属性の投票による分類の一手法

3AG－7

渡辺 博芳† 荒井 正之† 奥田 健三‡
†帝京大学 ‡作新学院大学

1. はじめに

例からの学習において，決定木 (Decision Tree) や実例に基づく学習 (Instance-Based Learning) などの手法の他に，属性の投票によるアプローチが考えられる．すなわち，各属性ごとに概念記述を学習し，分類において各属性の分類結果に基づいた投票を行い，最も高い得票を獲得したクラスを解とする方法である．CFP(Classification by Feature Partitioning)[1] はこのようなアプローチの１つであると考えられる．本稿では，数値属性をもつデータに対する属性の投票による分類の一手法を提案する．本手法は概念記述として，属性ごとに各クラスの下限値，上限値，属性値の分布を保持する．分類においては問題として与えられた例の近傍$\pm\Delta_f$に存在する例の数に基づく評価値を各クラスに投票する．パラメータΔ_fは GA を用いて属性ごとに決める．

2. 各属性の概念記述

CFP[1] では値の区間の集合を属性ごとの概念記述としている．属性ごとに概念記述を持つ場合，区間で表すか，IBL で例を保持するように値とクラスの組を保持するかという問題がある．また，区間で表す場合には区間のオーバラップを許すか許さないかという問題もある．CFP の区間はオーバラップしない．例えば，図 1(a) のようなインスタンスが与えられた場合，例の提示順序にもよるが，CFP では図 1(b) のような区間が生成される．オーバラップを許す場合は図 1(c) や (d) のような区間が考えられる．

図 1(d) のように１つのクラスの概念記述を複数の区間に分割する場合，どの程度の細かさに分割するかが問題となる．そこで，本手法では属性ごとの概念記述を，あるクラスに対して下限値，上限値，値の分布で表現する．値の分布は，そのクラスに含まれる例について，値とその値を持つ例の数の組の集合で表す．すなわち，区間自体は図 1(c) のようになり，その他に値の分布情報を持つ．１つのクラスの概念が複数の区間で表現されるケースについては，分類において値の分布情報を用いることで対応する．

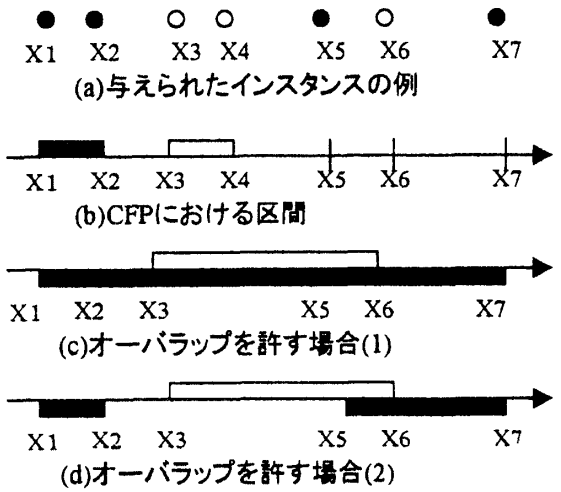

図 1: 属性ごとの概念記述の例

```
train(TrainingSet):
begin
  initializeCD();
  foreach e in TrainingSet
    foreach feature f
      if e_f value is known
        if e_f > upper then upper = e_f
        if e_f < lower then lower = e_f
        updateDistribution(f, e_f, e_class)
end
```

図 2: 学習アルゴリズム

3. 学習と分類

学習は学習用データを用いて，各属性各クラスごとに **2.** で述べた記述を作成する処理である．図 2 にそのアルゴリズムを示す．一方，分類では属性ごとに各クラスに得点を投票し，最も高い得点を得たクラスを分類結果とする．分類アルゴリズムを図 3 に示す．各属性の投票では，与えられた例の値が下限値と上限値の間にあれば，値の分布情報とパラメータΔを用いて

A Method of Classification by Voting among Features
Hiroyoshi Watanabe†, Masayuki Arai† and Kenzo Okuda‡

†) School of Science and Engineering, Teikyo University
‡) Business School, Sakushin Gakuin University

classification(e,Δ):
begin
  foreach class c $vote_c = 0$
  foreach feature f
    if e_f value is known
      foreach class c
        if $c_{lower} \leq e_f$ **and** $e_f \leq c_{upper}$
          then $vote_c = vote_c +$ getPoint(c,e_f,Δ_f)
  return class c with highest $vote_c$
end

図 3: 分類アルゴリズム

表 1: 実験に用いたベンチマークデータ

データ名	クラス数	属性数	インスタンス数
glass	6	9	214
ionosphere	2	34	351
iris	3	4	150
wine	3	13	178

得点を計算する.

各属性において，属性値の最大値と最小値の差を $rangeOfValue$ とし，そのクラスの値の分布より，与えられた例の属性値の近傍 ($\pm\Delta_f \times rangeOfValue$) の例の数を i，そのクラスに含まれる例の総数を t とすると，i/t をそのクラスの得点とする.

4. GA によるパラメータの調整

分類で用いられるパラメータΔは属性ごとに異なる値を指定するが，その調整に GA を用いる．実現した GA では，Δ_fの配列を遺伝子コードとする．初期値は乱数を用いて 0.0 から 1.0 の値を設定する．交叉は一様交叉とし，交叉する個体の選択にはトーナメント方式 (サイズ 4) を用いた．また，次世代の個体の選択はエリート戦略に基づき，世代間のギャップを 0.6 とした．突然変異の操作は遺伝子の値を 0.5 倍，または 1.5 倍する (突然変異の確率 0.1)．また，適合度はデータの一部を使って求めた分類精度とした.

5. 実験結果

UCI Repository のデータセットのうち，全てが数値属性である表 1 のデータを用いて実験を行った．実験では leave-one-out cross-validation による分類精度を求めた．本手法と同様に各属性の投票によって分類を行う手法である CFP との比較を表 2 に示す．BCFP はバッチモードで CFP の学習を行うアルゴリズムであり，文献 [2] における BCFP2-w の分類精度であり，CFP*は文献 [1] における CFP の分類精度である．これらも leave-one-out cross-validation による結果であり，実験条件は等しい．また，本手法において，全データの 20%を用いて GA によるパラメータの調整を行った．本手法では，GA で用いるデータに依存して結果が異なると考えられるので，GA で用いるデータを変えて 10 回の実験を行った．表 2 より，CFP と比較して本手法が有効であることがわかる.

表 2: 分類精度の比較

データ	分類精度 (%)				
	CFP*	BCFP	本手法		
			平均	最大	最小
glass	56.54	57.94	69.21	72.43	65.42
ionosphere	88.60	87.46	91.94	93.16	90.03
iris	95.33	94.00	93.40	95.33	88.00
wine	91.01	89.89	96.69	98.31	96.07

表 3: GA で用いるデータ量の違いによる分類精度

データ量 (%)	分類精度 (%)					
	glass			wine		
	最大	平均	分散	最大	平均	分散
10	72.90	63.69	23.65	98.31	96.52	0.61
20	72.43	69.21	4.24	98.31	96.69	0.59
30	72.43	69.53	3.34	98.31	97.25	0.87
40	73.83	71.87	5.18	98.88	97.76	0.83
50	74.77	72.48	2.50	98.88	98.03	0.37

表 3 に，glass と wine に対して，GA で用いるデータ量を 10%から 50%まで変化させた場合の分類精度を示す．おおまかな傾向としてはどちらのデータセットにおいても GA で用いるデータ量が多いほど，分類精度は高くなり，分散は小さくなると言える.

6. おわりに

属性の投票による分類の一手法を提案し，同様な手法である CFP に対する優位性を示した．今後，他のデータに対する実験や，GA 以外のΔの調整法の検討等を行いたい.

参考文献

[1] Guvenir,H.A. and Sirin,I. : Classification by Feature Partitioning, Machine Learning, 23, pp.47 - 67, 1996.
[2] 渡辺博芳，荒井正之，奥田健三 : CFP 手法の改善，人工知能学会全国大会 (第 11 回) 論文集，pp.117 - 180, 1997.