F-050

数値データベースに対するエキスパート付き決定木の構築
Construction of Expert Guided Decision Tree on Numeric Database

全 眞嬉[†] 金子 法正[§] 定兼 邦彦[‡] 徳山 豪[†]
Jinhee Chun Masanori Kaneko Kunihiko Sadakane Takeshi Tokuyama

1. はじめに

本論文では，予測精度の高い決定木を構築するための手法の提案を行う．決定木を構成する際の問題点として過学習がある．学習データに対して最適な決定木を用いると，未知データに対する予測精度が落ちてしまうといったトレードオフが生じる．そのため決定木の枝刈りが必要となるが，どのように枝刈りを行えば最適であるかはわからない．

予測精度の高い決定木を生成するためには，学習データの分割をできるだけ小さな誤差で行い，そして適切な枝刈りを行うことが必要である．

従来の数値データの 1 点分割手法は，分割時にデータの誤差問題がある．また，最適サポートルールの区間を用いた分割手法では，誤差の少ない区間分割は可能であるが，適切な枝刈りは困難である．

本論文では，分割時の誤差をより小さくするために，数値データを，[2] で提案された最適ピラミッド近似を利用して整理し，構築されたピラミッドをエキスパートとして用いた [3]，個々のデータに依存する柔軟な枝刈り手法を提案する．

これにより，最適な決定木が分からない場合でもそれに近い予測精度を得ることができる．

2. 数値属性ルール

結合ルールは「x ならば y である」(x:条件属性，y:目的属性）というデータベースのトランザクションに含まれた属性間の相関関係である．決定木とはデータベース中のある注目する属性（目的属性）に関する知識を木構造によるルールの組み合わせで表現したものである．学習データから抽出される目的属性の値を導く条件属性の結合ルールを各ノードの判定条件に利用して決定木を構築する．

データベースには離散値を持つカテゴリ属性データと連続値をもつ数値属性データがある．数値属性データを扱う場合は，数値データの 2 値化が必要である．ところが，数値属性の 2 値化で予測精度が悪くなる欠点がある．決定木による知識表現の場合は，2 値データに対しては良い予測精度をもつが，数値データに対してはうまくいかない．数値データの 2 値化には誤差が生じ，情報のロスがある．数値データに対して，分類精度を向上させるためには木を大きくすれば良いが，予測精度は低下していまう．

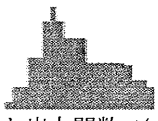

図 1: 入力関数 $f(x)$（左図）と出力関数 $\phi(x)$（右図）

本研究では，数値データの 2 値化時の誤差をより小さくするために数値属性データに対し階層構造であるピラミッド数値属性結合ルール [2] を利用して，構築されたピラミッドをエキスパートとして用いた，個々のデータに依存するより柔軟な枝刈り手法を提案する．

[†]東北大学大学院情報科学研究科 {jinhee,tokuyama}@dais.is.tohoku.ac.jp
[‡]九州大学大学院システム情報科学研究院 sada@csce.kyushu-u.ac.jp
[§]富士重工業株式会社 kaneko@dais.is.tohoku.ac.jp

3. 最適ピラミッド

本論文では，学習モデルとしてピラミッド関数（単峰な関数）を用いる．

定義 1 領域族 $\mathcal{F}$ に関するピラミッドとは，区間 $I=[0,1]$ から $\mathcal{F}$ への関数 P で，$t>t'$ のとき $P(t)\subset P(t')$ となるものである．

$\mathcal{F}$ の要素 R が領域ルール，t がルールの確信度に対応する．このような階層的な領域を用いることで，多くのモデルを表現できるが，過学習のおそれがある．そこで，領域族に制限を加えることで過学習を回避する．例えば，閉集合族に限定した場合，ピラミッド関数は単峰になる．

本論文では最適ピラミッドを用いたデータ学習アルゴリズムを提案する．

定義 2 関数 $f:R^d\mapsto R$ の，領域族 $\mathcal{F}$ に関する最適ピラミッドとは，$D(f,\phi)=\int_0^1(f(x)-\phi(x))^2dr$ を最小にするとき ϕ が f の最適ピラミッドである．

定理 1 *1* 次元最適ピラミッドは $O(n)$ 時間で計算できる．[2]

4. 最適ピラミッドを用いた決定木

本論文では最適ピラミッドを用いた決定木を提案する．ピラミッドだけでは任意の学習モデルを表現できないため，それらを組み合わせて決定木を作る．これは Morimoto らの 2 次元領域ルールを用いた決定木 [1] の拡張になっている．Morimoto らの決定木では，各ノードでの判定条件は $\hat{x}\in R$ という形をしている．R は 1 次元または 2 次元の領域で，x のある属性に対応している．$\hat{x}$ はその属性の値である．判定条件を満たすかどうかによって 2 通りに分岐し，決定木の葉まで到達すると Yes ($Y=1$) か No ($Y=0$) を返す．

4.1 決定木構成アルゴリズム

最適ピラミッドを用いた決定木の構成法を示す．

step1 学習データ T を読み込む．

step2 T のすべての事例が同一クラスならば葉とする．
そうでなければ，各属性に対して次の 1〜4 を行う．
1. 各属性値における確信度を求める．
2. 求めた確信度を元に最適ピラミッドを構築する．
3. 全ての高さに対して分割を行い，相互情報量を計算する．
4. 得られた各属性の相互情報量を比較する．
   (a) 閾値 g_ℓ 以上かつ最も大きな相互情報量が得られた属性があるならば，それをノードとし，T を T_1，T_2 に分割する．同時に，T_1，T_2 の確信度を求めておく．
   (b) もし，どの属性の相互情報量も g_ℓ 未満であるならば，T に最も多く現れるクラスラベルの葉とする．

step3 分割された事例集合に対し，全てが葉になるまで**step1**, **step2** を繰り返す．

生成された決定木の各ノードでは分割テストが行われる．例えば，根である属性 A テストは，全学習データ T の，属性 A について求められた確信度のヒストグラム f を持つ．また，f から求められた最適ピラミッド ϕ，そして，分割された T_1，T_2 の確信度 $conf(T_1)$，$conf(T_2)$ をそれぞれ持っている．この確信度はテストデータが分類され，枝を通るときに各テスト事例に対して与えられる．分割された事例集合 T_1，T_2 は，子ノードに入力され，子ノードも同様な構造を持つ．

また，決定木の末端である葉には，クラスラベルが与えられる．その葉に振り分けられた学習データの事例数 $|T_{11}|$，その中でクラスが C でない事例数 $|E_{11}|$ (エラー数) も記録してある．

4.2 エキスパート付き決定木

エキスパートは，決定木構築アルゴリズムの **step2**.4 の後に生成され，該当ノードにおける学習データ数 $|T|$，及び，T より構築された最適ピラミッド関数 $\phi(T)$ を持つ．エキスパートは決定木の各ノードに配置される．入力されたデータ i に対し，ノード A のエキスパートは，学習データ数 $|T|$，及び $\phi(T)$ におけるデータ i の確信度 $\alpha_T(i)$，を与える．そして，与えられた確信度と事例数のセットにより，データ i のクラスを決定できるかどうか判断する．あるノード A の情報の信頼性が大きいと判断し，そこでデータ i にラベルを与え，テストを終了する判断を行う．即ち，ノード A, B 間で枝刈りを行うことになる．ここで，比較に用いた $\alpha_T(i)$ の値は各データによって値が異なる．つまり，エキスパートは，個々のデータに依存した，データことに柔軟かつインプリシットに決定木の枝刈りを行う．

5. 実験

本実験では，C4.5 が生成した枝刈り前の決定木の各ノードに最適ピラミッドを構築し，エキスパートを配置した．各ノードのエキスパートは，テストデータに対し，個々のデータに依存する確信度を与える．その確信度がユーザの指定した範囲にあれば，その時点でラベルを与え，分類を終了し，次のデータの分類へと進む．

実験データは UCI で提供する Diabetes データセット (http://www.ics.uci.edu/ mlearn/MLRepository.html) で，糖尿病診断データである．クラスは健康 (0) または糖尿病 (1) の 2 値属性であり，全事例は 768 で，500 事例が健康，268 事例が糖尿病のデータである．属性数は 8 で，クラス数は 2 である．本実験では，数値属性データのみを実験対象にした．

5.1 実験方法

10-cross-validation 法を用いて実験を行った．まず，全データをシャッフルし，10 個のデータ集合に分割し，その 1 つをテストデータとし，残りを学習データとする．次に，学習データを元に決定木を構築し，テストデータを入力する．入力された個々のデータはそれぞれ，通るノードでクラス 1 に対する確信度を得る．その確信度が，ユーザの設定した閾値 α 以下または β 以上であればそこで分類をストップさせ，それぞれ 0, 1 をそのデータのクラスラベルとする．その他の場合は，分類を続行させ，ノードであれば上記と同様な処理を行い，データが葉に辿り付いた場合は，その葉に記述されているラベルを，データのラベルとする．予測誤りの合計数を E，全事例数を N とした時，E/N をテストデータの誤分類率とする．

以上の操作を分割した 10 個のデータ集合全てに対して行い，得られたそれぞれの誤分類率の平均値を，最終的なシステムの誤分類率として評価する．

5.2 実験結果

本実験では，$\alpha + \beta = 1$ とし，$(\alpha, \beta) \in \{(0, 1), (0.5, 9.5), (1.0, 9.0), (1.5, 8.5), (2.0, 8.0), (2.5, 7.5), (3.0, 7.0), (3.5, 6.5), (4.0, 6.0)\}$ とした．また，得られた結果を，C4.5 の枝刈り前，枝刈り後の結果と比較した．

表 1: 出力された決定木のサイズ

条件	木の大きさ	木の最大深さ	誤差分類率
枝刈り前	54.4	11.5	0.269
枝刈り後	46.4	10.5	0.265

最適ピラミッドを構築する際に，入力の $f(x)$ のヒストグラムの棒の個数 (バケット数)，各バケットの属性値幅，各バケットに入るデータ数 (サポート)，サポートの重みのが各異なる equi-depth,const-dist,const-dist-sup,dyn-bucket,dyn-bucket-sup の 5 通りの入力関数 $f(x)$ の構成手法で実験を行なった．

その結果を表 1 と図 2 に示す．equi-depth，const-dist, const-dist-sup では，β の値が 0.80 までは C4.5 と比べて良い精度を示しているが，それ以上になるとノード決定率が急激に下がり始め，C4.5 に近づくように誤分類率が増加している．特に，equi-depth 法での精度は著しく低下している．一方，dyn-bucket, dyn-bucket-sup では，β を 0.90 まで増加させても良い精度を示した．これは，各ノードにおいて，データ数に応じて動的にバケット数を決定することで，信頼性の高いエキスパートを生成できたためと考えられる．

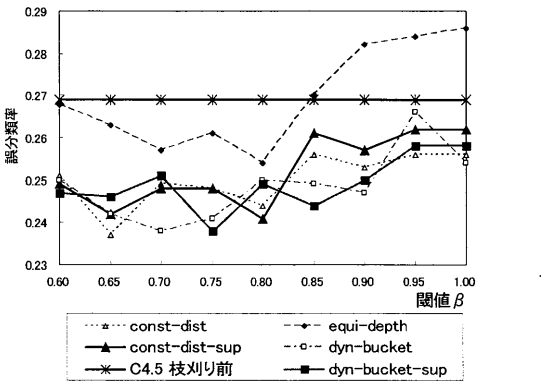

図 2: 誤差分類比較

6. まとめ

本論文では，最適ピラミッドを用いてエキスパート付き決定木の提案を行った．実験により，C4.5 の生成した決定木の各ノードに最適ピラミッドのエキスパートを用いて，適切な閾値を与えれば，より少ないテストで，精度の良いデータ分類を行えることがわかった．

参考文献

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

[2] J.Chun, K.Sadakane, and T.Tokuyama. Linear time algorithm for approximating a curve by a single-peaked curve. *Proc. 14th ISAAC, LNCS 2906* (2003), 6-16.

[3] 金子法正．決定木を用いたデータマイニングシステムの構築の研究，修士学位論文，東北大学，(2004)