情 報 学 基 礎 46－3
（1997．7．24）

数値データからの直交凸領域結合ルール発見

依田 邦和　福田 剛志　森本 康彦
森下 真一　徳山 豪

日本アイ・ビー・エム (株) 東京基礎研究所

データマイニングにおいて2次元結合ルールや決定木作成に役に立つ領域切り出し問題を考える。我々は以前2次元数値属性をもつデータから最適X-単調領域を計算する効率のよいアルゴリズムを示した．ここでX-単調領域とはy軸に平行な線との交わりが必ず連続か空になる領域である。しかし、切り出された領域はy軸方向のノイズに過剰に反応し、トレーニングデータセットにオーバーフィットするため、将来のテストデータに対しての予測精度が低いという問題が残った。そこで本研究では切り出す領域に直交凸領域を使用することを提唱する。直交凸領域とはx軸と平行な線との交わりが必ず連続か空で、かつy軸との交わりも必ず連続か空になる領域である。直交凸領域を用いるとトレーニングデータセットへのオーバーフィットはあまり起こらず、切り出された領域は将来のテストデータへの良い予測を与えることを示す。またこの最適直交凸領域を実際に計算するアルゴリズムを示す。

Finding Rectilinear Regions for Association Rules from Numeric Data

Kunikazu Yoda　Takeshi Fukuda　Yasuhiko Morimoto
Shinichi Morishita　Takeshi Tokuyama

Tokyo Research Laboratory, IBM Japan, Ltd.

We address the problem of finding useful regions for two-dimensional association rules and decision trees. In a previous paper we presented efficient algorithms for computing optimized *x-monotone* regions, whose intersections with any vertical line are always undivided. In practice, however, the quality of x-monotone regions is not ideal, because the boundary of an x-monotone region tends to be notchy, and the region is likely to overfit a training dataset too much to give a good prediction for an unseen test dataset. In this paper we instead propose the use of a *rectilinear region* whose intersection with any vertical line and whose intersection with any horizontal line are both undivided, so that the boundary of any rectilinear region is never notchy. This property is studied from a theoretical viewpoint. Experimental tests confirm that the rectilinear region less overfits a training database and thefore provides a better prediction for unseen test data. We also present a novel efficient algorithm for computing optimized rectilinear regions.

1 はじめに

近年のデータ収集技術の発達と記憶装置の低価格化によって金融や小売業者などが巨大な量のデータを簡単に蓄積できるようになった。そのような団体はこれらの巨大データベースから新しい市場戦略を得るための隠れた情報を発見することに関心を抱いている。データベースや人工知能の分野ではデータから興味ある規則を効率的に発見する技術に感心が集まって来ているが、これは従来のデータベースの機能ではできないことであった。

結合ルール

次のような結合ルールを考える。「もしデータのあるタップルで条件 C_1 が成り立つならば、ある確率 (これを "確信度" と呼ぶ) で条件 C_2 も成り立つ。」推定条件 C_1 と目的条件 C_2 の間のこのルールを $C_1 \Rightarrow C_2$ で表す。理想的には 100%の確信度のルールだけを見つければよいが、ビジネスのデータベースなどでは確信度があるしきい値 (例えば 50%) より大きいルールを見つけることが重要である。このようなルールを "確信がある" と呼ぶ。文献 [1] で Agrawal, Imielinski, そして Swami は確信があるルールを全て見つける方法を研究し効率的なアルゴリズムを示した。彼らが着目したのは (A = yes) という条件がいくつか継ったルールである。ここで A はブール値属性である。彼らはそのアルゴリズムをバスケットデータタイプの小売の取り引きに適用して、アイテム間の相関 (例えば $(Pizza = yes) \wedge (Coke = yes) \Rightarrow (Potato = yes)$ など) を得た。彼らのを改良したアルゴリズムも文献 [2, 11] で報告されている。

1 次元結合ルール

ブール値属性に加え、現実の世界では年齢や銀行預金残高などの数値属性が存在する。そのため数値属性についての結合ルールを発見することも重要である。文献 [8] で我々は $(Balance \in [v_1, v_2]) \Rightarrow (CardLoan = yes)$ というような形式の簡単なルールを発見する問題を考えた。このルールは、「預金額が区間 $[v_1, v_2]$ の範囲に入る顧客はクレジットカードローンを使っている可能性が高い。」という意味である。もし区間の確信度がある与えられたしきい値を超えるならば、その区間は "確信がある" と呼ばれる。区間の "サポート" を数値属性の値がその区間内に入るタップルの数とし、もしサポートがある与えられたしきい値を超えるならばその区間を "十分である" と呼ぶ。

我々が求めたいのは確信があり十分な区間のルールである。そのような区間は無数にありえるので、サポート最大の確信区域 (最適サポート区間と呼ばれる) そして確信度最大の十分な区間 (最適確信度区間と呼ばれる) を発見することを考える。文献 [8] で我々はこのような数値属性とブール値属性の関係を効果的に表す最適区間を計算する効率の良いアルゴリズムを与えた。

2 次元結合ルール

現実の世界では 2 つの属性だけのルールではデータセットの特徴を表すのには十分ではなく、そのため 2 つより多くの属性も扱いたい。文献 [7] で、我々はブール値属性をもつ目的条件へ数値属性の組が及ぼす依存度を与える "2 次元結合ルール" を発見する問題を考えた。各タップル t に対して $t[A], t[B]$ をその 2 つの数値属性の値とする。例えば $t[A]$ = "顧客 t の年齢"、$t[B]$ = "t の預金額" などである。そして t をユークリッド平面 E^2 上の点 $(t[A], t[B])$ へ写す。E^2 の領域 P について、タップル t が P の内部の点へ写されるとき、t は条件 "$(Age, Balance) \in P$" を満たすという。我々は $((A, B) \in P) \Rightarrow C$ のような形式のルール、例えば

$$((Age, Balance) \in P) \Rightarrow (CardLoan = yes)$$

などを見つけたい。

実際には、何百万ものタップルをもつ巨大なデータベースを考慮しなければならなく、何百万もの点を扱わなければならない。このような多くの点を扱うことを避けるため、問題を離散化する。すなわち、各数値属性の値を N 個のほぼ等しい大きさのバケットへ分配する。ユークリッド平面を $N \times N$ 個のピクセル (単位正方形) に分割して、各タップルを点 $(t[A], t[B])$ を含むピクセルへ写す。M を与えられたデータベースのレコード数とする。一つのピクセルへ写される平均レコード数は少なすぎず、多すぎないのが望ましい。実際には $\sqrt[3]{M} \leq N \leq \sqrt{M}$ を仮定し、1 ピクセル内へ写される平均レコード数が 1 から $\sqrt[3]{M}$ の間の数であることを保証させておく。ピクセルの合併集合を 2 次元結合ルールの領域として用いる。すると確信のある領域、十分な領域、最適確信度 (サポート) 領域がそれぞれ 1 次元結合ルールのときと同じようにして自然に定義できる。

良い結合ルールを得るには P の形が重要である。例えば、もし確信度があるしきい値以上のピクセルを全て集めて、P をそれらの合併とすると、P はたいがいの場合大きなサポートをもつ確信のある領域である。しかし、そのような P は多くのバラバラの連結成分の集まりかもしれず、特徴付けが大変困難でよってその妥当性が見えにくいような結合ルールをしばしば作り出してしまう。よって、簡潔に表現できたり視覚化を通じて特徴付けられるようなルールを得るには P は良い幾何学的性質を持った領域のクラスに属しているべきである。

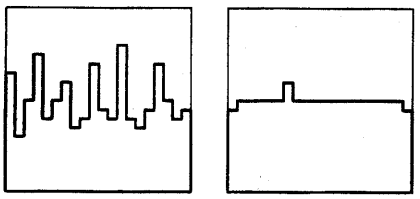

図 1: 最適確信度 X-単調 (左)、直交凸 (右) 領域

X-単調領域

以前の論文 [7] で, 我々は 2 つの幾何学的領域のクラスを考えた。長方形領域と X-単調領域である。Apte 等 [3] も予測のためのルールに区間と長方形を使うことを報告している。領域は y軸に平行な線との交わりが連続か空のとき "X-単調" と呼ばれる。最適 X-単調領域は $P = NP$ でない限り N と $\log M$ の多項式時間で計算することはできないが、我々は $O(N^2 \log M)$ 時間の近似アルゴリズムを示した [7]。

$N \leq \sqrt{M}$ と仮定すればアルゴリズムの計算量は $O(M \log M)$ である。我々は $O(N^3)$ 時間の最適確信度 (サポート) 長方形領域計算アルゴリズムも示した。

我々は X-単調領域を導入して、様々な形の領域が X-単調領域で表現できること、最適確信度 (サポート)X-単調領域の確信度 (サポート) が最適確信度 (サポート) 長方形領域の確信度 (サポート) よりもずっと高いことを示した。長方形は X-単調に含まれるからである。

しかし様々な理由により、X-単調領域は理想的ではなかった。一つの理由は X-単調領域の境界がギザギザになりやすいことである。図 1 (左) は最適確信度 X-単調領域の例である。

もう一つのより深刻な問題は、最適 X-単調領域がトレーニングデータセットを大きくオーバーフィットして、将来のデータに対する良い予測を与えないということである。もっと正確にいえば、たとえ最適確信度 X-単調領域をトレーニングデータセットから作ったとしても、将来のテストデータに対してのその領域の確信度がもとのトレーニングデータでの確信度より低くなるということである。この問題に関して、4節で幾つかの実験結果を与える。

主要結果

我々は 2 つの極端な領域のクラス、長方形領域と X-単調領域の間の適当な幾何学的領域のクラスを探し、直交凸領域を使うことを提案する。連結領域は y軸に平行な線との交わりと x 軸に平行な線との交わりが共に連続か空のとき "直交凸領域" と呼ばれる。この定義から、直交凸領域の境界はけっしてギザギザにはならない。

X-単調領域と直交凸領域の境界の形状について我々は文献 [17] で理論的解析を与え、直交凸領域の方が X-単調領域に比べてピクセル数の変化に対して安定していることを示した。

図 1 (右) は X-単調領域のと同じデータセットに対して切り出した最適確信度直交凸領域である。X-単調の場合と比べて最適直交凸領域はトレーニングデータセットをほんの少ししかオーバーフィットしないので、将来のテストデータセットに対して良い予測を与えることが 4 節の実験で確かめられる。我々は $O(N^3 \log M)$ 時間の近似アルゴリズムを 3節で示す。$N \leq \sqrt{M}$ と仮定すれば、これらのアルゴリズムの計算量 $O(M^{3/2} \log M)$ であり、実用的であるといえる。

関連する研究

数値属性の組の間の相関関係は統計学の主要な研究トピックである。例えば共分散や線推定量などは相互関係を表現する良く知られたツールである。しかし、これらのツールはデータセット全体の集合の相互関係を表しているだけであり、データセットの中の (強い相関のある) 部分集合の関係は表していない。これに関しては幾何学的クラスタリングのテクニックを使うヒューリスティック法がいくつか紹介されている [10, 18]。

他のいくつかの研究では数値属性を扱い、ルールを導こうとしている。Piatetsky-Shapiro [12] は、数値属性の値をソートし、ソートされた値をほぼ均等な大きさの区間に分割し、これらの固定された区間を用いて確信度がほとんど100%のルールを導く方法について研究した。固定された区間以外の区間については彼の研究の枠組では考えられていない。最近 Srikant と Agrawal [15] はいくつかの連続した区間をつないで一つの区間をつくる方法をつけ足すことで、Piatetsky-Shapiro の方法を改善した。しかし継った区間は数値属性全体の区間という当り前のルールを導くかもしれず、これを避けるため彼らは、継った区間の大きさがユーザが与えたしきい値以下になるような効率的な区間計算アルゴリズムを示した。彼らの方法ではハイパーキューブを生成することができる。

決定木をつくるという目的で数値属性を扱うテクニックも開発されている。ID3 [13, 14], CART [5], *CDP* [1], そして SLIQ [9] はギロチンカットと呼ばれる数値属性を 2 分割する方法である。このギロチンカットは Quinlan のエントロピー関数 [13] を最大にする分割である。我々の知る限り、領域をエントロピー関数を最大化するように分割するというアイディアは我々の [6] を除いて、今まで深く研究されていない。

2 2次元結合ルール

ピクセル領域

Definition 2.1 2つの数値属性 A と B を考える。A と B をそれぞれ N_A個 と N_B個 の等しいサイズのバケットに分配する。2次元の $N_A \times N_B$ ピクセル-グリッド G を考える。G は'ピクセル'と呼ばれる単位正方形 $N_A \times N_B$ 個から構成されている。$G(i,j)$ は (i,j)-ピクセルを表し、i と j はそれぞれ行番号、列番号 と呼ばれる。G の第 j-列 $G(*,j)$ は列番号が j である全てのピクセルからなる G の部分集合である。幾何学的には、列は垂直の縞である。■

$n = N_A \times N_B$ という表記をする。典型的な実用例では、N_A や N_Bの範囲は 20 から 500 であり、したがって、n は 400 から 250,000 である。簡単のため、今後 $N_A = N_B = N$ を仮定するが、この仮定は本質的なものではない。

Definition 2.2 ピクセルの集合に対して、それらピクセルの合併は平面領域を形成する。これを "ピクセル領域" と呼ぶ。ピクセル領域はその周囲が長方形をしているとき、"長方形" という。ピクセル領域はそれが連結で各列との共通部分が連続か空のとき "X-単調" という。ピクセル領域はそれが連結で、各列 (垂直な線) との共通部分が連続か空で、かつ各行 (水平な線) との共通部分が連続か空のとき "直交凸" という。■

最適2次元結合ルール

Definition 2.3 各タップル t に対して、$t[A]$ と $t[B]$ は数値属性 A と B の t での値である。もし $t[A]$ が 第 i バケットにあり、かつ $t[B]$ が第 jバケットにそれぞれあるとき、$f(t) = G(i,j)$ と定義する。こうして全てのタップルからグリッド G へのマッピングが出来上がる。

C を目的条件とする。タップル t は C を満たすとき'成功'タップルという。各ピクセル $G(i,j)$ について、$u_{i,j}$ を $G(i,j)$ に写されるタップルの数、$v_{i,j}$ を $G(i,j)$ へ写される成功タップルの数とする。領域 Pが与えられたとき、Pのピクセルへ写されるタップルの数$\sum_{G(i,j)\in P} u_{i,j}$ は P の "サポート", または $support(P)$ と呼ばれる。■

Definition 2.4 Pのピクセルへ写される成功タップルの数$\sum_{G(i,j)\in P} v_{i,j}$ を P の "ヒット" または $hit(P)$ と呼ぶ。Pに写される成功タップル数のタップル数に対する比、すなわち $hit(P)/support(P)$ を P の "確信度" または $conf(P)$ と呼ぶ。■

領域 Pに写されるタップルが確率 $conf(P)$ で条件 Cも満たすという結合ルールを次の記法で表す。

$$(A,B) \in P \Rightarrow C,$$

これを "2次元" 結合ルールと呼ぶ。

Definition 2.5 領域は確信度 (サポート) がある与えられたしきい値以上あれば "確信がある"("十分である") と呼ばれる。我々が見つけたいルールは確信度とサポートがともに高い領域である。"最適確信度" 領域とはサポートがしきい値以上で確信度が最大の領域である。"最適サポート" 領域とは確信度がしきい値以上でサポートが最大の領域である。■

Theorem 2.1 最適サポート (確信度) 直交凸領域を計算する $O(N^3M)$ のアルゴリズムが存在するが、$P = NP$ でない限り、N と $\log M$について多項式時間で計算できるアルゴリズムは存在しない。

Proof: 文献 [16] 参照。■

次の節では、$O(N^3 \log M)$ 時間で最適サポート (確信度) 直交凸領域を計算する近似アルゴリズムを示す。

3 最適直交凸領域を求める近似アルゴリズム

ハンドプロービング技術

この節では "ハンドプロービング" の技術について簡単に説明する。この技術は浅野ら [4] が画像分割のために発明し、福田ら [7] が最適 X-単調領域を切り出すために修正したものである。ここでは最適直交凸領域を計算するアルゴリズムも記述する。

Definition 3.1 各直交凸領域 P を ユークリッド平面上の "スタンプ" 点 $(support(P), hit(P))$ へ写像する。■

直交凸領域は 2^N個より多くあるので、全てのスタンプ点を実際に作ることはできないが、全てのスタンプ点が平面上にあるのを想像し、これらスタンプ点の上部凸包を考える。図 2にあるように最適確信度 (サポート) 直交凸領域に対応するスタンプ点 P_1は必ずしも凸包の上にのっていない。しかし実際は、非常に多くの数の点が凸包の上に密に分布しているので、凸包上の点 P_2 を P_1 の近くに見つけることができ、この点 P_2 を P_1 の近似解とみなすことができる。この "凸包仮定" に基づいて最適直交凸領域を計算する近似アルゴリズムが作られる。

次の問題は上部凸包上の点をどうやって計算するかである。ここにおいて我々は浅野ら [4] が与えた "ハンドプロービング" のテクニックを導入する。凸包上の各スタンプ点

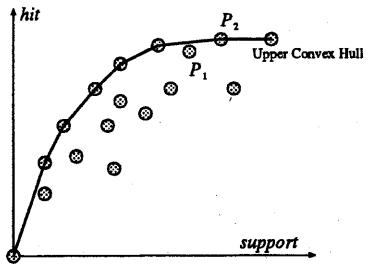

図 2: 凸包仮定

には接線が存在し、その傾きをτとする。この点は全てのスタンプ点の中で$y-\tau x$を最大化する点である。したがって、対応する直交凸領域Pも$hit(P)-\tau \times support(P)$すなわち$\sum_{G(i,j)\in P} v_{i,j}-\tau u_{i,j}$を最大化する。

Definition 3.2 $v_{i,j}-\tau u_{i,j}$をピクセル$G(i,j)$の利得値と呼び、$hit(P)-\tau \times support(P)$を$P$の利得値とそれぞれ呼ぶ。全ての直交凸領域の中で最適"利得"直交凸領域PはPの利得値を最大化する。∎

ハンドプロービング技術を使うと、凸包上のスタンプ点を効率良く調べる必要がある。我々は$O(\log M)$個の接線を使うことでそれができることを示した[7]。

次の節では傾きτが与えられたとして、最適利得直交凸領域を$O(N^3)$で計算するアルゴリズムを示す。

最適直交凸領域

いまPを直交凸領域とする。m_1, m_2をそれぞれPの左端、右端の列番号とし、$s(i)$と$t(i)$をそれぞれPの第i列の下端、上端の行番号とする。Pは直交凸領域だから、上端のピクセルの左から右への系列$G(i,t(k))$(for$i=m_1,\ldots,m_2$)は単調に増加するか($t(i)\le t(j)$ for $m_1\le i<j\le m_2$)、または単調に減少するか($t(i)\ge t(j)$ for $m_1\le i<j\le m_2$)、またはある列まで単調に増加してその後単調に減少する。同様に下端のピクセルの系列$(G(i,s(k))$(for$i=m_1,\ldots,m_2$)は単調に増加するか、単調に減少するか、ある列まで単調に減少してその後単調に増加する。

図3の一番上の絵は上端のピクセルの系列を表している。

Definition 3.3 直交凸領域の4タイプについて定義する。左から右へいくにつれ広がる(狭まる)領域をそれぞれ$W(N)$で表す。左から右へいくにつれ上昇(下降)する領域をそれぞれ$U(D)$で表す。

図3の真中左の絵は上系列が途中まで単調増加してその後単調減少し、下系列が途中まで単調増加してその後単調減少する場合を表している。これには2種類の直交凸領域がある。1つはW-typeとD-typeの組合せであり、WD-typeと記される。もう一つはU-typeとN-typeの組合せであり、UN-typeと記される。図3真中右の絵は上系列が単調増加するか単調減少するかし、下系列が途中まで単調減少してその後単調増加する場合を表している。これには2種類の領域がある。1つはWU-typeでもう1つはDN-typeである。図3の一番下の絵は上系列が途中まで単調増加してその後単調減少し、下系列が途中まで単調減少してその後単調増加する場合を表している。これには3種類の領域がある。WDN, WN, そしてWUN-typeである。

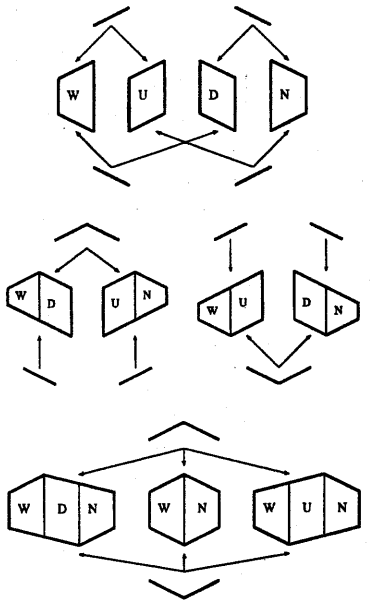

図 3: 直交凸領域の計算

Theorem 3.1 最適利得直交凸領域は$O(N^3)$で求めることができる。

Proof: いま、$f_W(m,[s,t])$を右端がm列の第s行から第t行までであるようなW-typeの直交凸領域の最大利得値とする。$g_{i,j}$をピクセル$G(i,j)$の利得値、すなわち$v_{i,j}-\tau u_{i,j}$とする。$f_W(m,[s,t])$の計算の前に、まず$g_{m,[s,t]} := \sum_{j\in[s,t]} g_{m,j}$を$m=1,\ldots,N$ $1\le s\le t\le N$に対して予め計算しておく。この計算は$O(N^3)$でできる。$m=1$のときは、$f_W(1,[s,t])=g_{1,[s,t]}$である。$m>1$に対しては、もし$s=t$ならば$f_W(m,[s,s])=\max\{g_{m,s}, f_W(m-1,[s,s])+g_{m,s}\}$. それ以外($s<t$)ば

次の再帰によって $f_W(m,[s,t])$ を決定する。

$$f_W(m,[s,t]) = \max \begin{pmatrix} f_W(m-1,[s,t]) + g_{m,[s,t]} \\ f_W(m,[s+1,t]) + g_{m,s} \\ f_W(m,[s,t-1]) + g_{m,t} \end{pmatrix}$$

次に、$f_U(m,[s,t])$ を右端が m 列の第 s 行から第 t 行までであるような U-type の直交凸領域の最大利得値とする。$m = 1$ のときは、$f_U(1,[s,t]) = g_{1,[s,t]}$ である。$m > 1$ に対しては、$\max_{i \le s} f_W(m-1,[i,t])$ と $\max_{i \le s} f_U(m-1,[i,t])$ を全ての $s \le t$ について予め $O(N^2)$ で計算する。そして次の再帰処理によって $f_U(m,[s,t])$ を決定する。

$$f_U(m,[s,t]) = \max \begin{pmatrix} \max_{i \le s} f_W(m-1,[i,t]) + g_{m,[s,t]} \\ \max_{i \le s} f_U(m-1,[i,t]) + g_{m,[s,t]} \\ f_U(m,[s,t-1]) + g_{m,t} \\ (\text{or } g_{m,t} \text{ when } s = t) \end{pmatrix}$$

次に、$f_D(m,[s,t])$ を右端が m 列の第 s 行から第 t 行までであるような D-または WD-type の直交凸領域の最大利得値とする。$m = 1$ のときは、$f_D(1,[s,t]) = g_{1,[s,t]}$ である。$m > 1$ に対しては、$\max_{i \le s} f_W(m-1,[i,t])$ と $\max_{i \le s} f_U(m-1,[i,t])$ を全ての $s \le t$ について予め $O(N^2)$ で計算する。そして次の再帰処理によって $f_D(m,[s,t])$ を決定する。

$$f_D(m,[s,t]) = \max \begin{pmatrix} \max_{t \le i} f_W(m-1,[s,i]) + g_{m,[s,t]} \\ \max_{t \le i} f_D(m-1,[s,i]) + g_{m,[s,t]} \\ f_D(m,[s+1,t]) + g_{m,s} \\ (\text{or } g_{m,s} \text{ when } s = t) \end{pmatrix}$$

最後に、$f_N(m,[s,t])$ を右端が m 列の第 s 行から第 t 行までであるような N-,UN-,DN-,WDN-,WN-, または WUN-type の直交凸領域の最大利得値とする。$m = 1$ のときは、$f_N(1,[s,t]) = g_{1,[s,t]}$ である。$m > 1$ に対しては、次の再帰処理によって $f_N(m,[s,t])$ を決定する。

$$f_N(m,[s,t]) = \max \begin{pmatrix} f_W(m-1,[s,t]) + g_{m,[s,t]} \\ f_U(m-1,[s,t]) + g_{m,[s,t]} \\ f_D(m-1,[s,t]) + g_{m,[s,t]} \\ f_N(m-1,[s,t]) + g_{m,[s,t]} \\ f_N(m,[s-1,t]) - g_{m,s-1} \\ f_N(m,[s,t+1]) - g_{m,t+1} \end{pmatrix}$$

この場合、$f_N(m,[s,t])$ を計算するのに $f_N(m,[s-1,t])$ と $f_N(m,[s,t+1])$ を使う。これらはより長い区間 $[s-1,t]$ or $[s,t+1]$ について計算されていなければならない。すなわち、まず最初に $f_N(m,[1,N]) = \max\{f_N(m-1,[1,N]) + g_{m,[1,N]}, g_{m,[1,N]}\}$. を計算する。

このようにして、簡単なダイナミックプログラミングによって計算量 $O(N^3)$ で $f_W(m,[s,t])$, $f_U(m,[s,t])$, $f_D(m,[s,t])$, and $f_N(m,[s,t])$ を全ての m と $s \le t$ について求めることができる。∎

4 実験結果

オーバーフィット

この節で我々は、最適 X-単調領域がトレーニングデータに大幅にオーバーフィットしやすく、そのため将来のデータセットに対する良い予測ができにくいということ、一方最適直交凸領域はこのオーバフィットの現象がそれほど大きくは生じないということを実験的に示す。

この実験には現実の典型的な場合を代表する人工データを作った。A と B を共に $[-1,1]$ の範囲をとる数値属性とし、C を目的のブール属性とする。我々は各タップルが以下の処理によって生成される人工データセットを作った。

1. ランダムな点を正方形領域 $[-1,1] \times [-1,1]$ の中に一様に生成していく。
2. 各点 $(t[A],t[B])$ について, $t[C]$ の値を以下のようにして決め、タップル t をデータセットに加える。

   $p(x,y)$ を $[-1,1] \times [-1,1]$ を $[0,1]$ へ写す関数とする。$t[C]$ は確率 $p(t[A],t[B])$ で 1 に、そうでなければ 0 にする。

我々は次の 2 つの関数を $p(x,y)$ に用いた。

- $p_{linear}(x,y) = \frac{1}{\sqrt{2\pi}} \exp(\frac{-(x-y)^2}{4})$, これは点 (x,y) の対角線 $y = x$ からの距離についての正規分布である。
- $p_{circular}(x,y) = \frac{1}{\sqrt{\pi}} \exp(-(x^2+y^2))$, これは点 (x,y) の原点からの距離についての正規分布である。

我々は共に 10,000 レコードの 2 つのデータセットを生成した。一方のデータセット D_{linear} は $p_{linear}(x,y)$ を使い、もう一方 $D_{circular}$ は $p_{circular}(x,y)$ を使って生成した。

X-単調領域と直交凸領域をオーバーフィットの観点から比較するため、N-フォールドクロスバリデーションについて説明する。

1. 与えられたデータセット D_{linear} または $D_{circular}$, をランダムに N個の等しいサイズの部分集合に分割する。
2. $N-1$ 個の部分集合の合併をつくり、領域を切り出すトレーニングデータセットとする。すなわち、このトレーニングデータから最低サポートしきい値 50%で最適確信度の X-単調領域、直交凸領域を切り出す。
3. そして残りの 1 つの部分集合をテストデータセットに使う。領域の中に入っているテストデータの数とそのうちの目的条件を満たすデータ数の比を計算し、これをテストデータに対するその領域の確信度と呼ぶ。

ピクセル数	トレーニング	テスト	差
8 × 8	47.77%	46.92%	-0.85%
16 × 16	48.22%	47.66%	-0.56%
32 × 32	48.30%	47.52%	-0.78%
64 × 64	48.42%	47.03%	-1.39%

(a) 最適長方形領域の確信度

ピクセル数	トレーニング	テスト	差
8 × 8	52.70%	51.38%	-1.31%
16 × 16	53.72%	51.76%	-1.95%
32 × 32	55.24%	51.69%	-3.55%
64 × 64	57.47%	51.00%	-6.46%

(b) 最適 X-単調領域の確信度

ピクセル数	トレーニング	テスト	差
8 × 8	52.70%	51.56%	-1.14%
16 × 16	53.49%	52.24%	-1.25%
32 × 32	53.96%	51.79%	-2.17%
64 × 64	54.43%	51.75%	-2.67%

(c) 最適直交凸領域の確信度

表 1: D_{linear}の結果

ピクセル数	トレーニング	テスト	差
8 × 8	39.02%	39.19%	0.17%
16 × 16	39.83%	39.83%	0.00%
32 × 32	40.15%	39.81%	-0.34%
64 × 64	40.32%	39.77%	-0.55%

(a) 最適長方形領域の確信度

ピクセル数	トレーニング	テスト	差
8 × 8	40.41%	39.81%	-0.60%
16 × 16	41.55%	40.24%	-1.30%
32 × 32	42.94%	39.72%	-3.22%
64 × 64	44.94%	38.21%	-6.73%

(b) 最適 X-単調領域の確信度

ピクセル数	トレーニング	テスト	差
8 × 8	40.41%	39.77%	-0.64%
16 × 16	41.31%	40.38%	-0.93%
32 × 32	41.97%	40.28%	-1.68%
64 × 64	42.67%	40.17%	-2.50%

(c) 最適直交凸領域の確信度

表 2: $D_{circular}$の結果

4. 上の3ステップを N回繰り返し、確信度の平均を計算する。

我々は 10-フォールドクロスバリデーションを D_{linear} と $D_{circular}$. に対して行った。表 1 は D_{linear} の結果であり、表 1(a) {(b), (c)} はそれぞれ最適確信度長方形 {X-単調、直交凸 } 領域のピクセル数 8 × 8 から 64 × 64 までの間の確信度である。

最初の列はピクセル数であり、第 2 列はトレーニングデータセットから作られた各最適領域の確信度であり、第 3 列はテストデータに対する各最適領域の確信度であり、第 4 列は第 3 列の値から第 2 列の値を引いた値である。

まず第一に、直交凸領域が 3 つの領域のクラスの中ではテストデータで最も高い確信度を与えていることがわかる。これは他の領域のクラスの代りに直交凸領域を使うことの主な利点である。

第 2 に、第 4 列は最適領域がどれほどトレーニングデータにオーバーフィットしてテストデータの予測に失敗しているかを表す。ピクセル数の増加につれて第 4 列の値は大きくなっていく。これらの増加は X-単調領域が他の領域族にくらべてずっと急激である。これは最適 X-単調領域が大幅にトレーニングデータにオーバーフィットしていることを示唆している。

表 1(b) の第 2 列をみると最適 X-単調領域のトレーニングデータに対する確信度はピクセル数に依存し、表 1(a) {(c)} の第 2 列からは最適長方形 { 直交凸 } 領域のトレーニングデータに対する確信度はピクセル数が増加してもほぼ安定していることがわかる。

領域計算のパフォーマンス

最悪の場合のパフォーマンスを計るため、次のようにして人工データを作った。まず区間 $[N^2, 2N^2]$ の中で一様に乱数を発生させ、$u_{i,j}$ に割り当てる。次に $1, 2, \ldots, N^2$ を右下のセルから中央のセルへ時計回りにらせん階段のように $v_{i,j}$ へ割り当てる。

実験は我々のプロトタイプシステム Database SONAR (System for Optimized Numeric Association Rules) を用いて行われた。プログラムは C++ で書かれ IBM SP2 の 1 ノード上で実行された。1 ノードは CPU が 66 MHz の Power2 で 2 MB の 2 次キャッシュと 256 MB のメインメモリーを持ち、OS は AIX4.1 である。

表 3 は最適確信度の長方形、直交凸、そして X-単調の領域を計算するのにかかった時間を示している。最小サポートは 50%でピクセル数は 8 × 8 から 128 × 128 まで変化させた。これらの実験は、最適長方形領域、X-単調領域、そして直交凸領域の計算時間がそれぞれ $O(n^{1.5})$, $O(n \ln M)$, そして $O(n^{1.5} \ln M)$ であることを確証している。ここで n はピクセル数、Mはタップル数である。

ピクセル数	時間 (秒)		
	長方形	X-単調	直交凸
8 × 8	0.001	0.014	0.012
16 × 16	0.008	0.089	0.087
32 × 32	0.051	0.378	0.746
64 × 64	0.364	1.830	6.502
128 × 128	2.747	7.983	63.018

表 3: 最適領域計算のための実行時間

参考文献

[1] R. Agrawal, T. Imielinski, and A. Swami. 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] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th VLDB Conference*, pp. 487–499, 1994.

[3] C. Apte, S. J. Hong, S. Prasad, and B. Rosen. Ramp: Rules abstraction for modeling and prediction. Technical Report RC-20271, IBM Watson Research Center, 1995.

[4] T. Asano, D. Chen, N. Katoh, and T. Tokuyama. Polynomial-time solutions to image segmentations. In *Proc. 7th ACM-SIAM Symposium on Discrete Algorithms*, pp. 104–113, 1996.

[5] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth, 1984.

[6] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Constructing efficient decision trees by using optimized association rules. In *Proceedings of the 22nd VLDB Conference*, pp. 146–155, 1996.

[7] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. 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.

[8] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. 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.

[9] M. Mehta, R. Agrawal, and J. Rissanen. Sliq: A fast scalable classifier for data mining. In *Proceedings of the Fifth International Conference on Extending Database Technology*, 1996.

[10] R. T. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. In *Proceedings of the 20th VLDB Conference*, pp. 144–155, 1994.

[11] J. S. Park, M.-S. Chen, and P. S. Yu. An effective hash-based algorithm for mining association rules. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pp. 175–186, May 1995.

[12] G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, pp. 229–248, 1991.

[13] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[14] J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[15] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, June 1996.

[16] K. Yoda, T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Computing optimized rectilinear regions for association rules. In *Proceedings of KDD'97*, Aug. 1997.

[17] K. Yoda, T. Fukuda, Y. Morimoto, S. Morisita, and T. Tokuyama. Computing optimized rectilinear regions for association rules. Technical Report RT0201, IBM Tokyo Research Laboratory, 1997.

[18] T. Zhang, R. Ramakrishnan, and M. Linvy. Birch: An efficient data clustering method for very large databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pp. 103–114, June 1996.