2K−05　大規模データベースへの適用を考慮した分類規則の学習

中安　とし子[†]　　末松　伸朗[‡]　　林　朗[‡]

[†]広島市立大学大学院情報科学研究科　　[‡]広島市立大学情報科学部

1　はじめに

近年，データベースからの知識獲得 (KDD: Knowledge Discovery in Databases) に関する研究が盛んになっている．古くから様々な分野で研究されてきたデータ集合から分類規則を学習する手法も，KDDへの適用が試みられている．しかし，これまで提案された学習法のほとんどは，大規模なデータ集合を扱うことに適していない．なぜならば，それらの学習法は，すべてのデータを用いる計算を行うため，大きなメモリ容量やストレージ入出力の工夫を必要とするからである．このような背景のもと，本稿では，限られたメモリ容量の内で，非常に大きなデータ集合から効率的に分類規則を学習する手法を提案する．

2　データ要約

本研究では，データを走査しながら，与えられたメモリ上において表現可能なデータ集合の要約を逐次的に構築し，その要約から分類規則を学習するというアプローチをとる．要約には，クラスタリング分野で提案されたCF tree[2] を拡張したものを用いる．CF treeの拡張に伴い，CF treeを利用した密度推定法であるCF-kernel法 [3] の拡張も提案する．

CF tree

CF treeは，クラスタ表現によってデータ集合を要約する木構造のデータ構造である．

CF treeはクラスタを，そのクラスタに属する“データ点の数”，“線形和”，“二乗和”の3つの情報に要約して保持する．これら3つの情報をCF(Clustering Feature) と呼ぶ．クラスタに属するすべてのデータを保持しなくとも，CFからクラスタの平均や標準偏差，クラスタ間の様々な距離を計算できる．また，クラスタに新たな点が追加されるとき，CFは逐次的に更新できる．

このようなCFの特徴を利用した木構造がCF treeである．CF treeの各ノードは，それが持つ子ノードに対応するサブクラスタのCF情報をエントリとして持ち，子ノードすべてによって形成されるクラスタに対応する．したがって，CF treeのルートは，すべてのデータで形成されるクラスタに対応する．葉ノードが持つエントリが最小単位のサブクラスタであり，その大きさはある閾値以下となるように制約されている．新しいデータは，距離関数を用いて，木のルートから最も近い子ノードに降りていく．そして，閾値が許すならば，葉ノードのエントリに吸収される．さもなければ，新たなエントリとして木に加えられる．木の占有する大きさがメモリ制限を越えたときには，閾値を変更し（サブクラスタの最小単位を大きくし），木を構築しなおす．

CF-kernel法

CF-kernel法とは，kernel法を近似した密度推定法である．kernel法がそれぞれのデータ点にkernel関数を配置する代りに，CF-kernel法はCF treeの葉ノードのエントリにkernel関数を配置し，密度関数を，

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{m} n_i CK_i(x) \qquad (1)$$

と推定する．ここで，[3] ではkernel関数CK_iとして，

$$CK_i(x) \approx \frac{1}{\sqrt{2\pi}\sqrt{\hat{\sigma}_i^2 + h^2}} e^{-\frac{(x-\hat{\mu}_i)^2}{2(\hat{\sigma}_i^2+h^2)}} \qquad (2)$$

を用いている．hは平滑化パラメータである．nはデータ点の総数，n_i，$\hat{\mu}_i$，$\hat{\sigma}_i$は，それぞれ葉ノードのエントリに対応するサブクラスタi $(i = 1, \ldots, m)$のデータ点の数，平均，標準偏差である．

2.1　CF treeとCF-kernel法の拡張

MCF

[2] の提案するクラスタ表現CFは，クラスタ分布の異方性の情報を切り捨てている．本研究では，より詳細なデータ要約を得るために，CFの内の“二乗和”の代りに，“積和行列” $\sum_{i=1}^{N} \mathbf{x}_i \mathbf{x}_i^t$（$\mathbf{x}_i$はクラスタに属するデータ点）を用いる．そして，“データ点の数”，“線形和”，“積和行列”の3つの情報を，MCF(Multivariate Clustering Feature) と呼ぶことにする．また，CFの代りにMCFを用いる木をMCF treeと呼ぶことにする．MCFが分かれば，CFからは得られなかったクラスタの共分散行列を求めることができる．

距離関数

MCF treeを構築するためには，異なる共分散行列を持つクラスタ間の距離，一つのデータ点とMCFで要約されたクラスタの距離を一貫して決定する距離関数が必要である．

A Classification Learning method for Very Large Databases
Toshiko Nakayasu[†], Nobuo Suematsu[‡], Akira Hayashi[‡]
[†]Graduate School of Information Sciences, Hiroshima City University,　[‡]Faculty of Information Sciences, Hiroshima City University

一つのデータ点も一つのクラスタであると解釈すると，クラスタ間の距離関数は，クラスタを合併することによって悪化するクラスタの質で表現できる．データ要約である CF tree において，クラスタの占める空間が大きくなるほど，データの要約情報が曖昧になるため，クラスタの質が悪くなると考えられる．

共分散行列 Σ をもつクラスタの大きさの定義として一般に使われているものに，$trace(\Sigma)$ や $|\Sigma|$ があるが，前者はクラスタの歪みを無視しており，後者は空間の次元数以下の点の数をもつクラスタにおいて常にゼロとなる．この問題を回避するために，共分散行列が示す楕円形の軸の長さの和をクラスタの質として定義する．

MCF-kernel 法

MCF 情報を反映させると，(1) 式の $CK_i(x)$ は，

$$MCK_i(\mathbf{x}) \approx \frac{1}{(2\pi)^{\frac{d}{2}}|\hat{\Sigma}_i + H|^{\frac{1}{2}}} \times \exp\left[-\frac{1}{2}(\mathbf{x}-\mu_i)^t(\hat{\Sigma}_i + H)^{-1}(\mathbf{x}-\mu_i)\right] \quad (3)$$

となる（証明は省略）．ここで，d はデータ点の次元数，H は $d \times d$ の平滑化行列，$\hat{\Sigma}_i$ はクラスタ i の共分散行列である．(3) 式を kernel 関数とする密度推定法を MCF-kernel 法と呼ぶことにする．

3 分類規則の学習

学習の対象となるデータ要約は，データのラベルにしたがって，クラス別に CF tree を構築することによって得られる．そして，データ空間において，各クラスがどのように分布しているのか知るために，MCF-kernel 法を用いて密度関数 $\hat{f}_{\omega_1}, \hat{f}_{\omega_2}, \ldots, \hat{f}_{\omega_C}$ を推定する．すると，データ $\mathbf{x}$ がクラス ω_i に属する確率は，

$$P(\omega_i|\mathbf{x}) = \frac{P(\omega_i)\hat{f}_{\omega_i}(\mathbf{x})}{\sum_{k=1}^{C} P(\omega_k)\hat{f}_{\omega_k}(\mathbf{x})} \quad (4)$$

で与えられる．ここで，$P(\omega_i)$ はデータがクラス ω_i に属する事前確率である．分類においてデータ $\mathbf{x}$ は，

$$\arg\max_{\omega_i} P(\omega_i|\mathbf{x}) \quad (5)$$

で求められるクラスに分類される．

4 実験

UCI の機械学習用データベース [4] のいくつかにおいて，本手法と他の学習法の分類精度を比較した．比較する手法として，多くの研究者が比較対象とする決定木アルゴリズム C4.5(Revision8) と，学習に使用するメモリ容量がデータ数に依存しない naive Bayes 法 (NB) を選択した．表 1 は 10-fold cross-validation で得られた分類誤り率である．C4.5 と NB は十分に機能できるメモリ環境において実行した．表中の“-”は，本手法がそのメモリ容量では実行不可能であることを示している．括弧内は“MCF tree の最小単位サブクラスタの数／データの数”(%) である．

表 1: 分類誤り率 (%)

データ集合	C4.5	NB	本手法／メモリ (K)		
			8	32	128
balance	21.3	9.3	8.3 (1.8)	7.3 (8.4)	7.5 (30.8)
breast	6.0	6.4	-	4.7 (0.8)	5.1 (2.3)
haberman	29.1	25.2	26.5 (6.0)	26.2 (22.7)	26.2 (73.9)
iris	4.7	4.7	3.3 (8.1)	3.3 (34.5)	3.3 (98.7)
liver	34.8	44.1	40.9 (1.9)	34.8 (7.9)	33.7 (32.0)
pima	26.1	24.2	26.3 (0.7)	24.7 (2.2)	26.3 (8.8)
vehicle	28.5	54.2	-	15.1 (1.1)	16.3 (3.7)
wine	7.3	2.3	1.1 (1.9)	0.6 (7.6)	0.6 (29.8)
sonar	28.4	32.2	-	-	25.4 (2.1)

表 1 において，本手法の分類性能は非常に小さなメモリ上においても，C4.5 や NB より優れていた．

5 おわりに

本稿で提案した手法は，データ数に依存しないメモリ容量で学習することができるので，大規模データベースへの応用が容易である．本手法は，データ要約から学習を行うため，学習にすべてのデータを必要とする手法より効率が良い．また，ここで提案した CF tree と CF-kernel 法の拡張は，他分野においても寄与が大きいと思われる．

今後の課題としては，本手法の性能のより詳しい検証を予定している．

参考文献

[1] Quinlan,J.R., “Improved use of continuous attributes in C4.5”, Journal of Artificial Intelligence Research, 4, 77-90, 1996.

[2] T.Zhang, R.Ramakrishnan, M.Livny., “BIRCH: An Efficient Data Clustering Method for Very Large Database”, Proceedings of Workshop on Research Inssue on Data Mining and Knowledge Discovery (in cooperation wiht ACM-SIGMOD'96), 1996.

[3] T.Zhang, R.Ramakrishnan., “Fast Density Estimation Using CF-kernel for Very Large Databeces”, Proceedings of the fifth ACM SIGKDD international Conference on Knowledge discovery and data mining, 1999.

[4] Blake,C.L., Merz,C.J.,(1998).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.