正確な学習よりも得する学習
ー誤分類コストを考慮する分類学習ー

(2) 解決編

鈴木 英之進
横浜国立大学大学院工学研究院
suzuki@ynu.ac.jp

はじめに

前回紹介した，クレジットカードの不正利用を警告するシステムの開発担当者は，使った決定木学習法が失敗する理由を，通常利用の例が不正利用の例に比べて圧倒的に多いためであると考えた．このことより，彼は決定木学習法を適用するデータを前処理することにした．方法は3通りあり，1) 通常利用に属す例集合の一部を取り除く，2)不正利用に属す例を複製する，3)例をクラスに応じて重み付けして用いる．この開発担当者は几帳面であり，パラメータをいくつか設定していろいろな値を試し，たくさんの分類モデル，すなわち決定木を学習で得た．得た決定木は，前回紹介した誤分類コストの評価法の1つである期待誤分類コストで優劣をつけた．このようにして最良と認められた決定木は，比較的単純であり，期待誤分類コストが低かった．ただし，この開発担当者は2つの間違いを犯しており，この決定木が実は性能が悪いことに気づく由もなかった．

片方の間違いは本稿の最後に述べるとするが，もう片方は単純である．誤分類コストが小さい分類モデルを構築する学習手法は近年盛んに研究されており，この開発担当者はこれらの手法を試すべきだった．前回述べたように，従来の分類学習手法は，多数派クラスを優遇するバイアスがかかっている上にクラスiの例をクラスjと予測する誤分類コスト$C(j|i)$をクラスi, jに無関係で一定としており，クレジットカードの不正利用予測問題などには不適切である．言い替えれば，これらの手法は，誤分類コストがクラスに依存しない0/1損失関数を仮定しており，一般的な損失関数を用いる問題では性能が悪い．この種の問題は，上記のように場当たり的な方法でも対処できるが，せっかく研究・提案されて有効性が示されている手法を試さないのは残念である．今回は解決編と題し，クラス当たりの例数を変更するサンプリングに基づく手法，例のクラスを書き換える方法，および学習アルゴリズムを適切に変更する方法に分けて，近年の研究成果を解説する．

サンプリングによるアプローチ

■サンプリングの効果

ここでのサンプリングとは，例数を増やしたり減らしたりする前処理を表し，学習問題によっては必要となる場合もある．たとえば，データを得るのが困難な場合，少数派クラスの例を実際よりも多く計測することがある．逆に，例がきわめて多い場合，保存コストや処理コストを考慮して，多数派クラスの例を削除することもある．手元にデータ集合がある場合，少数派クラスの例を複製することをオーバーサンプリング，多数派クラスの例を削除することをアンダーサンプリングと呼ぶ．前者は訓練データを大きくしてしまう上に過学習の危険性があり，後者は有用な例を捨ててしまう危険性がある．ただし，両者ともクラス分布が偏っていたり，誤分類コストがクラスに依存する分類学習において，有効となる場合がある．

前章における問題点は，サンプリングの基準がクラスだけに依存することである．クラスだけに基づくサンプリングの効果を，決定木学習について25種類のデータ

集合を用いた実験で調べた研究が存在する[10]．重要な結果として，評価指標が前回説明した正答率かAUC☆1の場合，分類学習に最良のクラス分布が本来の分布と異なるデータ集合が，それぞれ8個と13個あることが挙げられる☆2．分類学習では，本来のクラス分布を用いた学習結果，すなわち訓練データのクラス分布がテストデータのクラス分布と一致する場合の学習結果が最良であると思いがちである．手元のデータからその背景にある対象概念を推定する問題はそれほど単純ではなく，実験結果もこの思い込みが正しいわけではないことを示している．多くのデータ集合に共通する「推奨」クラス分布など，学習法を固定しても存在するわけがない．良いサンプリングは，分類学習問題ごとに異なり，簡単に得られるものではない．

クラスだけではなく誤分類コストに応じてサンプリングすればよいのではという考え方もある．たしかに，クラス数が3以上の場合でも，$C(j|i)=C(\cdot|i)$であれば妥当な割合が求められる．もっとも，クラスだけに基づくサンプリングは，対象概念を考慮しないので限界があると考えられる．

■クラス分布推定を用いる方法

そもそも，与えられた訓練データから手元にないテストデータのクラスを予測する分類学習は困難なタスクであり，クラス比を変更してうまくいく場合に限りがある．学習問題によっては，対象概念の構造，すなわち分類学習においては分類境界面を推定するなど，より適切な手続きを用いるべきである．

距離に基づくアンダーサンプリング[7]は，この考えに挑戦した手法である．著者らは，衛星写真から油田を発見する分類学習問題に取り組んだ際の経験を活かし，正例が負例に比較してきわめて少ない場合に有効な前処理手法を提案した．もっとも誤分類コストの非対称性は意識されているが明示的には扱われていない．前回述べたように，ROC (Receiver Operating Characteristics Curve) 曲線分析を用いる場合には，クラス割合と誤分類コストは学習結果の評価に同じ影響を及ぼせるが，この手法を実際の応用問題に適用するにあたっては修正が必要である．

この手法の対象は，属性がすべて連続値属性である2クラス分類問題である．対象問題においては，属性値やクラスが実際とは異なる値になってしまうノイズが存在すると仮定している．たとえば，**図-1**(a)のように正例（十字架）と負例（黒丸）が例空間に分布しており，真

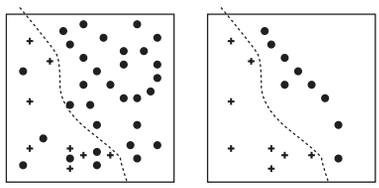


図-1　距離に基づくアンダーサンプリング[7]の発想．元のデータ集合(a)をアンダーサンプリングしたデータ集合(b)に変換し，点線で表される分類境界面を推定しやすくする．

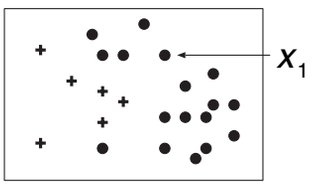
の分類境界面（学習者には未知）が点線であるとする．負例は正例に比較してきわめて多いため，例数が増えるにつれてノイズに犯された負例の影響が大きくなり，最小近傍法 (nearest neighbor method)，素朴ベイズ法 (Naive Bayes method)，および決定木学習法など，従来の分類学習手法をそのまま適用した場合，主に負例だけを予測する分類モデルが学習されてしまう危険性が高いことが直観的に分かる．

この前処理手法は，この問題に対し，代表的な負例だけを選択するアンダーサンプリングで対処する．図の例では，(a) における例をノイズ例，境界例，冗長例，および代表例に分け，すべての正例と負例の代表例だけを残そうとすることで，(b) の形式に変換する．ノイズ例，境界例，冗長例，および代表例はそれぞれ，ノイズによりクラスや属性値が変わってしまった例，分類境界面付近にある例，なくても分類境界面の特定に影響しない例，および分類学習に有用な例を表す．この研究では，境界例はノイズによって属性値が少し変化している場合，学習に悪影響を及ぼしやすいので除くべきと見なされている．境界例とノイズ例を特定するためには，Tomekリンクと呼ばれる概念を用いる．例$\mathbf{x}$と例$\mathbf{y}$間の距離を$d(\mathbf{x}, \mathbf{y})$で表す．クラスが異なる例$\mathbf{x}$と$\mathbf{y}$に関して，$d(\mathbf{x}, \mathbf{z})<d(\mathbf{x}, \mathbf{y})$か$d(\mathbf{y}, \mathbf{z})<d(\mathbf{x}, \mathbf{y})$を満たす例$\mathbf{z}$が存在しなければ，$(\mathbf{x}, \mathbf{y})$はTomekリンクに属すと定義される．Tomekリンクに属す例は，ノイズ例か境界例のどちらかである．アルゴリズムは，元々最小近傍法において保存事例を減らすために提案された手法に着想を得ており，次に示す通りである．

手続き：アンダーサンプリングに基づく前処理手法[7]
入力：訓練データS
返り値：アンダーサンプリングされたデータT

1　$A=$（S中のすべての正例とランダムに選択した負例1個の集合）
2　$B=A\cup$（SをAから学習した1-NNで分類した際に誤

☆1　AUCは，ROC曲線$g(x)$下の面積$\int_0^1 g(x)\,dx$を表す．
☆2　これらは，統計的有意性を考慮した実験結果である．



① バギングを使って各例 x について所属クラスを推定

たとえば,

正例である確率 $\hat{P}(+|x_1) = 0.4$

負例である確率 $\hat{P}(-|x_1) = 0.6$

② 右を各例について行う　たとえば, x_1 は正例と予測した方が期待誤分類コストは小さくなる ⟶ x_1 のクラスを正に書き換える

③ できたデータ集合から分類モデルを学習する

図-2　MetaCostの発想．ただし十字架は正例，黒丸は負例を表す

予測した例集合)

3　T= (BからTomekリンクに属すすべての負例を除いた例集合)

属性が多くなると，例空間において例がまばらになり，すべての例同士の距離がほぼ同じになり，学習が困難になることが知られている．このような場合，通常用いられる距離尺度 (distance measure) はしばしば直観に反する値となる上，ノイズの影響を受けやすい．この前処理手法は，属性が少なく対象概念が比較的単純である場合には有望であると考えられる．

MetaCost：クラス書き換えによるアプローチ

MetaCost[3] は，分類学習手法を誤分類コストを考慮するように変更するのではなく，学習データ中のクラスを書き換えて分類学習手法はそのまま使うという発想に基づいている．すなわち，分類学習手法をブラックボックスと見なして外側にメタ学習モジュールをかぶせるため，従来の分類学習手法を容易に用いることができる．

MetaCostを説明する前に，誤分類コストを考慮する分類学習の基礎をまとめておく．条件つきリスク$R(i|x)$は，例xがクラスiに属すと予測する際の期待誤分類コストであり，小さい方が望ましい．

$$R(i|x)=\sum_j P(j|x)C(i|j). \qquad (1)$$

xのベイズ最適予測は，$R(i|x)$を最小とするiであり，予測クラスとして望ましい．ベイズ最適予測は，期待誤分類コスト，すなわちすべてのxに対し$P(x)$で重み付けした値$\sum_x P(x)\ R(f(x)|x)$を最小化する予測である．ベイズ最適予測により例空間は，最適予測となるクラスに応じて互いに排反な領域に分割できる．誤分類コストを考慮する分類学習の目的は，この領域間の境界を見つけることに等しい．

前回述べたように，コスト行列によっては，真のクラスを予測しない方がよいことがある．たとえば信販会社は，クレジットカードでの支払いがほぼ合法的と見なせる場合でも，高額なために拒否する場合がある．我々はベイズ最適予測が分からないため，訓練データ中の例についてさえ，最適な予測クラスを知らないのである．MetaCostは，訓練データに属す例のクラスをコスト行列に基づいて書き換え，最適な予測境界を学習しようとする．

一番の問題は，$P(j|x)$が分からないので，これを推定する方法が必要なことである．MetaCostでは，ユーザが問題に適する分類学習手法を選択すると仮定し，この用途にバギング (bagging) を用いて推定値$\hat{P}(j|x)$を求める．MetaCostの発想を**図-2** に示す．バギングは，分類学習手法を複数個の例集合サンプルに適用して複数個の分類モデルを得，これらの投票でクラスを予測することで高い正答率を達成しようとする手法である．ただし用いる分類学習手法は，決定木学習手法などのように訓練データが少し異なるだけで大きく異なる分類モデルを出力するものに限る．複数個の分類モデルを同時に用いる学習は，アンサンブル (ensemble) 学習とも呼ばれる．訓練データの例数をsとすると，通常のバギングでは，s個の例を復元抽出して1個の分類モデルを得る．MetaCostでは学習時間の短縮を重視し，sより小さいn例を復元抽出する．さらにMetaCostは，分類学習手法が$P(j|x)$の推定値$\hat{P}(j|x)$を出力する場合，この使用をフラグpで指定することもできる．予測例xも再サンプルの対象にすることは功罪両面があるため，フラグqを使って選べるようになっている．

手続き：*MetaCost*

入力：訓練データS，分類学習手法L，コスト行列C，生成する再サンプル数m，各再サンプルの例数n，Lが生成する$\hat{P}(j|x)$を利用するか否かのフラグp，予測例xも再サンプルの対象にするか否かのフラグq

返り値：分類モデルM

1　**For**(再サンプル番号i) **from** 1 **to** m **Do**

2　Sから例数nの再サンプルS_iを生成
3　S_iにLを適用して分類モデルM_iを得る
4 **Foreach** S中の例 x **Do**
5　**Foreach** クラス j **Do**
6　$\hat{P}(j|x)=\frac{1}{\sum_i 1}\sum_i \hat{P}(j|x, M_i)$

ただし
If p **then** M_iで$\hat{P}(j|x, M_i)$を求める
Else M_iがxのクラスとして予測したjについては$\hat{P}(j|x, M_i)=1$，その他のjについては$\hat{P}(j|x, M_i)=0$
If q **then** iはすべてのM_iに関する
Else iは$x \notin S_i$であるM_iに関する

7　(xのクラス)$=\mathrm{argmin}_i \sum_j \hat{P}(j|x)\, C(i|j)$
8　SにLを適用してMを得る

MetaCostは，後で解説する誤分類コストを考慮するアンサンブル学習手法とは異なり，分類モデルが1個であるため可読性が良い．さらにコスト行列が変更されても$\hat{P}(j|x)$が使えるため，アルゴリズムの最後(7と8)だけをやり直せばよい．実験の結果，MetaCostは期待誤分類コストが低い分類モデルを高速に学習でき，複数個の分類モデルを平均化することによりノイズを除く効果があるので，ノイズに犯されたデータに強いことが分かった．

なお，MetaCostを他の誤分類コストを考慮する分類学習手法と実験で比較し，MetaCostの期待誤分類コストは高く，結果の可読性が唯一の取り柄であると結論付けた研究が存在する[9]．この種の実験的報告は，原論文からの実装の変更，実験条件，および評価方法などに注意して参考にすべきである．筆者は，分類学習手法の評価指標として，複数個のデータ集合についての期待誤分類コストの平均値は意味がないと考えており，単純な勝敗数も誤った結論につながりがちであると思っている☆3．実験的結果は，納得できる理由とともに示されてこそ，経験的事実として認知される．

アルゴリズム変更によるアプローチ

■決定木に基づく方法

決定木は，人工知能においては一般に，**図-3** に示すような木構造形式の分類モデルを指す．図の決定木の各内部ノードには属性test1とweightが，各葉ノードにはクラスTかFが，各エッジには属性値に関する条件が割り当てられている．たとえばtest1=Y, weight=39である例のクラスは，この決定木を用いるとTと予測される．決定木学習手法は通常，階層的な分類手続きを出力するために結果の可読性に優れ，ノンパラメトリック手法に属すため特定のデータ分布を仮定することなく適用でき，属性選択を行うため不要属性を比較的気にしなくても済むなどの利点がある．このため，最も頻繁に用いられる機械学習手法といっても過言ではないと思われる．実際，決定木は，工業，商業，農業，科学，医学，法学など種々の領域において用いられている．

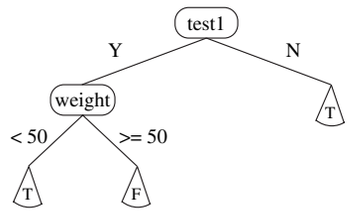


図-3　決定木の具体例

決定木は通常，まず根から順に大きな決定木を構築し，次に予測に有害と思われる部分木を葉ノードで置き換える枝刈り行うことで学習される．決定木の構築は，評価基準に基づく最良属性の選択を再帰的に行う貪欲法によって行われることが一般的である．評価基準としては，情報利得 $i(v)-\sum_{i=1}^{m_v} P'(v_i|v)i(v_i)$ などが知られている．ただしm_vはvの子ノード数，$P'(v_i|v)$は内部ノードvにある例が子ノードv_iに割り当てられる確率を表し，$i(v')=-\sum_j P_{v'}(j)\log P_{v'}(j)$，$P_v(j)$は内部ノード$v$におけるクラス$j$の確率である．

決定木学習手法を誤分類コストを考慮する分類学習用に拡張する研究はいくつかあり，興味深い知見が実験により得られている．Laplace修正に基づく枝刈りを提案し，頻度に基づく枝刈りや枝刈りを行わない手法と比較した研究[1]も，その代表例である．Laplace修正によれば，決定木の葉ノードvにおけるクラスiの推定生起確率$\hat{P}_v(i)$は，クラスがk個ありm_v例中$m_{v,i}$個がiに属す場合，

$$\hat{P}_v(i)=\frac{m_{v,i}+1}{m_v+k}. \qquad (2)$$

この研究では，枝刈りを行わない手法の期待誤分類コストとLaplace修正を用いた各種枝刈り手法の期待誤分類コストが，同程度であることが実験により示されている．この実験では学習時にコスト行列とクラス分布が分かっている場合を対象としているが，1）では分かっていない場合に ROC 曲線分析を用いた結果も同様である

☆3 実験で用いたデータ集合が他のデータ集合も代表するという考えは危険である．最善は学習手法が適するデータ集合の特徴を述べることであり，次善は極端な結論を導かないことである．

と述べている．

そのほか，決定木の分割テストにおいて，誤分類コストとクラス分布をまったく考慮しない評価規準が，これらを考慮する手法よりも良い可能性があるとの実験結果[5]が注目を浴びている．この評価規準は，上記のエントロピー関数$i(v')$を，クラス数2の場合，$2\sqrt{P_{v'}(+)P_{v'}(-)}$で置き換えたものである．ただしこれらの実験的評価の欠点として，得られた知見について納得できる理由を述べていないことがあげられる．確実に分かっていることとして，従来の決定木学習において用いられてきた枝刈りなどの手法は，正答率の向上と決定木サイズの減少には適しているが期待誤分類コストの低下などには適していない場合があることと，最良の手法はコスト行列，対象概念，および訓練データに依存することである．最近，これらの知見がクラスへの所属確率に応じて例を順位付ける問題についてもほぼ当てはまることが示された[8]．筆者は，正答率の向上を目的とする分類学習問題における知見が，より広い学習問題に関して一般化されていくと期待している．

■ブースティングに基づく方法

ブースティング (boosting) は，MetaCostの章で述べたアンサンブル学習手法の一種であり，学習アルゴリズムの正答率を増やす (boost) ことから，こう名付けられている．AdaBoostは，ブースティングの代表的なアルゴリズムであり，クラスがノイズに犯されていないデータに関して実験結果が良く，テストデータに関する正答率が高いことが証明されている．AdaBoostを，1990年代における機械学習の最高成果と言う人もいる．

AdaBoostは，各訓練例に重みをつけて「弱仮説」と呼ばれる比較的単純な分類学習モデルを学習することをT回行い，得られたT個の弱仮説を重み付けした線形和を，学習された分類モデルとする☆4．各回はラウンドと呼ばれ，そのラウンドで得られた弱仮説で各訓練例のクラスを予測し，実際のクラスと合えば次のラウンドにおいてその訓練例の重みを小さくし，逆であれば大きくする．各訓練例の重みは総和が1になるように決められ，各ラウンドでの弱仮説の学習を「確率分布の元で行う」とも表現する．**図-4**にAdaBoostの実行例を示す．図では各ラウンドにおいてクラス予測を間違えた例が丸で囲って示されており，これらの例についての重みは増やされる．各ラウンドでは弱仮説の重みが求められ，最終的な分類モデルは各弱仮説に該当する重みを乗じた線形和で表される．

AdaCost[6]は，AdaBoostを誤分類コストを考慮するように一般化したアルゴリズムであり，実験における性能が良いだけではなく，以下に述べるように理論的な裏付けもある．一般化とは，コスト修正関数βを導入して誤分類コストが高い例の重みを大きくしたことである．なおAdaCostは2クラス分類学習問題だけを対象としているが，誤分類コストが各例に依存する問題にも適用できる．

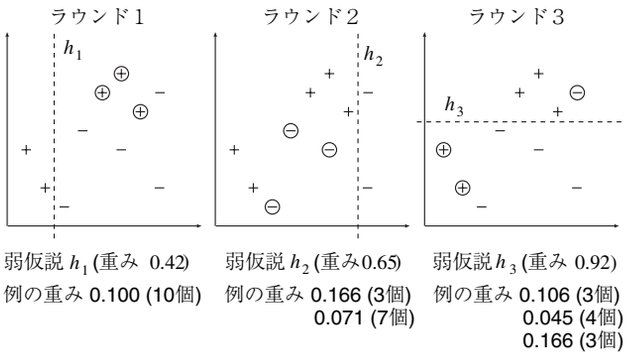


図-4　AdaBoostの実行例．ただし弱仮説h_1, h_2, h_3はそれぞれ，点線の両側をそれぞれの多数派クラスと予測する．

AdaCostのアルゴリズムを次に示す．ただしコスト修正関数$\beta(\mathrm{sign}(y_i h_t(x_i)), c_i)$を，前後から明らかな場合は$\beta(i)$や$\beta(c_i)$と書く場合がある．

手続き：*AdaCost*

入力：訓練データ$S=\{(x_1, c_1, y_1), (x_2, c_2, y_2), \ldots, (x_m, c_m, y_m)\}$ただし$x_i$, c_i, y_iは例iのそれぞれ属性値群，誤分類コスト，クラスラベル(+1か−1)

返り値：分類モデル$H(x)$

1　$D_1(i)$を初期化（たとえば$D_1(i)=c_i/\sum_{j=1}^{m} c_j$）

2　**For**(ラウンド番号t) **from** 1 **to** T **Do**

3　　確率分布D_tの元で弱仮説h_tを学習する

4　　実数値α_tと非負の実数値$\beta(i)$を選択

5　　$D_{t+1}(i)=D_t(i)\exp(-\alpha_t y_i h_t(x_i)\,\beta(i)/Z_t)$
　　ただし$\beta(i)=\beta(\mathrm{sign}(y_i h_t(x_i)), c_i)$はコスト修正関数．$Z_t$は$D_{t+1}$が分布をなすように選ばれた正規化要素

6　$H(x)=\mathrm{sign}(f(x))$ ただし$f(x)=\sum_{t=1}^{T} \alpha_t h_t(x)$

コスト修正関数$\beta(\mathrm{sign}(y_i h_t(x_i)), c_i)$を設定する際には，$x_i$の重みがコストの高低に応じて適切に更新されるようにすべきである．よって，$\beta(c_i)\geq 0$とし，$h_t(x_i)$のx_iについての予測が間違いすなわち$\mathrm{sign}(y_i h_t(x_i))=-1$の場合，$\beta(c_i)$は$c_i$に関して減少しないように設定すべきである．同様に$h_t(x_i)$の$x_i$についての予測が正しいすなわち$\mathrm{sign}(y_i h_t(x_i))=+1$の場合，$\beta(c_i)$は$c_i$に関して増加しないように設定する．

☆4 例の重みも弱仮説の重みも，直観的には重要度を表す．

AdaCostにおいて，訓練データについての誤分類コストに上限値があることが証明できる．s. t. で「を満たす」(such that)を表すと，

$$\sum_{i \text{ s.t.} H(x_i)\neq y_i} c_i \leq \sum c_j \prod_{t=1}^{T} Z_t \qquad (3)$$

よって，訓練データについての誤分類コストの上限値を抑えるためには，各ラウンドtにおいて，α_tをうまく選んでZ_tを最小化するように努める必要がある．α_tの選び方は2通り提案されている．1通り目は解析的な方法であり，

$$\alpha=\frac{1}{2}\ln\frac{1+r}{1-r} \qquad (4)$$

$$\text{where } r=\sum D(i)u_i,\ u_i=y_i h(x_i)\beta(i)$$

この場合，$Z\leq\sqrt{1-r^2}$ となることが示せる．2通り目は数値的な方法であり，Zの上限値としてはより厳密である．すなわち，αを次式の解とする☆5．

$$\frac{dZ}{d\alpha}=-\sum_i D(i)u_i\exp(-\alpha u_i)=0 \qquad (5)$$

実際には，最初の方法でαの初期値を定め，2番目の方法に基づき値を良くしていくのが現実的である．AdaCostの期待誤分類コストが高いことを示す実験結果も存在するが，MetaCostの章の最後と同様の指摘が当てはまる．AdaBoostにはいくつかのバージョンがあるため，AdaCost原論文のアルゴリズムを用いて比較すべきである．

SMOTEBoost[2]は，ブースティングを誤分類コストを考慮する分類学習用に変更するだけではなく，さらに一歩踏み込んで，少数派クラスに属す例を補間によって生成する．逆の見方をすると，単に少数派クラスに属す例を生成するだけでは，少数派クラスの予想が正確になるものの全体データに関する正確性が損なわれるので，ブースティングを用いている．この手法は，生成数のパラメータを指定するに当たって注意が必要だが，AdaCostよりも前回紹介したF値が良いことが実験で示されている．

おわりに

筆者らは，慢性肝炎患者データから肝硬変患者を予測する問題に取り組んでいる[11]が，これは誤分類コストを考慮する分類学習問題に属する．この問題では，コスト行列やクラス分布はもちろん未知であり，重症患者ほど頻繁に計測されているためデータが多いという問題を抱えている．評価法として，ROC曲線やコスト曲線が決定版というわけではなく，学習法としてもラプラス修正が決定木学習法の期待誤分類コストを高くする場合がある．もっとも，2回の解説論文で紹介してきたように，この研究分野では重要な発表が相次いでいる．たとえば，決定木の枝刈りが期待誤分類コストを増加しがちなど，限定された状況に関してではあるが傾向や特性が分かってきた．この分野がさらに発展し，一般的な損失関数を仮定する場合の学習理論が構築されることを切に願う．

分類学習は，手元にある訓練データから手元にないテストデータのクラスを予測する難しさがある．直観的には妥当に見える方法が，他の要因により失敗することもしばしば起こる．冒頭の開発担当者は，分類モデルの候補を多数調べると，訓練データを偶然うまく説明する分類モデルを見つけてしまうという過探索[4]の失敗も犯している．分類学習問題の奥深さに触れてこの分野を志す学生や研究者が出れば望外の喜びである．

謝辞　本研究の一部は，文部科学省科学研究費特定領域研究「アクティブマイニング」の援助を受けている．IBMの鹿島久嗣氏たちから有益なコメントを得た．記して感謝する．

参考文献

1) Bradford, J. P., Kunz, C., Kohavi, R., Brunk, C. and Brodley, C. E.: Pruning Decision Trees with Misclassification Costs, Proc. Tenth European Conf. on Machine Learning (ECML), pp.131-136 (1998).
2) Chawla, N. V., Lazarevic, A., Hall, L. O. and Bowyer, K. W.: SMOTEBoost: Improving Prediction of the Minority Class in Boosting, Principles of Data Mining and Knowledge Discovery, LNAI 2838 (PKDD), Springer-Verlag, pp.107-119 (2003).
3) Domingos, P.: MetaCost: A General Method for Making Classifiers Cost-Sensitive, Proc. Fifth Intl. Conf. on Knowledge Discovery and Data Mining (KDD), pp.155-164 (1999).
4) Domingos, P.: Process-Oriented Estimation of Generalization Error, Proc. Sixteenth Intl. Joint Conf. on Artificial Intelligence(IJCAI), pp.714-721 (1999).
5) Drummond, C. and Holte, R. C.: Exploiting the Cost (In)sensitivity of Decision Tree Splitting Criteria, Proc. Seventeenth Intl. Conf. on Machine Learning (ICML), pp.239-246 (2000).
6) Fan, W., Stolfo, S. J., Zhang, J. and Chan, P. K.: AdaCost: Misclassification Cost-sensitive Boosting, Proc. Sixteenth Intl. Conf. on Machine Learning (ICML), pp.97-105 (1999).
7) Kubat, M. and Matwin, S.: Addressing the Curse of Imbalanced Training Sets: One-Sided Selection, Proc. Fourteenth Intl. Conf. on Machine Learning (ICML), pp.179-186 (1997).
8) Provost, F. and Domingos, P.: Tree Induction for Probability-Based Ranking, Machine Learning, Vol.52, pp.199-215 (2003).
9) Ting, K. M.: An Empirical Study of MetaCost Using Boosting Algorithms, Proc. Twelfth European Conf. on Machine Learning (ECML), pp.413-425 (2000).
10) Weiss, G. M. and Provost, F.: The Effect of Class Distribution on Classifier Learning: An Empirical Study, Tech. Rep. ML-TR-44, Dept. Computer Sci., Rutgers Univ. (2001).
11) Yamada, Y., Suzuki, E., Yokoi, H. and Takabayashi, K.: Decision-tree Induction from Time-series Data Based on a Standard-example Split Test, Proc. Twentieth Intl. Conf. on Machine Learning (ICML), pp.840-847 (2003) (erratum http://www.slab.dnj.ynu.ac.jp/erratumicml2003.pdf).

(平成15年12月8日受付)

☆5　解が存在することは保証されている．