情 報 学 基 礎 46－2
（1997．7．24）

領域及び区間分割を用いた決定木の作成
-領域分割の有効性の検証-

森本 康彦，福田 剛志，森下 真一，徳山 豪

日本アイ・ビー・エム株式会社　東京基礎研究所

あらまし　今日，エントロピヒューリスティックに基づく決定木作成法が広く用いられている．この手法では，数値属性を扱う場合，従来,「ギロチンカット」と呼ばれるテストを用いて木を作成している．しかし，このような従来手法では，数値属性間に強い相関が存在した場合，良い決定木を生成することができない．我々は数値間の相関を，より良く決定木作成に反映させるため，２つの数値属性からなる平面上の領域を用いてデータ分割を行なう決定木を作成した．本論文では，領域算出のための効率的なアルゴリズムについて述べ，その有効性を実験を通じて検証した．

Constructing Decision Trees with Range and Region Splitting
- Evaluation of Effectiveness of Region Splitting -

Yasuhiko Morimoto,　Takeshi Fukuda,　Shinichi Morishita,　Takeshi Tokuyama

Tokyo Research Laboratory,　IBM Japan Ltd.

Abstract　Decision trees based on entropy heuristic are widely used. In order to handle numeric attributes, conventional trees use "guillotine cutting" tests. However, such conventional methods are inefficient if any numeric attributes are strongly correlated. We propose an extension the heuristic to handle arbitrary correlations among attributes. For each pair of numeric attributes with strong correlation, we compute a region with respect to these attributes and the objective attribute and use the region in decision trees. We have implemented efficient algorithm to compute the regions. Diverse experiments show that this method can produce compact and accurate trees.

1 はじめに

決定木

蓄積されている大量のデータをいかに有効利用するか，そんなデータからいかに有用な情報を発見するか，といったニーズから，近年，データマイニング技術が注目されている [1, 2]．なかでも，決定木を利用したクラス分け技術，データ分類技術は人工知能分野においても盛んに研究されている [5, 13, 16]．

病院において患者の過去の検査結果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，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，というニーズなどもある。

例えば，表1のような糖尿病診断データベースを考える．このデータベースには，血圧 (BP)，血中コレステロール (C)，尿糖 (S) などの診断データ，ならびにその患者が糖尿病 (Diabete) であったかどうかを示す二値属性が入力されているとする．このようなデータベースから，目的とする (糖尿病であるかないかといった) 2値属性の値によってデータを分類するための2進木データ構造として，代表的なものに図1のような決定木がある．図の決定木では，各レコードの「尿糖 (S) が＋か－か」が検査され，それが＋なら次に，「血圧 (BP) が 100 以上かどうか」が，－なら「血中コレステロール (C) が＋か－か」検査され，その結果をもとに糖尿病を診断する．

決定木は，この例のように，データベース中のレコードを，木のノードに対応する「尿糖 (S) が＋か－か」といったテストで再帰的に分割してゆく2進木構造で，一般的に，深さが小さく，頂点数も小さいのが理想的である．しかし，最小の決定木の構成問題は NP 困難である [11, 12]．従って，何らかの近似的な解法が必要になってくる．これに対するアプローチには様々なものがあり，代表的なものに Quinlan[16] のエントロピを用いたヒューリスティクスがある．この方法は決定木を上 (根の方) から作っていくが，各段階で行なうテストを選ぶ時に，各々の条件文によるテストで，データ集合がどのような分布に分割されるかを計算する．そこで，分割のエントロピを計算し，それが最小になる (すなわち，分割の各成分でクラス分布がもっとも偏っている) 条件文を選ぶ．実用上、この方法はかなり良い決定木を生成する。

数値属性を用いたデータ分割

上述の例で，テストの条件文に用いる属性のうち，血中コレステロール (C) および尿糖 (S) は，とり得る値が順序のない (＋あるいは－といった) 離散的な値である「範疇属性」である．一方，血圧 (BP) は順序のある連続値をとる「数値属性」である．データの属性が数値属性である時，一般的には，一つの数値属性，例えば「体重」に対して，(体重 $< Z$)?や (体重 $\in I$)?のような，「一次元ルール」，「ギロチンカット」などと呼ばれる条件文を用いる．条件分中の閾値 Z や区間 I は，エントロピーなどの基準で，よい値になるものを選び，そのようにして選ばれた条件文のなかから，最も良いものをノードのテストとしてデータ分割に利用する．

Patient	BP	C	S	Diabete
0001	140	+	+	○
0002	120	−	+	○
0003	110	−	−	×
0004	80	−	−	×
0005	90	−	+	×
0006	120	+	+	○
0007	90	+	−	○
0008	100	−	−	×
0009	100	−	+	○
0010	120	−	−	×

表 1: Health Check Records

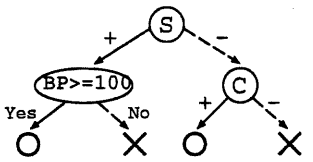

図 1: Decision Tree

ところが，数値属性を扱う場合の従来技術には，大きな欠陥があることが知られている．Quinlan は，数値属性を持つデータに対して，数値属性間に相関がある場合，従来の方法では不十分であることを指摘している．例えば，属性として「体重」と「身長」を考えてみる．この2属性を用いた健康診断では，図2にグレーで表される領域のルール

$$0.85 * 22 * (\text{身長})^2 < \text{体重} < 1.15 * 22 * (\text{身長})^2$$

がかなり良く「正常」の判定を行なう．しかし，従来の手法では，この診断のために図のようなギロチンカットでデータ分割を行なわなければならず，非常に大きな決定木を作らなければならないばかりか，このようにして作られる決定木のルール群から，データ中に本質的に存在していた，上記の相関は見つけにくい．

この欠陥を補うために，複数の数値属性の組を考え，対応する平面もしくは (多次元) 空間を，直線もしくは (超) 平面で二つの領域に切ることにより，データを分割する試みもなされている [4, 5, 6, 8]．しかしながら，直線や平面による分割は，計算時間が掛かる上に分割の自由度が低いため，あまり効果的とはいえない．

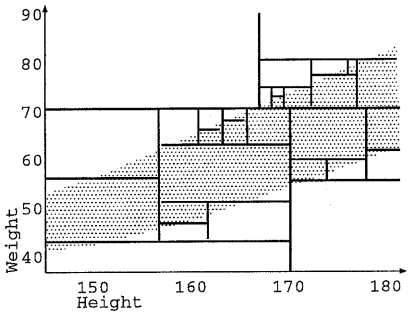

図 2: Healthy Region and Guillotine-cut

領域分割

このような問題に対処するため，決定木を 2 次元平面上の領域ルールによるテスト条件で分割してゆく手法を利用する．領域の高速な算出法は，浅野らにより画像切り出しの分野で発明され [3]，それを利用した 2 次元結合ルールの高速算出法が福田らによりデータマイニング技術として発表されている [9, 10]．本論文では，これらの技術を応用し，2 次元結合ルールのうち，エントロピを最適にする領域の効率的な発見手法を述べ，領域分割を利用した決定木の有用性について検証する．

分割したい n 個のレコードからなる集合があるとする．まず，それぞれの数値属性に対し，集合を $N \leq \sqrt{n}$ なる N個のほぼ均等なレコード数からなるバケットに離散化する．次に，すべての数値属性のペアに対し，$N \times N$ グリッド平面 G を用意する．その G 上での，X 単調 (x-monotone), あるいは直交凸 (rectilinear), あるいは長方 (rectangular) 領域族の領域を利用してデータを分割し，2 つの部分集合を作る．これらの部分集合を元に，この操作を再帰的に行なうことにより，領域分割を利用した決定木を作成できる．ここで X 単調領域族とは図 3のように，垂直方向には単一の区間ルールである連続領域とする．また，水平方向も同時に単一の区間ルールである領域を直交凸領域とする．

このように自由度の高い領域をテストとして利用すると，図 2 のような相関にも対応可能で，認識しやすく小さい決定木となる．図 3は，3章で利用した実際の糖尿病データベースから，(X 単調) 領域分割の手法で作られた決定木の，ルートノードでのテストを示す．この例では，横軸が糖検査値 (GlucoseConc)，縦軸が年齢 (Age) の三角形領域の内側が糖尿病で陽性を示す傾向が強いことを示している．

我々のアルゴリズムは，G 上で実効およそ $O(n \log n)$, 最悪のケースで $O(nN^2)$ の計算量でエントロピを最適化する X 単調領域 R_{opt} を算出する．直交凸および長方領域の場合はそれぞれ，$O(nN^3)$，$O(N^3 \log n)$ となる．

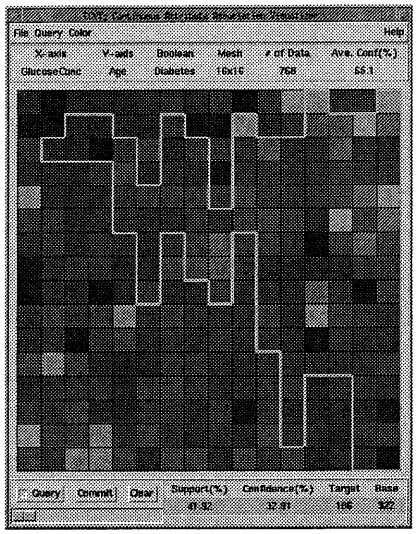

図 3: Region Splitting

2 最適領域

エントロピ

n 個のレコードからなるある集合 Sを分離することを考える．ある k個の値をとる目的属性に対する分離性を評価するエントロピ関数 $Ent(S)$ は，p_jを目的属性値の j番目の値をとる確率とすると

$$Ent(S) = -\sum_{j=1,\ldots,k} p_j \log p_j$$

で表される．この集合 Sを n_1個，n_2個のレコードからなる部分集合 S_1，S_2に分割した場合の，集合 Sのエントロピは

$$Ent(S_1; S_2) = \frac{n_1}{n} Ent(S_1) + \frac{n_2}{n} Ent(S_2)$$

で与えられる．決定木はこのエントロピ関数が最小になるような分割をテストとする．

ここで，$x_i = p_i n$, $s(X) = \sum_{i=1}^{k} x_i$, $f(X) = f(x_1, .., x_k) = \sum_{i=1}^{k} x_i \log(x_i / s(X))$ を用いてエントロピ関数を

$$Ent(S) = -f(p_1, ..., p_k) = -\frac{1}{n} f(x_1, .., x_k)$$

と書き換え，y_iを S_1内の目的属性値が i 番目の値であるレコード数とすると，分割後のエントロピは

$$Ent(S_1; S_2) = -\frac{1}{n}\{f(y_1, ..., y_k) + f(x_1 - y_1, ..., x_k - y_k)\}$$

となる．このときエントロピ関数は以下の性質を持つ．

Lemma 2.1 $f(X)$ は $X \geq 0$ $(i = 1, 2, .., k$なる i に対して $x_i \geq 0)$ なる領域において凸関数であり，$X > 0$ かつ $X + 2\mathbf{a} > 0$ であるいかなるベクトル $\mathbf{a}$ に対して

$$\frac{f(X) + f(X + 2\mathbf{a})}{2} \geq f(X + \mathbf{a})$$

を満たす．

ハンドプロービング

領域の算出法を，簡単のため，目的属性が2値 $(k = 2)$ である場合について考える．分割したい集合 Sに対応するグリッド平面 Gと，その G上にとりうる，ある領域族の領域が集合 Sの部分集合 S_1であるとする．ここで，$x(S_1)$ を S_1の全レコード数，$y(S_1)$ を目的属性の0番目の値をもつレコード数とし，この2つの値の張る平面を仮定する．各領域はこの平面上の点 $(x(S_1), y(S_1))$ として特徴づけられる．ここで，エントロピの最小値を持つ領域(点)に注目したならば，lemma 2.1から，目的とする領域 R_{opt}は，上述の平面上に存在し得る全ての点集合のうち凸包上の頂点に存在する．

凸包上の頂点とそれに対応する領域は，福田らによる2次元結合ルール [9] で使われている「ハンドプロービング手法」を利用して高速に求めることができる．ハンドプロービングとは計算幾何学で知られている，ある傾きθを持つ線と凸包との接点を「タッチングオラクル」と呼ばれる方法で求める手法 [7] である．この技術を応用して，Gが $N \times N$のグリッドである場合，X単調領域で $O(N^2)$ 時間，直交凸領域および長方領域で $O(N^3)$ 時間のダイナミックプログラミングで頂点とそれに対応する領域を求めることができる．

凸包検索

エントロピ値を最小にする領域は凸包の頂点に絞って，高速に求めることができるが，さらに，以下の性質を利用して高速化することができる．

凸包検索における現在までのエントロピの最小値を E_{min}とし，図 4にある凸包の上側にある傾き θ_l，θ_rの2つの接線で求めた2接点間の区間 $I = [v^+(\theta_l), v^+(\theta_r)] = [I(left), I(right)]$ について考える．点 $Q(I) = (x_{Q(I)}, y_{Q(I)})$ を2つの接線の交点とし，その交点のエントロピ値を $E(Q(I)) = E(x_{Q(I)}, y_{Q(I)})$ とする．ただし，2接線の交点が存在しない場合，$E(Q(I)) = -\infty$ とする．

Lemma 2.2 $I(left), I(right), Q(I)$ の三点を頂点とする三角形の内側の点 $Q' = (x', y')$ は，*lemmna2.1*から

$$E(x', y') \geq \min\{E(Q(I)), E_{min}\}$$

である．

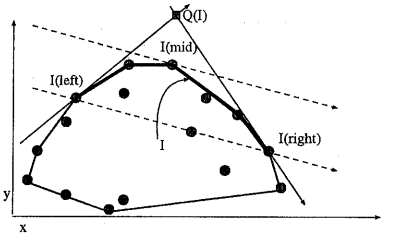

図 4: Hand Probe

この性質から，$E(Q(I)) \geq E_{min}$である区間 Iの如何なる点も E_{min} を与えない．従って，この条件を満たす区間は枝がりすることができる．そうでない場合は，図のように2端点を結ぶ傾きで，新たな接点 $I(mid)$ を求め，2つの新たな区間 $[I(left), I(mid)]$，$[I(mid), I(right)]$ を作る．凸包検索において，凸包上の全区間を $E(Q(I))$ の値を利用した順位キューで管理し，検索条件を満たす区間がなくなるまで，キューの最初の区間の細分化を繰り返す．E_{min}は単調に減少し，$E(Q(I))$ は単調に増加してゆくため，効率的に枝がりすることが可能となる．

3 実験結果

3.1 識別精度

領域分割による決定木の識別精度を測定するため，10フォールドクロスバリデーション実験を行なった．10フォールドクロスバリデーションでは，まず，対象となるデータ集合をランダムに10個のできる限り均等なサイズの部分集合になるよう分割する．次に，それらの部分集合の一つをテスト集合とし，残りの9個の和集合からなるトレーニング集合を用いて作成された決定木の精度を評価する．この評価手順を10個それぞれの部分集合に対して行なう．

トレーニング集合から作成される木は，分割を繰り返してゆくほど，そのトレーニング集合に対する識別精度が上げることができる．しかし，過度にトレーニング集合に対する，精度をあげる大きな決定木は，テスト集合での識別精度を低下させてしまう．我々は過度に大きな決定木を避けるため，分割を続けるかどうかをχ^2検定によってコントロールする [15]．

N個のタップルからなるノード Xを考える．ここで，Xが T，Fの2つのノードに分割され，それぞれタップル数が N_T，N_Fであるとする．$p(i)$ を目的属性値 i をとるタップルの X内での比率，$p_T(i), p_F(i)$ を，それぞれ分割された2つのノード内での i の比率とすると，この

分割のχ^2値は

$$\sum_i \frac{(N_T p_T(i) - Np(i))^2 + (N_F p_F(i) - Np(i))^2}{Np(i)}$$

で与えられる．このχ^2値が小さければ，Xでのiの分布と，分割後の2つのノードでのi分布が独立であるという帰無仮説を棄却できないため，分割の決定木内での意義が小さいと考えて，その時点で分割を止める．このχ^2値の閾値 (枝がりパラメータ) をより大きくとれば，木はより小さくなる．

この実験では，それぞれのデータ集合に対して，10フォールドクロスバリデーションを行ない，その10回のテスト集合に対するエラー率の平均値を，0から45までの枝がりパラメータによる木で計測し，その比較をおこなった．さらに，エラー率の最も低くなる枝がりパラメータでの木の大きさも比較した．分割に利用する領域族としては，X単調 (X-monotone)，直交凸 (Rectilinear)，長方 (Rectangular) の3つをそれぞれ用いた．

領域分割での各ピクセル上の平均タップル数 (ピクセル密度) は，領域分割による決定木の精度に影響を与える．ピクセル密度の小さい細粒度での領域は，トレーニング集合に過度にフィットしてしまう．逆に密度の大きい粗粒度での領域は，自由度の小さい領域となるため，同様にテスト集合での精度の低下を招く．経験的にはピクセル密度5から10程度の値が良い精度を得ることができる．そこで，領域による決定木は密度をやや細かい5，およびやや粗い10とし，精度の良い方の密度パラメータを用いた．決定木の深いノードにおいては，データ数が少なくなるため，充分なピクセル数を確保できなくなる．5×5以上のピクセル数を確保できないノードでは，領域でなく区間分割を行なった．

実験では「UCI Machine Learning Repository[14] の中から述語属性が数値データである表2のデータ集合を用いた．図5から図13に，それぞれのデータ集合に対する10フォールドクロスバリデーション実験の結果を示す．図中の例えば「Rectilinear(dens5)」は,「ピクセル密度5の直交凸領域」を用いた決定木の結果を表す．表3にこれらの実験結果の比較をまとめた．

各実験結果から，以下のような点が明らかになった．

- どんな決定木においても，小さい枝がりパラメータは過度に大きな木となり，実際に識別精度がよくない．
- ほとんどのケースで，領域分割を用いた決定木のほうが，従来手法に比べ識別精度が高く，木の大きさも小さい．
- X単調領域族は小さいピクセル密度では，過度にトレーニング集合にフィットしてしまうため，充分な密度を確保した方が良い．
- 直交凸および長方形領域族は，ピクセル密度が小さい場合でも比較的安定した精度を得ることができる．

Dataset	#tuples	#attr	#cls
balance scale	625	4	3
breast-cancer-wisconsin	699	9	2
german credit	1000	24	2
liver disorder	345	6	2
pima diabetes	768	8	2
segmentation	2310	19	7
vehicle	846	18	4
waveform	5000	20	3
waveform-+noise	5000	40	3

表 2: Dataset Summary

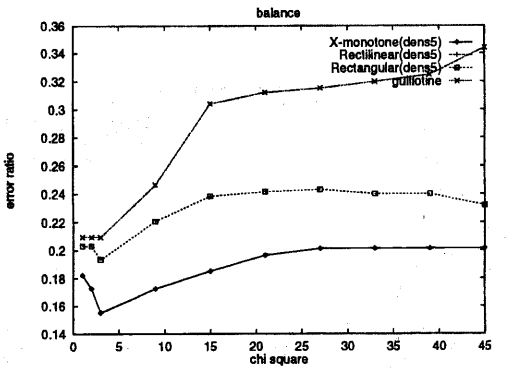

図 5: Accuracy for "balance scale"

- 長方形領域族は，領域の自由度が低過ぎ，数値属性間の様々な相関をうまく反映できないため精度および大きさの面で良い決定木を得られない．

3.2 計算速度

領域分割を用いた決定木の構築には，従来のギロチンカットによる手法以上に計算時間を要する．ここでは，計算時間の面から領域分割の影響を検証する．計算時間は全て，主記憶512MB，112MHz PowerPC 604 チップ搭載の IBM RS/6000 ワークステーションの AIX4.1 OS 上でのものである．

まず，決定木構築にかかる時間に大きく影響する，最適領域の計算時間を調べる実験を行なった．実験データとして$[N^2, 2N^2]$の範囲で一様分布になる乱数を生成し，それを$N \times N$グリットの行列の各ピクセルに「ピクセル内のレコード数」として割り当てる．さらに，その行列の各ピクセルに「ピクセル内の，あるクラスのレコード数」として$1, \ldots, N^2$の数を，外側から内側に大きくなる渦巻階段状に割り当てた．このように作られた実験データは凸包上の点がピクセル数の平方根Nに近い値になる．

表4に，Nが10から50までのサイズの実験データで，X単調，直交凸，長方の各最適領域の計算時間 (秒) および，領域発見に要したタッチングオラクルの回数を示す．結果から，タッチングオラクル回数の，サイズに

	X-monotone		Rectilinear		Rectangular		Guillotine	
Dataset	Err(%)	Size	Err(%)	Size	Err(%)	Size	Err(%)	Size
balance scale	15.52	34.7	15.52	34.7	19.34	44.7	20.95	48.8
breast-cancer-wisconsin	5.01	3.4	4.15	3.3	4.58	4.0	5.72	19.2
german credit	27.30	3.8	23.80	3.6	26.90	4.6	25.60	12.8
liver disorder	34.81	2.0	33.36	3.2	31.08	2.0	34.87	3.0
pima diabetes	24.47	4.5	25.12	3.0	23.69	4.6	26.82	4.4
segmentation	4.81	57.9	4.37	53.7	4.89	65.3	4.50	71.1
vehicle	30.02	17.0	28.47	12.2	27.65	38.9	26.23	94.0
waveform	21.74	46.9	20.98	33.2	22.36	65.1	22.74	91.7
waveform-+noise	22.54	30.3	21.32	38.6	22.94	67.7	24.36	91.7

表 3: Cross-Validation Results Summary

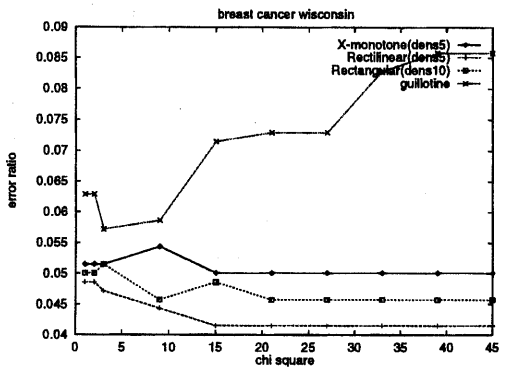

図 6: Accuracy for "breast cancer wisconsin"

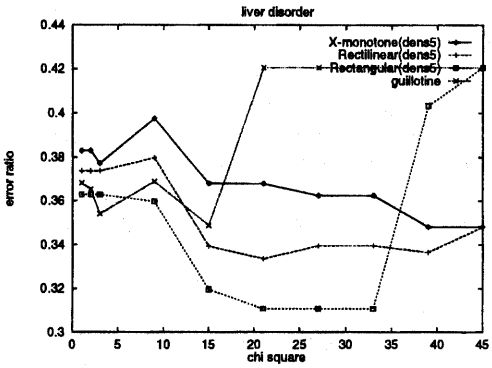

図 8: Error Ratios for "liver disorder"

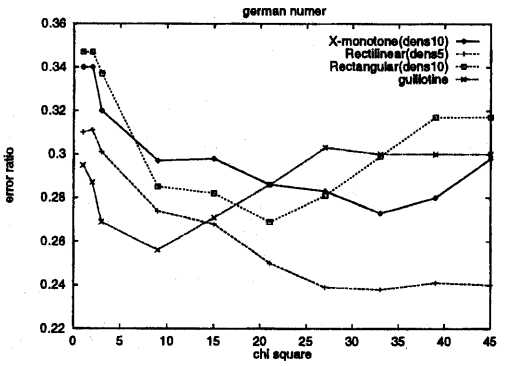

図 7: Error Ratios for "german credit"

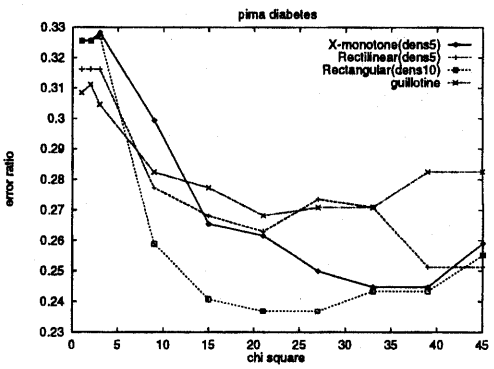

図 9: Error Ratios for "pima diabetes"

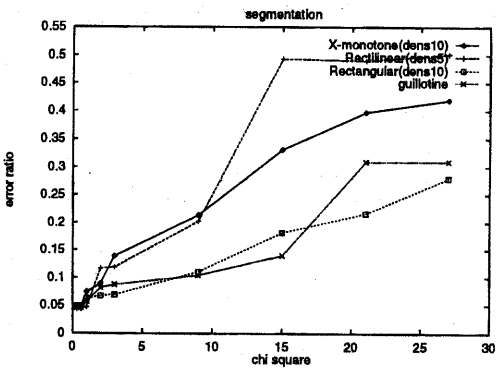

図 10: Error Ratios for "segmentation"

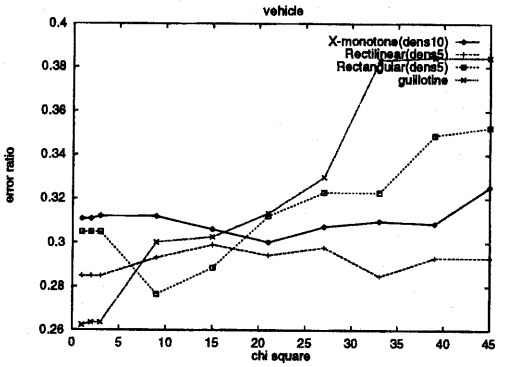

図 11: Error Ratios for "vehicle"

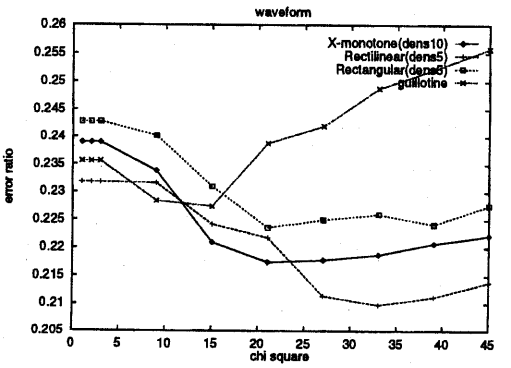

図 12: Error Ratios for "waveform" dataset

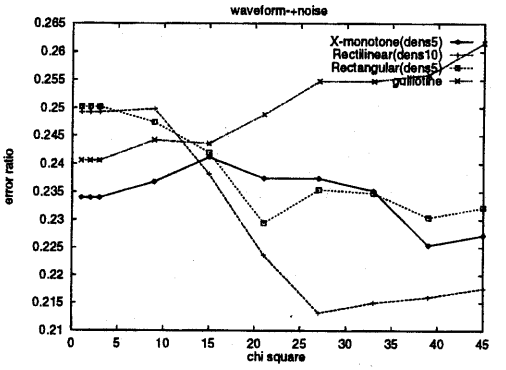

図 13: Error Ratios for "waveform-+noise"

	X-monotone		Rectilinear		Rectangular	
Size	sec.	#touch	sec.	#touch	sec.	#touch
10^2	0.08	26	0.05	24	0.01	16
20^2	0.36	27	0.30	24	0.05	23
30^2	1.03	32	1.30	31	0.14	25
40^2	1.78	33	3.19	30	0.37	29
50^2	2.83	34	6.97	31	0.73	30

表 4: Computing Optimal Region (1)

対する増加は非常に小さいことがわかり，我々の凸包検索が有効に機能していることがわかる．直交凸および長方領域の計算量は大きいが，X 単調に比べ定数分の計算時間が小さいため，サイズが小さい場合は X 単調より高速に計算できている．図 14から実質的な計算量評価が正しいことが確認できる．実際の決定木作成の際，ルートに近いノードでは大きなサイズの行列を対象とするが，分割が進むに従って，サイズは $N = 30$ 以下程度に小さくなるため，実質的な計算時間は，コストの大きな直交凸領域でもそれほど大きくはならない．

次に，データセットから領域分割を利用した決定木を構築するまでの全体の計算時間調べるため，タップル数，属性数など特徴の違うデータを用いて決定木を作成した．この実験では，精度の実験で利用したデータセット中でデータ数の最も大きい「waveform」を元に射影，選択に

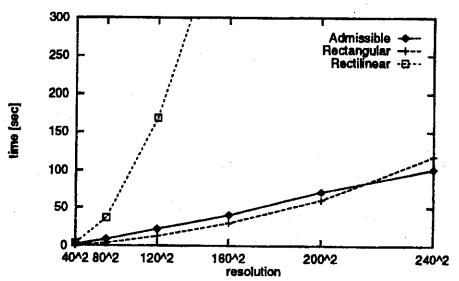

図 14: Computing Optimal Region (2)

#tuples	X-monotone	Rectilinear
1000	106	44
2000	310	134
3000	545	207
4000	764	285
5000	943	353

表 5: Tree Construction Time (1)

#attrs	X-monotone	Rectilinear
4	146	50
6	281	109
8	545	207
10	735	293
12	1117	383

表 6: Tree Construction Time (2)

より実験データを作成した.

表5に，タップル数の違うデータからの決定木構築に要した時間をまとめた．この実験では,「waveform」データからランダムに，1000から5000までのタップル数の違う5つの実験データを作り，その各実験データから決定木を作成する時間を計測した．また，決定木の述語属性として使用する数値属性は，実験の簡単化のため，最初の8属性のみとした．決定木の枝がりパラメータは精度の良かった$\chi^2 = 33$とし，領域のピクセル密度は5と固定した．一方，表6では，上記3000タップルの実験データから，最初のn個の属性を使用して決定木を作成し，nで与えられる属性数に対する計算時間の変化を計測した．

決定木構築の際，各ノードでは，まず領域の元となる行列を各数値属性の組み合わせ (X単調用には順列) に対して準備する．この行列の準備にはノード内の全タップルを1回スキャンする必要があるため，タップル数に対して決定木構築時間は (木の大きさの安定するタップル数以上の範囲で) ほぼ線形に増加する．また，各ノードで検査しなければならない行列数は，数値属性数が増加するとその組合せの数だけ増加してしまう．実験結果は，ほぼその影響を反映したものとなっている．

4 まとめ

領域分割を利用した決定木は精度，大きさともに従来手法に比べ優れていることがわかる．これは，データには本質的に何らかの相関が存在し，決定木を作成する際，それを考慮した方がよいことを示している．とくに，人間に認知できる大きさでの木で比較した場合，精度の差は図5から13からわかるとおり，非常にはっきりとあらわれている．領域分割は従来手法以上に，数値属性の組合せ数に比例した計算時間を必要とするが，多くのアプリケーションでは，数値属性の数が極端に多くない場合，これを考慮してなお有意義であると考える．さらに重要なことは，領域分割を利用することにより，従来手法では発見できない非線形な重要ルールを見つけ出すことが可能となったことである．

参考文献

[1] R. Agrawal, S. Ghosh, T. Imielinski, B. Iyer, and A. Swami. An interval classifier for database mining applications. In *Proceedings of the 18th VLDB Conference*, pages 560–573, 1992.

[2] R. Agrawal, T. Imielinski, and A. Swami. Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engineering*, 5(6):914–925, Dec. 1993.

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

[4] T. Asano and T. Tokuyama. Partial construction of an arrangement of lines and its application to optimal partitioning of bichromatic point set. *IEICE Transactions E*, 77:595–600, 1994.

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

[6] C. E. Brodley and P. E. Utgoff. Multivariate decision trees. *Machine Learning*, 19:45–77, 1995.

[7] D. Dobkin, H. Edelsbrunner, and C. Yap. Probing convex polytopes. In *Proc. 18th ACM Symposium on Theory of Computing*, pages 387–392, 1986.

[8] D. Dobkin and D. Eppstein. Computing the discrepancy. In *Proc. 9th ACM Symposium on Computational Geometry*, pages 47–52, 1993.

[9] 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*, pages 13–23, June 1996.

[10] 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*, pages 182–191, June 1996.

[11] M. R. Garey and D. S. Johnson. *Computer and Intractability. A Guide to NP-Completeness*. W. H. Freeman, 1979.

[12] L. Hyafil and R. L. Rivest. Constructing optimal binary decision trees is np-complete. *Information Processing Letters*, 5:15–17, 1976.

[13] 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.

[14] P. M. Murphy and D. W. Aha. *UCI Repository of Machine Learning databases*. [http://www.ics.uci.edu/~mlearn/MLRepository.html], Irvine, CA: University of California, Department of Information and Computer Science, 1994.

[15] J. R. Quinlan. The effect of noise on concept learning. *R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (eds.), Machine Learning An Artificial Intelligence Approach*, 2:149–166, 1986.

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