高次元ピラミッドを用いた数値属性ルールの生成とデータマイニングへの応用

全　眞嬉† 　DannyZ. Chen†† 　加藤直樹††† 　徳山　豪†

† 東北大学大学院情報科学研究科システム情報科学専攻　〒 980-8579 仙台市青葉区荒巻字青葉 09
† Dept. of Computer Science and Engineering, University of Notre Dame　Notre Dame, IN 46556, USA.
††† 京都大学大学院工学研究科建築工学専攻　〒 606-8501 京都府左京区吉田本町
E-mail: †{jinhee,tokuyama}@dais.is.tohoku.ac.jp, ††chen@cse.nd.edu, †††naoki@archi.kyoto-u.ac.jp

あらまし　福田らによって提案された最適領域ルールは数値属性データで構成されるデータベースに対する有効なデータマイニングツールである [11], [12]. しかしながら，従前の方法では 2 つの欠点がある：(1) 各ルールは高々2 変数の数値属性に対応でき (2) 判定は与えられたデータの正確な位置ではなく，単純に領域 R の中にあるか，外にあるかだけに基づき行われる。本論文では，これらの欠点を取り除くための新しい方法の提案をする．具体的にはグラフアルゴリズムの用いて，2 つ以上の属性を持つ最適数値属性結合ルールと階層構造の数値属性結合ルールを与える．又，本論文の手法は異常なデータの除去とデータクラスタリングに適切である．

キーワード　データマイニング, データの可視化, 最適化, 数値属性結合ルール, アルゴリズム

Higher-Dimensional Pyramid Construction Problem and Application to Data Mining

Jinhee CHUN†, Danny Z. CHEN††, Naoki KATOH†††, and Takeshi TOKUYAMA†

† Graduate School of Information Sciences,　TOHOKU University,　Aramaki-aza-aoba 09, Aoba-ku, Sendai, 980-8579 Japan.
† Dept. of Computer Science and Engineering, University of Notre Dame　Notre Dame, IN 46556, USA.
††† Graduate School of Engineering, Kyoto University,　Yoshida-Honmachi, Sakyo-ku, Kyoto,　606-8501 Japan.
E-mail: †{jinhee,tokuyama}@dais.is.tohoku.ac.jp, ††chen@cse.nd.edu, †††naoki@archi.kyoto-u.ac.jp

Abstract　Optimized region rules developed by Fukuda et al. [11], [12] are effective tools for data mining in databases with numeric data. However, there are two drawbacks in the previous methods: (1) each rule can contain at most two numeric conditional attributes, and (2) the decision is made based only on whether a given data is inside or outside a region R, but not on the exact position of the data. In this paper, we propose a new method for removing these drawbacks. Indeed, by applying graph algorithms, we give optimized numeric association rules with more than two attributes, and give layered-structure numeric association rules. Our method is also applicable to removal of exceptional data and data clustering.

Key words　Data mining,Visualization,Optimization, Association rule for Numeric Attribute,Algorithm

1. はじめに

データマイングの手法として「A ならば B である」といったデータベースの属性間の相関関係を求める結合ルールがあり，Apriori アルゴリズム [1] が提案された．現行のデータベースには性別，血液型といったカテゴリ属性と年齢，体重といった数値でとられる数値属性が含まれているが，Apriori アルゴリズムはカテゴリ属性間の結合ルールを求めるためのものであり，数値属性に対しては直接適用することはできない．なぜなら，数値属性は値に順序があるため，値を連続した区間で表すことが重要であるが，Apriori アルゴリズムではそのようなことは考慮していないからである．よって，数値属性を含む結合ルールを求めるには別のアルゴリズムを用いる必要があ る．

福田ら [12] は領域切り分けを用いて数値属性とカテゴリ属性

間の結合ルールの生成を行う手法を提案している．この手法では「$x \in R$ ならば B である」(R はいくつかの数値属性の値の空間の部分領域) のようなルールを発見している．

福田らによる領域として切り出しを行うと，領域に入っているかいないかだけで判定を行うため，境界線上のデータと中央部のデータが同じ扱いをされるといった位置情報の損失がある．領域切り分けを用いて生成された結合ルールを学習に用いると，制限が強く，学習データに入っているノイズ（離れ値）を排除してしまうため，学習データ以外のデータに対して生成されるルールは不自然な領域ルールを生成するといった，過学習の問題点がある．また，属性を 3 つ以上を与えると，計算量が多く計算が困難になる．すなわち，属性次元の制限の問題がある．ここで，本来は切り取るべきものは多次元正規分布のような特徴をもったデータ分布であり，領域として切り取るより，性質の良い関数として捉える事が望ましい．

本論文では，以上の問題を解決するために，最適ピラミッドによる結合ルールの表現法を提案する．つまり，領域として切り出すのではなく，性質の良い関数として切り出しを行う．数値属性または複数の数値属性に関しての最適な階層構造でデータの近似を行い，階層的な数値結合ルール，すなわち最適ピラミッドを用いた数値属性結合ルールの提案をする．

手段として，幾何学的なアルゴリズムを用いた最適ピラミッド構築問題への定式化を行う．最適ピラミッドとは，入力データが持っている位置情報の損失を最小にする（最小の近似エラーを持つ），すなわち，パラメトリックゲインを最大にする領域を積み重ねた単峰な図形への変更である．

2. 研 究 背 景

2.1 結合ルール

結合ルールは，タプル内の属性間の相関関係である．X,Y が属性に関する命題式のとき，$(X \rightarrow Y)$ を結合ルールと呼ぶ.「X ならば Y である」という意味である．Y は通常，単一属性に関する命題式である．

健康診断データベースを例として考える．正常体重域を超えた体重を肥満，正常血糖値を超えた血糖値を異常血糖値と呼ぶことにする.「肥満で異常血糖値である人は糖尿病の検査が必要である」ような診断法則があるとする．健康診断データベースの診断法則の結合ルールは (肥満 $= yes \wedge$ 異常血糖値 $= yes) \rightarrow ($ 糖尿病検査 $= yes)$ と表現される．

X が単一属性に関する命題式のとき，$(X \rightarrow Y)$ は 1 次元結合ルールと呼び，$(X_1 \wedge X_2 \rightarrow Y)$ のように前提条件に X_1, X_2 の 2 つの属性を用いた場合は 2 次元結合ルールと呼ぶ．

結合ルールの尺度である，確信度とサポートについて述べる．結合ルール $X \rightarrow Y$ の確信度 ($confidence$) とは，データベース $\mathcal{D}$ の X を含むトランザクションのうち，Y を同時に含む割合のことである．すなわち，$|X \wedge Y|/|X|$ が c%であることを，conf($X \rightarrow Y$)=c%と表記する.「結合ルール $X \rightarrow Y$ は c%の確信度である」という意味を持つ．結合ルール $X \rightarrow Y$ のサポート ($support$) とは，データベース $\mathcal{D}$ の (X) を含むトランザクションの，トランザクション全体に対する割合のことである．このとき，support($X \rightarrow Y$)=s%と表記する.「結合ルール $X \rightarrow Y$ は s%のサポートである」という意味を持つ．

各項目間の関連に説得力のある根拠を提示することで，抽出された規則はユーザに判断基準を与える説明性を持つ．このような規則の根拠を確信度とサポートを用いて表す．

例えば，肥満で血糖値も高い人はサポート s%，確信度 c%で糖尿病であることが判ったとする．サポートと確信度が分かるので，抽出されたルールは意味の有るルールなのか，信頼性が低いルールであるのかが客観的に判断できる．

サポートと確信度によって，知識判断の客観的な基準の提供が可能になり，抽出されたルールが知識として有効なのか，有効でないかの判断基準の尺度となる．

2.2 数値属性結合ルール

健康診断データベースを例として考える．血液型，病歴，性別といった 2 つ以上の値をとるカテゴリ属性と，年齢，身長，体重，血圧，血糖値といった数値を属性ととる数値属性がある．しかし，数値属性の場合，値の順序に意味があり，定義域が大きい特徴があることから，結合ルールを間単に求めるのは困難である．

血圧の場合，0mmHg から数百 mmHg までの 1mmHg 刻みで測定された値で構成される．このような数値データをカテゴリ属性のような形式では扱えにくい．血圧の場合は多くても数百だが，銀行の預金残高のようなデータは数千億円を 1 円刻みで表現しなければならないので，数値属性で構成されるデータは「$x_1 \quad \leqq$ 最大血圧 $\quad < x_2$」のように数値属性を区間の条件として取り入れ，区間を表す条件をアイテムとして扱うことにより，定義域が大きかった数値属性から結合ルールを効率的に求めることに役に立つ．

次に，2 次元数値属性の結合ルールについて考察する．次のような結合ルール

(年齢 $\in [55, 60]) \wedge$ (最大血圧 $\in [140, 160]) \Rightarrow$ (精密検査 $=$ 必要) は，(年齢 $\in [55, 60]$) と (最大血圧 $\in [140, 160]$) の二つの数値属性の条件を用いるこのような 2 つの数値区間の直積は，図 1 右上のように 2 つの数値属性が作る面上の軸に平行な矩形領域である．矩形領域は制限が強く，データの特徴をうまく表現できない．また，x 単調領域 (注1) は自由度が高い．d 但し，学習データに含まれているノイズをを排除するため，学習データ以外のデータに対して生成された最適な領域はキザキザした不自然な形になることもある．その両方の欠点を補うため，領域族として直交凸領域 (注2) （図 1 左下）を用いて 2 次元数値ルールを用いる事が提案されている（[8], [12]).

ここで問題になるのは，データから切り出すルールとして区間や領域でよいか？ という事である． 数値結合ルールを決定木の部品として使う場合，その区間や領域の内部に入れば yes そうでなければ no という判断を行うのが一般的である [8]. しかし，もともとの入力はデータ分布であり，領域の中央部分

(注1)：x-monotone region;x 軸に垂直な直線と交わりが一つの区間か空で有るような連結領域

(注2)：rectilinear convex region;x 単調かつ y 単調な連結領域

にあるか境界部分にあるかにより，yes-no の判定に揺らぎが生じているはずである．実際，領域の境界を決める場合に，エントロピーや GINI といった目的関数を用いるが，これは非常に恣意的な決定の手法である．本論文では，切り出すルール自体も単なる領域でなく，分布関数であり，かつその分布関数がルールとして有効な形をしているものを考える．具体的には，性質の良い単峰分布関数の族を考え，ルールをその族に入る分布関数として捉える．一つの有力なアプローチは，2 次元正規分布による近似であるが，本論文ではより精密なルールを目指し，自由度の高い単峰分布関数族を考える．

2.3 問題点と解決法

効果的な結合ルールの効率的生成はデータマイニングの主要なトピックである．本論文では数値データの結合ルールについて考察する．ブール，又はカテゴリ属性に対して結合ルールを生成させる為の良いアルゴリズムが Agrawal-Imielinski-Swami [1] で提案された．

数値属性を扱うために，個々の数値の属性を個別にし，ブール属性又は数値情報のないカテゴリ属性に変換を行う方法がある．ところが，このようなデータの変換は，多くの場合オリジナルデータの情報が失われる．他の幾何学的なアプローチは，各ルールがガウス分布であると仮定し，ガウス分布による，又は少数のガウス分布の結合によるデータ分布の最良近似をみつけることである．しかし，上記の方法は条件が強く，データの特徴をうまく表現できない．したがって，弱い条件を使い幾何学ルールを抽出するためにシステム構築を行いたい．

他のアプローチは福田らによって提案された [11], [12], [17]. d-タプルデータ $\mathbf{x}$($\mathbf{x}$ は d 数値属性から成る) を，d-次元のスペースの点と見なし，Fig. 1 左上図で見られるような確信度/サポート ($confidence/support$) 情報を示すボクセル ($voxel$) グリッド分布の構築を行う．その後，d-次元のスペースの適切な領域 R のためのルール $(\mathbf{x} \in R) \rightarrow (C = yes)$ を考える．そのようなルールを領域ルール ($regionrule$) と呼ぶ．ここで，ターゲット属性 C は数値 (通常の単調条件) の場合もある．領域ルールの右辺は $X(C) = f$ であり，$X(C)$ は確率変数が C に関係していることを示し，f は確率分布である [16]．領域 R は，与えられた領域族 $\mathcal{F}$ からルールのサポートおよび確信度で求める目的関数が最適になるようなものが選ばれる．ルールの正確さは領域族 $\mathcal{F}$ に依存する．また，直観的に R は「良い」形を持っていて，$\mathcal{F}$ は多くの領域を含んでいる豊富な族であるはずである．例えば，そのような領域の多項式数（ピクセルグリッドのサイズ）だけを持ち，ガウス分布のような典型的なデータ分布を表わすことはできないが，軸に平行な矩形領域が作る直交領域 (Fig. 1 の右上) は良い形をしている．

x-単調領域および直線の凸面の多角形の領域のようなより柔軟領域族は，福田ら [12], [20], によって提案された．また，領域ルールは正確な決定木の構築に適用される．それらの領域族は最も典型的なデータ分布のアウトラインを表わす様々な形を含んでいる．ところが，2 次元以上で領域ルールを得る方法を拡張することは計算上困難であり，主な欠点である．さらに，そのような領域ルールでは，領域 R の内のデータの実際の位置に関しての重要な情報は無視される; 言いかえれば，R の中央部近くの点は，R の境界近くの点として等しく扱われる．ルールが通常の確率的な特性に従うデータ分布であるので，これは多くの場合都合が良くない．したがって，ルールの影響は，高度にデータ点の位置に依存するべきである．データの位置情報の傾向を抽出する際に Fig. 1 の右下の図のような滑らかな階層構造の提供をしたい．

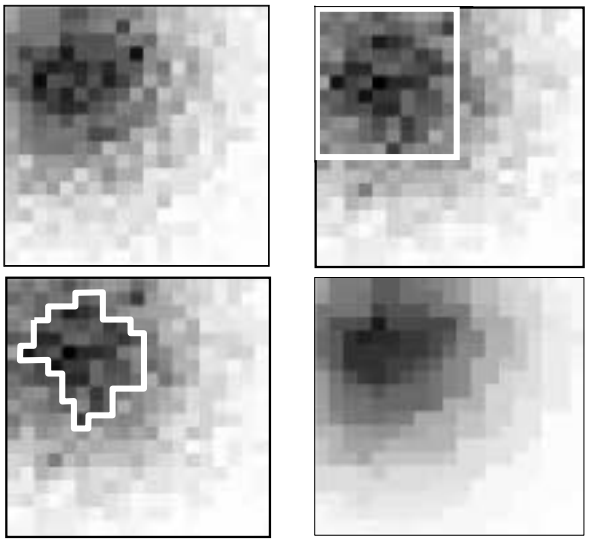


図 1　階層構造の領域ルール

本論文では，新しい領域族を階層最適化し，より高次元のルールの効率的な生成を行い，これらの欠点を補う．本研究の結果は，結合ルール生成だけでなくデータビジュアル化 [2] およびデータマイニングへのファジーアプローチにおいても有効に応用できる．階層構造は決定論的な決定ルール $(\mathbf{x} \in R) \rightarrow (C = yes)$ に比較して，強いルールの影響を縮小する方法を適用する．拘束力の弱いルールで判定をする，非決定性を持たせた柔軟な決定システムの構築を行う．

3. 最適ピラミッド問題

階層的結合ルール生成問題を定式化するためにより一般の幾何学問題を与え，データマイニングでの利用をその応用として行う．

μ と ρ を $n = m^d$ セルの d 次元ボクセルグリッド Γ 上の非負整数値関数とする．すべてのセル $c \in \Gamma$ に対し $\mu(c) \leqq n$ かつ $\mu(c) \geqq \rho(c)$ と仮定する．データマイニングへの応用では μ はセル c に対応するデータの数，すなわちサポート（$support$）の関数を表し，ρ はセル c において条件属性と目的属性を同時に満たすデータの数，すなわちヒット（hit）関数を表す．また，確信度（$confidence$）はサポートをヒットで割ったもので，$conf$(c)=ρ(c)/μ(c) とする．

N に領域族 $\mathcal{F}$ を固定する．一般性を失わず，$\emptyset \in \mathcal{F}$ および $\Gamma \in \mathcal{F}$ と仮定できる．

［Definition 1］ $P(t_0) = \Gamma$ および $t > t'$ のとき $P(t) \subseteqq P(t')$ を満たす $\mathcal{F}$ の領域の列 $\mathcal{P}$=$P(t_i)(i = 0, 1, 2, \ldots, h)$ を考える．こ

こで $t_0 < t_1 < t_2 < \cdots < t_h$ は高さと呼ばれる実数である．$P(t_{h+1}) = \emptyset$ で $conf(P(t_i) - P(t_{i+1})) = t_i$ の場合，$\mathcal{P}$ を μ に関して $conf = \rho/\mu$ を近似する．ρ に近似するピラミッド（あるいはピラミッド構造）と言う．

$\mathcal{P}$ の近似エラーは $\sum_{i=0}^{h} \sum_{c \in P(t_i) - P(t_{i+1})} (conf(c) - t_i)^2 \mu(c)$ によって定義される．これは確率密度関数として μ を考えたとき，確信度 ρ/μ と，$[f_P(x) = t_i \text{ if } x \in P(t_i) - P(t_{i+1})]$ によって定義されたピラミッドの表面関数 f_P の間の 2 乗された L_2 距離である．すべてのピラミッド中で最小近似エラーを持つとき，ピラミッド $\mathcal{P}$ を最適であると言う．

μ がユークリッド空間の体質を与える密度関数であるとき，最適ピラミッドは，位置ポテンシャルの損失を最小限にする ρ/μ の単峰な変更と見なすことができる．これは計算幾何学および地理学 (特に $d = 2$ の場合) の基礎的な問題となる．図 2 は，$d = 1$ の場合のピラミッド ($\mu \equiv 1$ の場合) へ関数 ρ の変更の例である．

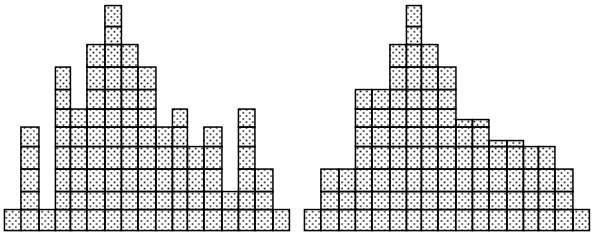


図 2　1 次元最適ピラミッド構築の入力と出力

最適ピラミッドの構築は領域切り出し問題の自然な拡張であり，データマイニングに加えて多くのアプリケーションにおいて有用になる．

3.1　最適ピラミッドとパラメトリックゲイン

［Lemma 1］ 領域を値に持つ写像 $P(t) : (0, \infty) \to \mathcal{F}$ で，単調条件［$t > t'$ のとき $P(t) \subseteq P(t')$］を満たし，目的関数 $J(P) = \int_0^\infty (\rho(P(t)) - t \cdot \mu(P(t)))dt$ を最大にする関数 $P(t)$ を考える．

このとき，$P(t)$ は $t \in (t_{i-1}, t_i]$ のとき $P(t) = P(t_i)$ 及び $t > t_h$ のとき $P(t) = \emptyset$ を満たす，高々 $n + 1$ の変化値 $0 = t_{-1} < t_0 < t_1 < \cdots < t_h$ を持ち，さらに $P(t_0), P(t_1), \ldots, P(t_h)$ から成る $\mathcal{P}$ は最適ピラミッドである．

Proof: Γ に $n = m^d$ 個のピクセルがあるので，$t \in (0, \infty)$ のとき $P(t)$ の中に高々 $n + 1$ の異なる領域がある．したがって，変化値の数が $n + 1$ 以下であることは明らかである．目的関数を最大にするために，$\rho(P(t_i)) - t \cdot \mu(P(t_i))$ と $\mu(P(t_i)) = \rho(P(t_{i-1})) - t \cdot \mu(P(t_{i-1}))$ は変化値 $t = t_{i-1}$ で等しい $t_{i-1} = conf(P(t_{i-1}) \setminus P(t_i))$ であり，$\mathcal{P}$ は ρ のピラミッドである．

2 乗された L_2 エラー

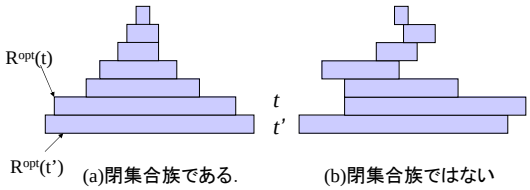
$E(\mathcal{P}) := \sum_{i=0}^{h} \sum_{c \in P(t_i) - P(t_{i+1})} (conf(c) - t_i)^2 \cdot \mu(c)$ は $E(\mathcal{P}) = \sum_{c \in N} \rho^2(c)/\mu(c) - 2J(P)$ と等しい．従って，$J(P)$ を最大にする関数 $P(t)$ は，$E(\mathcal{P})$ を最小にするピラミッド $\mathcal{P}$ を定義する．よって証明された．

したがって，直観的に最適ピラミッド $\mathcal{P}$ は，できるだけ大きなパラメトリックゲインを持つ横断面を積み重ねることにより得られる．パラメトリックゲイン $g_t(R, \rho, \mu)$ を最大にする $\mathcal{F}$ の領域 $R^{opt}(t)$ を考える．直感的には t が増加する場合，$R^{opt}(t)$ は減少する．$\{R^{opt}(t)\}$(正確には，関数 $R^{opt}(t)$ の閉包) がピラミッドを形成する場合，それは明らかに最適ピラミッドである．ところが，最大ゲイン領域 $R^{opt}(t)$ を積み重ねるとき，$R^{opt}(t) \subset R^{opt}(t')$ が $t > t'$ に対して成立するとは限らないので，それらは必ずしもピラミッドを形成するとは限られない（図 3 の右）．これは，最適ピラミッドを計算することを非常に難しくする．

3.2　閉集合族に対する考察

ある集合族 $\mathcal{S}$ が $G = [0, n]^d$ の閉集合族であるとは，$G \in \mathcal{S}$, $\emptyset \in \mathcal{S}$, $[X, Y \in \mathcal{S} \to X \cup Y \in \mathcal{S}]$ かつ $[X, Y \in \mathcal{S} \to X \cap Y \in \mathcal{S}]$ である事を言う．

［Theorem 1］ $\mathcal{F}$ が閉集合族であれば，各々の t に対し $\rho(R) - t \cdot \mu(R)$ を最大にする領域 $R \in \mathcal{F}$ を $\Psi(t)$ とすると，Ψ は単調条件を満たし，従って，最適ピラミッドになる．



$A = R^{opt}(t) \subseteq B = R^{opt}(t')$　$(t > t')$

図 3　ピラミッド問題で $R^{opt}(t)$ のイメージ

Proof: $t > t'$ のとき $A = \Psi(t) \subseteqq B = \Psi(t')$ である．$R \subset R'$ の場合，t に対して関数 $g_t(R, \rho, \mu) - g_t(R', \rho, \mu)$ は増加しない関数である．$A \setminus B$ が空集合でない場合，$0 \geq g_{t'}(A \cup B, \rho, \mu) - g_{t'}(B, \rho, \mu) = g_{t'}(A, \rho, \mu) - g_{t'}(A \cap B, \rho, \mu) \geq g_t(A, \rho, \mu) - g_t(A \cap B, \rho, \mu)$ である．$A = \Psi(t)$ なので，$g_t(A, \rho, \mu) \geqq g_t(A \cap B, \rho, \mu)$ になる．従って $0 \geqq g_t(A, \rho, \mu) - g_t(A \cap B, \rho, \mu) \geqq 0$ であり，$g_t(A, \rho, \mu) = g_t(A \cap B, \rho, \mu)$ になる．従って $A = A \cap B$ であり，$A \subseteqq B$ である．

従って，集合族 F が閉集合族で構築される場合，F の最適のピラミッドを計算するための効率的なアルゴリズムを設計が必要である．

4.　1 次元の場合

1 変数の場合，$G = [0, n]$ であり，$\mathcal{F}$ として，整数端点を持つ区間全体の集合 $\mathcal{I}$ を取ることが自然である．区間族 $\mathcal{I}$ は閉集合族ではない (2 つの区間の和集合は一般に区間ではない)．しかし，ある固定した $i \in [0, n]$ を含む区間全体（+空集合）$\mathcal{I}(i)$ は閉集合族であり，$\mathcal{I}$ に関する最適ピラミッドは，ある $\mathcal{I}(i)$ に関する最適ピラミッド (i はピラミッドの頂点に対応する）となる．

一方，$\rho(I) - t \cdot \mu(I)$ を最適にする区間 $I \in \mathcal{I}(i)$ が変化する

t をトレースする事は，計算幾何学での凸包上のトレースを (正確には左右 2 つの凸包で）行う計算に対応する [11]．ソートされた点の凸包計算は線形時間でできるので，この手法を用いると，$O(n^2)$ の時間で $\mathcal{I}$ に対する最適ピラミッドを求める事ができる．さらに下記の結果が示されている [6].

［Theorem 2］ 与えられた ρ, μ に対し，$\mathcal{I}$ に関する最適ピラミッドは $O(n \log n)$ 時間で計算する事ができる．

5. 2 次元での実用的な領域族

n をピクセルの数，N をサポート値の和（すなわち，データの数）とする．2 次元の領域族でもっとも基本的なのは軸方向長方形の族の場合である．しかし，この場合では $O(n^5)$ の計算時間のアルゴリズムしか知られていない [6]．ここでは，実用的な多くの領域族についてより効率的な手法を考える．

［Definition 2］ 2 次元ピクセル面 $G = [0, m]^2$（$m = \sqrt{n}$） 内で，適当な関数 $y = h(x)$ より真に下にあるピクセル全体の和になっている領域を下半切断領域と呼ぶ．

［Definition 3］ 2 次元ピクセル面 $G = [0, m]^2$ 内の 1 点 p に対し，p を含む長方形たちの和集合を p を中心にする矩形和楕円型領域（rectilinear ellipsoid）と呼ぶ．

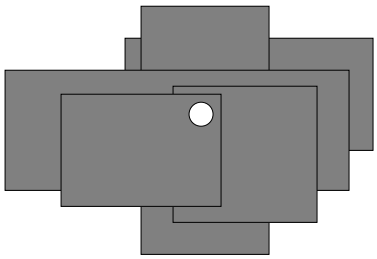


図 4　和矩形楕円形領域

点 p を中心とする矩形和楕円型領域（図.4）の族は，点 p を含む矩形全体の族の集合和，集合積に対する閉包になっている．矩形和楕円型領域の族は直交凸領域の族の部分族であるが，軸方向楕円の離散化を含む広い領域族であり，指数個の領域を持つ．例として，単調減少関数の下半切断領域は原点を中心とする矩形和楕円型領域となる．

［Lemma 2］ 点 p を中心とする矩形和楕円型領域全体の集合は閉集合族である．

Proof: 矩形和楕円型領域は点 p を含む長方形の和集合としてとられる領域である．点 p を中心とする矩形和楕円型領域 $\mathcal{F}$ とし，$\mathcal{F}$ の任意の領域 R と R' に対して，$R \cap R' \in \mathcal{F}$ および $R \cup R' \in \mathcal{F}$ である．すなわち，$\mathcal{F}$ の任意の領域 R と R' は和集合と共通集合に閉じているので，$\mathcal{R}^d$ の領域族 $\mathcal{F}$ は閉集合族である．

［Lemma 3］ 与えられた点 p を中心とする矩形和楕円型領域で与えられた t に対して利得を最大にする領域は $O(n)$ で計算できる．

Proof: まず，$p = (0, 0)$ の場合を考える．この場合，矩形和楕円型領域は単調非減少関数で区切られた階段状の領域になる．まず，最初の行における最初の i 個のピクセルのゲイン (図.6 の左) の和（Prefix 和）である $f(i, j) = \sum_{s=1}^{i} \rho((s, j)) - t\mu((s, j))$ を計算する (図.6 の右)．これはすべての (i, j) に対して $O(n)$ 時間で計算できる．

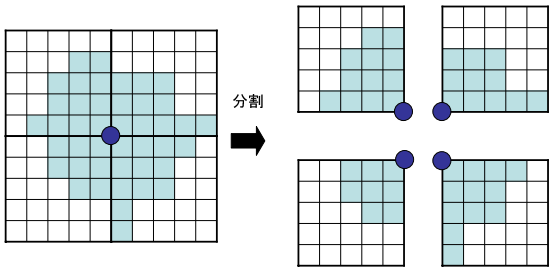


図 5　領域の 4 分割

$\alpha(i, j)$ を p が中心で (i, j) を含み，y 座標が j より大きい領域を含まないような矩形和楕円型領域の中でゲインが最大になるもののゲインと定義する．すると，p を中心とする矩形和楕円型領域の中でゲインが最大になるもののゲインは $\max_{j=1}^{m} \alpha(1, j)$ によって求まる．$\alpha(i, j)$ は（図　7）以下のダイナミックプログラミングによりすべての (i, j) に対して $O(n)$ 時間で求まる:

$$\beta(i, j) = \alpha(i - 1, j) + f(i, j) \tag{1}$$

$$\alpha(i, j) = \max\{\beta(i, j), \alpha(i, j + 1)\}. \tag{2}$$

そして，領域はダイナミックプログラミングプロセスのバックトラッキングにより計算できる．

一般の $p = (x_p, y_p)$ の場合は，ピクセル領域を $x = x_p$ と $y = y_p$ で 4 分し（図 5），各象現で上記と同じ操作を行い，得られた領域の和領域を求めればよい．

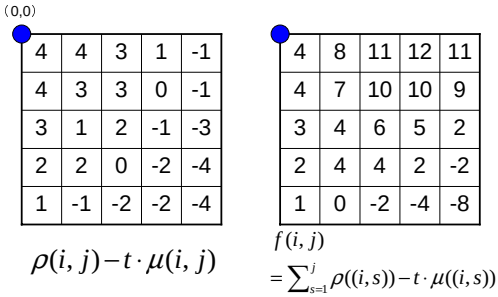


図 6　$\rho(i, j) - t \cdot \mu(i, j)$ と $f(i, j)$

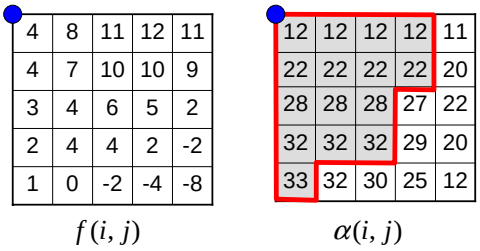


図 7　$f(i, j)$ と $\alpha(i, j)$

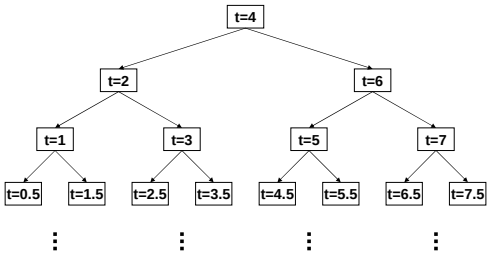


図 8　t

［Theorem 3］　指定された一点 p を中心とする矩形和楕円型領域全体の集合に関する最適ピラミッド（図 11 の右）は，$O(n \log N)$ 時間で計算される．ここで N は問題の出力精度であり，$1/N$ の近似精度で最大な積分値を持つ解が出力される．（従って，N が入力精度以上なら最適解を出力する）．

Proof: 閉集合族なので，各 t に関して，利得を最大にする領域を求めればよい．これは $O(n)$ 時間でできる．更に，ある t での最適領域を計算した時に，これが G を領域の内部と外部に分割するため，各部分を独立に，対応する部分に問題を縮小して解ける．従って分割統治が可能である．t はプロセスの分岐（図 8）を用いる事により，深さ $\log N$ の再帰で全ての部分での最適領域の計算が終了する．各深さでの全ての部分領域での最適領域は計算時間は $O(n)$ で求まる．従って，全体に関する最適ピラミッドは深さが $\log N$ なので $O(n \log N)$ 時間で計算できる．

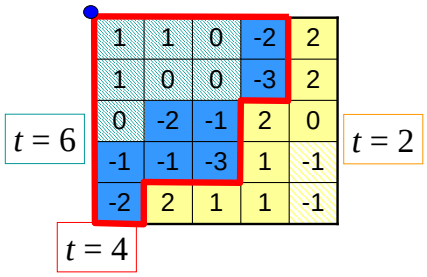


図 9　分割統治法による最適ピラミッド構築

アルゴリズムの実行例を図を用いて説明する．はじめに図 9 のように $t = 4$ で領域を 2 つに分割した後，左上の領域を $t = 6$ で再分割し，右下の領域を $t = 2$ で再分割している．具体的には，まず，$t = 4$ で最適領域を求める．ピクセルの数を n とすると $t = 4$ での最適領域の計算時間は $O(n)$ になる．２番目のレベルで，求まった領域は図 9，図 10 の青と黄色で色塗りしている．左上の青の領域の中で $t = 6$ の時の最適領域を求める．また，右下の黄色の領域の中での $t = 2$ の時の最大領域を求める．青い領域のピクセルの数を n_1，黄色い領域のピクセルの数を n_2 とすると最適領域を求めるためにはそれぞれ $O(n_1)$ 時間 $O(n_2)$ 時間かかり，このレベルでの総計算時間は $O(n)$ になる．次の３番目のレベルでは，４つの領域があり，ピクセルの数がそれぞれ n_1'，n_2'，n_3'，n_4' になる．このレベルでの最適領域計算時間は それぞれ $O(n_1')$ 時間，$O(n_2')$ 時間，$O(n_3')$ 時間，$O(n_4')$ 時間かかり，総計算時間は $O(n)$ 更に分割し同様に繰り返す．各レベルでは重なっていない領域に対して，最適領域を求めるため総計算時間は $O(n)$ になる．ある領域内で最適領域を求めるための計算時間は領域のピクセル数に比例する．各レベルでは重なっていない領域に対して最適領域を求めているため総計算時間は $O(n)$ になる．従って，各レベルでの計算時間は $O(n)$ で深さが $\log N$ なので，全体に関する最適ピラミッドの総計算時間は $O(n \log N)$ である．図 11 は入力（左図）に対し，最適ピラミッドの出力（右図）の例である．

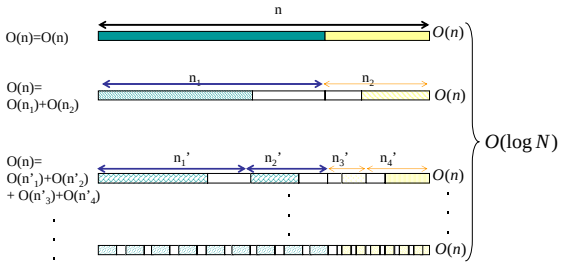


図 10　分割統治法による最適ピラミッド構築の計算時間

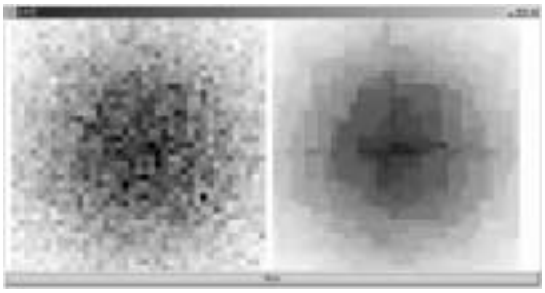


図 11　2 次元最適ピラミッド構築の入力と出力

6.　高次元の場合

6.1　小さい数の領域族の場合

$\mathcal{F}$ が M 個の異なる領域を持っている場合，最適ピラミッドは d 次元の場合 M と n についての多項式時間で計算できる．頂点集合と $\mathcal{F}$ とする閉路を持たない有向グラフ $H(\mathcal{F}) = (\mathcal{F}, E)$ を構築する．各 $\mathcal{F}$ のペア R と R' が $R \supset R'$ を満たすとき，グラフに有向辺 $e = (R, R')$ を加える．そして $\rho(R \setminus R') = t(e) \cdot \mu(R \setminus R')$ を計算する．値 $t(e)$ は e の高さラベルと呼び，$r(e) = t^2(e)\rho(R \setminus R')/2$ は e の利得と呼ぶ．

$t(e_{i-1}) < t(e_i)$ が $i = 1, 2, \ldots, q$ で成立する場合，有向路 $p = e_0, e_1, \ldots, e_q$ を許容有向路 (*admisible*) と呼ぶ．許容有向路の利得は辺の利得の総和である．

［Lemma 4］　最適ピラミッドは，(R, R') が経路上の辺である場合のみ $R \setminus R'$ がピラミッドの面である場合，$H(\mathcal{F})$ の最大利得の許容路（*admissible path*）に対応する．

したがって，最適ピラミッド問題を循環路を持たない有向グラフ $H(\mathcal{F})$ の最大重み経路 (*maximum-weight-path*) 問題にすることができる．$H(\mathcal{F})$ の各有向閉路が大部分で n 辺を持っていることに注意する．ダイナミックプログラミングアルゴリズムを使うことによって，次の結果を得る：

［Theorem 4］ M 個の異なる領域の $\mathcal{F}$ のための最適ピラミッドは $O(M^2n)$ 時間で計算できる．

しかし，上記のアルゴリズムは実用的ではない．例えば，長方形の領域族は $O(n^2)$ 領域を持っている．従って，上記の計算時間は $O(n^5)$ である．さらに，正確な階層状の領域ルール計算のために，M が n において幾何級数的に大きい族を考える．

したがって，特別な領域族のために，より効率的なアルゴリズムを考える．

6.2 直交領域の stabbed union

Γ の領域の典型的な閉集合族について考える．Γ の固定セル c については，各々が含んでいる直交領域の結合として R を表わすことができる，Γ の領域 R は c で直交領域の *stabbed union* と呼ぶ．セル c は R の中心セルと呼ぶ．2 次元の場合の例を図 **??**に示す．2 次元の場合は矩形和楕円型領域とほぼ同一で，相異はグリッド点の替わりにセルを考える点である．

セル c のすべての *stabbed union* 族が閉集合族であることは明らかである; 実際，c を含んでいるすべての長方形集合族に対して閉じている．与えられたピラミッドは，中心セルでの *stabbed union* 族に基づいてる．当然，中心セル (あるいは点) は，ピラミッドのピークである．最適ピラミッドを計算するためのアルゴリズムを設計するためには，*stabbed unions* 族の最大パラメトリックゲイン領域計算のために効率的なアルゴリズムを必要とする．上記の目的のために，問題を一般化し，グラフアルゴリズムを適用する．

6.3 グリッドグラフの推移閉包

ボクセルグリッド Γ において，中心セル $c = (c_1, c_2, \ldots, c_d)$ を固定し，頂点セットが Γ のすべてのボクセルから成る有向グラフ $G(c)$ を定義する．ボクセル $p = (p_1, p_2, \ldots, p_d)$ および $q = (q_1, q_2, \ldots, q_d)$ について，p と q の間の L_1 距離は $dist(p,q) = \sum_{i=1}^{d} |p_i - q_i|$ である．p の近隣のセルとは p からの L_1 距離が 1 のセルである．セル p およびその近隣 q について，有向辺が定義される. その方向は $dist(p,c) = dist(q,c) + 1$ の場合 (p,q) (つまり p から q まで) であり，そうでなければ (q,p) である．グラフ $G(c)$ は $d(m-1) \times m^{d-1} = O(n)$ 辺 (d は定数とする) の弱い連結有向グラフであり，また，c はそのユニークな *sink* 頂点 (つまり出発する辺のない頂点) である．

$G(c)$ の部分グラフ $H = (V, E)$ において，V の各頂点 v から c までの H に有向路が存在する場合，H を根付き部分グラフ (*rooted subgraph*) と呼ぶ．

$G(c)$ の根付き部分グラフ $H = (V, E)$ が与えられ，H に v から u までの有向路が存在する場合，頂点 u が頂点 v によって H において支配されると言う．頂点の集合 W において，W のすべての頂点 V に対し, V に支配される頂点がすべて W に含まれるとき，W を H-閉包と言う．各 H-閉包は Γ 中でセル c を含んでいる連結領域を定義する．

与えられた $G(c)$ の根付き部分グラフ H の，すべての H-閉包の集合族 $\mathcal{F}_H$ を考える．推移閉包であるという性質は集合和と集合積について閉じているので，次の命題が示せる:

［Proposition 1］ $G(c)$ の根付き部分グラフ H について，$\mathcal{F}_H$ は閉集合領域族である．

次の補題は，$G(c)$ および $\mathcal{F}_H$ の定義からなる．

［Lemma 5］ Γ の中の領域 R は，$G(c)$-推移閉包の場合のみ c の *stabbed union* である．

［Lemma 6］ H と H' を $G(c)$ の全域根付き部分グラフとする．このとき，H が H' の部分グラフならば $\mathcal{F}_H \supseteq \mathcal{F}_{H'}$ である．

6.3.1 $\mathcal{F}_H$ の最適ピラミッド計算アルゴリズム

$G(c)$ の根付き部分グラフ H を固定する．$\mathcal{F}_H$ に関しての最適ピラミッドの面の高さを定義するパラメーター値 t を考える．また，各ボクセル p(H の対応する頂点) に重さ $\rho(p) - t \cdot \mu(p)$ を与える．次の補題により，面の高さを与える t が小さな分母および分子を持つ有理数であると仮定できる．

［Lemma 7］ t がある領域族のための最適ピラミッド面を定義する高さである場合，t は N 以下の 2 つの整数の比によって表わされる有理数である．

Proof: t が面 P_i を定義すると仮定すると，$\rho(P_i \setminus P_{i+1}) = t \cdot \mu(P_i \setminus P_{i+1})$ である．ρ と μ は N 以下の整数値をとるので，補題は成り立つ．

定義によって，最大 (パラメトリック) ゲイン領域 $P^{opt}(t) \in \mathcal{F}_H$ は，領域のボクセル重みの合計を最大限にする H-閉包である．グラフ理論的な用語では，重み付き有向グラフ H の最大推移閉包であり，辺の集合 (*cut set*) の削除により c を含んでいる H の連結成分として得られる．$\mathcal{F}_H$ における $P^{opt}(t)$ の例および対応するカットセットが図 12 に示されている; 図 12 の左図の各ピクセル p の値は重さ $\rho(p) - t \cdot \mu(p)$ である．次の定理は Hochbaum [14] によるものであり，Wu と Chen [19] により本論文とは異なる幾何学セグメンテーション問題に適用された．

［Theorem 5］(Hochbaum [14]) 実数頂点重みを持つ n 頂点と m 辺からなる有向グラフ G の最大重み閉包は $O(T(n,m))$ 時間で計算できる．ここで，$T(n,m)$ は非負辺重みを持つ n 頂点と m 辺からなる有向グラフ H の最小 s-t カット計算時間である．

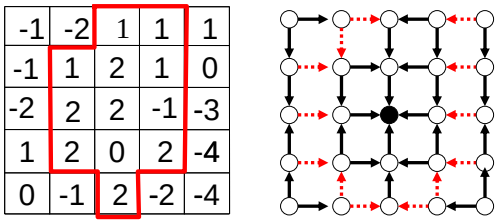


図 12 最適領域と $G(c)$ の *cutset*

［Theorem 6］ 集合族 $\mathcal{F}_H$ のための最適ピラミッドは $O(n^{1.5} \log n \log^2 N)$ 時間で計算できる．

Proof: N 以下の N 整数の対によって定義された有理数の集合 S の中のすべての可能な面の高さを知るために，2 分探索の一般化を適用する．プロセスは多くの部分プロセスに分岐するの

で，2 分分岐探索と呼ぶ．$t_0 < t_1$ に対し，$P^{opt}(t_0)$ と $P^{opt}(t_1)$ をすでに計算していると仮定する．

$P^{opt}(t_0) = P^{opt}(t_1)$ の場合，区間 $I = (t_0, t_1)$ を $inactive$ にする．そうでなければ，$active$ と呼ぶ．区間が $active$ ならば，t_{01} を $t(t_0, t_1)$ の中央値 (正確には，中央値近似する S の要素) とする．$\mathcal{F}_H$ 族が閉集合族なので，$P^{opt}(t_1) \subset P^{opt}(t_{01}) \subset P^{opt}(t_0)$ である．したがって，$P^{opt}(t_{01})$ の計算に際しては，H から $P^{opt}(t_0)$ の外側の頂点をすべて取り除き，$P^{opt}(t_1)$ の内部の頂点を単一頂点へ縮約できる．したがって，$|P^{opt}(t_0)| - |P^{opt}(t_1)|$ 個の頂点を持つ有向グラフ $H(I)$ の最大推移閉包を計算すれば十分である．

2 分分岐探索は各レベルにおいて，前のレベルで作られた $active$ な区間を 2 つの区間へ分離する．初期の全区間の長さは高々N であり，葉区間の長さは少なくとも $1/(N-1) - 1/N = 1/N(N-1)$ であるので，レベルの数は $O(\log N^3) = O(\log N)$ である．各レベルで，すべての $active$ な区間 I の $H(I)$ の頂点の数の合計は $O(n)$ である．大きさ n のグラフの最大推移閉包を求める計算時間は $T(n, n) = O(n^{1.5} \log n \log N)$ [13] である．したがって，各レベルを処理する計算時間は $O(n^{1.5} \log n \log N)$ である．したがって，総計算時間は $O(n^{1.5} \log n \log^2 N)$ である．

［Theorem 7］ 与えられたセル c の直交領域 *stabbed union* の集合族に対する最適ピラミッドは $O(n^{1.5} \log n \log^2 N)$ 時間で計算できる．

7. まとめと考察

ピラミッド構造はいくつかの分野で有用である．まず，確信度分布の傾向の良いビジュアル化を与える．

次に，しきい値の高さ t 以上のピラミッドの一部の選択によって (このオペレーションを切り取りと呼ぶ)，領域内部のその実際の幾何学的な位置に依存する各データに対するルールの影響に基づいた情報と一緒に結合ルールを与える領域を生成することができる．

高さ t は，サポートしきい値あるいは GINI 最大化のようないくつかの最適化基準の使用により決定される．[17]

第 3 に，一旦領域ルールを得れば，オリジナルのデータ分布からピラミッドを引き平均確信度をもって同じ先行のサポートを補充することにより，容易にこのルールの影響を削除することができる．

さらに，より弱いルールを抽出することができる．また，異なるピークを備えたピラミッドを同時に考えることにより，データから 2 つ以上の階層化ルールを抽出できる．これにより，各々のそのようなピラミッドからの高い一部分を切り取ることにより，データの大多数をカバーするクラスタリングを自動的に与える．第 4 に，すべてのピークでのピラミッド近似で離れ値であるデータアイテムを，例外的なデータ (恐らくある入力エラー) あるいはあいまいなデータと見なすことができるので，データクリーニングのために潜在的に使用することができる．適切な領域族の選択は非常に重要な問題である．しかし，ディジタル化された星形領域は，$d = 2$ ではリーズナブルな族であるように思える．$d \geqq 3$ については，より多くの枝を備えたグラフに対応する族が有用かも知れない．データマイニングアプリケーションのための良い族を定義することに役に立ち，$d \geqq 3$(特に $d = 3$) の時、頂点の各出次数、入次数の良いパラメーターを発見する実験を行う．

文　献

[1] R. Agrawal, T. Imielinski, A. Swami, Mining Association Rules between Sets of Items in Large Databases, *Proc. SIGMOD* (1993) 207–216.

[2] A. Amir, R. Kashi, N. S. Netanyalm, Analyzing Quantitative Databases: Image Is Everything, *27th Proc. VLDB Conference* (2001).

[3] T. Asano, D. Chen, N. Katoh, and T. Tokuyama, Efficient Algorithms for Optimization-Based Image Segmentation, *Int'l J. of Computational Geometry and Applications* **11**(2001) 145-166.

[4] I. Bloch, Spatial Relationship between Objects and Fuzzy Objects using Mathematical Morphology, in *Geometry, Morphology and Computational Imaging*, 11th Dagsthul Workshop on Theoretical Foundations of Computer Vision, April 2002.

[5] D. Chen,J. Chun, N. Katoh, and T. Tokuyama, Layered Data Segmentation for Numeric Data Mining, *Presented at Submitted*.

[6] J. Chun, N. Katoh, and T. Tokuyama, How to Reform a Terrain into a Pyramid, *Presented at DIMACS Workshop on Geometric Graph Theory* (2002).

[7] J. Chun, K. Sadakane, T. Tokuyama, Improved Algorithms for Constructing Pyramids from Terrains. *Submitted*.

[8] Y. Morimoto, T. Fukuda, S. Morishita, and T. Tokuyama, Implementation and Evaluation of Decision Trees with Range and Region Splitting, *Constraints* (1997) 402-427.

[9] S. Dasgupta, Learning Mixtures of Gaussians *Proc. 40th IEEE FOCS* (1999), pp. 634–644.

[10] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama, SONAR: System for Optimized Numeric Association Rules, *Proc. SIGMOD 1996* (1996) p.553.

[11] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama, Mining Optimized Association Rules for Numeric Attributes, *Journal of Computer and System Sciences* **58** (1999) 1-12.

[12] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama, Data Mining with Optimized Two-Dimensional Association Rules, *ACM Transaction of Database Systems* **26** (2001) 179-213.

[13] A. Goldberg and S. Rao, Beyond the Flow Decomposition Barrier, *Proc. 38th IEEE FOCS* (1997) 2–11.

[14] D. S. Hochbaum, A New-old Algorithm for Minimum Cuts in Closure Graphs, *Networks* **37** (2001) 171-193.

[15] D. T. Lee and F. P. Preparata, An Optimal Algorithm for Finding the Kernel of a Polygon, *Journal of the ACM* **26** (1979) 415-421.

[16] Y. Morimoto, H. Ishii and S. Morishita, Construction of Regression Trees with Range and Region Splitting, *The 23rd VLDB Conference* (1997) 166-175.

[17] Y. Morimoto, T. Fukuda, S. Morishita, and T. Tokuyama, Implementation and Evaluation of Decision Trees with Range and Region Splitting, *Constraints* (1997) 402-427 (a preliminary version in VLDB'96).

[18] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, 1995.

[19] X. Wu and D. Z. Chen, Optimal Net Surface Problems with Applications, *Proc. 29th International Colloquium on Automata, Languages and Programming* (2002) 1029-1042.

[20] K. Yoda, T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama, Computing Optimized Rectilinear Regions for Association Rules, *Proc. KDD97* (1997) 96-103.