

ニューラルネットワークの入出力間相互情報量を用いた

データ自動分割アルゴリズムに関する研究

2000年3月

菊地登志子



①

ニューラルネットワークの入出力間相互情報量を用いた
データ自動分割アルゴリズムに関する研究

2000年 3月

菊地 登志子

目次

第1章 序論.....	1
第2章 自動分割の観点からみた従来法の比較.....	6
2.1 階層的クラスタリング.....	6
2.2 非階層的クラスタリング.....	8
2.3 ニューラルネットワークによるデータ分割.....	9
2.3.1 競合学習.....	9
2.3.2 漏えい学習.....	12
2.3.3 Kohonen 自己組織化特徴写像.....	12
2.4 データ自動分割の手法としての比較.....	13
2.5 漏えい学習ニューラルネットワークによるデータ分割の例.....	15
2.5.1 アンケートデータの分割.....	15
2.5.2 分割結果の評価方法.....	16
2.5.3 k-means 法による分割結果との比較.....	17
2.6 まとめ.....	21
第3章 漏えい学習と競合学習を組み合わせた学習法.....	22
3.1 緒言.....	22
3.2 組合せ法.....	23
3.3 組合せ法と競合学習, Kohonen 自己組織化特徴写像との比較.....	24
3.4 結果.....	25
3.4.1 連続値モデルデータに対する結果の比較.....	25
3.4.2 2値モデルデータに対する結果の比較.....	30
3.4.3 アンケートデータに対する結果の比較.....	32
3.5 考察.....	34
3.6 まとめ.....	34
第4章 相互情報量により学習パラメータを調整した 効率的データ自動分割アルゴリズム.....	35
4.1 緒言.....	35
4.2 相互情報量により学習パラメータを調整する組合せ法 (CMAMI).....	36

4.2.1	CMAMI における相互情報量の計算.....	36
4.2.2	学習係数の減少率を決定する $\lambda(t)$ の設定.....	37
4.2.3	CMAMI の学習アルゴリズム.....	40
4.3	結果.....	41
4.3.1	サンプルデータ.....	41
4.3.2	r 値とデータの分布構造.....	44
4.3.3	s の値と g_w の減少方法に関する検討.....	45
4.3.4	他の手法との比較.....	49
4.4	考察.....	52
4.5	まとめ.....	53
第5章 ニューラルネットワークの入出力間相互情報量に基づく		
	データ自動分割システム.....	54
5.1	緒言.....	54
5.2	最適分割数の推定問題の背景.....	55
5.3	データの分布構造の特徴を捉えた分割数を推定する	
	データ自動分割システム.....	57
5.3.1	データの分布構造を捉えるための指標としての r 値.....	57
5.3.2	最適な分割数を推定するデータ自動分割システム.....	62
5.4	実データへの応用.....	64
5.4.1	地方銀行のデータ.....	64
5.4.2	地方銀行の分割結果.....	65
5.5	考察.....	71
5.5.1	スケーリングの方法.....	71
5.5.2	推定された分割数による分割結果の解釈.....	71
5.5.3	推定された分割数の検討.....	72
5.6	まとめ.....	73
第6章 結論.....		
		74
謝辞.....		
		76
文献.....		
		77

第1章

序論

データ分割は、画像、音声、文字などを対象にしたパターン認識の基本手法として、また最近では情報化社会から大量に発生するデータを対象にしたデータマイニングの手法としても注目されている[1][2][3][4]。データ分割とは、分割の基準となる特定の意味を考えず、データ集合がもつ分布の構造のみに基づいて距離の近いデータを同一グループとして分類する、その分類規則を探索する技法である。データマイニングの分野では、顧客、製品、消費者行動などのデータをデータ分割によりグループ化することで、今まで知られていなかった有用な情報を抽出し、企業意思決定や販売戦略に活用することが試みられている[5][6]。このようなデータ分割では、次から次へ大量に発生するデータを対象とするために、ほとんどユーザーの介入なしで、データ集合から特徴的情報を自動抽出できる発見型手法の提案が求められている[1][5]。しかし、増大するデータに対し、このような特徴抽出を行うのは大変困難な問題とされている[1][2][4]。

問題点の一つは、大量の多次元データを対象とすることにある。この問題に対しては、R.Agrawal や C.C.Aggarwal らが多次元空間では従来のクラスタリングアルゴリズムのほとんどが効果的な機能を果たさないことを指摘し、大量の多次元データに対する新しいクラスタリングアルゴリズムの提案を行っている[7][8]。二つめの問題点は、時々刻々発生するデータに対応する必要がある点である。企業戦略の重要なポイントとして、新しい変化への対応の速さが求められ、意思決定に有用な情報を如何に迅速に実際の戦略に活用できるかが問われている。そのためには、データ集合のもつ特徴を可能な限り機械的に抽出する、すなわちデータ分割の過程を可能な限り自動的に行うことが重要である。本研究では、この点を議論の中心に置いて、データの自動分割を行う手法に対する検討を行っていく。

データ分割の手法には、従来から様々な分野で用いられている統計学的な

手法と、データマイニングの分野で最近注目されているニューラルネットワークによるものがある。統計的な手法は、階層的クラスタリング[9][10]と、非階層的クラスタリング[9]の2つに大別される。階層的クラスタリングで最も広く用いられている手法は、データ数と等しいクラスタ数から始め、類似したものを統合することで最終的にデータ集合全体を1つのクラスタにまとめていくもので、中心的凝集法とよばれている。これに対し、非階層的クラスタリングはクラスタ数を固定した単一の分割結果が得られる手法である。最も広く応用されているk-means法では、クラスタ数を予め固定してデータ分割を行う。そのためクラスタ数が未知の場合、様々なクラスタ数を設定してクラスタリングを行い、結果の良否を判定する必要がある。k-means法の改良版ともいえるISODATA法では、予めクラスタ数の初期値は与えられるが、分割を行う過程の中で、クラスタ数を変更、決定していくことが可能である。いずれの手法も設定されたクラスタ数に対し、何らかの方法でクラスタ中心を与え、それぞれのデータを各クラスタ中心と比較し最も類似するクラスタに分類することでデータ分割を行う。

データ分割の自動化という観点からこれらの統計的手法を捉えた場合、いくつかの検討を要する問題点がある。階層的クラスタリングでは、類似度の定義をすれば、データ分割そのものは自動的に行うことができる。しかし、実データを対象にデータ分割を行う場合、すべての実データがこのような階層構造をもつとは考え難く、階層的クラスタリングから得られた分割結果が必ずしも有用とはいえない。この点では、非階層的クラスタリングを用いるほうが適切ではあるが、クラスタ数をいくつにするか、クラスタ中心の初期値をどのように設定するか（クラスタ中心の初期値により得られる分割結果が異なる）など、解決しなければならない点が多い。

一方、ニューラルネットワークを用いた手法では、分割の基準となる特定の意味を考えないデータ分割であることから、教師なし学習として知られる競合学習[11]、Kohonen自己組織化特徴写像[12][13]などの学習アルゴリズムが広く応用されている。これらの学習アルゴリズムでも、非階層的クラスタリングと同様、クラスタ数すなわち分割数の設定やクラスタ中心にあたるニューラルネットワークの結合荷重の初期値の設定などが必要とされる。しかし、入力データに応じて結合荷重の値を調整し、データ集合がもつ分布の特徴を自己組織的に抽出する学習アルゴリズムにより、統計的な手法よりはるかに有益な分割結果を得ることができる。

データ分割の手法を比較した研究としては、Snyderらのk-means法、競合学習、Kohonen自己組織化特徴写像の3手法を比較した研究や[14]、寺島らのk-means法とKohonen自己組織化特徴写像を比較した研究[15]などがあり、何れもKohonen自己組織化特徴写像がデータ分割に最も有効な手法と報告している。しかし、Kohonen自己組織化特徴写像では、いくつかのパラメータの設定が必要で、それらの初期値や学習の過程でどのように変化させるかは経験に基づく判断により行われている。いくつかの企業から、Kohonen自己組織化特徴写像を用いてデータ分割を行うユーティリティが既に商品化されているが[1][16]、必要なパラメータの調整はほとんど手動で行われているのが現状である。データ分割を自動的に行う手法として、Kohonen自己組織化特徴写像を採用するためには、このような調整をどこまで自動的に行えるかを検討しなければならない。

Kohonen自己組織化特徴写像では、近傍という概念を導入し近傍も含めて結合荷重の値を調整するため、データ集合の順序性や類似性が保存された分割結果を得ることができる。しかし、このような近傍学習を行うために、結合荷重の初期値によっては順序性の逆転が生じ、有効な分割結果が得られないことも起る[12][17]。データ集合のもつ分布構造の特徴を抽出するためのデータ分割では、得られる分割結果に順序性の保存は必要ではない。そこで、Kohonen自己組織化特徴写像のデータ分割の手法としての有効性をできるだけ損なうことなく、調整する必要のあるパラメータをできるだけ最小限にするために、本研究では、近傍のみではなくすべての結合荷重の値を入力データに応じて調整する漏えい学習[11]を用いることにした。

漏えい学習は、競合学習の修正型として知られている。競合学習では、入力データと最も類似性の高い結合荷重をもつグループを求め、そのグループの結合荷重の値を更に入力データとの類似性を高める方向へ調整する。このアルゴリズムでは、ある初期結合荷重が大半の入力データと高い類似性を示さない場合、その結合荷重の値はほとんど調整されずに学習が進むため、偏った分割結果が得られる可能性がある。このような問題に対処するために、学習の各ステップで最も類似性の高い結合荷重を調整すると同時に、その他の結合荷重の値も僅かずつ調整するよう修正した手法が漏えい学習である。このアルゴリズムによって、漏えい学習では、データの分布構造をありのままに反映した分割結果を得る確率が競合学習より高くなる。しかし、分布構造を反映した分割結果がほぼ得られた後も漏えい学習を続けることは、逆に

得られた分割結果を崩壊させることになる。分布構造を捉えた分割結果を高い確率で得るためには、学習が進むにつれて、漏えい学習から競合学習へ次第に移行していくことが望ましい。漏えい学習と競合学習を組合せたデータ自動分割アルゴリズムを本研究では組合せ法と呼ぶ[18]。このアルゴリズムによる分割結果と、Kohonen 自己組織化特徴写像、競合学習、k-means 法の分割結果を比較することにより、組合せ法の有効性を明らかにする。

組合せ法では、漏えい学習から競合学習へ次第に移行していくために、漏えい学習の学習係数を学習が進むにつれて、ある一定のシグモイド関数に従って緩やかに減少させる。しかし、一定のシグモイド関数を適用するのではなく、入力データの分布構造に応じて学習係数の減少のさせ方を調整すれば、より効率のよい分割アルゴリズムとなる。そこで、この調整を自動的に行うための指標として、「競合学習のアルゴリズムは入出力間の相互情報量を最大にするように働く」という Linsker の指摘に従い[19]、ニューラルネットワークの入力分布と出力分布間の相互情報量を用いる[20]ことを検討する。更に、この相互情報量により学習パラメータを調整した効率的データ自動分割アルゴリズムを構築する[21][22]。

効率的データ自動分割アルゴリズムでは予め分割数を設定する必要はあるが、データをネットワークの入力層に与えるだけで自動的に最適な分割結果が得られる。しかし、どの分割数の分割がデータの分布構造の特徴を最もよく捉えているかの判断は人間が行う必要があり、提案されている多くのユーティリティでも、指定された分割結果を視覚的に表示するのみにとどまっている。本研究の目的からすれば、データの分布構造の特徴を捉えた分割数の推定も自動的に行うことができれば非常に有用性が高い。そこで、効率的データ自動分割アルゴリズムの学習係数の調整に用いた入出力間相互情報量が、データのもつ特徴を最もよく捉えた分割数を推定する指標としても十分有効であることを示し[23]、この指標を用いて分割数の推定から最適な分割結果を得るまでのすべての過程を自動的に実行するデータ自動分割システムを構築する[24]。データ自動分割システムの応用例として、財務データによる地方銀行の分類を行い、推定された分割数の妥当性を主成分分析による結果と比較することで示す[24][25]。

以下第2章では、データ分割に用いられている従来の手法をデータ自動分割の手法としての観点から比較し、本研究では漏えい学習ニューラルネットワークを自動分割の基本的なアルゴリズムとして採用する根拠を示す。第3

章では、従来法より更に初期値依存性の低いアルゴリズムを構築する手法として、漏えい学習と競合学習を組み合わせた手法の有効性を示す。第4章では、効率よく最適な分割結果を得るために、相互情報量により学習係数を調整した効率的データ自動分割アルゴリズムを提案する。更に第5章では、効率的データ自動分割アルゴリズムをもとに、データ集合のもつ分布構造の特徴を捉えた分割数を推定する自動分割システムを構築する。

第2章

自動分割の観点からみた従来法の比較

本章では、データ分割の手法として、従来から様々な分野で用いられている統計学的手法と、ニューラルネットワークの教師なし学習について説明する。統計学的手法として、階層的クラスタリングと非階層的クラスタリング、ニューラルネットワークを用いた手法として、競合学習、漏えい学習、Kohonen 自己組織化特徴写像を取り上げ、データの自動分割アルゴリズムとしての観点からこれらの手法の比較を行う。

2.1 階層的クラスタリング

階層的クラスタリングには、凝集型と分割型がある[9][10]。凝集型では、1つのデータからなるクラスタから始め、最も近いクラスタを次々に統合していく。データ数と等しいクラスタ数の分割から、最終的にはデータ集合全体を1つのクラスタにまとめた分割まで、階層構造をもつクラスタが徐々に作成されていく。階層構造とは、すべてのクラスタの包含関係が定まっていることを表し、あるクラスタに属するすべてのデータはその親クラスタに属するという関係である。逆に、分割型ではすべてのデータを含むクラスタから始め、判別分析などを用いて各クラスタを最も離れた2つのクラスタに逐次分割していく。

階層的クラスタリングの手法として、多くの分野で広く一般的に利用されている凝集型のアルゴリズムを示す[10][26]。

- (1) N 個の n 次元データ $\mathbf{x}^h = (x_1^h, x_2^h, \dots, x_n^h)$ ($h=1, \dots, N$) が与えられたとき、すべて1つのデータからなる N 個のクラスタから始める。クラスタ数を K とした場合、 $K=N$ となる。
- (2) これらのクラスタ間の距離 d_{ij} ($i, j=1, \dots, K, i > j$) を求める。

- (3) 最も近接した 2 つのクラスタ (最小値の d_{ij} ($i \neq j$) だけ離れた 2 つのクラスタ i, j) を統合し, クラスタ数を 1 だけ減少させる ($K=K-1$).
- (4) 統合してできた新しいクラスタと, その他のクラスタ間の距離を計算する.
- (5) (3), (4)を繰り返す, $K=1$ になったら終了する.

統合後のクラスタ間の距離を計算する方法がいくつかあり, それぞれに一つの手法が対応している. 統合する前の 2 つのクラスタを p, q とし, 統合後の新しいクラスタを r とする. クラスタ r と任意のクラスタ t との距離 d_n を,

$$d_n = \min(d_{pt}, d_{qt})$$

と定義する. これを最短距離法, または単連結法と呼ぶ. 同様に,

$$d_n = \max(d_{pt}, d_{qt})$$

で定義される手法を, 最長距離法, または完全連結法と呼ぶ. これらの手法では, 統合するクラスタの最大または最小という極端な値によって, 新しいクラスタとの距離が定義されている. これに対し, 平均的な値によって定義しようとするのが群平均法である. クラスタ p, q に属するデータ数を n_p, n_q とするとき, 統合後のクラスタ r と任意のクラスタ t との距離 d_n を,

$$d_n = \frac{n_p d_{pt} + n_q d_{qt}}{n_p + n_q}$$

で定義する. 更に, クラスタ間の重心に基づいて d_n を計算する重心法, 順序で中間を取るメジアン法, クラスタ内の偏差平方和が最も小さくなるよう統合していくワード法などがあげられる. 一般的には, これらの手法による分割結果はそれぞれ微妙に異なるため, いろいろな手法を適用して得られた結果の中から最適なものを目的に応じて判断する必要があることが指摘されている[10].

2.2 非階層的クラスタリング

階層的クラスタリングでは, あるクラスタに属するすべてのデータはその親クラスタに属するという入れ子式の分割結果が得られるが, 非階層的クラスタリングはクラスタ数を固定した単一の分割結果が得られる手法である. 代表的な手法として, データ分割に最も広く応用されている k-means 法や ISODATA 法などがある[9]. 非階層的クラスタリングではクラスタ数が固定されるが, k-means 法のようにクラスタ数を予め設定してデータ分割を行うものや, ISODATA 法のようにデータ分割を行う過程でクラスタ数を変更, 決定するものがある. クラスタ数も含めて決定することが可能な ISODATA 法は, 予めデータの分布構造が判らないデータ集合を対象とするデータ分割には, 大変有効な手法と考えられる. しかし, クラスタ数を決定するために, クラスタの標準偏差の上限や, クラスタに属するデータ数の下限などの多くのパラメータに適切な値を設定しなければならず, アルゴリズムも複雑にならざるをえない.

非階層的クラスタリングでは, 自動分割のアルゴリズムとして k-means 法を取り上げ, 以下にそのアルゴリズムを示す.

クラスタ数を K 個とし,

- (1) K 個の初期クラスタ中心 $\mathbf{z}_1(1), \mathbf{z}_2(1), \dots, \mathbf{z}_K(1)$ を適当に決める.
- (2) N 個の n 次元データの $\mathbf{x}^h = (x_1^h, x_2^h, \dots, x_n^h)$ ($h=1, \dots, N$) を, 最も距離の近いクラスタ中心をもつクラスタに分類する. すべての $i=1, 2, \dots, K$ ($i \neq j$) について

$$\|\mathbf{x}^h - \mathbf{z}_j(k)\| < \|\mathbf{x}^h - \mathbf{z}_i(k)\| \quad \text{であれば} \quad \mathbf{x}^h \in S_j(k)$$

k は繰返し回数, $S_j(k)$ は $\mathbf{z}_j(k)$ をクラスタ中心とするクラスタである.

- (3) 新しいクラスタ中心をクラスタの重心として計算する.

$$\mathbf{z}_j(k+1) = \frac{1}{N_j} \sum_{\mathbf{x}^h \in S_j(k)} \mathbf{x}^h \quad j=1, 2, \dots, K$$

N_j は $S_j(k)$ に属するデータ数である.

- (4) (2)と(3)をクラスタの重心が変化しなくなるまで繰り返す.

$$\mathbf{z}_j(k+1) = \mathbf{z}_j(k) \quad j=1, 2, \dots, K$$

この方法による分割結果はクラス数 K の取り方に依存するため、種々の K について何度も試みる必要がある[27]. また、初期クラス中心の選び方によっても、収束する速度や分割結果が大きく異なることが指摘されている[26].

2.3 ニューラルネットワークによるデータ分割

ニューラルネットワークの学習法には、教師あり学習と教師なし学習がある。教師あり学習は、既に分類結果が分かっている入力データを用いて、出力から正しい分類結果が得られるようニューラルネットワークの結合荷重を修正するものである。一方、教師なし学習では入力データに対して目標となる分類結果が事前に与えられず、ニューラルネットワークは学習アルゴリズムに従って自己組織的に分類能力を学習していく。分割の基準となる特定の意味を考えずに、データ集合がもつ構造を見つけ出そうとするデータ分割には、この教師なし学習が適している。そこで、ここでは教師なし学習として知られる競合学習、漏えい学習、Kohonen 自己組織化特徴写像について説明する。

2.3.1 競合学習

競合学習ニューラルネットワークでは、基本的には入力層と出力層のみをもつ2層の階層型ネットワークが用いられている[11][17]. 入力層は入力データの次元（以下 n とする）と等しいユニット数を持ち、出力層はデータ集合の分割数（以下 K とする）と等しいユニット数からなる（図 2.1）. 出力層の各ユニットは入力層の全ユニットと完全結合している。

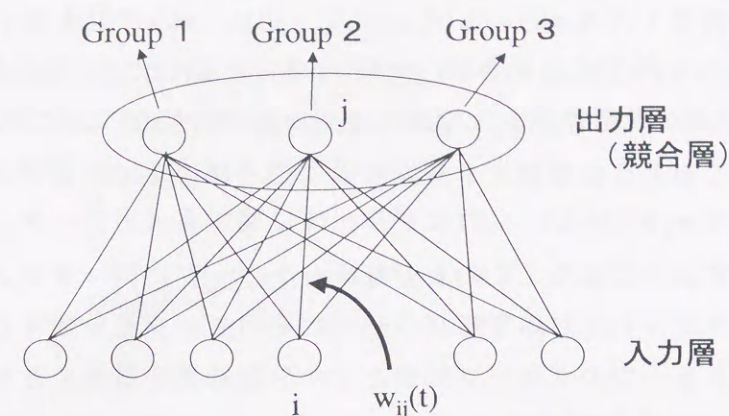


図 2.1 ニューラルネットワークの構造

競合学習のアルゴリズムを以下に示す。 $w_{ij}(t)$ を時刻 t における入力層ユニット i から出力層ユニット j への結合荷重とする。まず、この結合荷重をランダムに選出した入力データで初期化する。時刻 t における入力層ユニット i への入力を $x_i(t)$ とし、各 $x_i(t)$ から作られる入力ベクトル $\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_n(t))$ と、各入力ユニットから出力ユニット j への結合荷重ベクトルとの距離を次式で計算する。

$$d_j = \sqrt{\sum_{i=1}^n (x_i(t) - w_{ij}(t))^2} \quad (j=1, 2, \dots, K) \quad (2-1)$$

d_j が最小となる出力ユニットを j^* （勝者）とし、結合荷重を次式で決まる結合荷重変更量 $\Delta w_{ij}(t)$ だけ更新する。

$$\Delta w_{ij}(t) = \begin{cases} g_c (x_i(t) - w_{ij}(t)) & (j = j^*) \\ 0 & (j \neq j^*) \end{cases} \quad (2-2)$$

g_c は学習係数と呼ばれる ($0 < g_c \leq 1$)。出力ユニット j^* の結合荷重のみ更新されるのが競合学習の特徴である。競合学習のアルゴリズムは、入力層に提示された入力ベクトルに最も近い結合荷重ベクトルをもつ出力ユニットを探索し、この出力ユニットの結合荷重ベクトルを入力ベクトルに近づけるよう更

新する。入力ベクトルを $\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_n(t))$, $\mathbf{x}(t)$ に最も近い結合荷重ベクトルを $\mathbf{w}_1(t) = (w_{11}(t), w_{21}(t), \dots, w_{n1}(t))$ とすると, 図 2.2 に示すように, 式 (2-2) はこの $\mathbf{w}_1(t)$ が学習係数 g_c に比例した $\Delta \mathbf{w}_1(t)$ だけ $\mathbf{x}(t)$ に近づくことを意味している。このような学習を十分繰り返すことによって, 各グループの結合荷重ベクトル $\mathbf{w}_j(t) (j=1, 2, \dots, K)$ はグループに属する入力データの平均的なベクトルを表すようになる。このようなネットワークにデータを入力し, 勝者となる出力ユニットに対応するグループにそのデータを分類することで, 類似したデータを一つのグループにまとめた分割結果を得ることができる。

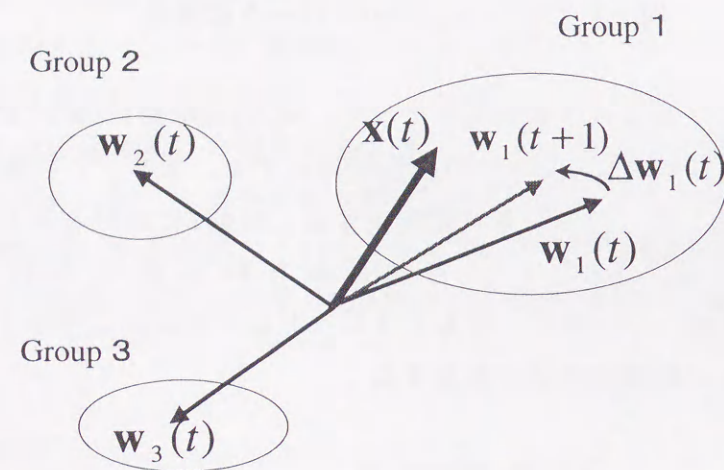


図 2.2 競合学習のメカニズム

2.3.2 漏えい学習

競合学習の修正型である漏えい学習は, 競合学習とそのアルゴリズムはほぼ同様で, 異なるのは結合荷重変更量の計算式のみである。漏えい学習では, 結合荷重変更量 $\Delta w_{ij}(t)$ は次式で計算される。

$$\Delta w_{ij}(t) = \begin{cases} g_w(x_i(t) - w_{ij}(t)) & (j = j^*) \\ g_l(x_i(t) - w_{ij}(t)) & (j \neq j^*) \end{cases} \quad (2-3)$$

g_w を勝者の学習係数 ($0 < g_w \leq 1$), g_l を敗者の学習係数 ($0 < g_l \leq 1$) と呼ぶ ($g_l \ll g_w$). g_w, g_l は, 学習の開始から終了まで変化しない定数である。 g_l を 0 とすると競合学習のアルゴリズムになる。競合学習では勝者の出力ユニットの結合荷重のみが更新されるため, ある初期結合荷重をもつ出力ユニットが大半の入力データに対して勝者にならない場合, その結合荷重はほとんど変更されずに学習が進み偏った分割結果が得られる可能性がある。このような初期値に対する依存性を回避するために, 漏えい学習では敗者の出力ユニットの結合荷重ベクトルも入力ベクトルに少しだけ近づけることで, 偏りを修正するものである。

2.3.3 Kohonen 自己組織化特徴写像

Kohonen 自己組織化特徴写像のアルゴリズムも, 競合学習, 漏えい学習と基本的にはあまり大きな違いはないが, 結合荷重変更量の計算に近傍という新しい概念が導入される。競合学習では, 勝者となる出力ユニットの結合荷重のみ更新される。漏えい学習では, 勝者となる出力ユニットの結合荷重が更新され, 更に敗者の出力ユニットの結合荷重も僅かずつではあるが更新される。Kohonen 自己組織化特徴写像では, 勝者の出力ユニットとその近傍にある出力ユニットの結合荷重が更新され, 近傍に含まれない出力ユニットは更新されない。Kohonen 自己組織化特徴写像の結合荷重変更量 $\Delta w_{ij}(t)$ は次式で計算される。

$$\Delta w_{ij}(t) = \begin{cases} \alpha(t)(x_i(t) - w_{ij}(t)) & (j \in N_c(t)) \\ 0 & (\text{それ以外}) \end{cases} \quad (2-4)$$

$\alpha(t)$ は学習係数で、学習が進むにつれて単調に減少させるのが一般的である。

$N_{j^*}(t)$ は、勝者の出力ユニット j^* の近傍領域である。近傍は、出力層ユニットが2次元グリッドとして構成されている場合、勝者のユニットを中心とする正方形内のユニットからなり、出力層ユニットが1次元で構成されている場合は、勝者ユニットの両隣の複数ユニットが近傍となる。近傍の大きさは、学習が進むにつれて次第に狭くする。この近傍の概念により、Kohonen 自己組織化特徴写像では入力データの分類能力が非常に高くなるとされている[28]。そのため、データ分割のアルゴリズムとして最も一般的に用いられているが、良好な分割結果を得るためには、学習係数や近傍の大きさを緩やかに減少させる必要があり多くの繰返し回数を必要とする。

Kohonen 自己組織化特徴写像では、ニューラルネットワークの入力層にデータを提示し、勝者となった出力ユニットとその近傍に含まれるすべての出力ユニットの結合荷重が、与えられた入力データに近づくよう更新される。これにより、類似した特徴をもつ2つの入力データに対し、出力層でも位置的に近い近傍の出力ユニットが勝者になる可能性が高くなる。つまり、入力空間の類似関係が、出力ユニットに対応する結合荷重ベクトル間の距離関係、即ち出力ユニットの距離関係に反映されることになる。これはトポロジカルマッピングと呼ばれ、このような写像特性は生体の視覚野や聴覚野などにも見られる[28]。

2.4 データ自動分割の手法としての比較

本節では、上に述べた様々なデータ分割の手法について、データの自動分割の観点から比較検討する。

凝集型の階層的クラスタリングでは、階層構造をもつクラスタが自動的に作成されていくため、自動分割の手法として適しているということもできる。しかし、この手法を用いたデータ分割では、最初に統合されたデータは、それ以降のデータ分割では分離して扱われることは全くない。これは最初に決めたことを不変のものと考えれば、検討を要する可能性の数は、完全に数え上げた場合に比べて大いに減少するという長所をもっている[9]。しかし、最初のうちに誤りを犯した場合、その後の修正は全く行えないということにもなる。実データを対象とする場合、すべてのデータ間に完全な階層構造が

存在するとは考え難く、このような方法から得られる分割結果が必ずしも有用であるとはいえない。

非階層的クラスタリングの k-means 法は、新しいクラスタ中心をクラスタの重心として計算するため、球状に分布するクラスタの抽出に適している。データ集合がもつ分布構造のみに基づいて、距離の近いデータをまとめていくデータ分割には適した手法といえる。k-means 法では予め分割数を設定する必要があり、分割数をいくつにするか、クラスタ中心の初期値をどのように設定するかなど問題点もあるが、設定さえすれば自動的に分割結果が得られることから、統計的な手法の中では k-means 法が最も自動分割には適したアルゴリズムといえる。

ニューラルネットワークを用いた手法も、非階層的クラスタリングと同様に、分割数の設定やクラスタ中心にあたる結合荷重の初期値の設定が必要とされる。しかし、k-means 法と同様に、これらの値の設定さえすれば自動的に分割結果を得ることができるため、自動分割の手法には適しているといえる。また、データを入力層に提示するごとに結合荷重の値を調整することにより、データ集合がもつ分布構造の特徴を自己組織的に抽出できるニューラルネットワークは、自動分割に大変有用な手法と考えられる。

データ分割の手法を比較した研究としては、Snyder らの k-means 法、競合学習、Kohonen 自己組織化特徴写像の3つの手法の能力を色の分割問題を用いて比較した研究があり、Kohonen 自己組織化特徴写像が分割結果と実行時間の両方の観点で最も良い結果を与えたと報告している[14]。また、寺島らは Kohonen 自己組織化特徴写像を用いて分割数の推定とデータ分割を実行する手法を提案し、k-means 法との比較から Kohonen 自己組織化特徴写像の有効性を示している[15]。これらの結果から、Kohonen 自己組織化特徴写像はデータ分割に最も有効な手法と考えることができる。しかし、このアルゴリズムでも学習係数や近傍のサイズの設定が必要で、それらに与える初期値や学習の過程での変化のさせ方は経験に基づく判断により行われている。データ自動分割の手法として、Kohonen 自己組織化特徴写像を取り上げるためには、このような調整をどこまで自動化できるかが問題となる。

Kohonen 自己組織化特徴写像では、近傍という概念によりデータ集合の順序性や類似性が保存された分割結果を得ることができる。しかし、結合荷重の初期値によっては順序性の逆転が生じ、有効な分割結果は得られないとも指摘されている[12][17]。データ集合のもつ分布の特徴を抽出するためのデ

ータ分割では、順序性の保存は必要ではない。そこで、本研究では、調整を必要とするパラメータを最小限にし、かつ Kohonen 自己組織化特徴写像のデータ分割に対する有効性を維持するために、近傍のみでなくすべての結合荷重の値を入力データに応じて調整する漏えい学習を、データ自動分割アルゴリズムを構築するための基本的なアルゴリズムとして取り上げていく。

2.5 漏えい学習ニューラルネットワークによるデータ分割の例

実際のアンケート調査データを対象にして、自動分割の基本的アルゴリズムとして取り上げた漏えい学習ニューラルネットワークと、自動分割に適した統計的手法の k-means 法を用いてデータ分割を行い、これらの手法から得られる分割結果の比較を行う。このような実データは、様々な要因により複雑な分布構造をしていることが多く、k-means 法では有用な分割結果を得るためにいくつかの初期条件で試行錯誤を繰り返さなければならない。一方、漏えい学習ニューラルネットワークでも、結合荷重の初期値や学習係数の値により異なる分割結果が得られる。本節では、これらの分割結果の比較から、データ分割に用いられる従来法の中では、漏えい学習ニューラルネットワークがデータ自動分割の基本的なアルゴリズムとして最も有用であることを示す[29][30][31]。

2.5.1 アンケートデータの分割

分割結果の比較に用いたデータは、男子大学生のライフスタイルについてのアンケートデータである。学生数は 163 人、質問項目の内容は年齢、衣食住、趣味、アルバイトに関するもので、質問総数は 253 項目である。各質問の内容に該当する場合は 1、しない場合は 0 とし、1 学生につき 253 ビットの入力データを用意した。

用いたニューラルネットワークは、入力層は 253 ユニット、出力層はそれぞれ分割数に応じて 2 から 5 ユニットをもつ 2 層の階層型ネットワークである。このネットワークによりアンケートデータの 2 分割から 5 分割を行うことができる。漏えい学習の勝者の学習係数 g_w に 0.1, 0.2, 0.3, 敗者の学習係数 g_l に g_w の 1/10 と 1/100 の値を設定しデータ分割を行った。漏えい学習ニューラルネットワークによる分割結果は、入力データの提示順序に依存する。

そこで、全入力データを 1 回ずつ入力層に提示するのを 1 サイクルとし、入力データは各サイクルごとにランダムな順序で提示した。k-means 法では、全入力データを各クラスに割り当てた後でクラスタの重心を更新するため、このような提示順序の問題は生じない。よって k-means 法では一定の順序でデータを提示した。

2.5.2 分割結果の評価方法

データ分割における結果の良さの評価は、次式で表される summed deviation を用いて行った。

$$\text{summed deviation} = \frac{1}{N} \sum_{h=1}^N \sum_{j=1}^K \sum_{i=1}^n v_{hj} (x_i^h - \mu_{ij})^2 \quad (2-5)$$

ただし、 N は入力データ数、 K は分割数である。各入力データを $\mathbf{x}^h = (x_1^h, x_2^h, \dots, x_n^h)$ ($h=1, \dots, N$) とし、 v_{hj} は h 番目の入力データが出力層の j 番目のユニットに対応するグループに属する場合 1、その他の場合 0 とする。 μ_{ij} は j 番目のユニットに対応するグループに属する入力データ集合の重心ベクトルの第 i 成分である。分割数を固定したときに得られる分割結果は、この summed deviation の値が小さいほど良好とする。すべての分割のうちで最小の summed deviation を示す分割が本来の最適解であるが、この真の最適解は実際は求まらないので、本研究では複数個の異なる初期値から得られた分割の summed deviation の最小値が一致する場合、その最小値に対応する分割を最適解とする。ただし、漏えい学習の場合、summed deviation の最小値は学習終了時点で得られるとは限らないため、学習の全過程で summed deviation を毎ステップで計算し得られる最小値を採用する。

分割数を固定した場合、各グループの重心とそのグループに属するデータとの距離の合計によって計算される summed deviation が小さな値を取るということは、互いに十分に分離されたグループが抽出されたことを意味する。特殊な目的のもとにデータ分割を行う場合を除外すれば、summed deviation を評価の基準とすることでデータの分布構造の特徴を捉えた最適解を得ることができる。本研究の目的は分布の構造上の特徴を自動的に捉える自動分割の手法を検討することであるため、summed deviation が最小値を示す分割を最適解と解釈することにした。

2.5.3 k-means 法による分割結果との比較

アンケートデータの分割を、上記の漏えい学習ニューラルネットワークで行った結果、2分割では g_w , g_l の値によらずすべて summed deviation の最小値が一致し、同一の分割結果が得られた。3分割から5分割では、表 2.1 に示すように、異なる学習係数に対し summed deviation の最小値が一致する結果は得られず、学習係数によって得られる分割結果がすべて異なっていた。

表 2.1 漏えい学習によるアンケートデータの分割結果から得られる summed deviation (網掛けは分割数ごとにみた最小値)

g_w	g_l	3 分割	4 分割	5 分割
0.1	0.01	24.9456	24.3783	23.8923
0.1	0.001	25.0838	24.3698	23.8903
0.2	0.02	24.9556	24.3984	23.9465
0.2	0.002	25.0938	24.3991	23.9571
0.3	0.03	24.9785	24.4693	24.0676
0.3	0.003	24.9730	24.4193	24.0120

このアンケートデータの場合、 $g_w=0.1$, $g_l=0.001$ とした漏えい学習から summed deviation が最小値となる分割結果が多く得られた。そこで、結合荷重に異なる初期値を与えた 100 通りのニューラルネットワークで $g_w=0.1$, $g_l=0.001$ とした漏えい学習を実行し、アンケートデータの3分割から5分割を行った。また、クラスタ中心として異なる 100 通りの初期値を与えた k-means 法を用いて、同様にアンケートデータの3分割から5分割を行った。得られた分割結果の summed deviation の値の分布を図 2.3 に示す。漏えい学習ニューラルネットワークによるデータ分割では、summed deviation の値の

分布にばらつきが少なく、どのような初期値に対してもかなり最適解に近い分割結果が得られることがわかる。

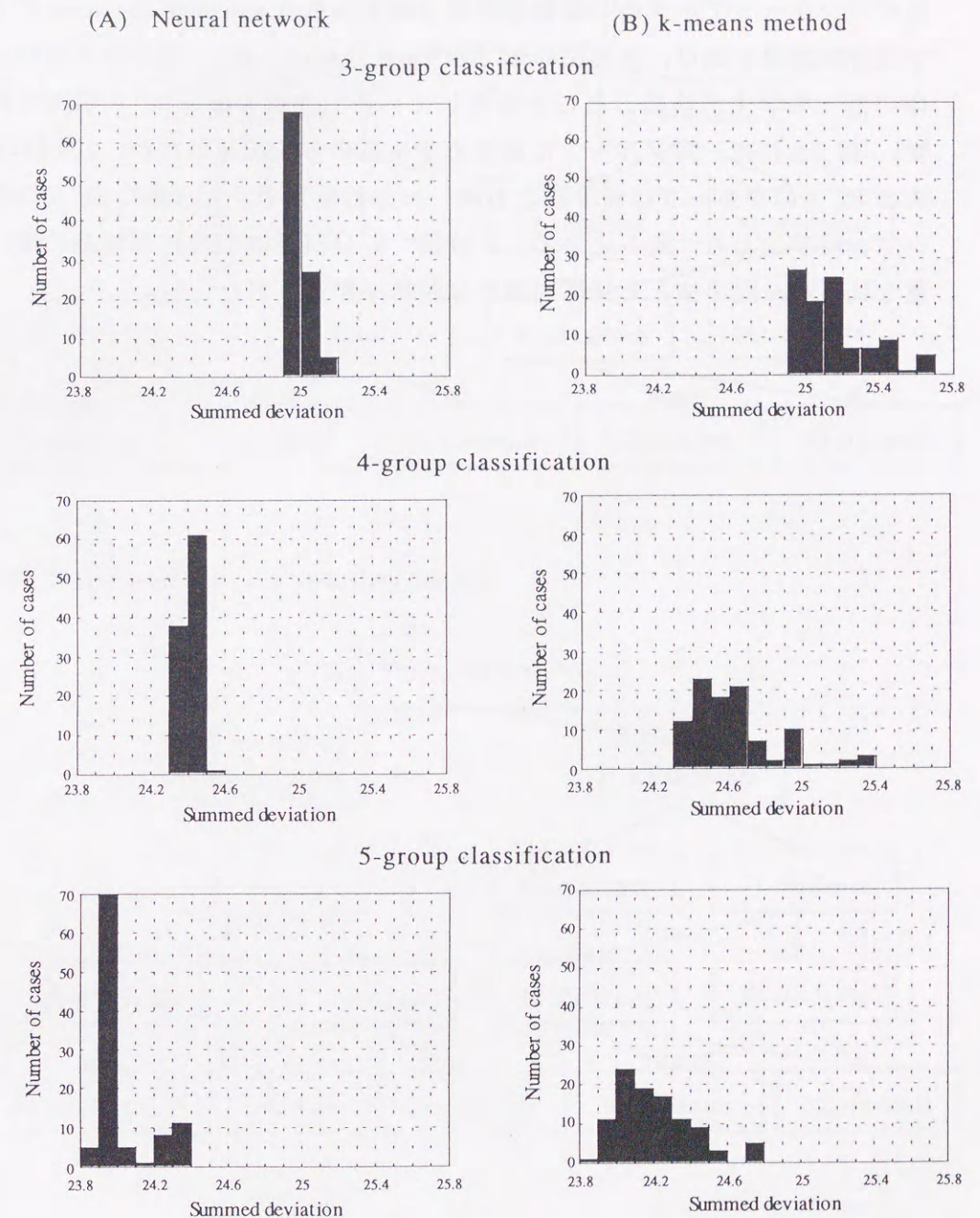
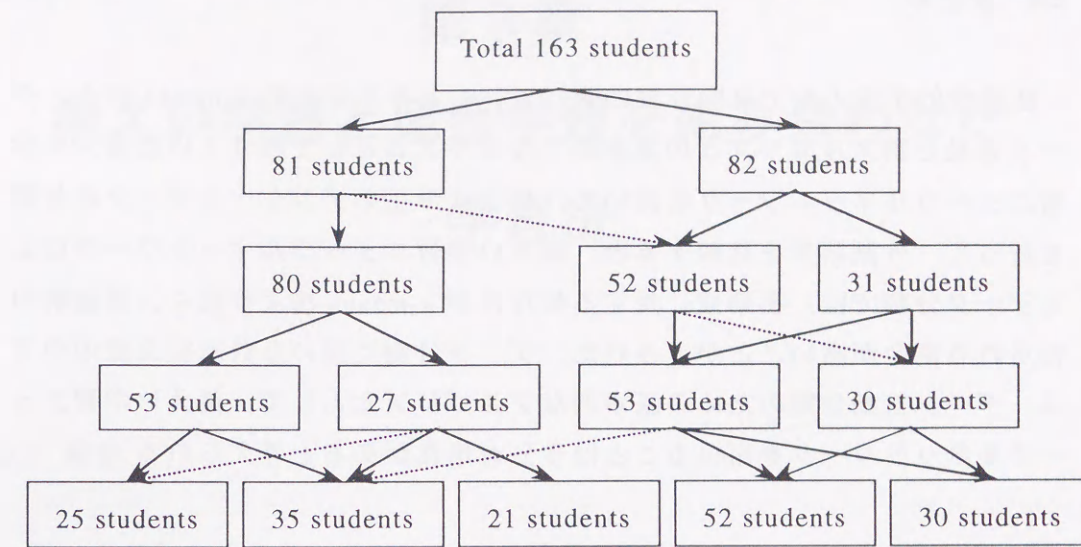


図 2.3 漏えい学習ニューラルネットワークと k-means 法による分割結果から得られた summed deviation の分布

k-means 法, 漏えい学習ニューラルネットワークから得られた各々 100 通りのアンケートデータの分割結果から, 最も小さな summed deviation に対応する分割結果を選び, 樹形図の形で示す(図 2.4 (A),(B)). 図中の矢印は, 矢印に沿って学生が移動したことを表している(点線の矢印は 1 名のみ移動). 図 2.4 には, 各グループに属する学生数のみを記載したが, 2 分割の 81 人と 82 人のグループは両手法とも同一の分割結果で, 3 分割の 31 人のグループも両手法とも一致している. 4 分割, 5 分割ではやや分割結果に違いが見られるが, 両手法ともほぼ同様の分割結果が得られた.

(A) Data classification by neural network



(B) Data classification by k-means method

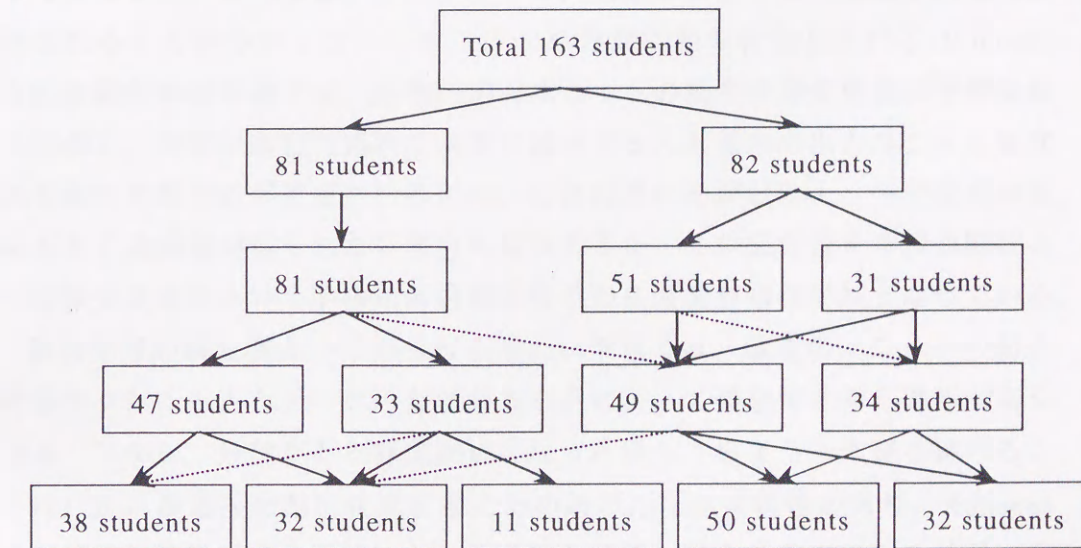


図 2.4 漏えい学習ニューラルネットワークと k-means 法によるアンケートデータの分割結果

2.6 まとめ

統計学的手法の中で自動分割に適していると考えられる k-means 法と、データ自動分割アルゴリズムの基本的アルゴリズムとして取り上げた漏えい学習ニューラルネットワークを用いて、男子大学生のアンケートデータの分割を行った。分割結果を比較すると、漏えい学習ニューラルネットワークによるデータ分割では、初期値に対する依存性が k-means 法より低く、最適解の得られる割合が高いことが示された。データ分割に用いられる従来法の中では、データ自動分割のための基本的なアルゴリズムとして、漏えい学習ニューラルネットワークを用いることは十分有用性があると考えられる。

第3章

漏えい学習と競合学習を組み合わせた 学習法

3.1 緒言

データ分割の代表的な手法として前章でも取り上げた k-means 法では、得られる分割結果はクラスタ中心の初期値に依存し、分割結果の summed deviation にはかなりばらつきがみられる。これに対し、漏えい学習ニューラルネットワークによるデータ分割では、初期値に対する依存性が大幅に改善されることがわかった。一方、データ分割に最も有効とされる Kohonen 自己組織化特徴写像では、近傍の出力ユニットの結合荷重変更量は学習係数に比例し、学習が進むに連れて次第に減少する。近傍内の出力ユニットも含めて結合荷重の値が変更されるため、結合荷重の初期値によっては位相の逆転が生じ最適解が得られない場合も起りえるが、この変更量を学習の進行と共に減少させるという手法は最適解を得るのに大変有効な手段となっている。

競合学習の修正型として知られる漏えい学習では、敗者のユニットの結合荷重ベクトルも入力データに近づけることによって最適解を得る確率が高くなる。しかし、分割がある程度安定になった後もこのような学習を続けることは、逆に最適な分割に収束することの妨げになる可能性があり、Kohonen 自己組織化特徴写像の近傍における学習と同様、結合荷重変更量を学習の進行と共に減少させれば有効な分割結果を得る確率が高くなると考えられる。漏えい学習の敗者の学習係数を学習の進行と共に減少させ 0.0 にすると、競合学習のアルゴリズムとなる。本章では上記の考えに基づいて漏えい学習と競合学習を組合せた学習法である、組合せ法(Combination Method; CM)[18]を説明し、更に Kohonen 自己組織化特徴写像、競合学習、k-means 法および CM の分割能力を比較し、CM は Kohonen 自己組織化特徴写像より更に最適

解が得られる確率が高いことを示す。

3.2 組合せ法

競合学習の(2-2)式の変更量を、本章では他の学習法の変更量と区別するため、肩に(1)をつけて $\Delta w_{ij}^{(1)}(t)$ と表す。

$$\Delta w_{ij}^{(1)}(t) = \begin{cases} g_c(x_i(t) - w_{ij}(t)) & (j = j^*) \\ 0 & (j \neq j^*) \end{cases} \quad (3-1)$$

(2-3)式の漏えい学習の結合荷重変更量には、肩に(2)をつけて $\Delta w_{ij}^{(2)}(t)$ とする。

$$\Delta w_{ij}^{(2)}(t) = \begin{cases} g_w(x_i(t) - w_{ij}(t)) & (j = j^*) \\ g_l(x_i(t) - w_{ij}(t)) & (j \neq j^*) \end{cases} \quad (3-2)$$

組合せ法 CM の結合荷重変更量 $\Delta w_{ij}(t)$ は、上の2つを用いて次の(3-3)式で定義される。

$$\Delta w_{ij}(t) \equiv \lambda(t)\Delta w_{ij}^{(1)}(t) + (1 - \lambda(t))\Delta w_{ij}^{(2)}(t) \quad (3-3)$$

$\lambda(t)$ はシグモイド型増加関数($0 < \lambda(t) < 1$)とする。 $g_c = g_w$ とすると、(3-3)式の結合荷重変更量は、次の(3-4)式のように表すことができる。

$$\Delta w_{ij}(t) = \begin{cases} g_w(x_i(t) - w_{ij}(t)) & (j = j^*) \\ (1 - \lambda(t))g_l(x_i(t) - w_{ij}(t)) & (j \neq j^*) \end{cases} \quad (3-4)$$

$\lambda(t)$ が時間(学習の進行)と共に増加するため、敗者のユニットの結合荷重変更量は時間と共に減少する。本章の実験では、 $\lambda(t)$ に(3-5)式を用いる。

$$\lambda(\tau) = \frac{1}{1 + \exp((25 - \tau)/10)} \quad (3-5)$$

ただし、全入力データをランダムな順序で一回ずつ入力層に提示するのを1

サイクルとし、ここでは同一サイクル内で $\lambda(t)$ を一定とするため、 $\lambda(t)$ を $\lambda(\tau)$ (τ はサイクル数)と表す。1サイクルで、 t は入力データ数だけ増加する。

3.3 組合せ法と競合学習, Kohonen 自己組織化特徴写像との比較

組合せ法 CM と、競合学習, Kohonen 自己組織化特徴写像による分割結果を比較する。ここで用いる Kohonen 自己組織化特徴写像の結合荷重変更量は、(2-4)式に変更を加えた(3-6)式を用いて計算することにする。

$$\Delta w_{ij}(t) = \begin{cases} \alpha(\tau)(x_i(t) - w_{ij}(t)) & (j = j^*) \\ \beta(\tau)(x_i(t) - w_{ij}(t)) & (j \neq j^* \text{ and } j \in N_{j^*}(\tau)) \\ 0 & (\text{それ以外}) \end{cases} \quad (3-6)$$

一般的には、近傍の範囲 $N_{j^*}(\tau)$ と $\alpha(\tau)$ は時間と共に減少させる。しかし、データ分割のグループ数(出力ユニット数)は非常に少数であるため、本研究では近傍の大きさを両隣のユニットに固定する。また、(3-6)式の学習係数は、近傍のユニットの結合荷重を変更する Kohonen 自己組織化特徴写像と、全ユニットの結合荷重を変更する CM の違いを明確にするため、CM と同じ $\alpha(\tau) = g_w$ 、 $\beta(\tau) = (1 - \lambda(\tau))g_l$ の値を用いる。その結果、両者の違いは漏えいを全出力ユニットにするか、両隣の出力ユニットにするかの違いのみになっている。各学習法を用いたニューラルネットワークで、全入力データをランダムな順序で入力層に提示し100サイクルの実験を行なう。

CM, 競合学習, Kohonen 自己組織化特徴写像による分割結果を比較するために、最適な分割が直接視覚的に決定できる次のような連続値と2値のモデルデータを用意し、これと第2章のアンケートデータを比較実験に用いることにした。

連続値のモデルとして、5つのグループからなる2次元ベクトルデータを作成した(29頁図3.2(c)の分割結果を参照)。各グループは、標準偏差0.5の正規乱数でグループ中心からのずれが決まる25個のデータからなり、この1グループ25個ずつの5グループへの分割を最適な分割とする。分割に用いたニューラルネットワークの入力層は2ユニット、出力層は5ユニットである。

2値のモデルとして、1から5までの4種類のトランプカード20枚を、同じ数字ごとの5グループに分割することを考え、それぞれのカードを表すための入力データは、トランプカードの色を表す2ビット(例えば赤:10, 黒:01), 種類を表す4ビット, 数字を表す15ビット(1から5までの数字をそれぞれ3ビットで表す), 合計21ビットの0, 1を21次元2値データとする。分割に用いたニューラルネットワークの入力層は21ユニット, 出力層は5ユニットである。

ニューラルネットワークに与えられる初期条件には、結合荷重の初期値と学習係数の値がある。結合荷重の初期値依存性を検討するために、学習係数 g_w , g_i を一定にし結合荷重に異なる初期値を与えた20通りのネットワークを用意する。これらのネットワークにKohonen自己組織化特徴写像, CMを適用し、それぞれで最適な分割が得られる割合を求める。更に、 $g_w=0.1, 0.2, 0.3$ の3通りに対して、 g_i を0.0005~0.05まで変化させて、学習係数の違いによるKohonen自己組織化特徴写像, CMのデータ分割性能比較を行なった。

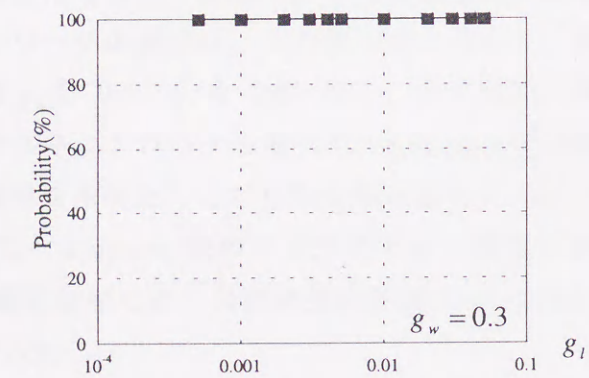
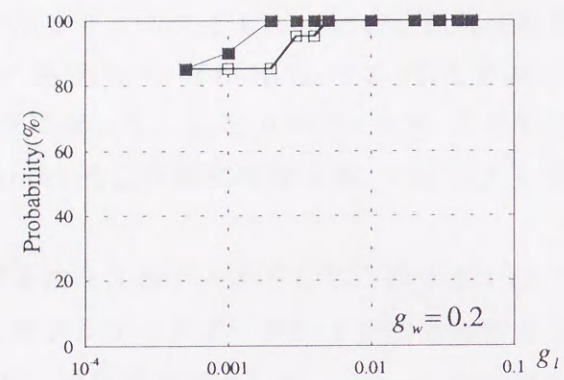
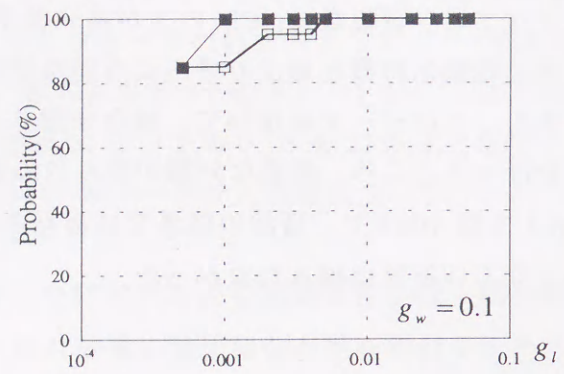
データの分布構造が未知な入力データとして、第2章で用いたアンケートデータを取り上げ、このアンケートデータの5分割を対象とし、Kohonen自己組織化特徴写像, CM, 更に従来法としてk-means法を加えた3つの手法による分割結果の比較を行なった。Kohonen自己組織化特徴写像, CMに用いたニューラルネットワークの構造は、入力層253ユニット, 出力層5ユニットである。学習係数 g_w を0.1, g_i を0.01とし、結合荷重の初期値を変えた100通りのニューラルネットワークを用いて、Kohonen自己組織化特徴写像, CMによる分割結果を求めた。これらの分割結果を、ニューラルネットワークの場合と同様に、k-means法のクラスタ中心に異なる初期値を与えた100通りの分割結果と比較した。分割結果の評価には、2.5.2で定義したsummed deviationを用いた。

3.4 結果

3.4.1 連続値モデルデータに対する結果の比較

連続値モデルデータを対象に、Kohonen自己組織化特徴写像とCMで最適な分割が得られる確率を比較した結果を図3.1に示す。このモデルデータで

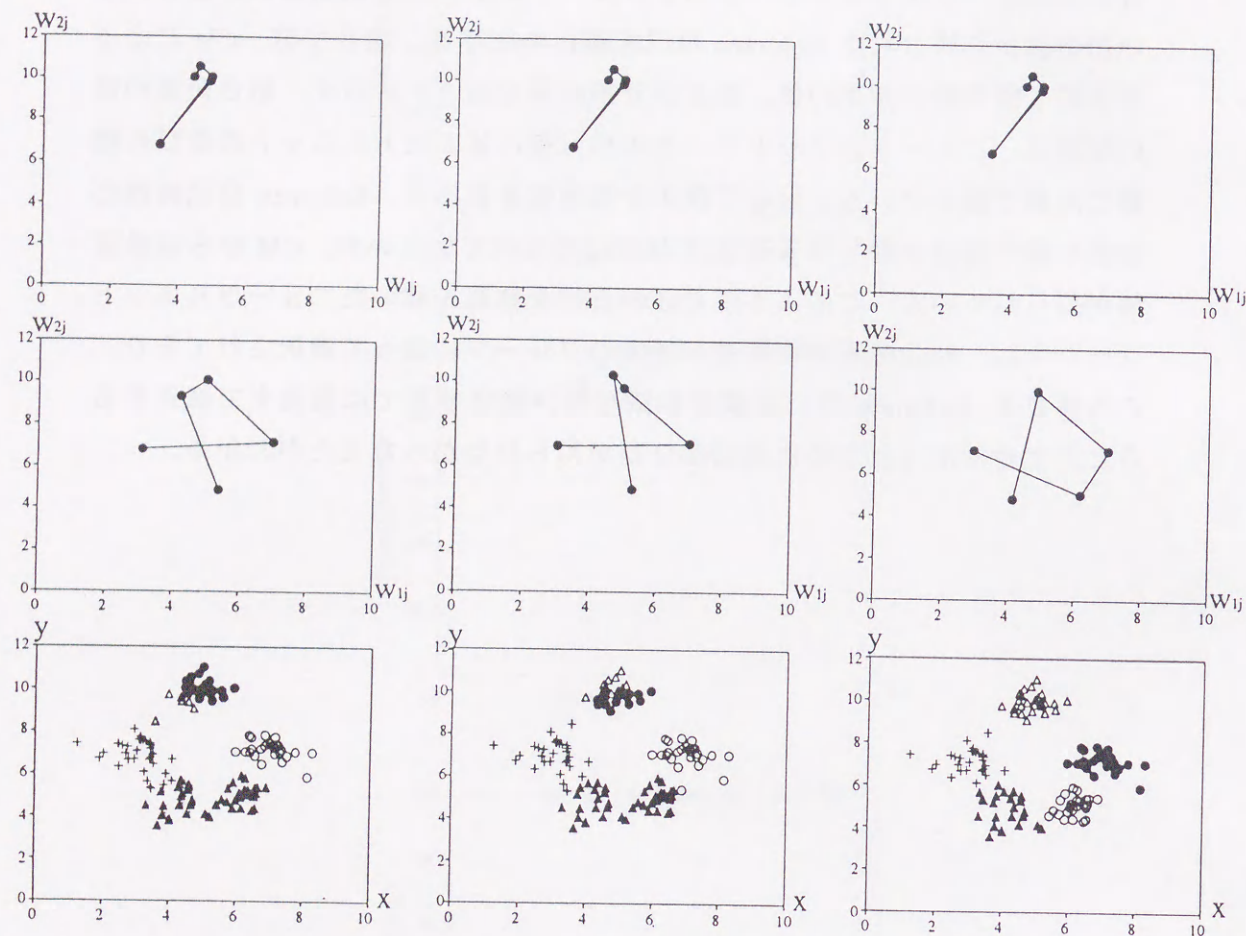
は、 g_w を0.3とすると g_i に関係なく、またCM, Kohonen自己組織化特徴写像どちらの手法でも100%最適解が得られる。 g_w が0.1, 0.2の場合では、 g_i の小さい値でKohonen自己組織化特徴写像のほうが不利になっている。実験を行なったすべての学習係数で、CMの方が高い確率で最適解が得られ、CMはKohonen自己組織化特徴写像よりさらに有効な学習法であることが明らかになった。また、このデータを用いて、競合学習ニューラルネットワークによる5分割を行ったところ、最適な分割が得られる確率は、 $g_w=0.1, 0.2$ では85%, $g_w=0.3$ では100%で、当然の結果ではあるがCM, Kohonen自己組織化特徴写像よりよい結果は得られなかった。



□ Kohonen 自己組織化特徴画像 ■ 組合せ法

図 3.1 組合せ法, Kohonen 自己組織化特徴画像で学習係数を変化させた場合の最適な分割が得られる確率 (連続値モデルデータ)

CM のほうが Kohonen 自己組織化特徴画像より最適な分割を得る確率が高い理由を検討するために, 同一の初期値に対し, CM の方が良好な分割結果が得られたニューラルネットワークを取り上げて, 結合荷重の値が学習と共にどのように変化したかを調べた. 1 例として, $g_w=0.1$, $g_l=0.001$ として学習したニューラルネットワークの 5 グループの結合荷重初期値の配置と, その初期値から得られた Kohonen 自己組織化特徴画像, 競合学習, CM による学習終了後の結合荷重の値, および分割結果を図 3.2 に示す. 結合荷重の値の配置は, ニューラルネットワークの出力層に並ぶ出力ユニットの並びの順番に直線で結んでいる. 図の下段の分割結果を見ると, Kohonen 自己組織化特徴画像や競合学習では最適な 5 分割は得られていないが, CM からは最適解が得られている. このように CM が良好な結果を導いたニューラルネットワークでは, 結合荷重の初期値が特定のグループに偏って選択されており, この偏りを Kohonen 自己組織化特徴画像, 競合学習では最後まで修正することができなかったために最適な分割が得られなかったことがわかる.



(a) Kohonen 自己組織化
特徴写像
($g_w = 0.1, g_l = 0.001$)

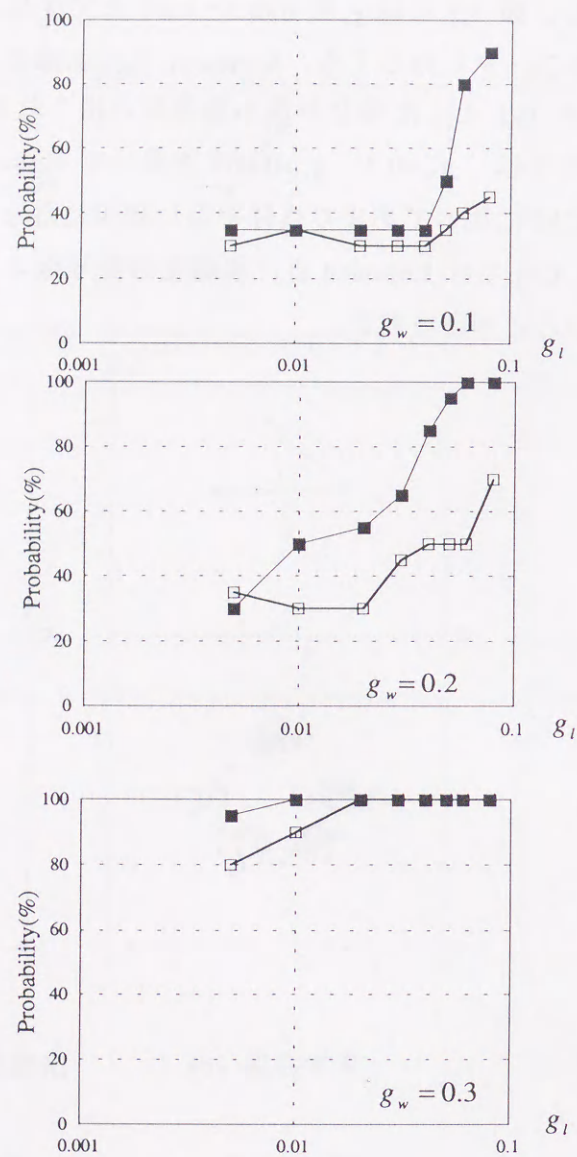
(b) 競合学習
($g_w = 0.1$)

(c) 組合せ法
($g_w = 0.1, g_l = 0.001$)

図 3.2 Kohonen 自己組織化特徴写像, 競合学習, 組合せ法による偏った結合荷重初期値からの連続値モデルデータの分割
(上段: 結合荷重の初期値, 中段: 学習終了後の結合荷重の値, 下段: 学習終了後の 5 グループ分割結果)

3.4.2 2 値モデルデータに対する結果の比較

トランプデータの 5 分割に対し, 連続値モデルデータと同様な方法により, Kohonen 自己組織化特徴写像と CM で最適な分割の得られる確率を比較した (図 3.3). ただし, 図 3.3 には g_l を 0.005~0.05 まで変化させた結果を表示した. このモデルデータに対しても, Kohonen 自己組織化特徴写像, CM 両手法ともに, g_w を 0.3 とした場合が最も最適解の得られる確率が高い. g_w が 0.1, 0.2 の場合では, $g_w=0.2, g_l=0.005$ を除いて Kohonen 自己組織化特徴写像のほうが CM に比べてあまり良好でない結果になっている. 2 値のモデルデータでも, CM では Kohonen 自己組織化特徴写像より最適な分割の得られる確率が高いことが示された.



□ Kohonen 自己組織化特徴写像 ■ 組合せ法

図 3.3 組合せ法, Kohonen 自己組織化特徴写像で学習係数を変化させた場合の最適な分割が得られる確率 (2 値モデルデータ)

3.4.3 アンケートデータに対する結果の比較

アンケートデータの 5 分割を, 結合荷重の初期値を変えた 100 通りのニューラルネットワークを用いて, Kohonen 自己組織化特徴写像, CM により実行しその分割結果を求めた. また, k-means 法のクラスタ中心に異なる初期値を与えた 100 通りの分割結果を求め, それぞれの結果に対し summed deviation を計算しその頻度分布を示した(図 3.4). 第 2 章の結果と同様に, k-means 法ではニューラルネットワークの結果に比べて summed deviation の値が広い範囲に分布し, 初期値によって得られる分割結果が大きく異なることを示している. Kohonen 自己組織化特徴写像, および CM の場合の分布は類似しているが, 同じ初期条件のニューラルネットワークで Kohonen 自己組織化特徴写像と CM による分割の summed deviation を比較したとき, その差が 0.1 以上のものが 3% 出現し, すべて Kohonen 自己組織化特徴写像の方が大きな値になった. これは, 図 3.4(a)で summed deviation の値が 24.05 以上に分布している部分である. これらのニューラルネットワークでは, 更に学習回数を増加しても Kohonen 自己組織化特徴写像の summed deviation が CM のそれに近づくことはなかった. これらは, 図 3.2 に示した例と同様に, 結合荷重の初期値の偏りを, Kohonen 自己組織化特徴写像のアルゴリズムでは最後まで修正することができなかったものである.

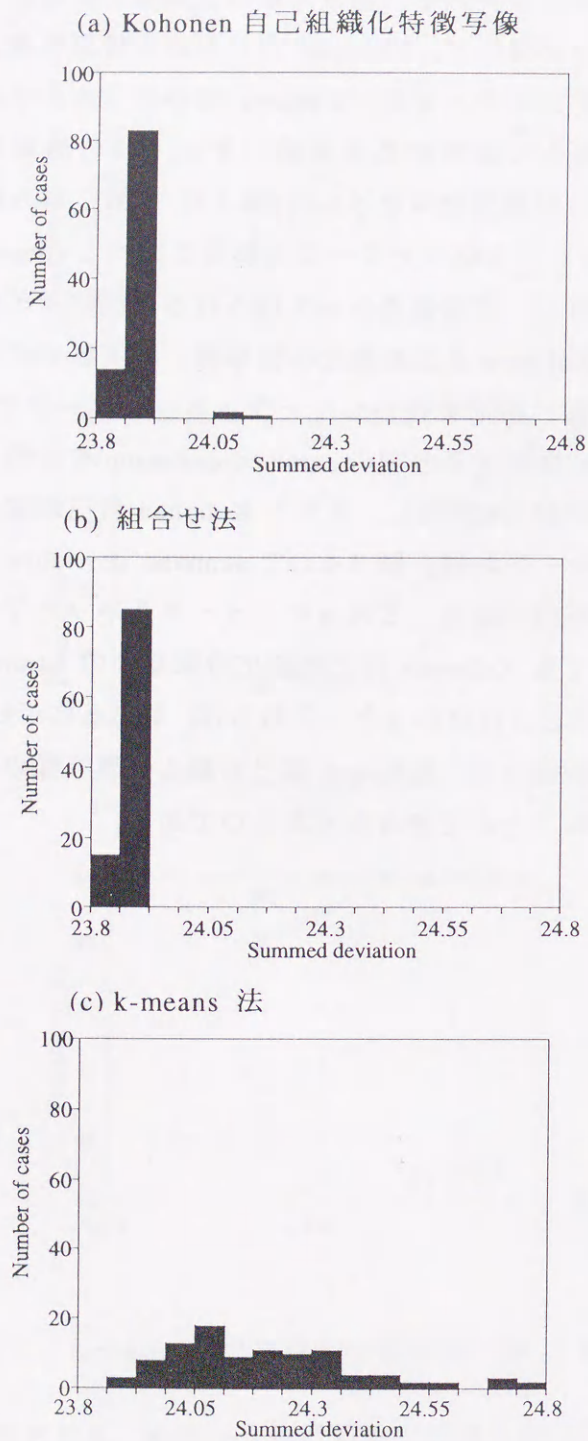


図 3.4 アンケートデータの 5 分割における summed deviation の分布

3.5 考察

競合学習と k-means 法は初期値に対する依存性が高くなり、実データでは最適な分割が得られにくい。これらと比較して、Kohonen 自己組織化特徴写像、CM は初期値によらず最適な分割を得る確率は高く、従って漏えいが初期値に対する依存性を低減する有力な手段であることが分かる。この結果は Kohonen 自己組織化特徴写像が競合学習よりよい結果を与えるという Snyder らの結論[14]とも一致している。またモデルデータおよびアンケートデータの両方で示されたように、CM では Kohonen 自己組織化特徴写像より初期値に対する依存性がさらに低い。この理由は、勝者とその近傍のみを入力ベクトルに近づけるという Kohonen 自己組織化特徴写像の近傍学習の方法が、結合荷重ベクトルの初期値が偏った場合、偏った結合荷重ベクトルの全空間への広がり方を妨げるように働くのに対し、全ユニットに漏えいを許す CM の方式は初期値の影響を緩和する方向に働くためである。一般的には、Kohonen 自己組織化特徴写像でも近傍の時間変化を工夫することによりこれを解決することは十分可能であるが、トポロジカルマッピングを必要としないアンケートデータのような分割では、最適な分割を得るために近傍のサイズを調整するのは不必要な努力と思われる。

3.6 まとめ

本章では、漏えい学習と競合学習を組合せた学習法である組合せ法について検討し、Kohonen 自己組織化特徴写像、競合学習、k-means 法と比較して、データ分割における初期値依存性が低い学習法であることを示した。特に、Kohonen 自己組織化特徴写像では結合荷重の初期値設定が偏っている場合偏りを修正することは難しいが、組合せ法ではこのような初期値に対しても最適解を得ることができる。この組合せ法を、人工的に作成したモデルデータと実際のアンケートデータの分割に適用し、その有効性を示した。

第4章

相互情報量により学習パラメータを調整した効率的データ自動分割アルゴリズム

4.1 緒言

第3章では、複雑な分布構造をもつデータの自動分割のアルゴリズムとして、漏えい学習と競合学習を組み合わせた学習法である、組合せ法 (Combination Method ; CM) について検討し、Kohonen 自己組織化特徴写像より更に初期条件依存性が低く、最適解が得られる確率が高い手法であることを示した。CM は漏えい学習の学習係数の大きさを、学習が進むにつれて、ある一定のシグモイド関数に従って緩やかに減少させる方法である。しかし、様々な分布構造をもつデータに対する自動分割のアルゴリズムとして CM を捉えた場合、一定のシグモイド関数に従って係数を変化させるのではなく、入力データに応じて学習係数の減少のさせ方を調整する方がより望ましい。本章では、この学習係数の調整を自動的に行うための1つの指標を提案する。

CM の学習アルゴリズムは、競合学習を基本にしている。Linsker によれば、競合学習のアルゴリズムは入出力間の相互情報量を最大にするように働く方式である[19]。このことは、基本的に競合学習アルゴリズムである Kohonen 自己組織化特徴写像によるトポロジカルマッピング形成の過程を見ることで、視覚的に捉えることができる。入力データ間の秩序が、学習が進むに連れて次第に出力層に組織化され、入出力間の相互情報量は増大していく。従って、入出力間の初期の相互情報量が大きければ短い学習で最適な分割が得られ、相互情報量が小さい場合には最適分割を得るのに十分な時間が必要であると考えられる。そこで、この入出力間の初期の相互情報量の大きさを、CM の学習係数を自動的に調整するための指標として用いることを検討し、更に、この指標に基づいた自動分割のアルゴリズムとして、相互情報

量により学習パラメータ（学習係数や学習の繰返し回数）を調整した組合せ法（Combination Method Adjusted by Mutual Information ; CMAMI）を提案する。CMAMI を用いたニューラルネットワークでは、データ集合をネットワークの入力層に提示するだけで、最適解を得るための適切な学習パラメータが入力データと初期条件に応じて自動的にネットワークに設定され、任意の初期条件から最適な分割が高い確率で得られる。いくつかのサンプルデータを用いて、初期条件依存性と、最適解（または最適近似解）が得られるまでの繰返し回数（学習効率）を他の手法と比較し、CMAMI の有効性を示す。

4.2 相互情報量により学習パラメータを調整する組合せ法（CMAMI）

CM を用いた対象データの分割では、入力データが非常に単純な分布構造をもつ場合には、全入力データを一回ずつ学習した時点でほぼ最適な分割が得られる。しかし、複雑な分布構造をもつ入力データでは、多くの繰返し回数を費やして結合荷重の修正を行わなければ、最適な分割は得られない。これらの経験的事実から、全入力データを一回ずつ学習した時点でどの程度最適な分割に近づいたかを測る指標を提案し、これを用いて学習係数および繰返し回数を調整する学習アルゴリズム CMAMI を構築する。CMAMI は、最適な分割にどの程度近づいたかを測る指標として入出力分布間の相互情報量を用い、CM の敗者の学習係数 g_l および勝者の学習係数 g_w の減少のさせ方をこの値に依存させて自動的に決めるアルゴリズムである。以下に、相互情報量の計算方法、および CMAMI の学習アルゴリズムについて述べる。

4.2.1 CMAMI における相互情報量の計算

次のような2つの確率事象系 A , B を考える。

$$A = \begin{pmatrix} a_1, & a_2, & \dots, & a_\alpha \\ p(a_1), & p(a_2), & \dots, & p(a_\alpha) \end{pmatrix}$$

$$B = \begin{pmatrix} b_1, & b_2, & \dots, & b_\beta \\ p(b_1), & p(b_2), & \dots, & p(b_\beta) \end{pmatrix}$$

$p(a_l)$ は事象 a_l ($l=1, \dots, \alpha$) の出現確率、 $p(b_m)$ は事象 b_m ($m=1, \dots, \beta$) の出現確率とする。このとき、2つの事象系 A , B 間の相互情報量は次の(4-1)式で定義される[32]。

$$I(A; B) = -\sum_{l=1}^{\alpha} p(a_l) \log_2 p(a_l) + \sum_{l=1}^{\alpha} \sum_{m=1}^{\beta} p(a_l, b_m) \log_2 p(a_l | b_m) \quad (4-1)$$

$p(a_l, b_m)$ は事象 a_l と事象 b_m の結合確率、 $p(a_l | b_m)$ は、事象 b_m のもとでの事象 a_l の条件付確率とする。相互情報量は A と B に関して対称 ($I(A; B) = I(B; A)$) であり、事象系 A と B に共通に含まれる情報の量とみなされる。

本研究では、入出力分布間の相互情報量を、入力層の各ユニットへの入力データの分布と、ニューラルネットワークによる分割結果のグループとの間で、入力ユニットごとに計算する。本来は、入力データの分布の特徴を十分記述しうる部分空間に射影した分布と出力分布の間の相互情報量を計算するのが望ましいが、自動的に処理するという点からはどの部分空間に射影するのが適切かを判断するのは難しいため、各入力ユニットごとの分布を用いる。入力層ユニット i に提示される全入力データの値 x_i^h ($h=1, \dots, N$) の取り得る範囲を、 M 等分したそれぞれの区間を事象 a_l ($l=1, \dots, M$) とし、その区間に入る入力データ x_i^h ($h=1, \dots, N$) の頻度分布から $p(a_l)$ を計算する。 N は入力データ数である。ニューラルネットワークの入力層に全入力データを提示して、その出力から各入力データの属するグループ b_m を求める。このグループを事象 b_m と考え、各グループに属するデータ数の分布から $p(b_m)$ を計算する。各グループ b_m ($m=1, \dots, K$) ごとに入力ユニット i の入力データ分布との結合確率 $p(a_l, b_m)$ 、条件付確率 $p(a_l | b_m)$ を求め、(4-1)式に基づいて入力ユニット i のデータ分布と出力分布との相互情報量 I_i ($i=1, \dots, n$) を計算する。全部で n 個の相互情報量が計算される。

4.2.2 学習係数の減少率を決定する $\lambda(\tau)$ の設定

各入力ユニットにおける分布と出力分布の間の相互情報量の最大値を求めることで、入力データの分布の特徴を最もよく表現している入力ユニットを選択することができる。しかし、各入力ユニットごとに計算される相互情報量が取り得る値の最大値は分割数に依存するため、CM の学習係数の変化のさせ方 $\lambda(\tau)$ を設定するための指標としてあまり適していない。そこで、 n 個

の相互情報量 I_i ($i=1, \dots, n$)の最大値 I_{\max} と、分割数 K の対数との比を次の(4-2)式で計算し、以後これを r 値とよぶ[20][21].

$$r = \frac{I_{\max}}{\log_2 K} \quad (4-2)$$

相互情報量 I_i が取り得る値の最大値は $\log_2 K$ となるため、 r 値は 0 と 1 の間の値をとる.

競合学習のアルゴリズムは入出力間の相互情報量を最大にするように働くという Linsker の指摘から[19], ここでは第 1 サイクルの学習終了後に r 値を計算し、この r 値を現在の分割が最適な分割にどの程度近づいているかを測る尺度として用いる. 以後、第 1 サイクルの学習終了後に計算した r 値を、単に r 値と表す. r 値が大きい時は 1 サイクルの学習で最適な分割がほぼ得られていることを表し、 r 値が小さい時は最適な分割が得られるまでには更に多くの繰返し回数を必要とすることを示している. 学習係数も、 r 値が大きい時は急激に減少させた方が、最適解が少ない繰返し回数で得られ効率がよいが、 r 値が小さい時は緩やかに減少させなければ最適な分割は得られない. そこで、CM における学習係数の減少の割合を決定する $\lambda(\tau)$ をシグモイド型増加関数とし、その形状をこの r 値を用いて次の(4-3), (4-4)式により決定する. ただし、 s , a は正の定数である.

$$\lambda(\tau) = \frac{1}{1 + \phi(\tau)} \quad (4-3)$$

$$\text{ただし、} \quad \phi(\tau) = \exp\left\{\frac{a(s(1-r)-\tau)}{s(1-r)-1}\right\} \quad (4-4)$$

s は与えられたどのようなデータ集合に対しても、最適な分割が得られるような繰返し回数を決めるための定数で、後の実験で検討を加えるが一般的には数十という値に設定される. また、 $\lambda(\tau)$ が増加関数になるためには(4-4)式の指数関数の指数部分の分母は正でなければならない. しかし、分母を $s(1-r)-1 = s(1-1/s)-r$ と変形し、 s に数十という値が設定されるという条件を考えると、 r が 1 に近い値になるとこの分母は 0 または負になることがわかる. そこで、 r 値がほとんど 1 に近い場合は競合学習でも十分最適解が

得られることから、 $r \geq 1-1/s$ の場合は $\phi(\tau) = 0.0$, 即ち $\lambda(\tau) = 1.0$ として競合学習を行うものとする. a の値は、第 1 サイクルにおける $\lambda(\tau)$ の値 $\lambda(1)$ が

$$\lambda(1) = \frac{1}{1 + \exp(a)} \quad (4-5)$$

となることから、 $(1-\lambda(1)) > 0.99$ を満たすよう $a = 5.0$ とする. この条件は、設定した g_i が $\lambda(\tau)$ によって第 1 サイクルの学習から減少することを避けるために設定した. このような値を設定したシグモイド型増加関数 $\lambda(\tau)$ は、図 4.1 に示すように r 値に応じて形状が変化する. ただし、図中の $\lambda(\tau)$ は、 $s = 40$ として計算している.

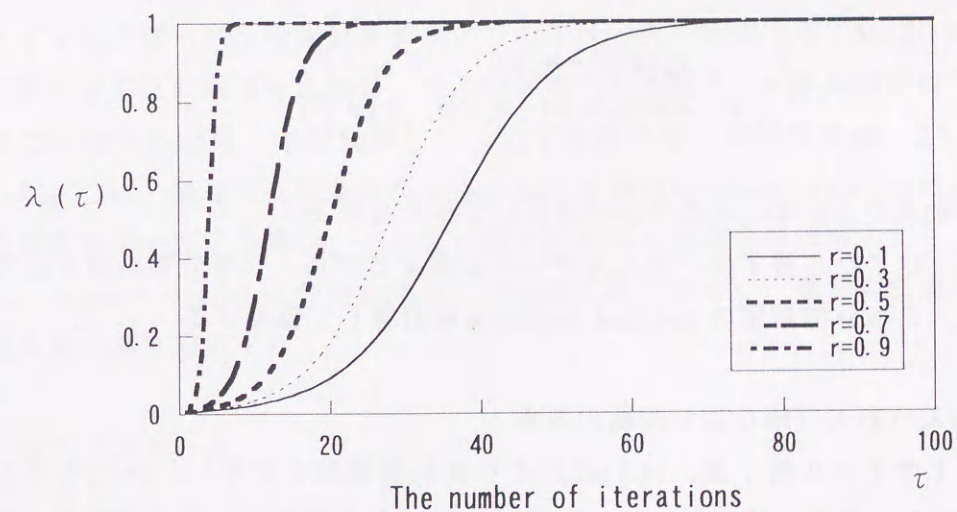


図 4.1 異なる r 値による $\lambda(\tau)$ の変化

更に、このように決定した関数 $\lambda(\tau)$ は、 $\tau = s(1-r)$ のとき $\lambda(\tau) = 0.5$, $\tau = 3 \cdot s(1-r)$ のとき $\lambda(\tau) \approx 1.0$ となることから、 τ が $3 \cdot s(1-r)$ 以上ではほとんど漏えいがない状態になる. この状態を更に繰返しても分割の変化は起こらないため、最適解を得るために必要な繰返し回数を $3 \cdot s(1-r)$ として学習を行う.

4.2.3 CMAMI の学習アルゴリズム

CMAMI は、組合せ法 CM のアルゴリズムを基礎としている。ネットワークは、CM と同様、2 層の階層型ネットワークを用いる。以下に、新しく提案する CMAMI の学習アルゴリズムを示す。

(1) 初期化

結合荷重を、ランダムに選出した入力データで初期化する。

(2) 第 1 サイクル

時刻 t における入力ベクトルを $\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_n(t))$ とし、出力層ユニット j への結合荷重ベクトルと入力ベクトルとの距離 $d_j (j=1, \dots, K)$ を(2-1)式で計算する。 d_j が最小となる出力ユニットを j^* とし、それぞれの結合荷重の値を次の(4-6)式で決まる $\Delta w_{ij}(t)$ だけ更新する。

$$\Delta w_{ij}(t) = \begin{cases} g_w(x_i(t) - w_{ij}(t)) & (j = j^*) \\ (1 - \lambda(1))g_l(x_i(t) - w_{ij}(t)) & (j \neq j^*) \end{cases} \quad (4-6)$$

(4-6)式の $\lambda(1)$ は(4-5)式で $a=5.0$ として与えられる。

第 1 サイクル終了後、各入力データの属するグループを分割結果として保存し、この分割結果の summed deviation を計算して保存する。

(3) $\lambda(\tau)$ および繰り返し回数の設定

第 1 サイクル終了後、(4-1)式により相互情報量を計算し、(4-2)式で r 値を求める。この r 値を用いて、(4-3)、(4-4)式から関数 $\lambda(\tau)$ を決定する。また、繰り返し回数を $3 \cdot s(1-r)$ と設定する。

(4) 第 2 サイクル以降

入力データをランダムな順序で入力層に提示し、次式に従って結合荷重の更新を行う。 j^* は(2)と同様、 d_j が最小となる出力ユニットである。

$$\Delta w_{ij}(t) = \begin{cases} (1 - \mu(\tau))g_w(x_i(t) - w_{ij}(t)) & (j = j^*) \\ (1 - \lambda(\tau))g_l(x_i(t) - w_{ij}(t)) & (j \neq j^*) \end{cases} \quad (4-7)$$

ただし、 $\mu(\tau)$ は r 値が大きい時は 0.0 として g_w を一定にして学習を行い、 r 値が小さい時は $\lambda(\tau)$ として g_w を g_l と同様に減少させる。この切り替えの値は別途検討する(4.3.4 参照)。

これを $3 \cdot s(1-r)$ サイクル繰り返し、各サイクル終了時点ごとに、得られている分割の summed deviation を計算し、保存されている summed deviation より値が小さい場合は、分割結果と共に更新する。

学習終了後、summed deviation の最小値が得られた分割結果を得ることができる。

4.3 結果

4.2.3 で提案した CMAMI について、データの分布構造と r 値の関係、適切な s の値、 g_w の減少方法と r 値との関係等を明らかにするため、データの分布構造の複雑さが段階的に変化するサンプルデータを用いて実験を行う。第 1 サイクルで得られた分割結果を用いてパラメータを設定する CMAMI の分割結果と従来法の結果を比較し、最適解が得られる割合と、最適解が得られるまでの繰り返し回数(学習効率)について検討する。本研究では、2.5.2 に示したように、異なる初期条件から得られた分割の summed deviation の最小値の複数個が一致する場合、その最小値に対応する分割を最適解とする。ただし、summed deviation の最小値は、学習終了時ではなく、学習の全過程にわたる最小値を選択する。

4.3.1 サンプルデータ

データの分布構造の複雑さが、 r 値にどのように反映されるかを示すために、段階的に分布構造が変化するサンプルデータを用意する。最初に、各成分が一様分布となる 10 次元のデータ(データ数は 200)を作成する。10 成分を x_1, x_2, \dots, x_{10} としたとき、次のようにして 6 通りのサンプルデータを作成する。まず、 x_1 だけ一様分布を 3 つの正規分布(平均値 0.5, 1.5, 2.5, 標準偏差はすべて 0.1)を合成した分布と置き換えたサンプルデータを作成する。更に、 x_1 と x_2 の 2 つの成分を 3 つの正規分布を合成した分布と置き換えたサンプルデータ、 x_1 と x_2 と x_3 を 3 つの正規分布の合成分布と

置き換えたサンプルデータ, ..., x_1 から x_6 までを 3 つの正規分布の合成分布と置き換えたサンプルデータの合計 6 通りのサンプルデータを用意する。また, 正規分布のオーバーラップの度合いを変えるために, 標準偏差を 0.1 から 0.5 まで変化させた 5 通りの正規分布を用意し, それぞれのサンプルデータの作成にこれらの 5 通りの正規分布を用いた全部で 30 通りの入力データ集合を作成する。 x_1 と x_2 の一様分布を 3 つの正規分布の合成分布と置き換えたサンプルデータの分布を, 図 4.2 に示す。 x_4 から x_{10} までの分布は, x_3, x_4 の分布に示すような一様分布になる。標準偏差が 0.1 のサンプルデータでは (図 4.2(a)), x_1, x_2 の分布は容易に 3 分割が可能な分布構造をもち, 標準偏差が 0.5 の場合は (図 4.2(b)), 3 分割の分布構造の特徴はあまり明確ではない。標準偏差が 0.2, 0.3, 0.4 では, 図 4.2(a) の分布から次第に図 4.2(b) の分布へ近づく。更に, 合成した正規分布をもつ成分を 1 つずつ増加することにより, 段階的に分布構造が変化するサンプルデータを作成することができる。明らかに, これらのデータ集合は, 標準偏差が小さいほど, また一様分布を 3 つの正規分布の合成分布で置き換えた次元数が多いほど, 容易に 3 分割が可能となる。

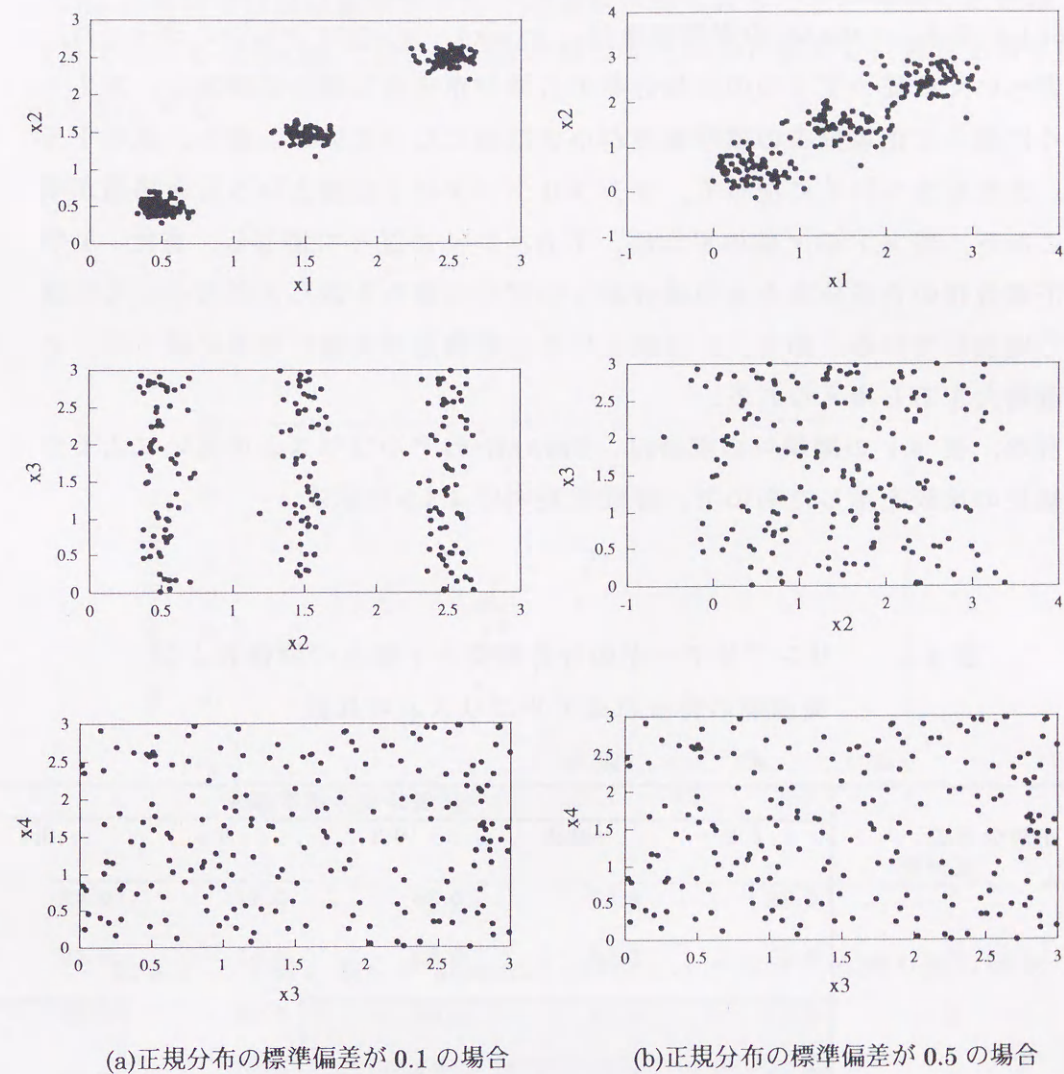


図 4.2 サンプルデータの分布

4.3.2 r 値とデータの分布構造

異なる 10 通りの初期条件を与えた CMAMI を用いて、4.3.1 で説明したサンプルデータの 3 分割を行い、第 1 サイクル終了後に計算した r 値の平均を表 4.1 に示す。CMAMI の学習係数は、 $g_w=0.1$ 、 $g_l=0.01$ とした。表 4.1 は、下方へいくに従って 3 つの正規分布の合成分布をもつ成分が増加し、左方へいくに従って正規分布の標準偏差が小さな値になっている。即ち、表の下方へ、また左方へいくに従って、サンプルデータの 3 分割という分布構造が明確になる。表 4.1 の r 値の平均は、下方へいくに従って増加し、また、3 つの正規分布の合成分布をもつ成分が 1 つだけの場合を除くと左方へいくに従って増加している。即ち、3 分割という分布構造が明確になるに従って、 r 値は増大すると考えられる。

なお、表 4.1 の網掛けの部分には、CMAMI のアルゴリズムの違いによる分割結果の比較を示したもので、詳細な説明は 4.3.3 に示す。

表 4.1 サンプルデータの分布構造と r 値との関係および最適解の得られるアルゴリズムの比較

正規分布の成分数	正規分布の標準偏差				
	0.1	0.2	0.3	0.4	0.5
1	0.36	0.37	0.36	0.37	0.42
2	0.55	0.56	0.53	0.51	0.49
3	0.67	0.66	0.60	0.56	0.53
4	0.97	0.95	0.84	0.69	0.59
5	0.97	0.97	0.86	0.72	0.62
6	1.0	0.97	0.87	0.73	0.63

- (I) すべてのアルゴリズムで最適解が得られる
- (II) g_w 一定、 g_l のみ減少させたほうが最適解が得られる
- (III) g_w, g_l 共に減少させたほうが最適解、精度のよい近似解が得られる

このようなサンプルデータの 3 分割を CM を用いて行い、最適解が得られたもののみを対象とし、 r 値と最適解が得られるまでの繰り返し回数の関係を示したのが図 4.3 である。ここで用いた CM では、 $g_w=0.1$ 、 $g_l=0.01$ 、 $\lambda(\tau)$ を(3-5)式に設定し、50 学習サイクルの繰返しを行った。この結果から、 $r > 0.8$ の範囲ではほぼ数サイクルで最適解が得られることが判明したため、以降で用いる CMAMI では、 $r > 0.8$ の場合 $\lambda(\tau)=1.0$ と固定し、競合学習を行うものとする。

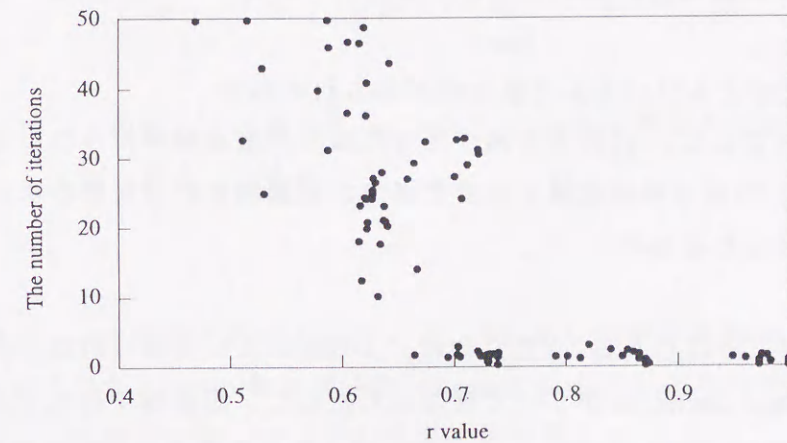


図 4.3 r 値と最小の summed deviation が得られる繰返し回数

4.3.3 s の値と g_w の減少方法に関する検討

CMAMI では、 g_w も g_l と共に減少させるアルゴリズムを提案している。両方の学習係数を減少させるのは Kohonen 自己組織化特徴写像を参考にしたものであるが、 r 値が大きな値の場合、 g_l と共に g_w も急激に小さくするとすべての結合荷重ベクトルがほとんど更新されなくなり、最適解を得るの

が困難になる可能性がある。従って、 r 値に応じて、 g_w を減少させるかどうかを判断する必要がある。また、繰り返し回数を決定する s の値は、ここでは $s=30,40,50$ について検討する。 s の値を検討する範囲は、一般的に最適な分割を得るためにはどの程度の繰り返し回数が必要かという点から決まる。あまり繰り返し回数が少ないと最適解を得るのが困難、また繰り返し回数が多いと最適解を得るのに時間がかかるだけということから、 $s=30,40,50$ を妥当な範囲とした。各々の s の値について、 g_w 、 g_l の両方を共に減少させる、 g_l のみを減少させるという 2 つのアルゴリズムで、異なる 10 通りの初期条件を与えたネットワークによりサンプルデータの 3 分割を行った。ただし、 g_w 、 g_l の両方を共に減少させる場合の g_w は、 $\mu(\tau)=\lambda(\tau)$ に従って減少させるものとする。ここで、 g_w 、 g_l を同じ関数に従って減少させるよう設定したのは、自動的に調整しなければならない要素を最小限にするためである。結果は、表 4.1 の網掛けの違いに示したように、入力データ集合は

- (I) すべてのアルゴリズムで最適解が得られるもの
- (II) g_w を一定にし、 g_l のみを減少させたほうが最適解が得られるもの
- (III) g_w 、 g_l の両方を共に減少させたほうが最適解または精度のよい近似解が得られるもの

の 3 グループに分けられる。(III) には、 s の値によって得られた分割結果の最小の summed deviation がすべて異なったために、最適解ではなく近似解(複数個の最小 summed deviation が一致せず、単一の最小 summed deviation の場合近似解と呼ぶ)が得られたケースも含まれている。表からそれぞれの場合の r 値をみると、(II) と (III) ではほぼ 0.6 のあたりを境にしていることから、ここで用いる CMAMI は、 $r > 0.6$ では g_w を一定にし、 g_l のみを減少させる、 $r \leq 0.6$ では g_w も g_l と同様、 $\mu(\tau)=\lambda(\tau)$ に従って減少させるというアルゴリズムを採用する。

(III) に含まれる 11 個のデータ集合を対象にして、適切な s の値の検討を行った。これらのデータ集合の 3 分割を、異なる 10 通りの初期条件を与えたネットワークにより行い、最小の summed deviation が得られるまでの繰り返し回数の平均と s の値の関係を図 4.4 に示す。

図中の●、■、△などの記号は、(III) に含まれる 11 個のデータ集合を区別するために使用している。繰り返し回数の平均は、 s の値に比例して増加

し、 s の値が小さい方が必要な繰り返し回数は少なくなっている。

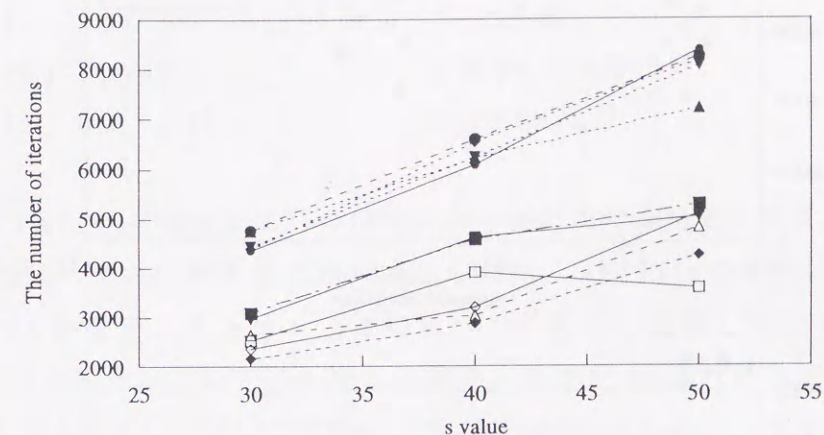


図 4.4 s の値による最小の summed deviation が得られる繰り返し回数(平均)の変化

更に、 $s=30,40,50$ それぞれについて、10 通りの初期条件を与えたネットワークから得られた 10 個の最小の summed deviation とそれが得られるまでの繰り返し回数を比較した。その代表的な例を図 4.5 に示す。 $r=0.36$ では、最小の summed deviation の分布に s の値による差は余りない。 $r=0.37$ では、 $s=30$ から得られる最小の summed deviation の分布(△印)は、 $s=40,50$ の場合に比べて大きな値をとる傾向にあり、より精度の高い解を得るには繰り返し回数が不足しているとみられる。 $r=0.56$ では、最小の summed deviation の分布は、 s の値の増大に伴って分散が小さくなり、最適解は $s=50$ (○印) で最も多く得られる。全体として、 $s=30$ では得られる結果にばらつきがあり、 $s=50$ では最適解が得られる可能性は高くなるが繰り返し回数も多く非効率的である。これらの結果から、 $s=40$ とすれば最適解が得られる割合と効率の両面でバランスがよいと判断した。

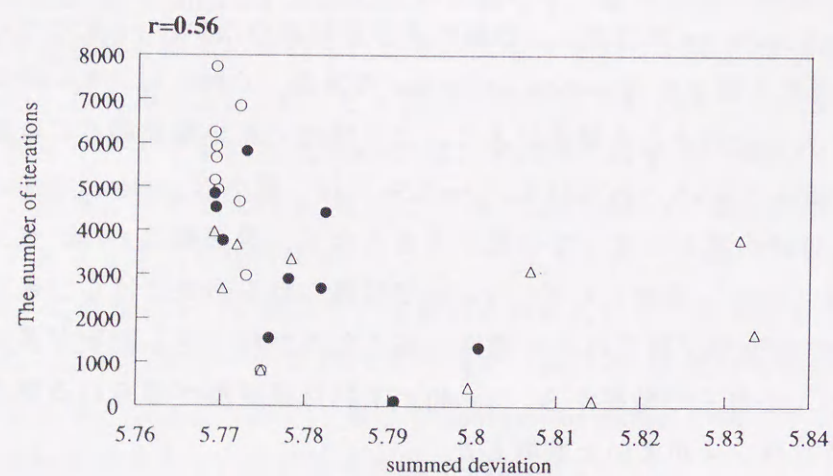
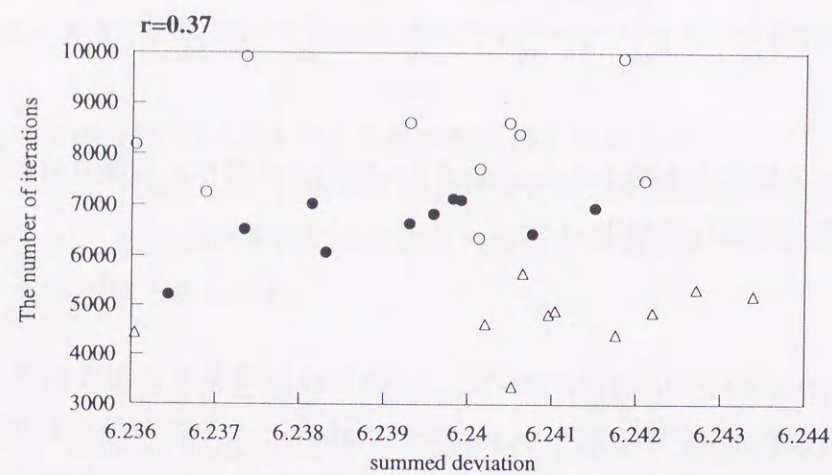
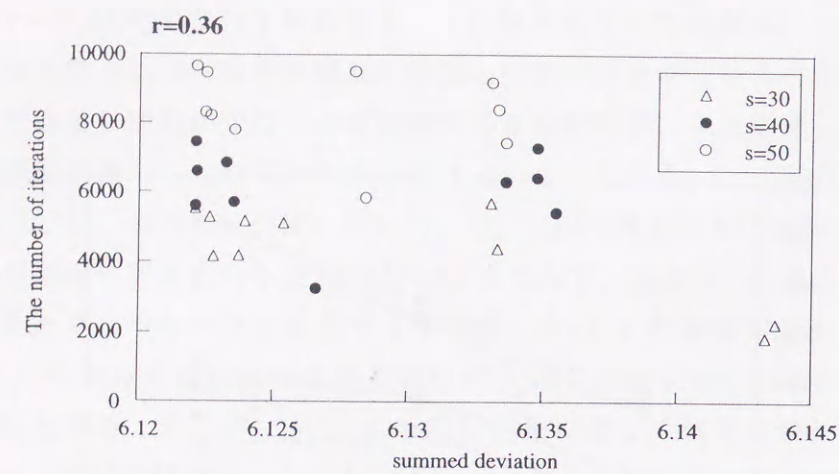


図 4.5 最小の summed deviation と繰り返し回数の比較

4.3.4 他の手法との比較

前節の結果をもとに、4.2.3 の CMAMI における s の値を 40 とし、4.2.2 に従って $\mu(\tau)$ および $\lambda(\tau)$ を以下のように設定する。

- (i) $\mu(\tau)=0.0, \lambda(\tau)=1.0$ ($r>0.8$)
- (ii) $\mu(\tau)=0.0$ ($0.6<r\leq 0.8$)
- (iii) $\mu(\tau)=\lambda(\tau)$ ($r\leq 0.6$)

ただし、(ii),(iii) の場合の $\lambda(\tau)$ は (4-3), (4-4) 式により求める。また、(4-3), (4-4) 式では $r \geq 1-1/s$ のとき $\lambda(\tau)=1.0$ としたが、 $r \geq 1-1/s$ の条件は $s=40$ の場合 $r>0.8$ に含まれるため、ここでは (i) に含めて表現している。この CMAMI によるサンプルデータの 3 分割を、異なる 50 通りのランダムな初期条件を与えたネットワークで行い、最小の summed deviation とそれが得られるまでの繰り返し回数を求める。比較のために、3 章で提案した CM のアルゴリズムと、自動的にデータ分割可能な手法である k-means 法を取り上げ、これら 3 手法による結果を比較した。得られた結果の中で、3 手法の違いが比較的捉えやすい $r=0.36$ の結果を図 4.6(a) に示す。k-means 法は極端に少ない繰り返し回数で収束するが、得られた最小の summed deviation は非常にばらつきが大きい。CM と CMAMI は、k-means 法に比べると得られた最小の summed deviation のばらつきは非常に小さくなり、最適解が得られる割合が高いことが分かる。更に、CMAMI は最適解を得る割合や最適解が得られるまでの繰り返し回数の両方で、CM より良好な結果を示すといえることができる。

$r=0.51$ と 0.66 の場合の CM と CMAMI の比較を、図 4.6 (b),(c) に示す。図 4.6(a) ほど明確ではないが、 $r=0.51$ でも最適解を得る割合や最適解が得られるまでの繰り返し回数の両方で、CMAMI の方がかなりよい結果が得られている (図 4.6 (b))。一方、 $r=0.66$ の場合には、やや CMAMI の結果にばらつきはあるが、CM と比較して最小の summed deviation が多く得られているように見える (図 4.6 (c))。このことを確認するために、図 4.6(a),(b),(c) のそれぞれ 50 個の最小の summed deviation 中で、最適解が得られたものの割合を計算した結果を表 4.2 に示す。最適解が得られる割合は、すべての r の値に

対し、やはり CMAMI が最も高いことがわかる。CMAMI は他の手法に比べ入力データの分布構造に応じて柔軟に分割アルゴリズムを調整し、最適解を効率よく発見しているといえる。

表 4.2 50 通りのネットワークから得られた最適解の割合

	r=0.36	r=0.51	r=0.66
k-means 法	0%	0%	0%
CM	0%	0%	22%
CMAMI	40%	8%	34%

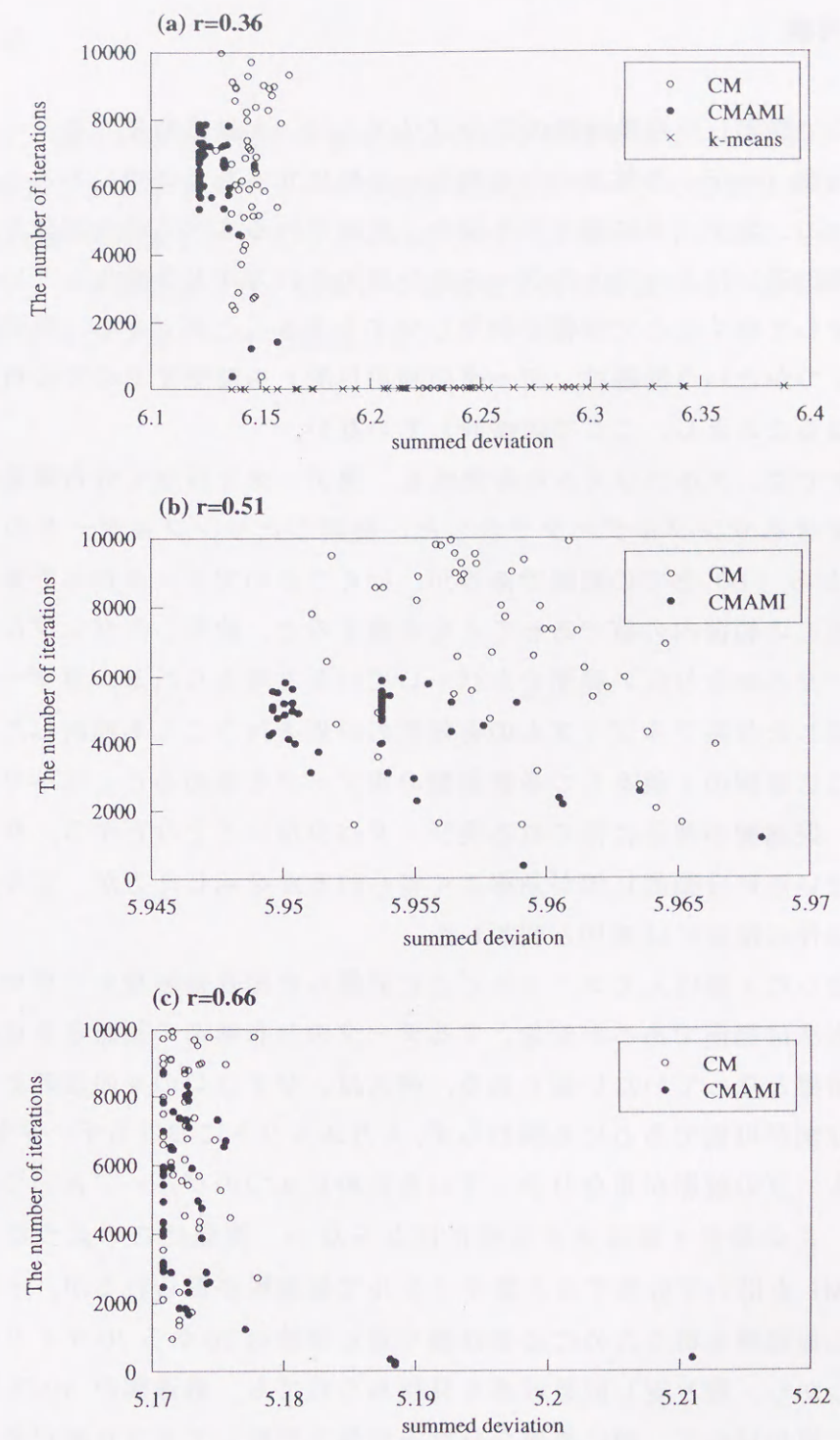


図 4.6 k-means 法, CM, CMAMI により得られる分割結果

4.4 考察

ここで提案した自動分割のアルゴリズムは、大量に発生するデータの大まかな特徴(=データ集合の分布構造)を抽出するために用いることを前提にしている。実データに適用する場合、想定されるすべての分割数を仮定し、分割数の違いによって入力データの分割のされ方がどう変化していくのかを予め示しておくことで情報を利用しやすくすることができる。最適な分割数はいくつかという問題は、データの利用目的とも関係するので分析者の判断に委ねることとし、ここでは検討していない。

本章では、アルゴリズムの有効性を、実データではなく分布構造が明確に理解できるサンプルデータで示した。使用したサンプルデータの r 値は約 0.35 から 1.00 までの範囲であるが、いくつかの実データから計算した r 値もほぼこの範囲内の値であることを考慮すると、使用したサンプルデータは実データのかなり広い範囲をカバーしていると考えられる。実データを用いて提案した分割アルゴリズムの有効性の評価を行うことも検討したが、このような広範囲の r 値をもつ多数種類の実データを集めることは不可能に近いこと、最適解が容易に得られる実データは少ないことなどから、サンプルデータでいかに自動的に解が効率よく得られるかを示した方が、アルゴリズムの有効性の検証には適切と判断した。

提案した r 値は入力ユニットごとに計算した相互情報量から求めるため、計算方法は簡便であるが対象とするデータの分布構造の複雑さを正確に反映する指標となっていない面もある。例えば、サイコロの 4 の目のように明白な 4 分割が可能であるにも関わらず、入力ユニットにおけるデータ分布では、各グループの射影が重なり合っているために 4 つのグループという構造は見えず、この場合 r 値は大きな値にはならない。実際にこのようなデータを CMAMI を用いて分割すると数サイクルで最適解が得られるが、 r 値から計算した最適解を得るために必要な繰り返し回数は 20 から 30 サイクルにもなる。しかし、繰り返し回数は多く見積もられても、最適解が 100% 得られることに違いはなく、逆に複雑な分割を容易と判断して適当な解が最適解と判断される誤りを犯すよりはるかによい。また実データの場合、すべての座標軸への射影に複数のグループが重なり合うという状況はほとんど起り得ないため、実際には r 値によりデータの分布構造を捉えることは十分可能である。

4.5 まとめ

本章では、漏えい学習と競合学習を組み合わせた学習法に、入出力分布間の相互情報量による学習パラメータの調整を加えたデータの自動分割アルゴリズム CMAMI を提案した。この CMAMI では、入力データに応じて自動的に学習パラメータが設定され、効率よく最適解または精度のよい近似解が得られることをサンプルデータで確認した。

第5章

ニューラルネットワークの入出力間相互情報量に基づくデータ自動分割システム

5.1 緒言

データマイニングの分野では、データ集合からそのデータ集合に内蔵する種々の情報を自動抽出できる発見型手法の提案が必要とされている[1][5]。即ち、データの分布構造の特徴をありのままに客観的に捉え、一般的な特徴を抽出するデータ分割の手法である。従来データ分割には k-means 法が用いられているが、データマイニングでは自己組織的にデータの分類を実行するニューラルネットワークによるデータ分割が注目されている[16]。既に、いくつかの企業がニューラルネットワーク・ユーティリティを商品化しているが[1][16]、データ分割に用いられているのは一般的に Kohonen 自己組織化特徴写像が多い。Kohonen 自己組織化特徴写像はデータ分割に大変有効な手法とされているが[14]、パラメータの調整など分析者の判断に依存する部分も多い。そこで第4章では、このような分析者の介入をほとんど必要としない、ニューラルネットワークの競合学習に基づいた効率的データ自動分割アルゴリズムを提案した。

データ分割では、グループ内のデータ間の類似度が高くなる分割結果を得ることが、データ分布構造の明確な特徴抽出につながる。しかし、グループ内の類似度は分割数の増加と共に高めることが可能なため、異なる分割数の結果に対する評価基準としては適当でない。例えば、隣接する2つのグループからなるデータ集合を3分割する場合、グループ内の類似度を最大化する3分割の結果を得ることは可能で、しかもグループ内の類似度は2分割の場合のそれより高められる。しかし、3分割の結果が、データ集合の分布の特徴を捉えているとは言えない。データの分布構造の特徴を捉えた分割数を推定するためには、様々な分割数に対する最適分割結果の中で、どれが最もよ

く特徴を捉えているかを判断しなければならない。

本章では、データの分布構造の特徴を捉えた分割数を自動的に推定するための指標を提案し、この指標をもとにデータ自動分割システムを構築することを目的とする。効率的データ自動分割アルゴリズムでは、学習係数を自動的に調整するための指標として r 値を提案した。 r 値は入力データ集合と分割結果との間の相互情報量であるため、 r 値を分割数ごとに比較した場合、データの分布構造の特徴を最もよく捉えた分割数では大きな値をとると考えられる。そこで、 r 値をデータの分布構造の特徴を捉えた分割数を推定するための指標として用いることを検討し、 r 値による分割数の推定まで含めた新しい統合的なデータ自動分割システムを提案する。更に、このシステムを財務データによる地方銀行の分類に適用し、推定された分割数の妥当性を主成分分析の結果を用いて示す。

5.2 最適分割数の推定問題の背景

画像の領域分割のような問題では、分割数自身が未知であり、何らかの方法でいくつに分割するのが適切かを判断しなければならない。第2章にまとめたデータ分割の既存の手法は、予め分割数を設定するか、または階層的クラスタリングのように、想定されるすべての分割数を前提としたものであり、何れも最適分割数の判断まで行えるものではない。

一定の条件のもとで、最適分割数を決定する手法は、いくつか知られている。統計学的手法の ISODATA 法[27]では、グループに属するデータ数の下限や、グループの標準偏差の上限、クラスタ間距離の下限など多くの条件に適切な値が設定されると、それらを満たす最適分割数を決定することができる。これは、クラスタ数の初期値を与え k -means 法によりクラスタリングを行い、条件を満たしていないクラスタを統合・分割するアルゴリズムに基づいて実現される。また、Gonzalez のアルゴリズム[33]を用いると、グループの直径を一定の値以下にする分割数を決定することができる。しかし、これらの手法では、分割数の決定のために何らかの条件を設定する必要があり、得られる分割結果もこれらの条件に依存する。

一方、このような条件設定をせずに、評価関数を用いて最適分割数を推定する手法がある。一つはクラスタの直径を評価基準とし、全クラスタの直径の最大値を最小化し、この最小値が得られる分割数を最適分割数とする方法

である。この最適分割数を決定する問題は、2次元以上のデータに対しては NP 困難となることが知られている[34]。これに対し、Gonzalez は直径の最大値が最小の解の高々2倍で抑えられる近似的解法を提案している[33]。比較的簡単なアルゴリズムであるためクラスタ中心の初期値設定にも利用されているが、この方法ではクラスタ内のデータ分布の状態まで評価することはできない。評価関数を用いて最適分割数を推定するもう一つの方法は、クラスタ内のデータ分布の状態を示すクラスタ内分散を評価基準として用いる方法である。本研究では、分割結果の評価を summed deviation で行っており、これはクラスタ内分散による評価に等しい。しかし、クラスタ内分散は分割数の増加と共に減少する評価基準であるため、クラスタ内分散の最小化のみで最適分割数を推定することは難しい。そこで、クラスタ内分散と共にクラスタ間分散も用いて評価する方法がいくつか提案されている。パターン認識の分野で、画像圧縮や領域分割に用いられている自動しきい値選定法では、1次元データに対し、グループ内分散とグループ間分散の比を最大にする分割数を最適分割数として選定している[35]。これを多次元に拡張する手法も提案されているが、単純に適用すると各次元ごとの最適分割数の次元数倍という大きな値の分割数になるため、更に階層的クラスタリングによってそれぞれのクラスタを統合することが行われている。そして、このような拡張をした場合の最適分割数の評価は、非常に困難な問題とされている[36]。多次元データに対しては、Coleman らが多次元特性をもつ画像の領域分割の手法として、 k -means 法を用いて分割を行い、分割結果のグループ内分散とグループ間分散の積を最大とする基準により最適分割数を推定する方法を提案している[37]。しかし、グループ内分散とグループ間分散の和は全分散であることから、この分散の積の最大値はそれぞれの分散が全分散の $1/2$ になるときに達成される。このような相補的關係にあるグループ内分散とグループ間分散を評価基準に用いることは、分布構造のありのままの特徴を捉えるという目的に対してあまり適しているとは言えない。何れにしても、制約条件を設定せずに、多次元データの最適分割数を推定する問題は、未解決の難しい問題の一つである[36]。このような背景から、本章ではデータ集合がもつ分布構造の特徴を、最もよく捉えた分割数を推定するための1つの指標を提案し、その有効性を示す。

5.3 データの分布構造の特徴を捉えた分割数を推定するデータ自動分割システム

効率的データ自動分割アルゴリズムでは、データの分割数は予め設定されることを前提としている。ここではこのアルゴリズムをもとに、データの分布構造の特徴を捉えた分割数の推定まで含めた統合的なデータ自動分割システムの構築を提案する。

最初に、データの分布構造と r 値の関係を考える。 r 値は、入力データの各成分の分布と、効率的データ自動分割アルゴリズムの第 1 サイクル終了後に得られる分割結果との相関を表す相互情報量から計算される。様々な分割数を指定してデータ分割を行った場合、入力データの分布の特徴を捉えた分割数では、学習の早い段階で最適な分割結果に近い出力が得られると考えられる。言い換えれば、この場合の分割結果は、他の分割数の場合の第 1 サイクル終了後の分割結果より、入力データと高い相関をもち、その r 値は他の分割数の r 値より大きな値を示す。このようなことから、 r 値はデータの分布構造の特徴を捉えた分割数を推定するための指標として、十分有効であると考えられる。そこで、 r 値によって分割数をどの程度推定することが可能かを、人工的に作成したサンプルデータを用いて調べる。サンプルデータは 5 次元のデータで、4.3.1 で説明したものと類似の方法で作成した。

5.3.1 データの分布構造を捉えるための指標としての r 値

各成分が一樣分布となる 5 次元のデータ（データ数は 400）の 2 分割から 8 分割までを、効率的データ自動分割アルゴリズムにより実行した。ネットワークの結合荷重に 10 通りの異なる初期値を与え、得られた分割結果の summed deviation を計算する。それぞれの分割数ごとに、10 通りの初期値から得られた分割の中から、summed deviation が最も小さな値となったものを選択し、その r 値と summed deviation の値を分割数ごとに示す（図 5.1）。使用したデータは、分布構造に特徴はなく特徴を捉えた分割数は特定できない。この場合、 r 値も summed deviation も、分割数の増加に伴って単調に減少している。

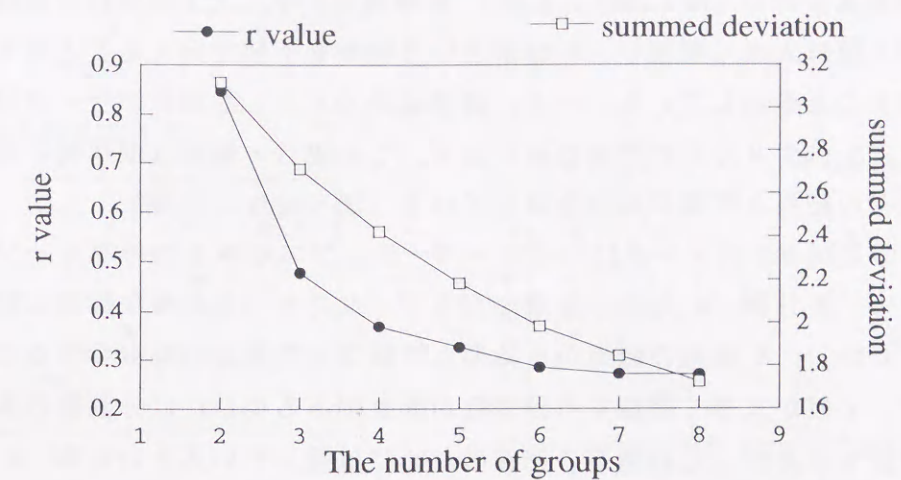


図 5.1 一樣分布 5 次元データの分割数と r 値

次にデータの分布構造がわかっているデータに対し、いくつかの分割数を指定してデータ分割を行い r 値の変化を調べてみる。各次元の成分が 8 つの正規分布（それぞれの平均値を 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0 とし、標準偏差はすべて 0.1 とした）を合成した分布に従う 5 次元のデータ（データ数は 400）をもとに、8 つの正規分布の標準偏差を 0.3, 0.5 と大きくした分布に従うデータを作成し、合計 3 種類のデータを用意する。標準偏差 0.1 としたデータでは、完全に分離した 8 つのグループからなり（図 5.2(a), 左側）、これらの 8 つのグループが得られる 8 分割がデータの分布構造の特徴を捉えた分割数となる。標準偏差が大きくなるにつれ各グループの分布が広がり、標準偏差 0.5 では隣接するグループ間にかなりのオーバーラップが生じており（図 5.2(c), 左側）、散布図から分布の特徴を捉えた分割数は判断できない。

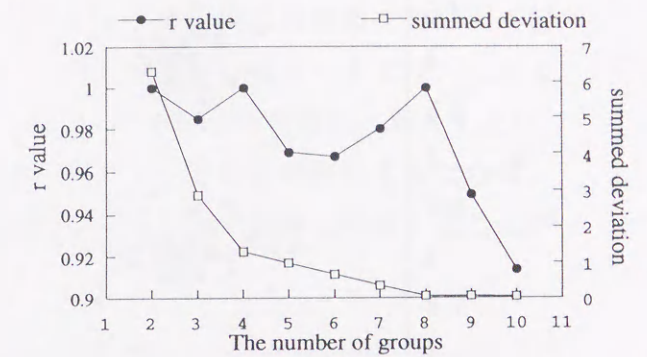
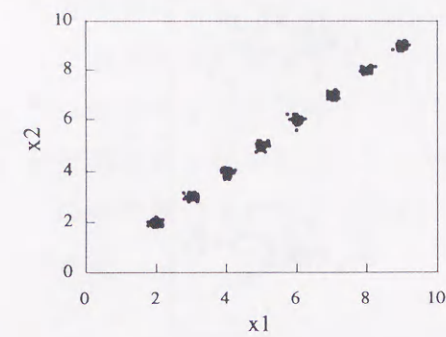
このデータの 2 分割から 10 分割を一樣分布に対するのと同様の方法で行い、その r 値と summed deviation を分割数ごとに示す（図 5.2, 右側）。何れの場合も、summed deviation は単調に減少し、summed deviation ではデータの分布構造を捉えた分割数を推定することはできない。8 つの正規分布の標準偏差が 0.1 のデータでは、2 分割、4 分割、8 分割で r 値が 1.0 となり、 r 値は一樣分布のデータのように単調に減少する傾向はなく増減を繰り返している（図 5.2(a), 右側）。標準偏差が 0.3 のデータでは、6 分割、8 分割に r 値

の増加が見られる(図 5.2(b), 右側). 標準偏差が 0.1, 0.3 の何れの場合も 8 分割の r 値は大きく増加し, 8 分割という特徴を r 値で捉えることが十分可能であることを示している. 一方, 標準偏差 0.5 とした場合のデータ分布の構造ではもはや 8 分割の特徴は存在せず, この場合 r 値はほぼ単調に減少し一様分布の結果と同様の傾向を示している(図 5.2(c), 右側).

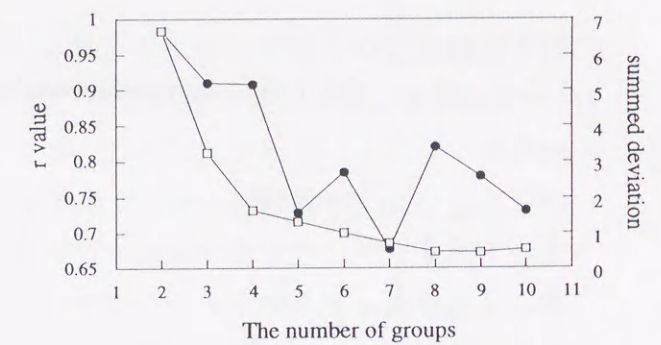
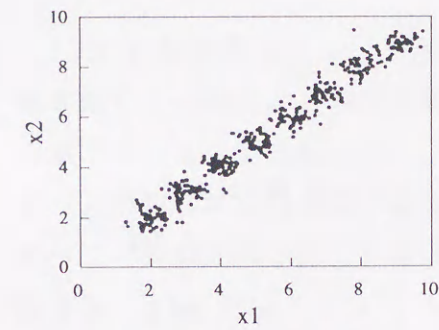
標準偏差が 0.1 のデータは全くオーバーラップのない 8 つのグループからなるため, 2 分割, 4 分割の結果からもデータのもつ大まかな特徴は捉えられる. しかし, 8 分割の結果からは最も詳細な分布構造の特徴を得ることが可能で, r 値が大きく増加する分割数が複数個ある場合には分割数の多いほうを選択する方が, 分布構造の把握のためには適しているといえる. 8 分割から更に分割数が増加すると, 8 つのグループの何れかが 2 つに分割され相互に比較的類似性の高いグループが複数個現れることになり, このような分割結果からデータの特徴を捉えるのはあまり適切とは言えず, r 値も 9 分割, 10 分割では増加していない.

r 値は, 入力データの各成分ごとに計算された相互情報量の最大値から計算されるが, 最大値を選択することで分割結果と最も相関の高い成分を見つけ出している. 即ち, この成分の分布と分割結果の相関を評価していることになる. 標準偏差 0.3 の場合の結果は, 入力データの各成分で分割結果のグループがかなり近接した分布を示しても, 分布構造の特徴を捉えた分割数を r 値により推定することが可能であることを示している. 一方, 8 分割という特徴をもたないと見られる標準偏差 0.5 のデータでは, 8 つの正規分布から作成したデータであるにも関わらず 8 分割のところで r 値は増加していない.

(a) standard deviation 0.1



(b) standard deviation 0.3



(c) standard deviation 0.5

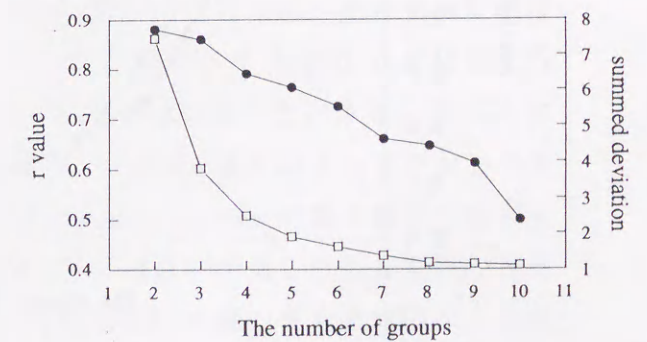
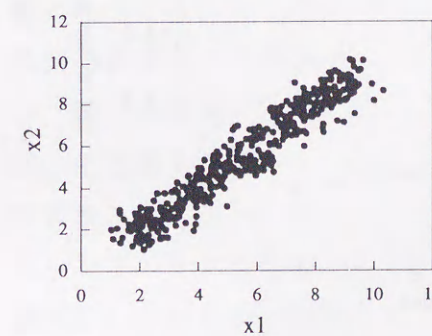


図 5.2 合成された正規分布 5 次元データの分布と r 値

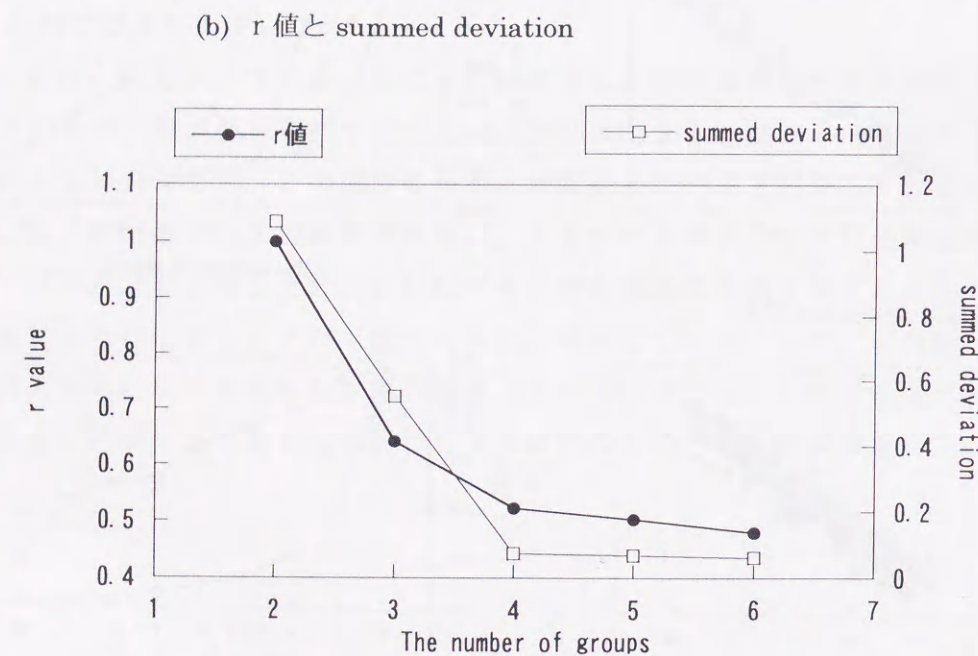
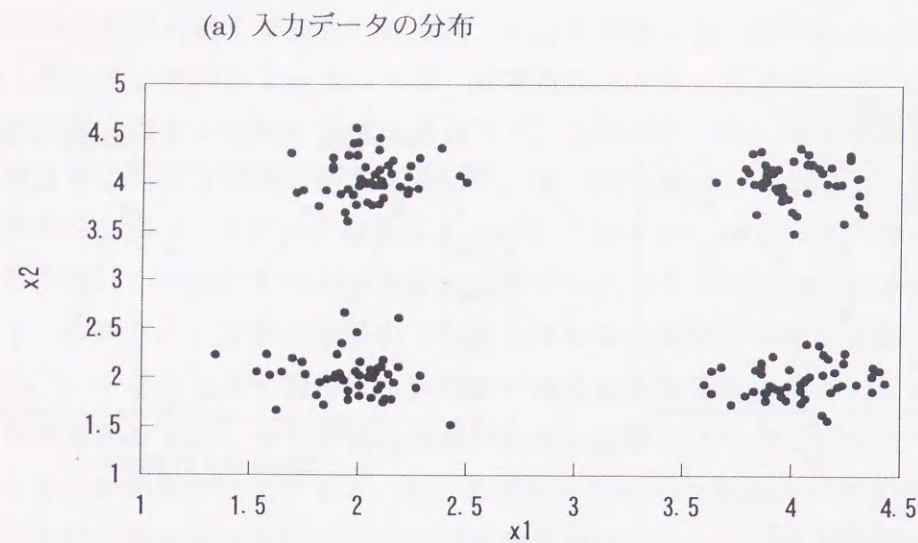


図 5.3 r 値により分割数を特定することができない例

一方、図 5.3(a)のようなグループからなるデータの分布構造の場合では、各成分分布がいくつかのグループで互いにほとんど重なり、どの成分分布にも 4 グループという分布構造が現れないため、r 値では分布構造の特徴を捉えることはできないと考えられる。実際、このデータの 2 分割から 6 分割を

行った結果、それぞれの分割数における最適解の r 値と summed deviation は図 5.3(b)に示すようになった。2 分割における r 値は 1.0 となったが、他の分割数の r 値は単調に減少し、データがもつ 4 分割という分布構造の特徴は抽出することはできなかった。このように、各成分分布が複数個重なるような分布構造をもつデータ集合では、r 値により分布構造の特徴を最もよく捉えた分割数を推定することは難しい。しかし、多くの実データでは、このような特異的な分布構造をもつことは少なく、例外的な分布構造と考えてよい。各分割数の r 値を比較したとき、r 値の増加する分割数が存在するならば、その分割数はデータの分布構造の特徴をよく捉えた分割数と判断することができる。

5.3.2 最適な分割数を推定するデータ自動分割システム

5.3.1 の結果をもとに、r 値により対象とするデータ集合の分布構造の特徴を最もよく捉えた分割数を推定する、データ自動分割システムを構築する。提案するシステムの概略を図 5.4 に示す。このシステムは、入力データ集合を与え想定される分割数の最大値 K を設定すると、2 分割から K 分割までのすべての分割を第 2 章に示した効率的データ自動分割アルゴリズムにより実行する。2.5.2 に示したように、本研究ではいくつかの異なる初期値から得られた summed deviation の最小値が一致する場合、その最小値の分割を最適解とみなしている。そこで、データ自動分割システムでは、結合荷重に異なる初期値を与えた複数通りのネットワークを用いて 2 分割から K 分割を行う。以下の実験では 20 通りのネットワークを用いることにする。図 5.4 に示した自動分割のループ回数 1 から 20 は、この 20 通りのネットワークで実行することを意味している。そして、summed deviation の最小値が一致するネットワークが複数個得られた分割数、即ち最適解が得られた分割数のみを選択し、これらの分割数に対しそれぞれ最適解の r 値の平均を求める。分割数を 2 から K まで次第に増加させたときの平均 r 値の変化を調べ、この平均 r 値が増加する分割数を見つけ出し、データの分布構造の特徴を捉えた分割数とする。この分割数の平均 r 値、summed deviation および分割結果を表示して終了する。ただし、summed deviation の最小値が一致するネットワークが複数個得られない分割数に対しては、最適解が得られなかったものと判断する。また、平均 r 値が増加する分割数が存在しなかった場合、入力デー

タ集合の分布構造の特徴を捉えた分割数は推定できなかったとして終了する。

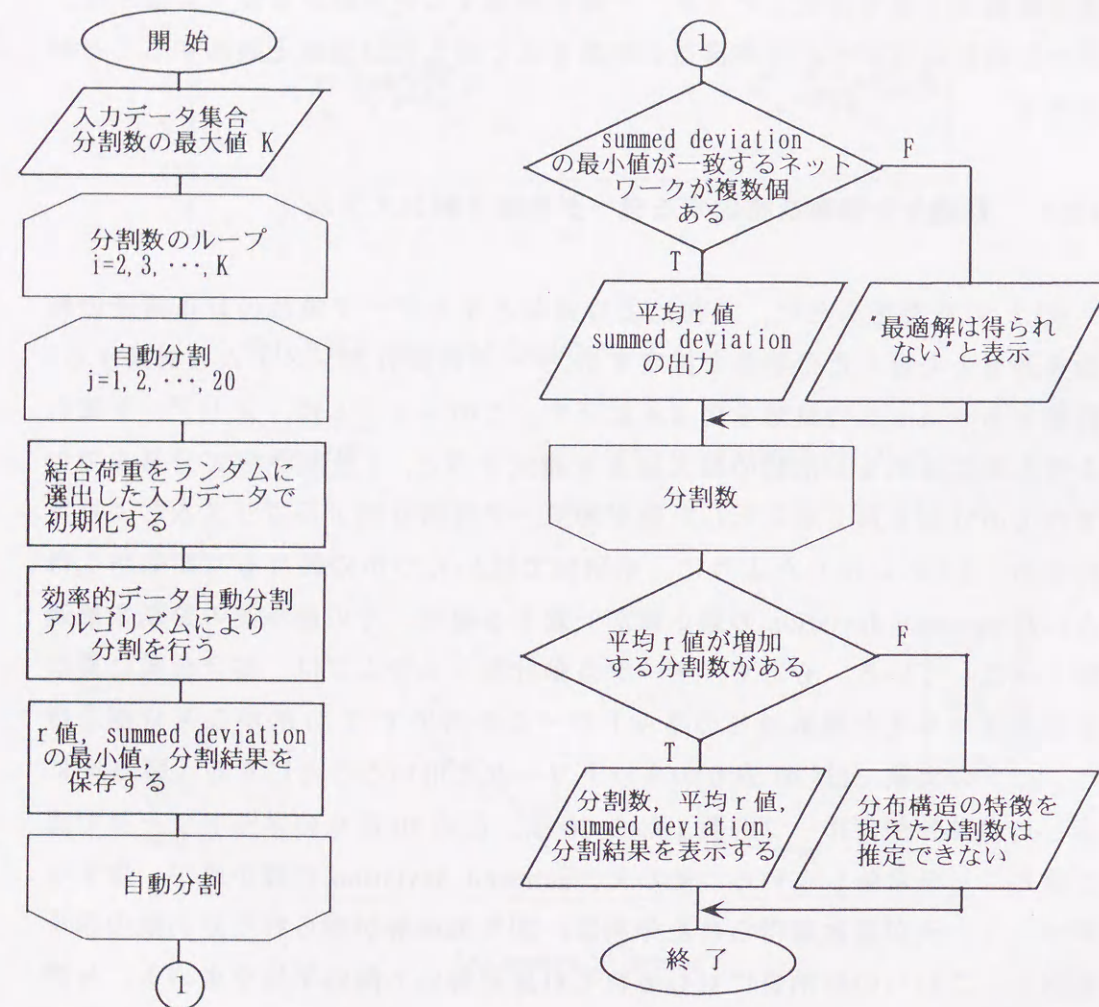


図 5.4 データ自動分割システムの概略

5.4 実データへの応用

提案するデータ自動分割システムを、財務データによる日本の地方銀行の分類に適用し、 r 値によりデータの分布構造の特徴を捉えた分割数の推定が可能であることを示す。5.1 に記したように、多次元実データ集合に対し、その分布の特徴を捉えた分割数を何らかの方法で決定するのは、大変困難な問題である。そこで、提案するシステムが推定した分割数の妥当性を検討するために、主成分分析の結果を用いることにした。対象データ集合の主成分分析を行うことで、データ集合がもつ特徴を最もよく反映した部分空間を決定することができる。この部分空間に、各分割数に対する分割結果を表示し、視覚的に捉えられるようにした。本章では、地方銀行のデータに対し主成分分析を行い、各分割数に対し提案するシステムから得られた分割結果を主成分平面上に表示して、どのようなグループが抽出されたとき r 値が増加するのかを調べることで推定された分割数の妥当性を示す。

5.4.1 地方銀行のデータ

日本の地方銀行の 1996 年 3 月、1997 年 3 月、1998 年 3 月期決算における財務データ 13 項目を入力データに使用した。1996 年のデータ数は 1 部、2 部上場の 94 行、1997 年は新たに 2 部上場となった 4 銀行を加えた 98 行、1998 年は 1997 年 11 月に破綻を表明した銀行を除いた 97 行である。用いた財務データは、以下の 13 項目である。

- ① 資本金
- ② 総資産 貸借対照表資産の部の合計で、企業の規模を表す
- ③ 株主資本 資本金、法定準備金、剰余金の合計で、純資産、自己資本ともいう
- ④ 株主資本比率 (%) 株主資本を総資産で除した百分率、自己資本比率
- ⑤ 外部負債 日本銀行および他の金融機関からの調達資金
- ⑥ 資金量 直近決算期末残高
- ⑦ ROE (%) 当期利益を期末株主資本で除した百分率、株主資本を使ってどれだけの利益を稼いでいるかをみる指標

- ⑧ 含み損益 有価証券の時価と貸借対照表計上額との差額
- ⑨ 最高経常利益 経常利益の過去最高額
- ⑩ 経常収益 1年間の売上高
- ⑪ 業務純益 本来的な業務から発生する利益
- ⑫ 経常利益 本業の利益に営業外損益を加減した基本的な利益
- ⑬ 利益 税引後利益

ここでは、これらのデータを次の(5-1)式のシグモイド関数を用いて 0.0 から 1.0 の範囲にスケールした。

$$y = \frac{1}{1 + \exp(-a(x - \bar{x}))} \quad (5-1)$$

x は 13 項目のそれぞれの数値、 \bar{x} は各項目の平均値、 a は各項目の桁数を調整するための正の定数である。この変換から得られる y の値をデータ自動分割システムの入力データとして使用した。

5.4.2 地方銀行の分割結果

5.4.1 で作成した 1996 年、1997 年、1998 年のデータをもとに、分割数の最大値 K を 15 として地方銀行の 2 分割から 15 分割までを行った。summed deviation の最小値が一致するネットワークが複数個得られた分割数は、

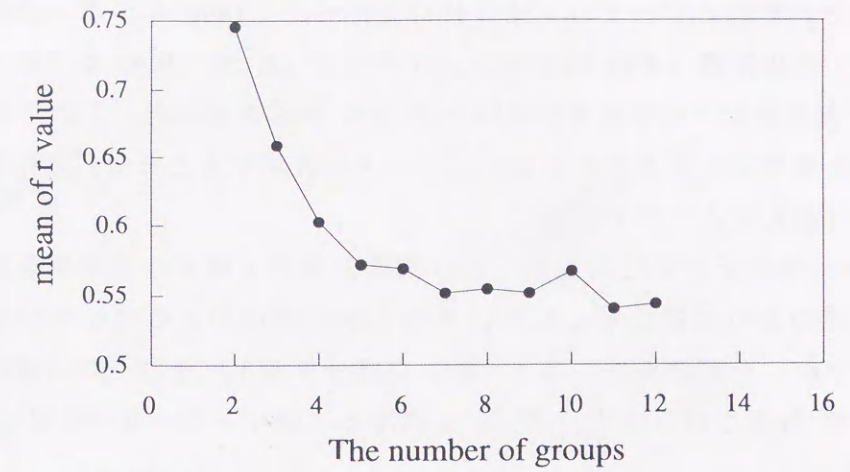
1996 年	2 分割から 12 分割まで
1997 年	2 分割から 10 分割までと 12 分割
1998 年	2 分割から 12 分割まで

であった。これらの各分割における平均 r 値の変化の様子を図 5.5 に示す。平均 r 値の増加は

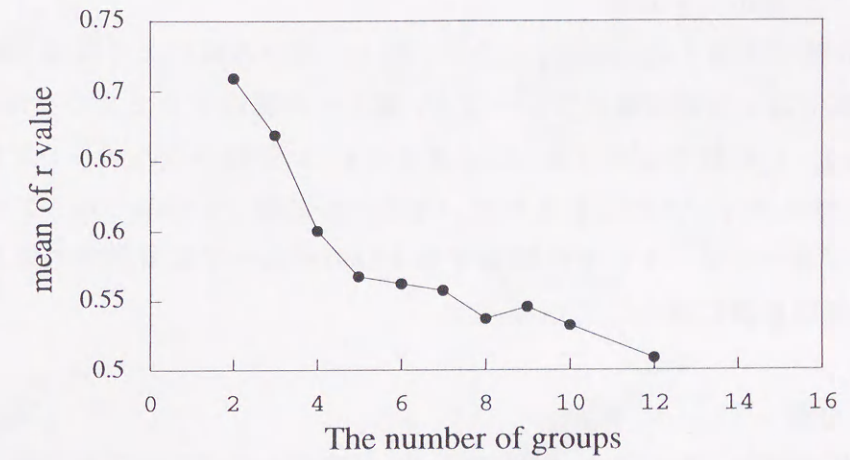
1996 年	10 分割に (図 5.5(a))
1997 年	9 分割でやや増加 (図 5.5(b))
1998 年	5, 6 分割も少し増加しているが 10 分割に顕著な増加 (図 5.5(c))

が見られた。従って、データ自動分割システムは 1996 年のデータでは 10 分割、1997 年のデータでは 9 分割、1998 年のデータでは 10 分割を分布構造の特徴を捉えた分割数として推定した。

(a) 1996 年



(b) 1997 年



(c) 1998 年

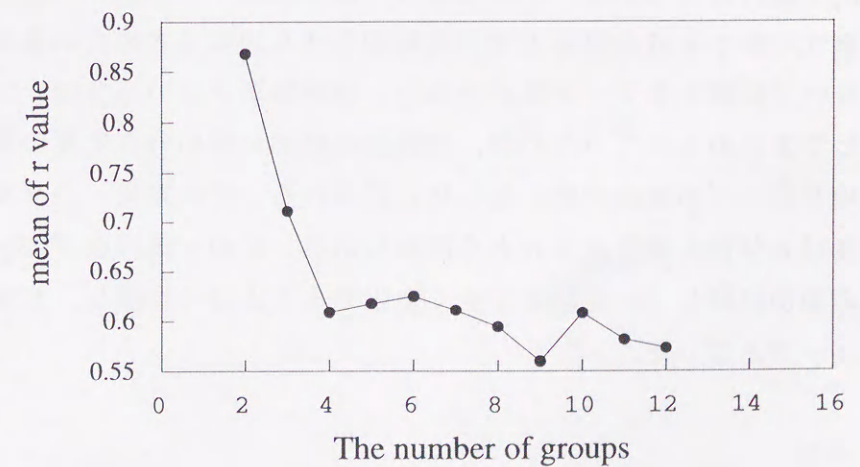


図 5.5 地方銀行の自動分割で得られた分割数と r 値

次に、データ自動分割システムが推定した分割数の妥当性を検討するために、それぞれのデータの主成分分析を行った。1996年のデータの第2主成分までの累積寄与率は88.1%、1997年では88.1%、1998年では85.5%で、第3主成分までの累積寄与率は3年とも90%を越えた。これらの3つの主成分を座標軸とする平面上に入力データを表示することで、分布構造の特徴は十分捉えることができる。

1998年のデータに対して、各分割数の平均r値と分割結果を比較し、分割がどのような変化をした時に平均r値の増加が見られるのかを検討した(図5.6)。分割結果は、第1、第2主成分平面上にそれぞれの銀行の主成分得点を、分割されたグループごとに異なるプロットマーカを用いて表した。

(1) 2分割から4分割

2分割では図5.6(a)に示すように、第2主成分の値によりほぼ2分された。3分割では2分割の■のグループが、■と+の隣接する2つのグループに(図5.6(b))、4分割では図5.6(c)に示すように2分割の◇のグループが、◇と▲の2つのグループに分割された。即ち、3分割、4分割では、2分割で得られたグループが、それぞれ隣接する2つのグループに分割されており、この間r値は単調に減少している。

(2) 5分割

r値が増加に転じた5分割では、4分割の▲のグループが分裂し、図中に囲った3銀行のみのグループが生じた(図5.6(d))。4分割での▲のグループは、第1、第2主成分平面上では比較的大きな領域を占めているが、その割にグループに属するデータ数が少ない。分割数が少ないうちは一つのグループとしてまとめられていたのが、分割数の増加に伴い分布密度の高い少数データのグループが抽出可能になったと見られる。その結果、各グループ間の分離度は4分割の場合より大きく高められた。この3銀行のグループは、分割数の増加に対し12分割まで全く変化することなく存続し、非常に安定したグループとなった。

(3) 6分割

r値が更に増加した6分割では◆のグループが形成され、2分割で得られた2つのグループの中間に位置する銀行が、図中に囲った1つのグループと

してまとめられた(図5.6(e))。その結果、各グループの分離度は更に高められている。

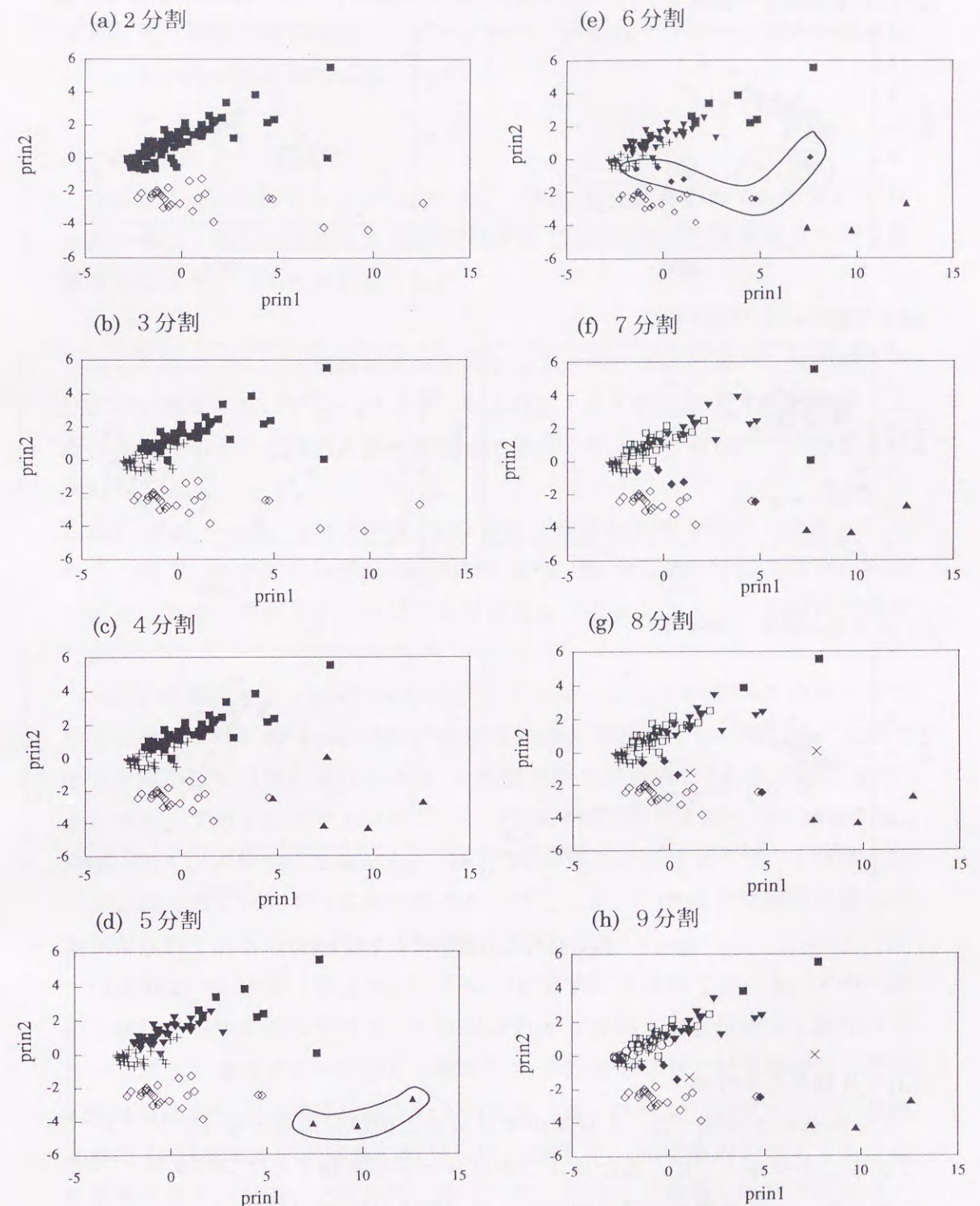
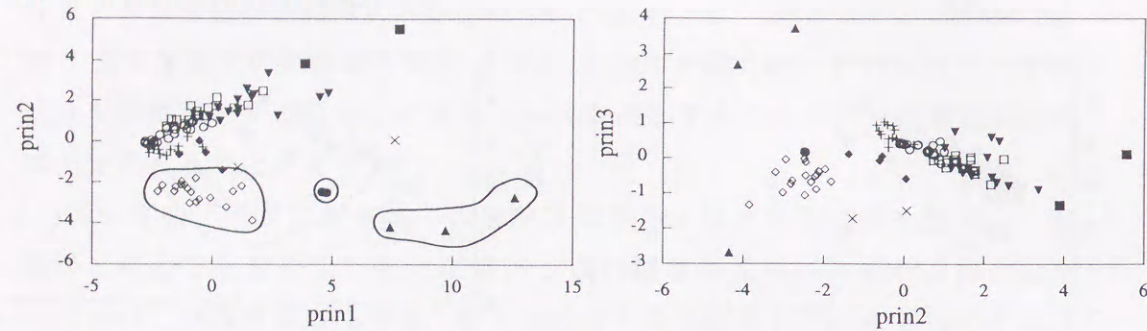
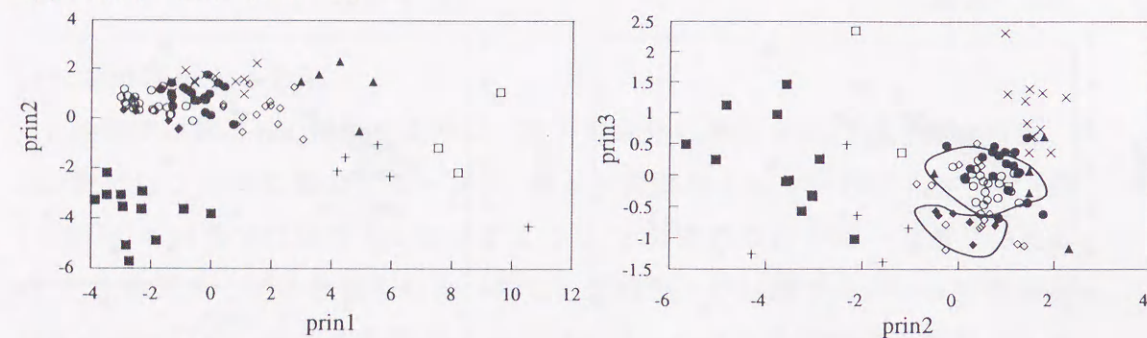


図5.6 地方銀行(1998年)データの分割結果と主成分得点

(a) 10 分割の結果 (1998 年)



(b) 9 分割の結果 (1997 年)



(c) 10 分割の結果 (1996 年)

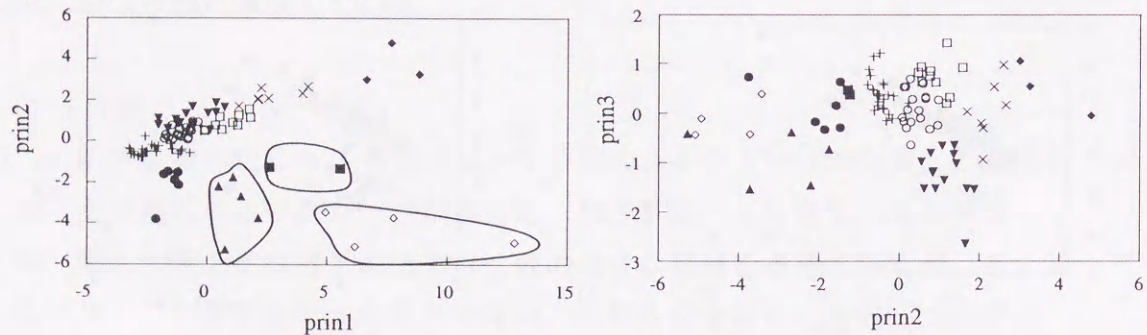


図 5.7 地方銀行の分割結果と主成分得点

(4) 7 分割から 9 分割

7 分割から 9 分割では、2 分割から 4 分割と同様に隣接する 2 つのグループが 3 つのグループに、または 1 つのグループが隣接する 2 つのグループに分割され、 r 値の増加は見られなかった (図 5.6(f),(g),(h)).

(5) 10 分割

r 値が最も大幅な増加を示した 10 分割では、図 5.7(a)の左側に示すように、2 分割の◇のグループが図中に囲った 3 つのグループに分割された。結果として、密度の高い少数データのグループが生じ、グループ間の分離度の高い分割結果が得られている。

(6) 11 分割から 12 分割

分割結果は図の形で示していないが、 r 値が減少に転じた 11 分割では 10 分割の◆の 1 銀行が分離、12 分割ではやはり 10 分割の○と▼のグループが隣接する 3 グループに再分割された。

これらの結果から、分割数が増加することで非常に密度の高い、他のグループとの分離度の高いグループが生じた場合に r 値の増加が見られ、このようなグループがすべて抽出された段階で r 値が最も大きく増加していることが分かる。

1997 年の 9 分割、1996 年の 10 分割の分割結果を図 5.7(b),(c)に示す。ただし、第 1、第 2 主成分平面 (左列) と第 2、第 3 主成分平面 (右列) 上に各銀行の主成分得点をグループごとに異なるプロットマーカーを用いて表示した。

1997 年のデータ (図 5.7(b)) は全くオーバーラップのない 2 グループという特徴をもつため 1998 年ほど顕著な r 値の増加は見られないが、それでも 8 分割から 9 分割へ移行する際、8 分割では 1 グループとして存在する○、◆のグループが 9 分割で 2 つのグループに分割され、ここで r 値が増加した。第 2、第 3 主成分平面で見ると、図 5.7(b)の囲みに示すように、1 つのグループと捉えるよりは 2 つの異なるグループとしたほうがより分離度の高い分割が得られている。

1996 年の 10 分割 (図 5.7(c)) では、10 分割に至るまでに、◆、×の一部の 5 銀行が 1 つのグループとして抽出された 6 分割、および●の 6 銀行のグループが+と▲のグループから分離された 8 分割でやや r 値が増加し、更に 9 分割では 2 グループに分割されていた▲、■、◇が 3 つのグループに分割された 10 分割で r 値が顕著に増加した。1996 年、1997 年のデータでも、より密度の高い、グループ全体の分離度の高いグループが得られた分割数で r 値が増加すると共に、このようなグループがすべて抽出された段階で r 値が

最も大きく増加した。

5.5 考察

5.5.1 スケーリングの方法

一般的に、ニューラルネットワークへの入力データは何らかの変換により、0.0 から 1.0 の範囲内にスケーリングされる。データ集合がもつ特徴をありのままに捉えるためには、このような変換として一次変換が最も望ましい方法と考えられる。しかし、地方銀行のデータには、例外的に巨額の損益が計上されるなど特異的なデータが存在する。このようなデータも含めて一次変換を行うと、他の大半のデータが同じような値になり、差異が認められなくなる可能性がある。このような特異値の影響を避けるため、上限と下限にあたるしきい値を設定する意味で、本研究では(5-1)式に示したシグモイド関数による変換を行った。しかし、入力データの変換はほとんどシグモイド関数が直線的に変化する部分で行っている。

最適なスケーリングの方法は、何を目的としてデータ分割を行うかによって異なると考えられる。異なるデータ変換を行った場合、その分割結果からは異なる側面から捉えたデータ集合の特徴が抽出される。特別な目的のためにデータ分割を行うのなら、異なるスケーリング方法から得られる分割結果の比較検討を行う必要があるが、本章の目的は一般的な特徴を自動抽出する自動分割システムの構築であるため、スケーリングの方法に関する詳細な検討は行わなかった。

5.5.2 推定された分割数による分割結果の解釈

分割結果の解釈はスケーリングの方法の詳細な検討をした上で行う必要があるが、本章で得られた分割結果にも銀行の現状が十分捉えられていることを若干の例をあげて示す。主成分分析により抽出された第1、第2主成分は、1996年、1997年、1998年のデータで大きな差はなかった。第1主成分は、資本金、総資産、資金量、経常収益などの係数が正の大きな値をもち、銀行の規模を表す成分と見られる。また、第2主成分は、株主資本比率、ROE、利益の係数が正の大きな値をもち、銀行の安全性を表す成分と見られる。各

成分にこのような意味付けを行った場合、図5.7の第1、第2主成分平面で左下にある銀行は規模も小さく安全性も低い銀行と解釈することができる。ちなみに1996年(図5.7(c))の左下の飛び離れて存在する●は、1997年11月に破綻した銀行である。1997年3月期決算では、■のグループで第2主成分の値が最上位にあり、経営不安説解消のために取られた対応の効果が現れている。また、図5.7(b)の■のグループの第2主成分の最低値は1999年6月に破綻した銀行である。この銀行は、1998年(図5.7(a))の分割結果では◇のグループに属し、第2主成分の値は下から2番目であった。これらのグループに属する銀行には、他の銀行と合併した銀行も含まれており、何らかの方法で安全性を高める必要がある銀行群とみなすことができる。

5.5.3 推定された分割数の検討

分割数が多くなると類似したデータをまとめたグループが更に再分割され、本来データがもっている特徴がまとまりとして捉え切れなくなる可能性がある。逆に分割数が少ないうちは、あまり類似していないデータも一つのグループとしてまとめなければならず、やはりデータの特徴は十分には捉えられない。

地方銀行のデータ分割では、5.5.2に示した主成分の意味付けによると、1998年のデータの2分割は「安全性の高い」銀行群と「安全性の低い」銀行群への分割である(図5.6(a))。2分割により、このような大まかな特徴を抽出することができる。4分割では銀行の規模によりそれぞれが更に2つに分割され、2分割の大まかな特徴から「規模も大きくて安全性も高い」というような、より詳細な特徴を捉えることができる。r値が最も大きく増加する10分割では、安全性の低い銀行群が図5.7(a)の第1、第2主成分平面に示したように、規模に応じて3つの段階に分割された。この結果、規模も小さく安全性も低いという特徴がより詳細に得られている。中規模の●を◇と同一グループにして安全性を問うよりは、◇の銀行に対して検討するほうが妥当である。また、このような特徴をもつグループが抽出されたところで、r値が最も大きく増加することは、分割数を推定する指標としてr値が非常に有用性が高いことを示している。更に分割数が増加した場合、例えば◇のグループが2つに分割されるというように、類似性の高いグループの再分割のみが起り始める。最終的には、1銀行が1グループという状態になり、も

はや分布構造の特徴を捉えることは不可能になる。データの分布構造の特徴を最も詳細に捉えている分割数は、このような類似性の高いグループの再分割のみが起る直前の分割数であり、この分割数は r 値が最も大きく増加する点として捉えられる。

5.6 まとめ

ニューラルネットワークの入出力間で求めた相互情報量に基づく r 値を、データの分布構造の特徴を最もよく捉えた分割数を推定する指標として提案し、この r 値を用いて分割数の推定から最適な分割結果を得るまでのすべての過程を自動的に行うデータ自動分割システムを構築した。提案したシステムを地方銀行財務の実データに対して適用した結果、データの分布構造が r 値に十分反映され、分布の特徴を捉えた分割数の推定が十分可能であることが明らかになった。

第6章

結論

コンピュータネットワークの発達に伴い、情報化社会から大量に発生するデータは、機械的にデータベースに蓄積されていく。このようなデータベースから、意思決定に有用な情報を如何に迅速に抽出し実際の戦略に活用できるかが、企業戦略の重要なポイントとなっている。そのためには、データ集合全体の特徴を把握するために必要な情報を、可能な限り機械的に抽出すること、即ち特徴抽出の過程の自動化が必要である。

このような背景から、本研究では、ニューラルネットワークによるデータ分割を基本とし、データの分布構造の特徴を最もよく捉えた分割数を自動的に推定するデータ自動分割アルゴリズムを構築した。

第2章で、データ自動分割アルゴリズムを構築するための基本的な手法として漏えい学習を採用することの根拠を示し、第3章では、漏えい学習と競合学習を組合せた組合せ法について検討した。組合せ法は、Kohonen 自己組織化特徴画像、競合学習、k-means 法などと比較して、データ分割における初期値依存性がより低い学習法であることを示した。また、組合せ法を実際のアンケートデータの分割に適用し、その有効性を示した。

第4章では、組合せ法の学習パラメータを自動的に調整するために、ニューラルネットワークの入出力間相互情報量を用いた指標、 r 値を提案した。更に、この r 値を用いたデータの自動分割アルゴリズム CMAMI を構築した。 r 値は CMAMI により全入力データを一回ずつ学習した、第1サイクル終了後に計算される。第1サイクルの学習を行ったことで、データ集合と出力層から得られる分割結果との間にどの程度の相関が得られたかを調べている。複雑な分布構造をもつデータ集合の場合、すべてのデータを一回だけ用いる最初のサイクルの学習では入出力間にさほど強い相関は得られない。相関の程度を表す r 値によってその後の学習を調整すれば、対象データ集合の分布構造に応じた学習計画を自動的に設定することができる。CMAMI では効率よく最適解または精度のよい近似解が得られることを、いくつかのサンプルデータを用いて示した。

第 5 章では、ニューラルネットワークの入出力間で求めた相互情報量に基づく r 値を、データの分布構造の特徴を最もよく捉えた分割数を推定する指標として用いることを検討した。あるデータ集合に対し、様々な分割数を指定してデータ分割を行った場合、一般的には分割数の増加に伴って r 値は減少する。しかし、分布構造を最もよく捉えた分割数では、第 1 サイクルの学習後でもデータ集合と分割結果の間に強い相関が得られ、単調減少に反してその分割数の r 値の値は増加する。従って、 r 値により分布構造の特徴を最もよく捉えた分割数を自動的に推定することができる。この r 値を用いて分割数の推定から最適な分割結果を得るまでのすべての過程を自動的に行うデータ自動分割システムを構築した。最後に、提案したシステムを財務データによる地方銀行の分類に適用し、 r 値によりデータの分布構造の特徴を捉えた分割数の推定が十分可能であることを示した。

今後の課題として、問題点の一つとして序章にも記載した、大量の多次元データに対応するアルゴリズムへの提案手法の改良が挙げられる。本来、ニューラルネットワークは人間の脳の情報処理を模倣したものであり、並列処理を得意としている。本研究で構築したアルゴリズムも、入力層ユニットごとの並列処理を行うことで、多次元データへの対応が可能になる。しかし、この実現にはハード・ソフト両面での大幅な改良が必要であり、今後の大きな課題として取り組んでいきたいと考えている。

一方、大量のデータという点では、本研究のアルゴリズムが、結合荷重変更量の計算を入力データごとに行うという直列的な処理を基本としているため、あまり大きな改善は望めない。R.Agrawal や C.C.Aggarwal らも、距離の計算方法に工夫を凝らす程度の対応しかしていない[7][8]ことを考慮すると、この問題はコンピュータの処理能力に頼らざるを得ない面もあると考えられる。

やや近い将来の課題として、このシステムにユーザーフレンドリーなインターフェースを作成して、自動的に得られた分割結果を可視化情報として提供できるようにすることがある。本研究では主成分分析の結果を利用して可視化を行っているが、大量の多次元データを対象とする場合は、必ずしもこのような方法が適用できるとは限らない。最近 Kohonen が自己組織化特徴写像の応用としていくつか提案している可視化情報探索の手法[38]を、本研究の自動分割システムにも是非取り入れていきたいと考えている。

謝辞

本研究を進めるにあたり、終始懇切丁寧な暖かいご指導を頂きました宇都宮大学工学部情報工学科宮道壽一教授に、心から感謝の意を表します。また、自治医科大学薬理学教室岸浩一郎講師には、長年にわたり懇切丁寧なご指導を頂きましたこと、心より感謝いたします。

本研究をまとめるにあたり、有益なご助言、ご指導を頂きました、宇都宮大学工学部情報工学科加藤茂夫教授、鎌田一雄教授、山田芳文教授、電気電子工学科松村和仁教授に深く感謝いたします。また、宇都宮大学工学部電気電子工学科松岡孝栄助教授、自治医科大学看護短期大学竹田俊明教授、博士後期課程在学中にご指導を頂きました宇都宮大学工学部情報工学科青木恭太助教授、東海林健二助教授にも深く感謝いたします。

そして、自治医科大学物理学教室青野修教授、物理学教室の皆様方には、長年にわたり暖かいご支援と大変居心地のよい研究環境を快く提供していただきましたこと、心より感謝いたしますとともに、厚くお礼申し上げます。

文献

- [1] P. Adriaans and D. Zantinge, "DATA MINING," Addison-Wesley, 1996.
- [2] U.M.Fayyad, G.Piatetsky-Shapiro, P.Smyth, and R.Uthurusamy, "Advances in Knowledge Discovery and Data Mining," MIT Press, Cambridge, Massachusetts, 1996.
- [3] R. Mattison, "データマイニング戦略と活用," 富士通経営研修所, 1997.
- [4] 梅村恭司, "個人のためのデータマイニング," bit, vol.30, no.6, pp.2-8, 1998.
- [5] P.Cabena, P.Hadjinian, R.Stadler, J.Verhees, and A.Zanasi, "Discovering Data Mining From Concept to Implementation," Prentice Hall, 1998. 日本アイ・ビー・エム株式会社, ナショナル・ランゲージ・サポート訳, "データマイニング活用ガイド," トッパン, 1999.
- [6] M.J.A.Berry, and G.Linoff, "Data Mining Techniques For Marketing, Sales, and Customer Support," John Wiley & Sons, Inc., 1997.
- [7] R.Agrawal, J.Gehrke, D.Gunopulos, and P.Raghavan, "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications," Proc. of the ACM SIGMOD International Conference on Management of Data, 1998.
- [8] C.C.Aggarwal, C.Procopiuc, J.L.Wolf, P.S.Yu, and J.S.Park, "Fast Algorithms for Projected Clustering," Proc. of the ACM SIGMOD International Conference on Management of Data, 1999.
- [9] M.R.Anderberg, "Cluster Analysis for Applications," Academic Press, New York, 1973.
- [10] 田中豊, 垂水共之, 脇本和昌, "パソコン統計解析ハンドブック II," 共立出版, 1984.
- [11] D.E.Rumelhart, J.L.McClelland, and The PDP Research Group, "Parallel Distributed Processing," MIT Press, Cambridge, Massachusetts, 1986.

- [12] T. Kohonen, "Self-Organized Formation of Topologically Correct Feature Maps," *Biological Cybernetics*, vol.43, pp.59-69, 1982.
- [13] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol.78, no.9, pp.1464-1480, 1990.
- [14] W. Snyder, D. Nissman, D. Van den Bout, and G. Bilbro, "Kohonen Networks and Clustering: Comparative Performance in Color Clustering," *Advances in Neural Information Processing Systems*, pp.984-990, Morgan Kaufmann, San Mateo, 1991.
- [15] 寺島幹彦, 白谷文行, 山本公明, "自己組織化特徴マップ上のデータ密度ヒストグラムを用いた教師なしクラスタ分類法," *電子情報通信学会論文誌*, vol.J79-D-II, no.7, pp.1280-1290, 1996.
- [16] J. P. Bigus, "Data Mining with Neural Networks," McGraw-Hill, 1996.
- [17] J. E. Dayhoff, "Neural Network Architectures," Van Nostrand Reinhold, New York, 1990.
- [18] 菊地登志子, 松岡孝栄, 竹田俊明, 岸浩一郎, "競合学習ニューラルネットワークによる自動分割," *電子情報通信学会論文誌*, vol.J78-D-II, no.10, pp.1543-1547, 1995.
- [19] R. Linsker, "How to Generate Ordered Maps by Maximizing the Mutual Information between Input and Output Signals," *Neural Computation* vol.1, pp.402-411, 1989.
- [20] 菊地登志子, 松岡孝栄, 竹田俊明, 岸浩一郎, "相互情報量により学習係数を調整した競合学習アルゴリズム," *日本神経回路学会第7回全国大会講演論文集*, pp.96-97, 1996.
- [21] 菊地登志子, 岸浩一郎, 宮道壽一, "漏洩学習の学習係数を自動的に調整する指標に関する検討," *電子情報通信学会総合大会講演論文集*, D-2-16, p.22, 1998.
- [22] 菊地登志子, 岸浩一郎, 宮道壽一, "相互情報量により学習パラメータを調整した効率的データ自動分割アルゴリズム," *電子情報通信学会論文誌*, vol.J82-D-II, no.4, pp.660-668, 1999.
- [23] 菊地登志子, 岸浩一郎, 宮道壽一, "ニューラルネットワークの入出力間相互情報

- 量に基づくデータ分割数の推定," *電子情報通信学会総合大会講演論文集*, D-2-13, p.20, 1999.
- [24] 菊地登志子, 岸浩一郎, 宮道壽一, "ニューラルネットワークの入出力間相互情報量に基づくデータ自動分割システム," (*日本社会情報学会学会誌* 印刷中)
- [25] 菊地登志子, 岸浩一郎, 宮道壽一, "ニューラルネットワークを用いたデータ自動分割システム—地方銀行の分類への応用—," *日本社会情報学会全国大会研究発表論文集*, vol.14, no.1, pp.147-152, 1999.
- [26] 尾崎弘, 谷口慶治, 小川秀夫, "画像処理—その基礎から応用まで," 共立出版, 1998.
- [27] 長尾真, "パターン情報処理," コロナ社, 1998.
- [28] 馬場則夫, 小島史男, 小澤誠一, "ニューラルネットワークの基礎と応用," 共立出版, 1994.
- [29] T.Kikuchi, T.Matsuoka, T.Takeda, and K.Kishi, "Automatic Classification of a Large Volume of Questionnaire Data by Means of Competitive Learning," *Proc. of the WCNN*, vol.I, pp.212-215, 1993.
- [30] 菊地登志子, 松岡孝栄, 竹田俊明, 岸浩一郎, "競合学習によるアンケート調査データのクラスタリング," *電子情報通信学会春季大会講演論文集*, D-43, p.6-43, 1993.
- [31] 菊地登志子, 松岡孝栄, 竹田俊明, 岸浩一郎, 渡辺則生, 今泉忠, "競合学習ニューラルネットワークによるクラスタリング," *日本分類学会研究報告予稿集*, pp.21-25, 1994.
- [32] 中川聖一, "情報理論の基礎と応用," 近代科学社, 1992.
- [33] T. Gonzalez, "Clustering to Minimize the Maximum Intercluster Distance," *Theoretical Computer Science*, vol.38, pp.293-306, 1985.
- [34] R.J.Fowler, M.S.Paterson, and S.L.Tanimoto, "Optimal Packing and Covering in the Plane are NP-complete," *Information Processing Letters*, vol.12, no.3,

pp.133-137, 1981.

- [35] 大津展之, "判別および最小 2 乗基準に基づく自動しきい値選定法," 電子通信学会論文誌, vol.J63-D, no.4, pp.349-356, 1980.
- [36] 大津展之, 栗田多喜夫, 関田敏, "パターン認識," 朝倉書店, 1996.
- [37] G.B.Coleman and H.C.Andrews, "Image Segmentation by Clustering," Proc. IEEE, vol.67, no.5, pp.773-785, 1979.
- [38] G.Deboeck and T.Kohonen(Eds),"Visual Explorations in Finance with Self-Organizing Maps," Springer-Verlag, New York, 1998.

