名なものに Quinlan⁹⁾ のエントロピーを用いたヒューリスティクスがある。Mehta らによる SLIQ⁹⁾ は同様のアプローチをとり、数千万件のデータに対しても現実的な時間で動作するシステムの構築に成功し注目されている。SONAR システムではさらに、テスト文として（少し変更した）最適結合ルール

を用いて数値属性を効果的に扱う。数値属性たちの間に強い相関がある場合、2次元最適結合ルールを判定に用いると、決定木の大きさは激減する。

6. データマイニングの今後

現状でのデータマイニングの研究は初期のステップにあると思われる。結合ルールの発見やクラスタリングなどは、ある意味でオペレーションズ・リサーチにおける主要問題をそのままデータベース解析の世界に写し替えたものと捉えることができる。考える集合がデータベースの要素であり、異なった構造をしているにもかかわらず、多次元の結合ルール生成では大域最適化理論などの OR の最近の手法をかなり利用することができる。したがって、OR や離散アルゴリズム理論とデータベースの2つの分野の相互の影響がより良いシステムの開発に繋がるとと思われる。

データマイニングが知識獲得システムである関連上、学習機能、あるいは曖昧な判断を行うことも重要な要素である。たとえば、一般に、生成された結合ルールのうちで多くのものは既知のルールであり、いまさら掘り起こしても意味がないものである。そして、これらの「常識的な」ルールの影響が、隠れている重要な、しかし比較的小さいルールをさがしにくくしている。現在の SONAR システムでも、既知のルールの影響を除去して、隠れたルールを見出すことを行う。しかし、ルールが既知であるか、価値があるかどうかの判定は、原則的にユーザ任せである。学習や、類推のような概念を使って、よりユーザに負担のないシステムが本来は望ましい。また、出てきたルールを「表現する」方法としても、様々なアイデアがあり得るであろう。

さらには、たとえば商品取り引きなどで、生成された結合ルールをどう使うかということまで考えると、経済予測モデルなどの理論との融合が必要になってくる。この延長線は、すなわち「マイニング（鉱山業）」のみでなく精錬や加工まで行う重工業を視野に入れることに近く、夢のような話であるが、近代の産業革命、工業発展の歴史と同じように、将来のデータマイニングはこのように発展していくのではないであろうか。

謝辞 貴重なコメントをいただいた読者の

方々に深謝いたします。

参考文献

- 1) Agrawal, R., Imielinski, T. and Swami, A.: Mining Association Rules between Sets of Items in Large Databases, In Proceedings of the ACM SIGMOD Conference on Management of Data, pp. 207-216 (May 1993).
- 2) Agrawal, R. and Srikant, R.: Fast Algorithms for Mining Association Rules, In Proceedings of the 20th VLDB Conference, pp. 487-499 (1994).
- 3) Asano, T., Chen, D., Katoh, N. and Tokuyama, T.: Polynomial-time Solutions to Image Segmentations, In Proc. 7th ACM-SIAM Symposium on Discrete Algorithms, pp. 104-113 (1996).
- 4) Fukuda, T., Morimoto, Y., Morishita, S. and Tokuyama, T.: Data Mining Using Two-Dimensional Optimized Association Rules: Scheme, Algorithms, and Visualization, In Proceedings of the ACM SIGMOD Conference on Management of Data, pp. 13-23 (June 1996).
- 5) Fukuda, T., Morimoto, Y., Morishita, S. and Tokuyama, T.: Mining Optimized Association Rules for Numeric Attributes, In Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pp. 182-191 (June 1996).
- 6) Mehta, M., Agrawal, R. and Rissanen, J.: Sliq: A Fast Scalable Classifier for Data Mining, In Proceedings of the Fifth International Conference on Extending Database Technology (1996).
- 7) Piatetsky-Shapiro, G. and Frawley, W. J. editors: Knowledge Discovery in Databases, AAAI Press (1991).
- 8) Quinlan, J. R.: C4.5: Programs for Machine Learning, Morgan Kaufmann (1993).
- 9) Stonebraker, M., Agrawal, R., Dayal, U., Neuhold, E. J. and Reuter, A.: DBMS Research at a Crossroads: The Vienna Update, In Proceedings of the 19th VLDB Conference, pp. 688-692 (1993).

(平成8年2月9日受付)



福田 剛志 (正会員)

1966年生。1991年早稲田大学大学院理工学研究科修士課程修了。同年日本アイ・ビー・エム(株)入社。現在、東京基礎研究所副主任研究員。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。ACM会員。



森本 康彦 (正会員)

1967年生。1991年広島大学大学院博士課程前期修了。工学修士。同年より日本アイ・ビー・エム(株)東京基礎研究所勤務。データマイニング、オブジェクト指向データベースなどの研究に従事。IEEE, ACM各会員。



森下 真一 (正会員)

1960年生。1983年東京大学理学部情報科学科卒業。1985年同大学院修士課程修了。同年日本アイ・ビー・エム(株)入社。東京基礎研究所配属。1990年理学博士(東京大学)。1990-1992年スタンフォード大学計算機科学科客員研究員。データベースの理論とその実用化・製品化に興味を持つ。著書「知識と推論」(共立出版)。



徳山 豪 (正会員)

1957年生。1979年東京大学理学部数学科卒業。1985年同大学院博士課程修了(理学博士)。1986年日本アイ・ビー・エム(株)入社。現在東京基礎研究所主任研究員。1992-1993年IBM T.J. Watson研究所研究員。1992年本会研究賞。計算幾何学を中心に、アルゴリズムと計算理論を研究。著書「はみだし幾何学」(岩波)、「離散構造とアルゴリズム I」(近代科学社、共著)。