



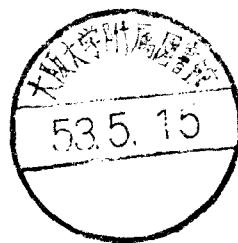
Title	パターン認識における連想と学習に関する研究
Author(s)	辰巳, 昭治
Citation	大阪大学, 1978, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/2366
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

パターン認識における 連想と学習に関する研究



1977年12月

辰巳昭治

序 文

本論文は、筆者が大阪大学大学院工学研究科（通信工学専攻）に在学中に行った、パターン認識における連想と学習に関する研究を5章に分けてまとめたものである。

第1章は緒論で、本研究の目的と関連した諸研究の現状を概説し、本研究の位置づけを行っている。

第2章では、誤り制御機能による連想方式を提案している。認識対象を量子化した時、その対象は1つの符号を構成していると考えられる。ここでは、符号化された入力情報に誤り訂正の復号手法を用いて、あいまいな入力パターンから正確な連想出力パターンを得る連想方式を考案している。この連想パターンを得るため、記憶パターンの生成方法と多くの記憶パターンから入力パターンに関連の深いパターンを想起する方法について述べている。

第3章では、被覆を用いた識別方法について論じている。パターン集合に対し連想被覆を定義し、連想被覆による連想モデルについて考察している。特に被覆間の细分関係に着目し、细分関係を持っている連想機構、细分関係と学習との関係について述べている。さらに、実際に被覆を構成する方法として、強制学習法を用いる手法と、Prime Event法を用いる手法について論じ、それらの手法によって構成された被覆を用いて、高品質手書き文字を対象として識別実験を行い、その結果を示している。

第4章では、教師なし学習の一つとして、種々な分布のデータ集合をノンパラメトリックな手法によりクラスタ分割するアルゴリズムを提案している。与えられた各サンプル点において、その点を中心として適当な半径をもつ近傍内の数個のサンプル点のみを考慮し、Parzen Estimate法を用いてその点に

おける分布の擬似的な勾配（擬似勾配）を定義する。この擬似勾配を用い、元のデータ集合より縮少したデータ集合を抽出し、この集合に対して **Minimal Spanning Tree** を構成し枝を切断することによりクラスタ分割を行う。これを基に、元のデータ集合の各サンプル点のクラスを決定する。また、提案した手法は種々のデータ集合に適用可能であり、計算機シミュレーションによつてこのことを検証している。

第5章は結論で、本研究で得られた結果をまとめ、今後に残された問題について述べている。

関連発表論文

- (1) 辰巳, 手塚: 「誤り制御機能による連想」
情報処理学会 第15回大会 (昭和49)
- (2) 辰巳, 手塚: 「被覆による連想の一方式」
信学会 全国大会 (昭和50)
- (3) 辰巳, 真田, 手塚: 「被覆を用いた連想記述に関する一考察」
信学会 パターン認識と学習研資 PRL75-23 (1975-7)
- (4) 辰巳, 真田, 手塚: 「有限観測可能のもとでの連想構造に関する考察」
信学会 総合全国大会 (昭和51)
- (5) 永田, 辰巳, 真田, 手塚: 「擬似勾配を用いたノンパラメトリック クラスタリングの一手法」
信学会 パターン認識と学習研資 PRL76-61 (1977-1)
- (6) 辰巳, 真田, 手塚: 「被覆による識別の一手法について」
信学会 総合全国大会 (昭和52)
- (7) 永田, 辰巳, 真田, 手塚: 「擬似勾配を用いたノンパラメトリック クラスタリングの一手法」
信学会 総合全国大会 (昭和52)
- (8) 辰巳, 永田, 真田, 手塚: 「擬似勾配を用いたノンパラメトリック クラスタリングの一手法」
電子通信学会論文誌D 昭和53年1月掲載予定

目 次

第 1 章 緒 論	1
第 2 章 誤り制御機能による連想方式	5
2 . 1 緒 言	5
2 . 2 誤り制御機能による連想方法	6
2 . 3 記憶方法	9
2 . 3 . 1 記憶方法 (A) (パターン・チェック法)	10
2 . 3 . 2 記憶方法 (B) (ライン・チェック法)	10
2 . 4 想起方法	13
2 . 4 . 1 想起方法 (A) (パターン・チェック法)	13
2 . 4 . 2 想起方法 (B) (ライン・チェック法)	14
2 . 5 結 言	14
第 3 章 被覆を用いた識別方式	17
3 . 1 緒 言	17
3 . 2 被覆による連想モデル	18
3 . 2 . 1 連想被覆	18
3 . 2 . 2 細分関係を持つ連想構造	20
3 . 2 . 3 細分関係と学習	22
3 . 2 . 4 有限観測可能時の連想構造	24
3 . 3 被覆による識別方法	25
3 . 4 被覆集合の生成方法と識別機の構成法	27
3 . 4 . 1 強制学習法による識別機の構成法	27

3 . 4 . 2 Prime Event 法による識別機の構成法	30
3 . 4 . 3 実験結果	32
3 . 5 結 言	39
第4章 教師なし学習によるクラスタリング	41
4 . 1 緒 言	41
4 . 2 クラスタリングの一般的表現	43
4 . 3 擬似勾配によるクラスタリング	45
4 . 3 . 1 擬似勾配	45
4 . 3 . 2 クラスタ分割の手法	49
4 . 3 . 3 計算機実験とその検討	56
4 . 4 結 言	65
第5章 結 論	67
謝 辞	69
文 献	71

第 1 章 緒論

高度の情報化社会になるとともに、情報量は飛躍的に増大し、雑多な情報を高信頼度、高速度で処理するために、計算機の利用が活発に行われている。超高速計算が可能となった現在、計算時間と比較して膨大な処理時間を要する従来の入力媒体の取り扱いが問題となり、図形・文字等を直接入力できる装置、即ちパターン認識機械の開発が要求されている。通常、パターン認識機構をモデル化すると、図 1.1 となる。本論文では、図 1.1 のパターン処理内のクラスタリングと決定機構に連想機能を付加する方法について論じる。

デジタル計算機を用いてパターン認識を行う時、人間の頭脳と電子計算機の間には、その情報処理方式において根本的な違いがあると指摘されている。

(1) その違いの一つとして、現在の計算機には連想機能がないことが挙げられる。連想機能を工学的に実現するため、人間の脳の生理学的、解剖学的考察が加えられた。その結果、脳は神経細胞を基本要素として多層構造をなす回路網であることが明らかにされ、神経回路網の生理学的、心理学的現象を説明するため W. S. McCulloch, W. Pitts⁽²⁾ は神経細胞のモデル化を行い、また E. R. Caianiello⁽³⁾ は神経回路網における神経方程式および記憶方程式を定式化している。また F. Rosenblatt^{(4)～(5)} および H. D. Block^{(5)～(6)} は神経細胞のモデルを素子として用い、2 次元パターンの認識装置であるペーセプトロンを提唱し、この装置の動的な動きおよび能力などについて研究している。連想を神経回路網的に取り扱った研究として、Post のモデル⁽⁷⁾、中野の提唱したアソシアトロン^{(8), (9)}、福島のコグニトロン⁽¹⁰⁾ 等がある。従来の電子計算機では、記憶情報を導出するのにアドレスを介在し、読み出し書き込みを行ってきたが、アソシアトロンはアドレスを介在とせず、ある記憶事項からほ

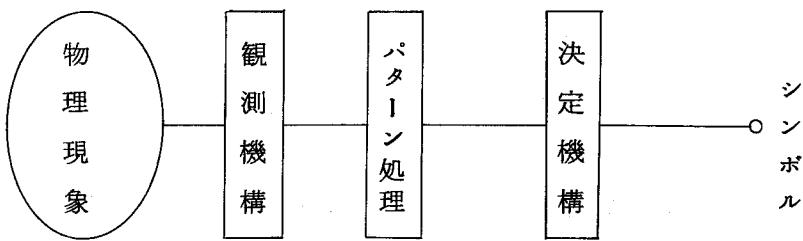


図 1.1 パターン識別機構 ⁽²⁴⁾

かの記憶事項を「連想」するという形で行われている。本論文では、連想をアドレスを介在としない形態であると考え、2, 3 の方法について考察している。

第 2 章では、連想機能を誤り制御機能によって実現する方式を提案している。

(11) アソシアトロン等の連想システムは、機能上は図形から図形、パターンからパターンへの変換または処理という形態であり、構造はペーセプトロンを多数並列に配列していると考えられる。上坂らの論文⁽¹²⁾は、この変換機能を記憶を重点に置いて論じ、磯道は学習システムの立場から論じている。⁽¹³⁾又、手書き文字のオンライン認識において、リー誤り訂正符号を用いて、記憶パターンに対応してインデックス・パターンを作成し、インデックス・パターンをも記憶する。読み出しを行う時、インデックス・パターンを参照として記憶パターンを想起する方法が提案されている。⁽¹⁴⁾ここでは連想という変換機能を以下の様に考える。即ち、何らかの歪みを受けたパターンを、伝送路に生じた雑音によって歪みを受けた符号と考え、その誤りを検出し、訂正することにより歪みを受けたパターンを認識することを連想という。よって、誤り訂正符号^{(15), (16)}、特に、代数的に取り扱い易い巡回符号^{(17), (18)}を用いて連想認識する方法を提案する。

第 3 章では、被覆を用いた識別方式^{(19), (20)}を提案している。この方式は、脳の生理学、心理学を考慮して発表された、Zeeman の脳のトポロジー論⁽²³⁾を基礎にしている。Zeeman の論文では、脳内の視野空間と外界の視覚領空間と

は同じ位相構造を持つことを証明している。しかし、Zeemanが定義した許容関係が明確化していないため、現実のパターン認識には意味を持っていない。許容関係を考慮し、パターン集合の要素間の位相構造を明確化するため、パターン集合に対し連想被覆族を定義する。あるパターンのクラスの決定には、そのパターンが連想被覆族のどの要素によって被覆されるか、その被覆した要素のクラスをそのパターンに名付けることにより識別する方法を提案する。そして、印刷文字に近い高品質手書き英数字を対象として識別実験を行い、その有効性を確かめる。

第4章では、もう一つのテーマである教師なし学習によるクラスタリングについて述べる。観測によって得られた量（観測パターン）を定まった目的に合うように処理する段階、すなわちパターン処理が必要である。パターンの処理は、扱う対象や観測機構によって異なるが、通常図1.2に示すように、量子化操作、正規化操作等を含んだ予備的な処理と、各種の特徴抽出、観測点の選択、類別（クラスター化）を含んだ本質的な処理の2段階に分けて考えることができる。

一般に1個のシンボルは、発生、観測の過程を通ることによって非常に多くの異なったパターンとして観測され、正規化によって、その数は減るが、そのままでは決定機構に導くことはできない。そのため、なんらかの性質を共

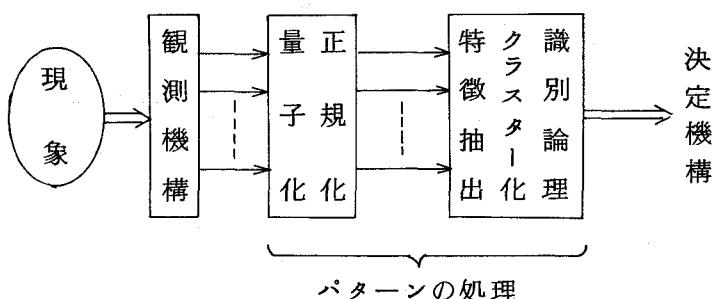


図1.2 パターンの処理 (24)

有しているパターン同志を集めて、同一の類に入れて分類していくこと（クラスタリング）が必要となってくる。クラスタリングの応用分野として、生物学における、生物の発生、進化過程による分類作業、医学における症候群の分類、人工衛星より電送されてくる写真の分析、言語、文字の文類などが挙げられ、これらの情報を迅速、適確に分類することが要求される。

クラスタ分析に関する論文は多数報告されている。G . H . Ball⁽²⁵⁾及びG . Nagy⁽²⁶⁾はパターン認識におけるクラスタ分析の役割を詳細に論じ、G . J . Zahn⁽²⁷⁾は network の概念に基礎をおき、グラフ理論からのアプローチにより構造の複雑なクラスタの検出に有効な手法を示し、S . C . Johnson⁽²⁸⁾は階層的クラスタ分析の手法を論じ、mass の概念に基礎をおく非階層的クラスタ分析の手法^{(29)～(32)}が幾つか報告されている。しかし、このような手法は、一般に、データ間距離の記憶や、繰り返し時間等の処理時間を多く要する。ここでは、多数のサンプルからなるデータを処理対象とし、Parzen Estimate 法を用いて、各サンプル点の擬似的な勾配を近傍内のサンプル点のみを用いて算出し、モードを包含する集合を抽出する。その抽出された集合に Minimal Spanning Tree を構成し、適当な切断条件を満たす枝を切断し、それによって生じた Sub - Tree を一つのクラスタを構成していると決め、抽出された集合のクラスタ分割を行う。これをもとに、元のデータ集合のクラスタを順次決定する。提案した手法は、種々のデータ集合に適用可能であり、計算機シミュレーションによってこのことを検証する。

第5章では、本論文で得た諸結果を検討し、今後の方向について述べる。

第 2 章 誤り制御機能による連想方式

2.1 緒 言

人間に特有な情報処理機能の 1つとして、あいまいな入力情報からできるだけ正確な情報を想起したり、数多くの記憶情報の中から、特定の情報と何らかの関連を有する情報を直感的に選び出す機能があり、これは一般に連想機能と呼ばれている。現代の電子計算機を用いて、記憶事項を読み出すには、アドレス情報を介在とする必要があり、また、あいまいな入力情報では、誤った情報しか得られない。そこで連想の機能をもつ記憶装置の研究がなされ、有名な連想モデルとして、Post のモデル、中野のアソシアトロン、福島のコグニトロンが挙げられる。これらのシステムは Distributed Perceptron と呼ぶことが出来る。機能上からみれば、パターンからパターンへの変換または処理という形態をとっている。人間は高度な連想能力を持っているが、文字認識において、歪んだパターンから正しいと思われるパターンを連想することが必要となってくる。ある部分に歪みが生じることは、その場所が雑音によって誤りを受けたと考えることができる。

一方、通信方式のなかで、伝送路に発生する雑音によって生じた誤りを検出訂正する方法が考案されている。この方法は、情報記号系列に適当な記号系列を付加することによって行っている。この理論についての基礎的研究は Shannon によって与えられたが、その後 Prange によって巡回符号が見い出され、画期的な発展を遂げることになる。

本章では、上で述べたように連想操作の中に変換機能の存在することに着目し、部分的に欠損したパターンは、雑音により誤りを生じたものと考え、誤り訂正操作による連想方式を提案する。誤り訂正符号としては、特に代数的に取

り扱い易い巡回符号を用い、記憶方法、想起方法について述べる。

2.2 誤り制御機能による連想方法

パターン \mathbf{A} を $m \times n$ のメッシュに量子化、各画素を $0, \dots, q-1$ の q 元で表示する。

パターン \mathbf{A} は次のようにベクトル表現できる。

$$\mathbf{A} = (a_{11}, a_{12}, \dots, a_{1n}, a_{21}, \dots, a_{mn}) \quad (2.1)$$

ただし、 a_{ij} は (i, j) 要素の明るさを示す。

この時、パターン \mathbf{A} は q 元の $N (= m \times n)$

次元符号ベクトルとみなせる。

ここで提案する連想方式は、誤り訂正符号として巡回符号を用いる。連想方法を述べる前に、巡回符号について簡単に説明する。

いま、 N 項からなる部分空間 H を考え、 H に属する各々のベクトル

$$v = (a_0, a_1, \dots, a_{N-1}) \quad (2.2)$$

に対して、それを巡回置換させたベクトル

$$v' = (a_{N-1}, a_0, a_1, \dots, a_{N-2}) \quad (2.3)$$

がまた H に属しているとき、これを巡回的部分空間といい、巡回符号はこの性質を保存し、次のように定義される。

[定義 2.1] ⁽³³⁾

N 次元ベクトル空間 V の部分空間 H が、巡回的部分空間であれば、部分空間 H の各要素は巡回符号である。

巡回符号の特徴として次の点が挙げられる。(a)線形符号の一種である。(b)線形帰還シフトレジスタで簡単に発生できる。(c)バースト誤りを訂正するような符号も容易に作成できる。(d)符号長が短かいときは比較的能率が高い。

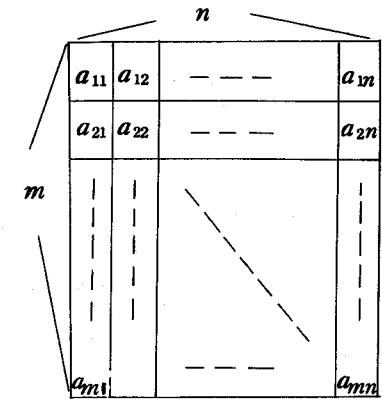


図 2.1 パターンの量子化

部分空間Hの元vは、多項式では

$$V(x) = a_0 + a_1 x + \dots + a_{N-1} x^{N-1} \quad (2.4)$$

と表わされる。ただし、ここで多項式の各係数 a_i ($i=0, \dots, N-1$) はガロア体 $G F(2)$ の上の任意の元を表わしている。この部分空間Hはイデアルであることが知られているから、巡回符号のすべての元すなわち符号点を割り切り、かつ $1 + X^N$ を割り切ることができる元 $G(x)$ が存在する。この多項式 $G(x)$ を特性多項式または生成多項式という。

符号長 n 、情報記号長 k の巡回符号において、情報記号 (a_1, a_2, \dots, a_k) は次のように多項式 $I(x)$ で表現できる。

$$I(x) = a_1 + a_2 x + \dots + a_k x^{k-1} \quad (2.5)$$

そして、 $I(x)X^{n-k}$ を巡回符号の生成多項式 $G(x)$ で割算を行ったときの商を $Q(x)$ 、その剰余を $R(x)$ とすれば、

$$I(x)X^{n-k} + R(x) = G(x)Q(x) \quad (2.6)$$

と表現できる。この左辺を $W(x)$ で示すと、

$$W(x) \equiv 0 \pmod{G(x)} \quad (2.7)$$

となり、この $W(x)$ が巡回符号である。したがって、特性多項式 $G(x)$ で情報点 $I(x) \cdot X^{n-k}$ を割り、その余り $R(x)$ を検査点として付加すれば、巡回符号が構成できることがわかる。

本章で考えている連想としては、図 2.2(b) のようにパターンの一部分が欠落している時、図 2.2(a) の標準パターンを想起することとする。各画素の明るさを 2 値化し、字の部分を “1” として、地の部分を “0” とし、字の部分の割合は地の部分の割合より少とする。また、欠落している部分は、“-1” で示すこととする。入力データである欠落パターン (I) に対し、欠落パターン (II) の形に交換する。この欠落パターン (II) において、字の欠落部分は雑音により誤りが生じ “1” が “0” に変化したものと判断し、その誤り個所を探索し

訂正することが連想と考える。

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	0	0	0	1	0	
0	0	0	0	1	0	0	
0	0	0	1	0	0	0	
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	0	0	0	1	0	
0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	
-1	-1	-1	-1	-1	-1	-1	0
-1	-1	-1	-1	-1	-1	-1	

0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0
0	1	0	0	0	1	0	
0	0	0	0	1	0	0	
0	0	0	1	0	0	0	
0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0

(a) "2" の標準パターン

(b) (a)の欠落パターン(I)

(c) (a)の欠落パターン(II)

図 2 . 2 "2" の標準パターンと欠落パターン

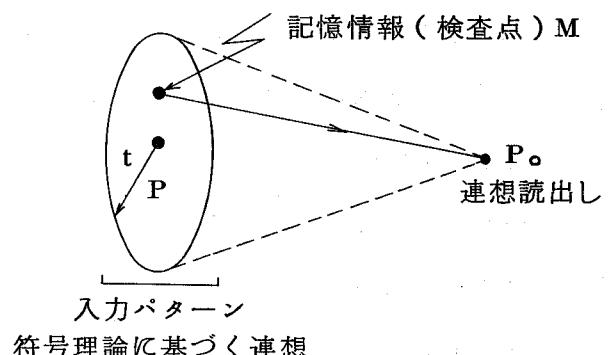
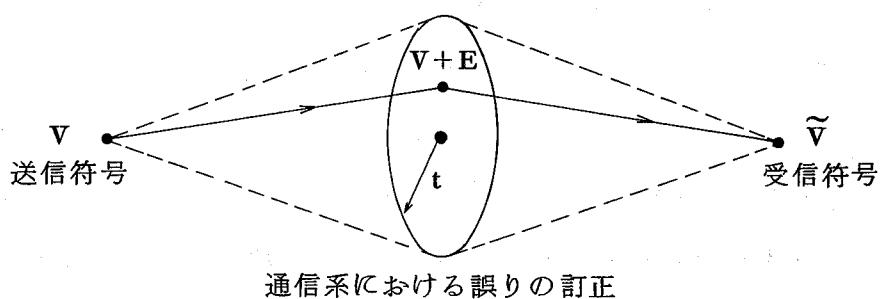


図 2 . 3 符号理論に基づく連想メカニズム

通信系における誤り訂正と、符号理論に基づく連想メカニズムを対比して論じる。 t 誤り訂正符号を用いた通信系においては、図 2.3 に示すように、送信符号 V に対する受信符号 \bar{V} ($= V + E$) の誤りパターン E の重みが t 以下で、 \bar{V} が V を中心とする誤り半径 t の円内に入っていれば、 \bar{V} は正しく V に復号される。ここで、送信符号 V は情報記号 I と検査記号 R とからなる。

$$V = \underbrace{[a_1, a_2, \dots, a_k]}_I, \underbrace{[a_{k+1}, \dots, a_n]}_R \quad (2.8)$$

このことは、図 2.3 に示すように、連想メカニズムと次のような対応をもつ。情報記号列 I に標準パターン P を対応させる。もし、標準パターンの一部分が欠損した時、この欠損部分を復元するには、検査記号列 R を参照すればよい。このことはまた以下のように説明できる。欠損パターンを $P'(x)$ と表現すると、

$$P'(x) = P(x) + E(x) \quad (2.9)$$

$$\begin{aligned} P'(x) + R(x) &= P(x) + E(x) + R(x) \\ &= (P(x) + R(x)) + E(x) \\ &= V(x) + E(x) \end{aligned} \quad (2.10)$$

となる。もし、 $E(x)$ が誤り訂正能力内の誤りであれば、 $E(x)$ を $G(x)$ で割った余りを手がかりにして、 $E(x)$ の形を見い出し、訂正することができる。すなわち、欠損部分の字の部分が復元できることとなる。

2.3 記憶方法

本節では、2種類の記憶方法について説明する。一つは標準パターン全体を一つの情報記号列とみなし、検査点を生成する方法である。もう一つは、標準パターンを m 行に分け、各行を一つの情報記号列とみなし、検査点を生成する方法である。

2.3.1 記憶方法(A) (パターン・チェック法)

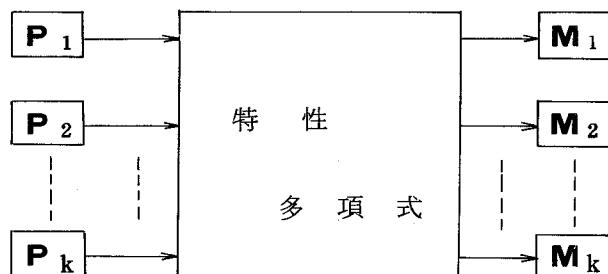
図2.1のように $m \times n$ 個の要素 a_{ij} ($\in (0, 1)$) ($i = 1, 2, \dots, m, j = 1, 2, \dots, n$) より構成されている標準パターン P_k ($k = 1, 2, \dots, K$)

$$P_k = (a_{11}, a_{12}, \dots, a_{1n}, a_{21}, \dots, a_{mn}) \quad (2.11)$$

に対し、巡回符号を構成するように検査点 $(M_k)_1$ を作成する。標準パターン P_k に対応して、検査点 $(M_k)_1$ を記憶装置に記憶する。

$$P_k \leftrightarrow (M_k)_1 \quad (k = 1, 2, \dots, K) \quad (2.12)$$

この記憶方法を用いた場合、符号長が長いため欠損によって発生した誤り個所が多くなると考えられる。よって、この連想方式は復元する時間が多くかかる懸念があり、また誤り訂正能力を高くとる必要が生じ、検査点のビット数が多くなる。この欠点を除去するため、各行が一つの符号を構成していると考え各行ごとに各検査点を生成し記憶することが考えられる。



標準パターン

検査パターン
(記憶内容)

図2.4 記憶方法(A) (ライン・チェック法)

2.3.2 記憶方法(B) (ライン・チェック法)

標準パターン P_k ($k = 1, 2, \dots, K$) を次のように表現する。

$$\mathbf{P}_k = \begin{pmatrix} a_{11}, \dots, a_{1n} \\ a_{21}, \dots, a_{2n} \\ \vdots & \ddots \\ a_{m1}, \dots, a_{mn} \end{pmatrix} = \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_m \end{pmatrix} \quad (2.13)$$

$$\mathbf{a}_j = (a_{j1} \ a_{j2} \ \dots, \ a_{jn}) \quad (2.14)$$

一つの行 \mathbf{a}_j を一つの情報点列とし、それに対応して生成した検査点を \mathbf{C}_j とする。

$$\mathbf{a}_j \longleftrightarrow \mathbf{c}_j$$

$$\mathbf{c}_j = (C_{j1}, \dots, C_{jk}) \quad (2.15)$$

標準パターン \mathbf{P}_k に對し、

$$(\mathbf{M}_k)_2 = \begin{pmatrix} \mathbf{C}_{1k} \\ \vdots \\ \mathbf{C}_{mk} \end{pmatrix} \quad (2.16)$$

を記憶パターンとして記憶する。

文字を認識対象とした場合、ある行に字の部分“1”が集中することが多い。この時、一つの行が欠損すると誤り数が多くなり、符号長の短い符号にしては、高い訂正能力が要求されることになる。この欠点を除去するため、次の記憶方法を提案する。

標準パターン \mathbf{P}_k の名 (i, j) に對し、次のような数字 t に変換する。

$$t = (i-1) \times n + j \quad (2.17)$$

すなわち、 $\mathbf{S} = (1, 2, 3, \dots, m \times n)$ が得られる。この行に對し、次の変換 T を考える。

$$\mathbf{T} = \begin{pmatrix} 1, 2, \dots, t, \dots, m \times n \\ b_1, b_2, \dots, b_t, \dots, b_{mn} \end{pmatrix} \quad (2.18)$$

ただし、 b_1, b_2, \dots, b_{mn} はランダムな系列とする。

この変換に従い、画素 t の明るさ a_t は、要素 b_t の明るさを表わすことにする。

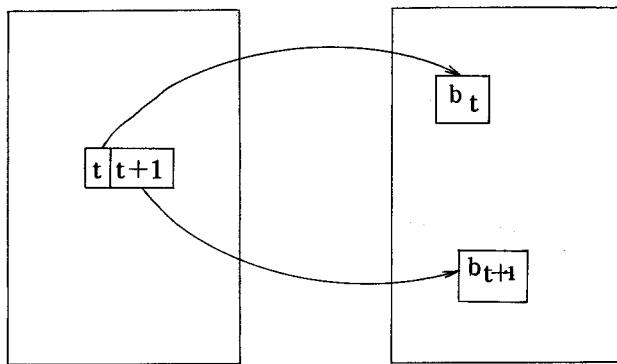
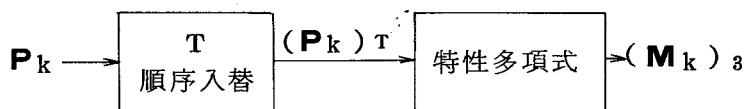


図 2 . 5 ランダム順序入替

このことは、隣接した要素間の相関をたち切り、各要素をランダムに配列することにより、一つの行に“1”の要素の個数が集中することを防ぎ、各行に平均して“1”の要素を分散したことになる。このことにより、低い誤り訂正能力を持つ符号を用いてよいことにもなる。また、同じ高い訂正能力を用いても、より多くの欠損個所を検出することが可能となる。

のことより、記憶方法は次の様になる。標準パターン \mathbf{P}_k の変換後のパターンを $(\mathbf{P}_k)_T$ として、 $(\mathbf{P}_k)_T$ の各行を一つの情報点列とみなし、 m 個の検査点 $\mathbf{C}_1', \mathbf{C}_2', \dots, \mathbf{C}_m'$ を生成し、これらを記憶する。

$$(\mathbf{M}_k)_3 = \begin{pmatrix} \mathbf{C}_{1k}' \\ \mathbf{C}_{2k}' \\ \vdots \\ \mathbf{C}_{mk}' \end{pmatrix} \quad (2.19)$$



標準パターン 記憶パターン

図 2 . 6 記憶方法(B) (ライン・チェック法)

2.4 想起方法

2.4.1 想起方法(A) (パターン・チェック法)

記憶方法(A)で記憶されたチェック・パターンを使用して想起する方法について述べる。入力パターン \mathbf{U} としては、図2.2(c)で示した欠落パターンを考え、記憶の場合と同様に多項式 $U(x)$ で表現する。 \mathbf{U} の提示部分（欠落していない部分）には誤りが生じていないと仮定する。記憶装置に記憶されているパターン $M_i(x)$ ($i=1, 2, \dots, K$) を式(2.20)に示す様に欠落パターン $U(x)$ に付加し、 $U_i'(x)$ を作成する。

$$U_i'(x) = U(x)X^{n-k} + M_i(x) \quad (i=1, 2, \dots, K) \quad (2.20)$$

この $U_i'(x)$ を復号化回路にかけ、 $U_i'(x)$ の誤り訂正を実行する。各 $M_i(x)$ に対応して誤り訂正を実行するため、一つの入力パターン $U(x)$ に對し K 個の訂正パターン $U_i(x)$ が得られる。もし誤り（欠落した部分の字の部分）の数が訂正能力範囲内であり、 $U(x)$ に適切な $M_r(x)$ が付加されるならば、適切な $M_r(x)$ に對応する標準パターン $P_r(x)$ が唯一一つ得られる。また、入力パターン \mathbf{U} の提示部には誤りがないと仮定しているので、希望される連想パターン P_r と \mathbf{U} との相関値は、他の訂正出力パターンより大きくなるゆえに、一つの $U(x)$ と K 個の $U_i(x)$ との相関をとり、最大相関値を示す $\hat{U}_o(x)$ が $U(x)$ に對応して想起されるべき連想パターンであると判定する。

図2.7にブロック図を示す。

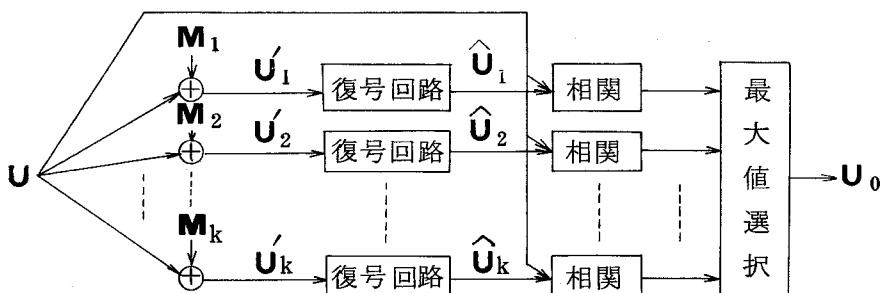


図2.7 想起方法(A) (パターン・チェック法)

2.4.2 想起方法(B) (ライン・チェック法)

2.3で述べた記憶方法(B)で記憶されたパターン(\mathbf{M}_k)₃を使用して想起する方法を述べる。入力パターン \mathbf{U} は図2.2(c)のパターンとする。入力パターン \mathbf{U} に、式(2.18)の変換 T を施し、 $(\mathbf{U})_T$ を得る。 $(\mathbf{U})_T$ の各行 d_i に対し、記憶しているパターン C_{ij} を付加する。

$$d_{ij}(x) = d_i(x)x^{n-k} + C_{ij}'(x) \quad (i=1, \dots, m), (j=1, \dots, K) \quad (2.21)$$

$$\mathbf{D}'_j = \begin{pmatrix} d'_{1j} \\ d'_{2j} \\ \vdots \\ d'_{mj} \end{pmatrix} \quad (j=1, 2, \dots, K) \quad (2.22)$$

K 個の \mathbf{D}'_j の各行ごとに復号化を施し、 \mathbf{E}_j を得る。 \mathbf{E}_j の各画素に対し、式(2.18)の逆入替 T^{-1} (2.23)を用いて変換し、得られたパターンを \mathbf{F}_j とする。2.4.1と同様、入力パターン \mathbf{U} と \mathbf{F}_j ($j=1, 2, \dots, K$)との相関をとり、最大相関値を示す \mathbf{F}_0 が \mathbf{U} に対する連想パターンとする。図2.8にプロック図を示す。

$$T^{-1} = \begin{pmatrix} b_1, b_2, \dots, b_t, \dots, b_{m \times n} \\ 1, 2, \dots, t, \dots, m \times n \end{pmatrix} \quad (2.23)$$

2.5 結 言

本章では、誤り制御機能を用いた連想方式の提案と考察を行った。この方式では標準パターンを記憶内容とせず、標準パターンに対して、巡回符号を生成するシフト・レジスターで構成可能な特性多項式を用いて検査パターンを発生させ、標準パターンに代る記憶パターンとして記憶した。また、想起について、雑音によって歪みを受けたパターン、または欠落パターンを入力として、記憶装置に蓄えられている各検査パターンを付加し、符号として、復号回路により

元の標準パターンを得た。このことは連想を一種の変換機能とみなして装置を構成している。また記憶方法として、パターン全体に対応して検査パターンを生成し、記憶するパターン・チェック法と、パターンの各要素をランダム順序入替して得たパターンの各行ごとに検査行を生成し、検査行を集めた検査パターンを記憶するライン・チェック法の2つの方法を考案した。本方式は、地

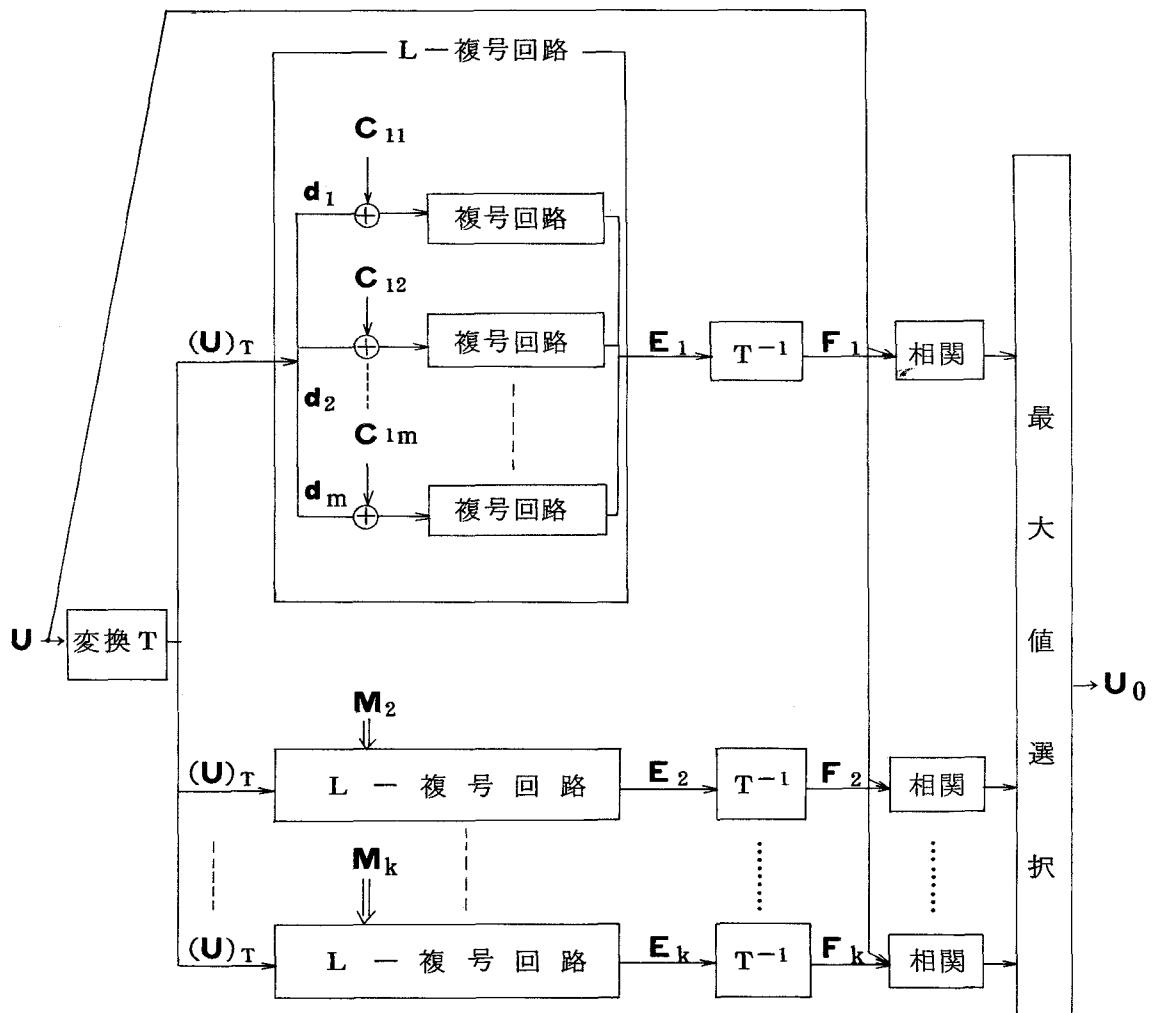


図2.8 想起方法(B) (ライン・チェック法)

(“0”)の部分の割合が字(“1”)の部分の割合より大と仮定しているので、
文字認識に対して有効な方式であることを示した。

第 3 章 被覆を用いた識別方式

3.1 緒 言

第 2 章では、符号理論を用いた連想方式を論述したが、パターンを認識するとき、入力パターン（情報）を 2 値等の有限個の組合せである論理情報だけで認識することは、実際的立場から不可能に近く、パターン間の遠近関係の情報を持つ位相情報を積極的に活用する研究が成されなければならない。

本章では連想を被覆の概念を用いて論じる。パターン集合を数学的に定式化するためには、集合の要素間の関係を明確に表現する必要がある。集合の要素間の関係、すなわち集合の構造には、代数的構造、位相的構造および順序構造がある。ここでは、被覆族に着目し、被覆族間の順序構造のなかの細分関係を使用して、連想モデルを考える。また、被覆集合が有限であるとき（有限観測可能時）の連想構造の性格について論及する。被覆族間に細分関係を導入したとき、学習と細分関係とはどのような関係にあるかについて述べる。

本章の後半において、印刷文字に類似した高品質文字（ここでは英数字 35 カテゴリー）を探り上げ、上で述べた連想モデルの実際的立場からの有効性について論じる。現在、計算機によって情報処理を行う場合に、印刷文字もしくは印刷文字に近い文字を読み取る場合が多く、こうした文字を効率良く識別するアルゴリズムが必要である。そのため、まず被覆による文字の識別方法について述べ、その被覆集合を生成するために、強制学習法を用いる方法と Prime Event 法⁽³⁴⁾による方法の 2 つの方法について述べ、これらの方法による文字識別機を構成する。被覆を用いた識別方法の有効性について、計算機による実験を行い、検討を行った。

3.2 被覆による連想モデル

本節では、連想を位相的見地から論じる。すなわち、パターンの位相的特質を考慮し、集合の被覆の定義、それに基づく連想被覆の定義を与え、連想の一モデルとして、被覆族間の細分関係を用いた連想モデルを与える。

3.2.1 連想被覆

〔定義 3.1〕

集合 X_0 に對して、 X_0 の部分集合の作る集合族

$$\mathcal{U}_0 = \{ E_{\lambda_0} ; \lambda_0 \in A_0 \} \quad (3.1)$$

が X_0 の被覆であるとは、

$$X_0 = \bigcup_{\lambda_0 \in A_0} E_{\lambda_0} \quad (3.2)$$

が成立することをいう。

上の定義に基づき、連想機構を明らかにするため、パターン集合 X 上に次のような連想被覆を定義する。

〔定義 3.2〕

パターン集合 X に對し、 X の部分集合の作る集合族

$$\mathcal{U} = \{ E_{\lambda} ; \lambda \in A \} \quad (3.3)$$

がパターン集合 X の連想被覆であるとは、

$$X = \bigcup_{\lambda \in A} E_{\lambda} \quad (3.4)$$

が成立することをいう。

ここで、連想の一モデルとして、次の様に考えることができる。連想とはパターン集合 X に對し、 X の連想被覆族 \mathcal{U}_{μ} ($\mu \in M$) が得られることである。パターン集合 X の要素 A を連想入力とした時、 X の連想被覆族 $\mathcal{U}_1, \mathcal{U}_2, \dots, \mathcal{U}_{\mu}$ の要素で、集合 A を包含する、すなわち、 $U_{A1} (\in \mathcal{U}_1) \supseteq A, U_{A2} (\in \mathcal{U}_2) \supseteq A,$

…, $U_{A\mu} (\in \mathcal{U}_\mu) \supset A$ を満たす $U_{A1}, U_{A2}, \dots, U_{A\mu}$ が A に対する連想出力とする。換言すれば、連想入力 A に対し A を包含する様な部分集合を要素として持つ連想被覆族がいくつか選び出され、その中の要素が連想出力として得られる。

上述のことを命題関数を用いて表現する。まず、連想被覆族を指定する関数を F_{U_i} とすると、

$$F_{U_i}(A) = \begin{cases} 1 & A \in U_i \lambda \quad (\text{ある } \lambda) \\ 0 & A \notin U_i \lambda \quad (\text{すべての } \lambda) \end{cases} \quad (3.5)$$

となる。この命題関数を用いると、 $F_{U_i}(A) = 1$ を満足する連想被覆 \mathcal{U}_i がとり出される。次に \mathcal{U}_i の要素中、連想入力 A に対応して出力される要素を選び出す操作を次の命題関数 $P_{U_A i}(A)$ を使用して表現する。

$$P_{U_A i}(A) = \begin{cases} 1, & A \in U_{A i} \\ 0, & A \notin U_{A i} \end{cases} \quad (3.6)$$

$P_{U_A i}(A) = 1$ を満足する $U_{A i}$ が連想出力として得られる。

パターン集合 X に対する連想被覆族間には、種々な関係を導入することができるが、ここでは同値、部分、細分について述べる。

〔定義 3.3〕⁽³⁵⁾

パターン集合 X に対して二つの連想被覆 $\mathcal{U} = \{U_\lambda ; \lambda \in \Lambda\}$, $\mathcal{V} = \{V_\mu ; \mu \in M\}$ が存在し、

(Ⅰ) $\Lambda \subset M$

(Ⅱ) すべての $\lambda \in \Lambda$ に対して, $U_\lambda = V_\lambda$ である。

の条件を満足している時、 \mathcal{U} は \mathcal{V} の部分をなすという。

〔定義 3.4〕

パターン集合 X に対して二つの連想被覆 $\mathcal{U} = \{U_\lambda ; \lambda \in \Lambda\}$, $\mathcal{V} = \{V_\mu ; \mu \in M\}$ が存在し、

(i) $\lambda \in A$ に對し, $x (\in X)$ に無關係に $\mu \in M$ を適當にとると,
すべての $x \in X$ に對し, $V\mu(x) \subset U\lambda(x)$ が成立する。

(ii) (i) の逆が成立する。

この時, U と V は同値であるといふ。

[定義 3 . 5]

パターン集合 X の二つの連想被覆 $U = \{U\lambda; \lambda \in A\}$, $V = \{V\mu; \mu \in M\}$
において, 任意の U は或る $V\mu$ に含まれているとき, U を V の細分といふ,

$$U > V \quad (3.7)$$

と表わす。

次節では, 細分関係をもとに連想モデルについて考察する。

3 . 2 . 2 細分関係を持つ連想構造

前節では, 連想はパターン集合から直接連想する方法について述べたが, ここではパターン集合より得られる連想被覆族間で行われる連想について述べる。

パターン集合 X の二つの連想被覆 U , V との間に, U が V の細分関係にある時, V は U より上層にあると定義する。この細分関係は順序関係を表わすこととなり, 連想被覆族 U_1, U_2, \dots, U_n の間に次のように表わす関係が成立するものが存在する。

$$U_1 < U_2 \dots < U_n \quad (3.8)$$

このときの細分は全順序である。

ここにおいて, 観測集合により連想被覆 U_n の要素 $U_{n,i}$ が連想された時, 次にこの $U_{n,i}$ を連想入力として, 連想被覆 U_n より上層にある U_{n-1}, \dots, U_1 のうちある適當な被覆族 U_j の要素 $U_{j,i}$ が連想出力として得られる。また, この $U_{j,i}$ を入力として U_j より上層にある $U_\kappa (\kappa < j)$ の要素が連想される。これらのこととを図示すると, 図 3 . 1 が得られる。

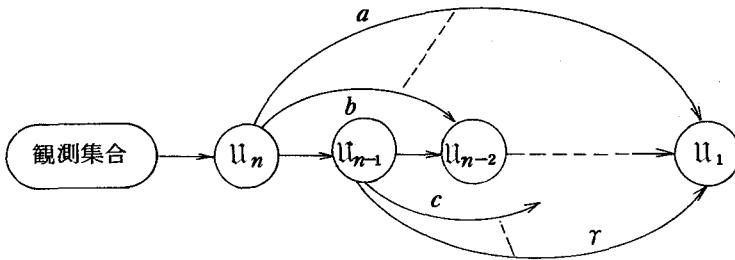


図 3 . 1 全順序関係にある被覆族間における連想

細分関係は順序関係であるので、被覆族間の関係を半順序、全順序等、種々の形で論じることができるが、制限の一番ゆるい半順序で論ずれば、あいまいさをより多く含んでいるため、連想として興味ある事実が生じてくると思われる。

被覆族数が有限である場合には、それらの間の関係を Hasse 線図で表現することができる。図 3 . 2 に一例を示す。図において Vertex は被覆を表わし、上の位置にある被覆は下の位置にある被覆を覆っている。下層の被覆から順次上層の被覆にある要素が連想されることとなる。

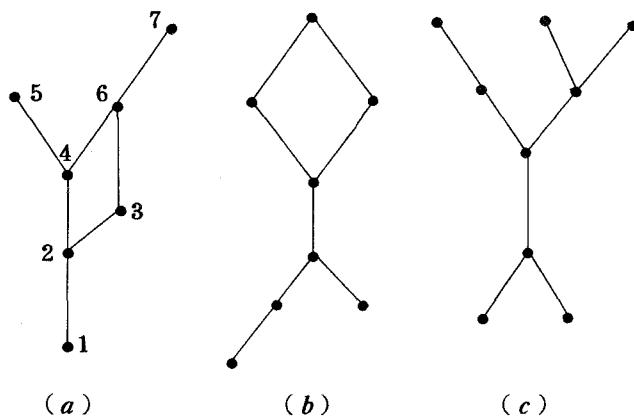


図 3 . 2 被覆族間の関係の一例

3.2.3 細分関係と学習

連想構造の一つである細分関係をもつ連想モデルにおいて、細分関係と学習との関係について論じる。

学習について論じる前に、位相について説明する。パターン集合の位相は、そのパターン集合の族として与えられる。より詳細に述べると、いまパターン集合 X は有限個の元よりなるとする。 X の位相とは次の性質をもつ X の部分集合の族のことである。

$$O \text{ I } X \in \mathcal{O} \text{ および } \emptyset \in \mathcal{O}$$

$$O \text{ II } O_1 \text{ および } O_2 \in \mathcal{O} \text{ ならば, } O_1 \cap O_2 \in \mathcal{O}$$

$$O \text{ III } \text{すべての } O_\lambda \in \mathcal{O} (\lambda \in \Lambda) \text{ ならば, } \bigcup_{\lambda \in \Lambda} O_\lambda \in \mathcal{O}$$

に属する X の部分集合を (X の) 開集合という。

集合 X に二つの位相 $\tau_x, \tau_{x'}$ が定義されたとき、 $\tau_x, \tau_{x'}$ の位相の強弱に関しては次の〔定義 3.6〕で与えられる。

〔定義 3.6〕

集合 X において定義された 2 つの位相において、 τ_x に属する任意の V がまた $\tau_{x'}$ に属しているとき、

$\tau_{x'}$ は τ_x より強い (τ_x は $\tau_{x'}$ より弱い) 位相

といい、記号

$$\tau_{x'} \geq \tau_x \quad (3.9)$$

で表わす。

位相の強弱を用いて、〔定義 3.7〕で分離可能(識別可能)について述べる。

〔定義 3.7〕⁽³⁶⁾

τ_x を X の位相とする。 X の二つの元 x, y ($x \neq y$) に對して、 V 又は V' ($\in \tau_x$) が存在して、

$$x \in V, \quad y \notin V$$

$$\text{または } x \notin V, \quad y \in V'$$

となるとき、 x, y は位相 τ_x のもとで分離可能（または識別可能）である、という。また、 x, y は集合 V （または V' ）によって分離（識別）されるという。

以上の定義をもとに学習過程について論じる。いま、 x, y ($x \neq y$) $\in X$ が位相 τ_x のもとで分離可能でないとする。このとき、 x, y を分離するような集合を含む新しい位相 $\tau_{x'} (\geq \tau_x)$ を作ることを考える。このことを学習ということができる。すなわち、学習過程は X の位相を強化していく過程であると考えることができる。このことによると、被覆族間の部分、細分は位相の強化の一例であると考えられる。

細分の関係にある 2 つの被覆 $\mathcal{U} = \{U_\lambda ; \lambda \in \Lambda\}$, $\mathcal{V} = \{V_\mu ; \mu \in \nu\}$ において、次のような位相を導入する。いま \mathcal{U} を \mathcal{V} の細分であるとし、 \mathcal{V} に対して \mathcal{V} の部分集合の族を位相 τ_v として導入する。 \mathcal{U} に対して、位相 τ_u を以下のように決める。

τ_v の任意の要素 V_μ は τ_u の要素となる。

かつ、任意の U_λ ($\lambda \in \Lambda$) は τ_u に属す。

以上のように τ_v , τ_u を導入すれば、

$$\tau_u \geq \tau_v \quad (\tau_u \text{ は } \tau_v \text{ より強い位相}) \quad (3.10)$$

が成立する。証明は自明であるので省略する。

よって、細分の関係にある 2 つの被覆 \mathcal{U}, \mathcal{V} の細分関係と位相の強弱との関係は

$$\mathcal{U} > \mathcal{V} \rightarrow \tau_u \geq \tau_v \quad (3.11)$$

となり、細分化は学習過程の一つであると考えられる。

また、部分の関係にある 2 つの被覆 \mathcal{U}, \mathcal{V} に対しても同様に考えられ、 \mathcal{U} が

の部分である時には \mathcal{U} に導入した位相 τ_u の方が、 \mathcal{V} に導入した位相 τ_v より強いと言え、 部分化も学習過程の一つである。

3.2.4 有限観測可能時の連想構造

ここでは、 パターン集合 X に位相 τ が導入され、 コンパクト位相空間 (X, τ) を成していると仮定する。 (X, τ) がコンパクトであれば、 パターン集合 X は有限個の連想被覆で覆われることになり、 文献(39)で論じられているように有限観測可能（有限個の観測値を有する観測機能が存在する。）であるといえる。以下において (X, τ) がコンパクトである時の連想構造について述べる。

〔補題 3.1〕

X をコンパクト空間、 $\mathcal{U} = \{U_\alpha ; \alpha \in A\}$ を X の有限開被覆とする。 A の部分集合 S に対し、 s に属する添数をもつ開集合の共通部分 $\cap \{U_\alpha ; \alpha \in S\}$ を U_s と書く $U_s \neq \emptyset$ を満足する α からなる A の部分集合を頂点の集合とする一つの抽象複体をつくる。

抽象複体が与えられると、 これから一意的に単体的複体が定まる。⁽³⁸⁾

〔補題 3.2〕

X をコンパクト空間、 $\mathcal{U} = \{U_\alpha ; \alpha \in A\}$ 、 $\mathcal{V} = \{V_\beta ; \beta \in B\}$ をそれぞれ X の有限開被覆とし、 \mathcal{V} が \mathcal{U} の細分である ($\mathcal{U} \subset \mathcal{V}$) 時、 $\beta \in B$ に対し $V_\beta \subset U_\alpha$ となる $\alpha \in A$ を対応させる写像を $\pi : B \rightarrow A$ とすれば、 π は単体写像 : $X_{\mathcal{V}} \rightarrow X_{\mathcal{U}}$ をひきおこす。

上の事柄をまとめると、 X がコンパクト空間の時、 連想被覆は抽象複体を構成し、 細分関係にある 2 つの連想被覆 \mathcal{U}, \mathcal{V} には単体写像が存在する。

3.3 被覆による識別方法

前節で述べた連想被覆を具体的なクラス識別に用いる手法について考察する。ここでは、2値パターンを対象として議論を行う。

カテゴリー集合を $C = \{C_1, C_2, \dots, C_m\}$, カテゴリー C_i から発生したパターンの集合を $A_i = \{A_{i1}, A_{i2}, \dots, A_{in}\}$ と表現する。

〔定義3.8〕

パターン \bar{A}_{ik} とパターン A_{ik} において、 \bar{A}_{ik} のすべての要素が A_{ik} の対応する要素の値と等しい時、パターン \bar{A}_{ik} はパターン A_{ik} を被覆するという。(ただし、don't care の要素は他の値のいかんにかかわらず等しいものとする。)

〔定義3.9〕

カテゴリー C_i より生じたパターン集合 A_i のうち、いくつかのパターンを被覆するパターンを被覆パターンと定義する。

〔定義3.10〕

カテゴリー C_i のパターン A_{ij} を全て被覆することができる必要最少限の被覆パターン $R_{i\alpha}$ の集合を被覆集合 P_i と定義する。

上の被覆パターンの定義の方法は2つのパターン間の距離に注目したものであり、パターン間の距離がパターンの類似度をはかる基準となっている。しかし、被覆集合 P_i の選び方は一意的には定まらない。

次に要素が“1”ある部分集合についての包含関係に着目して、前述の〔定義3.8〕よりも広義の被覆パターンの定義を行う。

〔定義3.11〕

パターン \bar{A}_{ik} とパターン A_{ik} において

$A_{ik} \subset \bar{A}_{ik}$ (\subset は値が1である成分についての包含関係を意味する)

が成り立つ時、パターン \bar{A}_{ik} はパターン A_{ik} を被覆するという。

〔定義 3 . 12〕

カテゴリー C_i より生じたパターン集合 A_i のすべてのパターンを被覆するパターンを被覆パターンと定義する。すなわち被覆パターン $R_{i\alpha}$ は

$R_{i\alpha} = \bigcup_{k=1}^n A_{ik}$ ($\bigcup_{k=1}^n A_{ik}$ は各成分について論理和をとった
パターンを表わし、 n は集合 A_i の要素の数
である。)

と定義する。

上のように定義された被覆集合を用いて観測パターンを識別する、一般的な識別手順を以下に示す。

〈被覆集合を用いた識別方法〉

- (i) 各カテゴリー C_i から発生したパターン A_{ij} の集合に對し、被覆パターンを生成し、パターンの集合 A_i に對して被覆集合 P_i を生成する。
- (ii) 観測パターン X に對し、 X を被覆する被覆パターン R_i を探し、 R_i のカテゴリーを観測パターン X に割り当てる。
- (iii) 観測パターン X がどんな被覆集合にも被覆されない時、そのパターンは棄却される。

しかし実際の文字等の識別においては、上の識別手順(ii)を満足しなくなる可能性が多い。これを補うために適当なしきい値を設け、識別を行う。次のような手順(iii)' に変更する。

- (iii)' 観測パターン X と被覆集合 P_i の被覆パターン $R_{i\alpha}$ との距離を計算し、
$$dis(X, R_{i\alpha}) < \theta \quad (\alpha = 1, 2, \dots, K_{\alpha}^i)$$
かつ $dis(X, R_{j\alpha}) > \theta \quad (j \neq i, j = 1, 2, \dots, m, \alpha = 1, 2, \dots, K_{\alpha}^j)$

(3 . 12)

が成立する時、観測パターン $X \in R_i$ のカテゴリーを割り当てる。

(iii)' (ii)' が満足されない場合、そのパターンは棄却される。

また、こうした識別法による識別誤りは次の 2 種類がある。

- (1) カテゴリー C_j からの観測パターンが、カテゴリー C_i ($i \neq j$) の被覆集合のある被覆パターンによって被覆されたと判定することにより生じる誤り。
- (2) カテゴリー C_j からの観測パターンが、カテゴリー C_j の被覆集合の被覆パターンによって被覆されないことによる誤り。

3.4 被覆集合の生成方法と識別機の構成法

ここでは、強制学習法と Prime Event 法を用いて被覆集合を生成し、識別機を構成する方法について説明する。

3.4.1 強制学習法による識別機の構成法

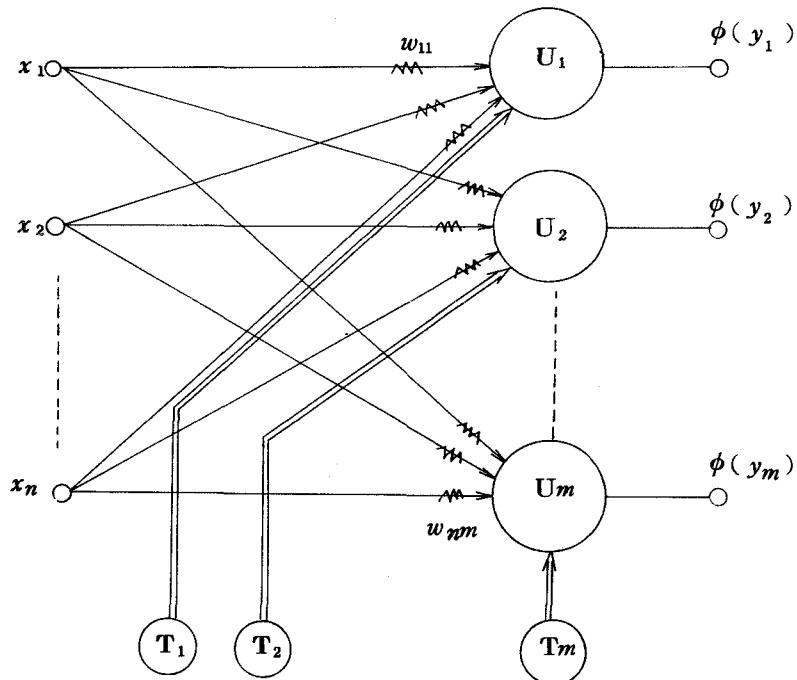


図 3.3 多クラス分類機

強制学習法は教師情報をもとにして結合係数を逐次改変していく教師付き学習方法の一つであり、この学習方法により収束した荷重行列を被覆パターンとして用いる。図3.3に学習システムを示す。ここで用いる記号について以下に説明する。

素子 U^α ($\alpha=1, 2, \dots, m$)

U^α の値 θ^α

パターン $X = (x_1, x_2, \dots, x_n)$ ただし、 $0 \leq x_i \leq x_{wax}$ とする。

パターンのクラス C^α ($\alpha=1, 2, \dots, m$)

荷重行列 $W^\alpha = (w_{1\alpha}, w_{2\alpha}, \dots, w_{n\alpha})$ ($\alpha=1, 2, \dots, m$)

ただし、 $0 \leq w_{1\alpha} \leq w_{max}$ とする。

また、クラス C^α に属するパターンを特に

$$X^{\alpha, \kappa} = (x_1^{\alpha, \kappa}, x_2^{\alpha, \kappa}, \dots, x_n^{\alpha, \kappa}) \quad (d=1, 2, \dots, m) \quad (\kappa=1, 2, \dots, \kappa_\alpha)$$

と表わす。ただし、 K^α は C^α に属するパターンの数を表わす。

U^α の教師情報発生素子 T^α ($\alpha=1, 2, \dots, m$)

T^α の出力 d_α^* (Oまたは θ^α の値をとる)

U^α の入力の総和 y_α^*, y_α

ただし、 y_α^* と y_α はそれぞれ教師情報が与えられている場合といない場合の入力の総和を表わし、それぞれ次式で与えられるものとする。

$$y_\alpha^* = (W^\alpha, X) + d_\alpha^* = \sum_{i=1}^n W_{i\alpha} x_i + d_\alpha^* \quad (3, 13)$$

$$y_\alpha = (W^\alpha, X) = \sum_{i=1}^n w_{i\alpha} x_i \quad (3, 14)$$

素子 U^α は $y_\alpha^* \geq \theta^\alpha$ あるいは $y_\alpha \geq \theta^\alpha$ で出力 1, それ以外では出力 0 を出すものとする。また、教師情報の入り方は $X \in C^\alpha$ なる入力に対しても $d_\alpha = \theta^\alpha$, $d_\beta = 0$ ($\beta \neq \alpha$) とする。すなわ

ち、一つのクラスに対して一つの素子が対応するよう教師情報が供給される。

このシステムの学習方式は次式で与えられる。

$$W_{i\alpha}(t + \Delta t) = W_{i\alpha}(t) + \eta x_i(t) \Phi(y_{\alpha}^*(t)) - \delta \quad (3.15)$$

ただし、 $0 \leq W_{i\alpha} \leq W_{max}$

$\eta (\eta > 0)$ 増減係数

$\delta (\delta > 0)$ 忘却のための減衰係数

$$\text{2値関数 } \Phi ; \quad \Phi(x) = \begin{cases} 1 : x \geq \theta \\ 0 : x < \theta \end{cases}$$

初期状態の荷重行列 $W_{\alpha}(0)$ ($\alpha = 1, 2, \dots, m$) をすべて零ベクトルとし、 η が η_{max} に比べ十分小さいとすれば、十分な学習時間を経た後、 $W_{i\alpha}$ の値は

$$\begin{aligned} W_{i\alpha} &= W_{max}, \quad \overline{\eta x_i \Phi(y_{\alpha}^*) - \delta} > 0 \text{ のとき} \\ &= 0 \quad , \quad \overline{\eta x_i \Phi(y_{\alpha}^*) - \delta} \leq 0 \text{ のとき} \end{aligned} \quad (3.16)$$

となる。従って荷重行列 W_{α} は

$$W_{\alpha} = W_{max} (\Phi_0(\overline{W_{1\alpha}}), \Phi_0(\overline{W_{2\alpha}}), \dots, \Phi_0(\overline{W_{n\alpha}})) \quad (3.17)$$

で与えられる。

図 3.3 システムで

$$\left. \begin{array}{l} (W_{\alpha}, X^{\alpha}, \kappa) \geq \theta_{\alpha} \quad (\kappa = 1, 2, \dots, K_{\alpha}) \\ (W_{\beta}, X^{\alpha}, \kappa) < \theta_{\beta} \quad (\kappa = 1, 2, \dots, K_{\alpha}, \beta \neq \alpha) \end{array} \right\} \quad (3.18)$$

がすべての α について成立する時、 C_{α} ($\alpha = 1, 2, \dots, m$) は分類可能なクラスであると考えられる。このシステムがパターンを正しく分類するための条件について、次の〔補題 3.3〕が与えられる。

〔補題 3.3〕 (40)

$X^{\alpha}, \kappa \in C_{\alpha}$ が入力されたとき

$$y_\alpha \geq \theta \quad \text{かつ} \quad y_\beta < \theta \quad (\beta \neq \alpha, \beta = 1, 2, \dots, m)$$

が成立する時, X^α, κ は正しく分類されたという。 $C_\alpha (\alpha = 1, 2, \dots, m)$ に属するすべてのパターンが正しく分類されるための必要十分条件は,

$$X^{\beta, l} \not\subseteq \bigcup_{\kappa=1}^{k_\alpha} X^{\alpha, \kappa} \quad (\beta \neq \alpha, \beta = 1, 2, \dots, m, l = 1, 2, \dots, K_\beta)$$

(3.19)

が成立することである。ただし, $\bigcup_{\kappa=1}^{k_\alpha} X^{\alpha, \kappa}$ は X^α, κ の各成分について論理和をとったパターンを表わし, \subset は値が 1 である成分の集合についての包含関係を意味する。

3.4.2 Prime Event 法による識別機の構成法(34)

Prime Event 法では被覆パターンとして Prime event を採用し, 一つのカテゴリーに割り当てられる色々な観測パターンを被覆するパターンを Prime event とする。さらに被覆集合にあたるもの definition set と呼ぶ。

識別手順は次のように書ける。

- (i) 各カテゴリーごとに Prime event を生成し, 適当な definition set を設定する。
- (ii) 観測パターンは, 観測パターンを被覆する Prime event を含む definition set のカテゴリーに割り当てられる。
- (iii) 観測パターンは, どんな definition set にも被覆されない時棄却される。

definition set は一意的には定まらないので, カテゴリーごとに生成された Prime event の最小サイズの集合を definition set として選択する。以下に Prime event の生成手順を示す。

<Prime Event 生成手順>

(I) 観測パターンの三つのリスト L, NL, LP をつくる。

リスト L ; Prime event を生成すべきある特定のカテゴリーに割り当
てられる観測パターンの集合のリスト

リスト NL ; 上記のカテゴリーに割り当てることのできない他の全ての
観測パターンの集合のリスト

リスト LP ; Prime event となりうる可能性をもつパターンを一時貯
えておくためのリスト

(II) リスト L に含まれる二つのパターンに merging (併合操作)* を施
して新しいパターンを作る。

(III) 新しく生成されたパターンがリスト NL のパターンを被覆しているか
どうかを調べ、もしリスト NL のパターンのうち一つでも被覆していれば
この新しいパターンは棄却され、リスト NL のパターンの全てを被覆
していなければリスト LP に加えられる。

(IV) (II), (III) の手順をリスト L に含まれる全てのパターンについて繰り
返す。

(V) リスト L の他のいかなるパターンと merging されても、生成された
パターンがリスト LP に加えられないようなパターンがリスト L に存在
する場合、このパターンをリスト LP に加える。

(VI) リスト L とリスト LP の内容が等しくなったらこの手順を終える。

(VII) (VI) が成立しなかった場合、リスト LP の内容を新たにリスト L の内
容として手順 (II) に戻る。

* merging は 2 つのパターンの対応する要素の値が等しい時にはその値を
等しくない時には、don't care を新しいパターンの要素の値とする。

手順 (vi) が成立した場合、リスト L あるいはリスト LP に入っているパターンを Prime event として用いる。次に Prime Event 法を用いた識別法の特色を挙げる。

- (Ⅰ) 観測パターンの確率分布は未知である。
- (Ⅱ) 構造情報を抽出するアルゴリズムである。
- (Ⅲ) 推定パラメータに対して統計的信頼度が高い。
- (Ⅳ) 構造情報を与え、かつ Prime event はクラスを識別する能力をもつ。

文字識別などの実際的な問題においては、観測パターンの確率構造は未知であり、しかも得られるサンプル数は一般に少ない。したがって Prime Event 法をこの種の問題に適用する場合、特徴抽出を効果的に行うために次の二つのしきい値 θ_A , θ_B を導入する。 θ_A , θ_B はともに有限サンプルサイズに対する不足を補償するために利用される。

(1) しきい値 θ_A

θ_A はリスト L のパターンがリスト NL のパターンを被覆するかどうかを調べる時に使用する。リスト L のあるパターンがリスト NL のパターンのうち $\theta_A\%$ 以下のパターンしか被覆していない場合には、このリスト L のパターンはリスト NL のパターンを被覆していないと考える。通常 θ_A の値は $\theta_A = 0$ である。

(2) しきい値 θ_B

リスト L のあるパターンの要素とリスト NL のパターンの要素との異なる数が θ_B 未満である時、このリスト L のパターンはリスト NL のパターンを被覆すると考える。通常 θ_B の値は $\theta_B = 1$ である。

3 . 4 . 3 実験結果

(A) 強制学習法

強制学習法を用いた文字識別実験を行った。

<実験方法>

- (1) 観測パターンはメッシュの数が縦 12 × 横 8 の 2 値パターン
- (2) 対象とするカテゴリーは数字 0 ~ 9, アルファベット A ~ Z
(ただし O は除く) の合計 35 クラス
- (3) 各カテゴリーに属する 10 個のパターン合計 350 サンプルのうち半分を訓練パターンとして荷重行列を収束させるために用い, 荷重行列が収束した後 350 個の全てのサンプルをテストパターンとして識別実験する。
- (4) 変数 ; 増減係数 $\mu = 0, 1$, 減衰係数 = 0.0001
- (5) 訓練パターンの提示の仕方 ;
 $X^{1,1}, X^{1,2}, X^{1,3}, X^{1,4}, X^{1,5}, X^{2,1}, X^{2,2}, \dots, X^{3,5,4}$
 $X^{3,5,5}$
 $X^{1,1}, X^{1,2}, \dots$
 $X^{\alpha, k}$; カテゴリー α の k 番目のパターン
- (6) W_α ($\alpha = 1, 2, \dots, m$) が収束したか否かの判定 ;
$$\max_\alpha \sum_{i=1}^{96} W_{i\alpha}(t)(1 - W_{i\alpha}(t)) < 1 (\alpha = 1, 2, \dots, m)$$
 が成立した時, 荷重行列 W_α ($\alpha = 1, 2, \dots, m$) は収束したものと考える
- (7) テストパターン X と荷重行列 W_α ($\alpha = 1, 2, \dots, m$) との相関 (内積) をとった結果, 内積値が最も大きいカテゴリーにテストパターンを割り当てる。

この結果, 荷重行列 W_α ($\alpha = 1, 2, \dots, m$) は 2151 回で収束した。文字識別実験に用いた訓練パターン, テストパターンおよび収束した荷重行列の例を, 図 3, 4 に示す。

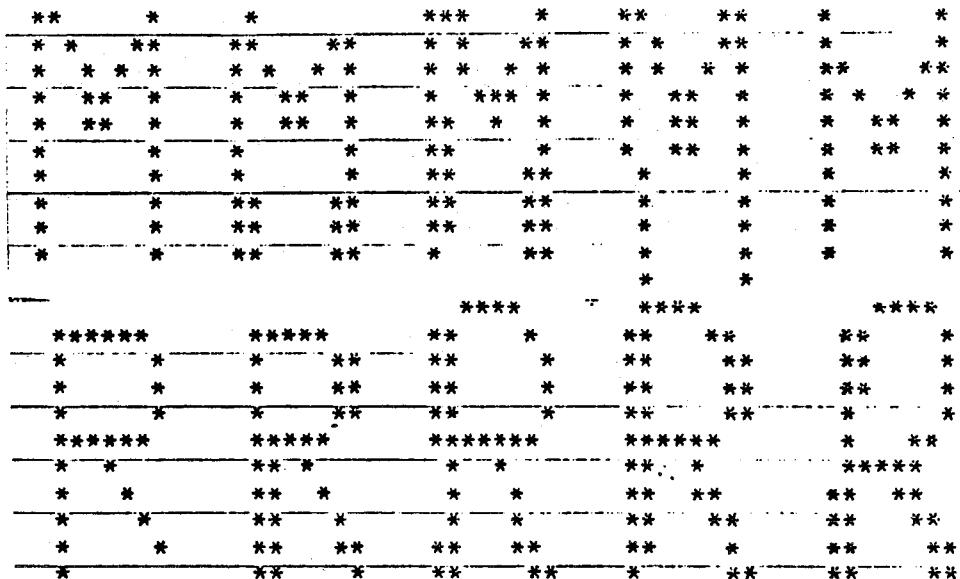


図 3 . 4 (a) 訓練パターン例

00110110	-	11111100	-	00000111	-	11100011	-	00000000
01111101	-	11111110	-	11111111	-	11100011	-	01111110
11111111	-	11111111	-	11100011	-	11110111	-	11111111
11111110	-	11101111	-	11100000	-	10111101	-	11100011
11101110	-	11111111	-	11100000	-	110111001	-	11100011
11111110	-	11111111	-	11111110	-	110111001	-	11100011
11111111	-	11111111	-	11111100	-	11000011	-	11000011
11111111	-	00111111	-	11100000	-	11000011	-	11111110
11111111	-	00111110	-	11100000	-	11000011	-	11111100
01111110	-	00111100	-	11100011	-	11000011	-	11101100
00111100	-	01111100	-	11111111	-	01000001	-	11100110
00011100	-	01111100	-	00000111	-	00000000	-	11100111
								11100011

4

9

E

M

R

図 3 . 4 (b) 収束した荷重行列 (4.9.E.M.R) 例

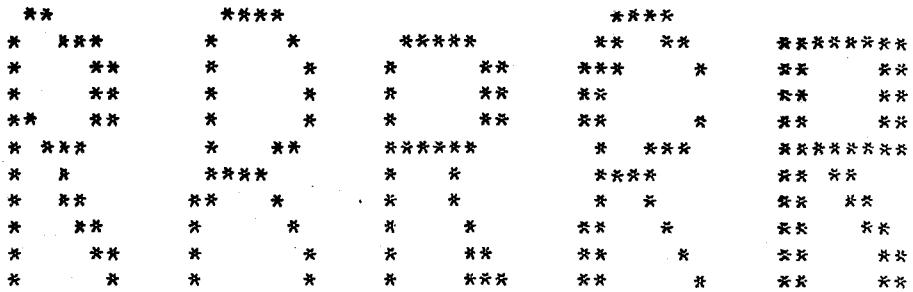


図 3 . 4 (c) テスト・パターン例

(B) Prime Event 法

<実験方法>

- (1) 観測パターン、対象とするカテゴリー数は (A) と同じ。
- (2) 各カテゴリーに属する 10 個のパターン合計 350 サンプルのうち半分を訓練パターンとして definition set を形成するのに用い、その後 350 個の全てのサンプルをテストパターンとして識別実験する。
- (3) しきい値 θ_A , θ_B はそれぞれ $\theta_A = 0$, $\theta_B = 10$ とした。

生成された Prime event の一例を図 3,5 に示す。

(C) 識別結果

表 4 . 1 被覆を用いた識別方式の識別結果

	識別率	識別速度
強制学習法	74.8 %	580 文字／分
Prime Event 法	97.4 %	450 文字／分

1 * **	* 1 * *	** 1 * *	1 1 1 1 1 1	1 1 1 1 1 1 *
* 1 * *	1 1 * *	* 1 1 * *	1 1 * 1 1 *	1 1 * 1 1 *
1 1 * *	1 1 * *	1 1 * *	1 1 1 * 1 1 *	1 1 1 * 1 1 *
1 1 * 1 1 *	1 1 * 1 1 *	1 1 * 1 1 *	1 1 1 1 1 1 *	1 1 1 1 1 1 *
1 1 * 1 1 *	1 1 * 1 1 *	1 1 * 1 1 *	1 1 1 1 1 1 *	1 1 1 1 1 1 *
1 1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1 1	1 1 1 1 1 1 *	1 1 1 1 1 1 *
* * 1 1 1 1 1	1 1 * * 1 1 1 *	1 1 * * 1 1 1 *	* * 1 1 *	* * 1 1 *
* 1 * *	* 1 * *	* 1 * *	* * 1 1	* * 1 1
* *	* 1 *	* 1 *	* * 1 *	* * 1 *
		**	**	**

(9)

(4)

* 1 1		* 1	* *	
* * 1 1 1 1 1 1		1 * * 1 1		
* 1 * * 1		1 * * 1 1		
* 1 *		1 1 1 1 1		
* 1 *		1 1 1 1 1		
1 1 1 1 1 1 *		1 1 1 1 1		
1 1 1 1 1 *		1 1 1 1 1		
1 1		1 1 1		
* 1 *		1 1 1		
* 1 * 1 1		1 1 1		
* * 1 1 1 1 1		1 1 1		
* 1 1				

(M)

(E)

*** *	* 1 1 1 *	1 1 1 1		
1 1 * * * *	* 1 * * *	1 1 1 1		
1 * * 1	* 1 * 1	1 1 1 1		
1 * * 1	* 1 * 1	1 1 1 1		
1 * * 1	* 1 * 1	1 1 1 1		
1 * * * 1 *	* 1 * * * 1 1	1 1 1 1 1 1		
1 * * 1 *	* 1 * * 1 *	1 1 1 1		
* 1 * * 1	* 1 * 1	1 1 1 1		
* 1 * * 1	* 1 * 1	1 1 1 1		
* 1 * * 1	* 1 * 1	1 1 1 1		
* 1 * * 1	* 1 * 1	1 1 1 1		

(R)

図3.5 生成されたPrime eventの例 (*はdon cave)

3.5 結 言

本章では、パターン集合を覆う被覆族において、細分関係を持つ連想構造について考察し、後半において被覆の具体的構成として強制学習法と Prime Event 法を用い、文字識別実験を行った。

パターン集合に対して連想被覆を定義し、連想被覆族間に部分、同値、細分の関係があり、特に細分関係をもつ連想被覆族間で、上層、下層の関係を設け下層の連想被覆の要素から上層の連想被覆の要素を連想する連想モデルを提案した。そして、有限個の観測値で表現されるパターン集合の連想構造は抽象複体、さらに単体的複体を構成し、細分関係をもつ単体的複体間には単体写像が存在することを指摘した。また学習と位相の強弱において、位相の強化が学習の過程であり、より詳しく細分することは位相の強化となり、細分化が学習の過程となることが明らかとなった。

被覆集合を具体的に構成する手法として強制学習法と Prime Event 法を挙げ、この 2 つの識別法の識別能力を評価するために、英数字 35 クラスを対象として文字識別実験を行った。その結果、強制学習法の場合、識別率 74.8 % 識別速度は 1 分間に 580 文字、Prime Event 法の場合、識別率 97.4 %、識別速度は 1 分間に 450 文字であった。Prime Event 法と強制学習法を比較すると、他のカテゴリーの情報も取り入れて被覆集合を構成している点、Prime Event 法の方が優れており、このことが識別率の差となって現われている。以上の結果、高品質手書き文字の識別機を構成する際、文字の構造的特徴を考慮に入れない本手法の有効性が認められた。

4.1 緒 言

情報処理における基本的な問題の一つとして、クラスタリングがある。クラスタを同じ混合成分から生成される観測集団と考えると、クラスタリングとは、未分類データを事前情報（クラスタの個数、クラスタの構造など）が一切得られていない状態で、幾つかのクラスタに分割することをいう。すなわち、同じクラスタ内では類似の度合いが大きく、異なるクラスタ間ではこれが小さくなる様に幾つかの部分集団に分割することである。このことより、クラスタリングはパターン認識の分野において、認識対象であるパターンが生成するパターン空間におけるサンプルパターンの分布に関する情報を教師なしで習得する技法となる。また、クラスタリングは医療情報の分類、言語や文字などの分類、計量心理学における因子分析などのデータ処理などの幅の広い応用をもつ情報処理である。

クラスタリングについて考察する時、次のような問題点があげられる。

- (1) 取扱うデータは多種多様、非標準的である。
- (2) データとして得られた n 個のサンプルを r 個のクラスタに分割する方法は第2種の Stirling 数* で与えられ、一般には相当の計算量を必要とする。
- (3) 事前情報としては距離情報だけであると考えてよい。

従来まで種々のクラスタリングの手法が提案されているが、その代表的なものは次のようにあげられる。

* $S_2(n, r) = \sum_{k=0}^r (-1)^{r-k} \cdot K^n (K! (r-k)!)$

- (1) 与えられた各サンプル点間に、適當な類似性を示す尺度（たとえばユークリッド距離の逆数）を与え、類似度の大きなものから各点を階層的に順次統合し、樹形図を形成し、適當な基準により各点をクラスタ分割する。⁽⁴⁵⁾
- (2) 各サンプル点間を結合して枝を構成し、各枝に非類似性を示す尺度による重みを与え、各枝に対する重みの和が最小となるように、各点を Tree で連結する（このようにして構成された Tree を Minimal Spanning Tree(M.S.T.) という）。構成された Treeにおいて、ある枝の重みが、その周辺の枝の重みと比較し、十分大きい場合、その枝を除去する操作を行い、いくつかの Sub-Tree に分割し、各 Sub-Tree を一つのクラスタとみなしがクラスタ分割を終える。⁽⁴⁶⁾
- (3) 各クラスタの代表点を仮定し、各サンプル点をその代表点との距離が最小となるクラスタに属するとする。生成された仮のクラスタにおいて、クラスタの中心を新たな代表点とし、それを用いて再びクラスタ分割を行う。この操作を、クラスタの代表点が収束するまで繰り返し、クラスタ分割を終える。^{(46), (47)}
- (4) サンプル点を用いて、空間におけるサンプルの分布関数の推定を行い、それをもとに密度関数およびその勾配を求め、密度関数の極小点または谷を求め、それをクラスタの境界として、各サンプルのクラスタ分割を終える。⁽⁴⁸⁾
- その他、適當な評価基準を定め、それを最適化にする方式、分布のモードを求め遂次クラスタを構成する方式などがある。しかし一般にこれらの手法は、高次元、多サンプルの時、多くの複雑な計算時間、多くの記憶容量が予想され、クラスタ分割は困難であると思われる。
- 本章では以上の観点から、多サンプルの時に大きな問題となるであろう問題点(2)を軽減し、クラスタリングを行う方式を提案する。クラスタリングを行う

時、サンプルが一様な分布である場合を除いて、クラスタの中心付近には情報を担っているサンプルが密になっていると考えられる クラスタの中心としてモードを推定し、クラスタを構成する方法が提案されているが、ここではモードを包含する集合（峰クラウド）を抽出し、その抽出された峰クラウドの集合にM.S.T. 法を適用しクラスタ分割を行う方法を提案している。この方法では M.S.T. 法を適用する峰クラウドの集合がもとのサンプル集合に比較し非常に減少しているために、多サンプル集合でも比較的計算時間も少なく、容易にクラスタ分割が行える。峰クラウドの抽出方法としては、分布の峰付近では勾配が緩やかであることを利用して抽出することが考えられる。サンプルを用いて、その分布の勾配を算出する方法としては、Koontz 等で提案されている手法がある。⁽⁴⁹⁾ この手法では、各点の近傍密度を求め、2点間の勾配は2点間の距離の比で定義し、サンプル点 X_i での勾配は、 X_i の r -近傍内のサンプル点 X_j との勾配のうちの最大値で与えている。しかし、この方法では多サンプルの時非常に多くの計算時間を必要とする。ここでは文献⁽⁴⁸⁾に考察されている Parzen Estimate 法によって分布の密度関数を推定し、その一次微分で勾配を算出する方法をさらに簡略化し、正確な勾配に代る擬似勾配を容易に算出する方法を提案する。

4.2 クラスタリングの一般的表現 ^{(41), (42)}

クラスタリングは次の2つの段階を経て遂行される。

(i) データ表現

- (ア) クラスタ分割される対象点 X はどんな空間を構成しているか、を決定もしくは仮定する。（初期データ構造）
- (イ) 対象 X の観測結果は n 個のパラメータ (x_1, x_2, \dots, x_n) のリストで表現され、パラメータ x_i は量的数又は色、病名などの質的なものを表わす。

- (ウ) 初期データ構造において距離(非類似度)の計算式を定義する。
- (エ) 初期データ構造に与えるラベル名、及びラベル数を決定する。
- (オ) 初期データ表現から情報を減少もしくは特徴抽出をし、それらを記号表現する。

(ii) 識別 ; ϵ

(i) の (オ) で表現されたデータを用いて、識別を実行する。すなわち、サンプル集合 E からクラスタ分割 P への写像 ϵ を行う。

分割されたクラスタの特性を普遍的に表わすことは困難であるが、次のような (i) クラスタの同質性 (Homogeneity) と (ii) クラスタ分割の安定性 (Stability) が考えられる。

(i) 同質性 (Homogeneity)

クラスタ P_s が同質であるための必要十分条件は、

$X_i, X_j \in P_s$ かつ $X_k \notin P_s$ である時、

$$d(X_i, X_j) \leq d(X_i, X_k) \text{かつ} d(X_i, X_j) \leq d(X_j, X_k)$$

が成立することである。

分割 $P = (P_1, P_2, \dots, P_s, \dots, P_N)$ が同質であるとは、上の条件がすべての P_s に対し成立することをいう。

クラスタ及び分割の同質性を論じる時、次の鎖距離 (Chain distance) が通常用いられる。

(ア) 鎖距離 (Chain distance) δ

X_j, X_k 間の鎖 (Chain) を C_{jk} 、 X_j と X_k 間のすべての鎖の集合を ϵ_{jk} と表現する。

$$C_{jk} = X_j, \dots, X_q, X_{q+1}, \dots, X_k$$

すべての q に対して、次の $\lambda(C_{jk})$ を定義する。

$$\lambda(C_{jk}) = \sup[d(X_q, X_{q+1})]$$

この $\lambda(C_{jk})$ を用いて、鎖距離 $\delta(X_j, X_k)$ は、

$$\delta(X_j, X_k) = \inf [\lambda(C_{jk})] \quad (\text{ただし, } C_{jk} \in \varepsilon_{jk})$$

と定義される。この δ は ultradistance の条件を満たす。

(ii) クラスタ及び分割 P の安定性を普遍的に論ずることは困難であるが、次のような形で述べることが可能である。

$X_i, X_j \in P_s$ に対して、

$$\text{Sup } P_s = \max [d(X_i, X_j)]$$

とする。クラスタ P'_s はクラスタ P_s 内のある対象点を除去し、新しい対象点を付け加えて得られたクラスタとする。

もし、 $\text{Sup } P_s > \text{Sup } P'_s$

が成立すれば、 P_s は P'_s より適切なクラスタと言う。ゆえに、 P_s より適切な P'_s が存在しない時、 P_s は安定であると言うことが出来る。

4.3 擬似勾配によるクラスタリング

本節では、擬似勾配を定義し、それを用いてクラスタ分割のアルゴリズムを明らかにする。

4.3.1 擬似勾配

ここでは、観測によって得られたサンプル点の集合は n 次元ユークリッド空間を構成すると仮定する。多くのクラスタリングの問題においては、得られたサンプルの真の分布密度関数の事前情報はまったくなく、そのため分布の勾配を得るために、ノンパラメトリックな手法で分布密度関数を推定しなければならない。ここでは、ノンパラメリックな手法として、Parzen Estimate 法が有効であると考え、この手法を用いて分布密度関数を推定し、擬似勾配を与える。

N 個の独立なサンプル \mathbf{Y}_i ($i = 1, 2, \dots, N$) が得られた時、密度関数 $P(\mathbf{X})$ の推定された関数を $\hat{P}_N(\mathbf{X})$ と表現すれば、次のようになる。

$$\begin{aligned}\hat{P}_N(\mathbf{X}) &= \frac{1}{N} \sum_{i=1}^N \left(\frac{1}{h_N^n} \right) k\left(\frac{\mathbf{X} - \mathbf{Y}_i}{h_N} \right) \\ &= \frac{1}{N} \sum_{i=1}^N \left(\frac{1}{h_N^n} \right) k\left[\frac{x_1 - y_{1i}}{h_N}, \frac{x_2 - y_{2i}}{h_N}, \dots, \frac{x_n - y_{ni}}{h_N} \right] \quad (4.1)\end{aligned}$$

ただし $\mathbf{X} = (x_1, x_2, \dots, x_n)^T$

$\mathbf{Y}_i = (y_{1i}, y_{2i}, \dots, y_{ni})^T$ である。

ここで $k(\mathbf{Y})$ は次式 (4.2) ~ (4.5) を満足するスカラー関数である。

$$\sup_{\mathbf{Y} \in \mathbb{R}^n} |k(\mathbf{Y})| < \infty \quad (4.2)$$

$$\int_{\mathbb{R}^n} |k(\mathbf{Y})| d\mathbf{Y} < \infty \quad (4.3)$$

$$\lim_{\|\mathbf{Y}\| \rightarrow \infty} \|\mathbf{Y}\|^n k(\mathbf{Y}) = 0 \quad (4.4)$$

$$\int_{\mathbb{R}^n} k(\mathbf{Y}) d\mathbf{Y} = 1 \quad (4.5)$$

ただし、 $\|\cdot\|$ はユークリッドノルム、 \mathbb{R}^n は n 次元ユークリッド空間である。またパラメータ h_N はサンプル数 N の関数であり、条件を与えることで、 $\hat{P}_N(\mathbf{X})$ が $P(\mathbf{X})$ に対し、 N が無限大の時、漸近的に不偏、一致推定となることが保証されている。したがって、式 (4.1) で与えられる推定値より、密度関数 $P(\mathbf{X})$ の勾配の推定値は、式 (4.1) を微分することにより次式で与えられる。

$$\text{grad}_{\mathbf{X}} \hat{P}_N(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^N \frac{1}{h_N^n} \text{grad}_{\mathbf{X}} k\left(\frac{\mathbf{X} - \mathbf{Y}_i}{h_N} \right) \quad (4.6)$$

ここで、 $\text{grad}_{\mathbf{X}} k(\cdot)$ は、関数 $k(\cdot)$ の、変数 \mathbf{X} に関する勾配を表わし、 $k(\mathbf{X})$ は微分可能であるとする。

ここで核関数 $k(\mathbf{X})$ として、次のような形の関数と仮定する。

$$k\left(\frac{\mathbf{X} - \mathbf{Y}_i}{h_N} \right) = k_1((\mathbf{X} - \mathbf{Y}_i)^T (\mathbf{X} - \mathbf{Y}_i)) \quad (4.7)$$

正規分布などが、この関数の例として挙げられる。この時、式(4.6)は次式(4.8)となる。

$$g^{\text{rad}} \mathbf{X} k\left(\frac{\mathbf{X}-\mathbf{Y}_i}{h_N}\right) = -2 \cdot k'((\mathbf{X}-\mathbf{Y}_i)^T (\mathbf{X}-\mathbf{Y}_i)) \times (\mathbf{Y}_i - \mathbf{X}) \quad (4.8)$$

また、

$$k\left(\frac{\mathbf{X}-\mathbf{Y}_i}{h_N}\right) = k_2(|\mathbf{X}-\mathbf{Y}_i|) \quad (4.9)$$

と仮定すれば、次式となる。

$$g^{\text{rad}} \mathbf{X} k\left(\frac{\mathbf{X}-\mathbf{Y}_i}{h_N}\right) = -k_2'(|\mathbf{X}-\mathbf{Y}_i|) \times \frac{(\mathbf{Y}_i - \mathbf{X})}{|\mathbf{Y}_i - \mathbf{X}|} \quad (4.10)$$

したがって、式(4.6)は、 $k(\mathbf{X})$ が式(4.7)または式(4.9)を満足するような関数である場合には、次式のように書き直すことができる。

$$g^{\text{rad}} \mathbf{X} \hat{P}_N(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^N \frac{1}{h_N} K(\mathbf{X}-\mathbf{Y}_i) \mathbf{e}(\mathbf{y}_i - \mathbf{x}) \quad (4.11)$$

ただし、 $\mathbf{e}(\mathbf{y}_i - \mathbf{x}) = (\mathbf{Y}_i - \mathbf{X}) / |\mathbf{Y}_i - \mathbf{X}|$ である。

以上のことより、擬似勾配を次のように定義する。

[定義 4.1]

サンプル点 \mathbf{X} での擬似勾配 \mathbf{V}_x を次式で定義する。

$$\mathbf{V}_x = \frac{1}{m} \sum_{\mathbf{y}_i \in N_{xr}} f(d) \cdot \mathbf{e}(\mathbf{y}_i - \mathbf{x}) \quad (4.12)$$

ただし、 $N_{xr} = \{ \mathbf{Y} \mid |\mathbf{X}-\mathbf{Y}| \leq r \}$ で点 \mathbf{X} での r -近傍

r ; 任意定数

m ; 集合 N_{xr} の要素数

$\mathbf{e}(\mathbf{y} - \mathbf{x})$; 点 \mathbf{X} から 点 \mathbf{Y} の方向く単位ベクトル

$f(d)$; \mathbf{X}, \mathbf{Y} 間のユークリッド距離 d の関数

である。

ここで定義した擬似勾配は、点 \mathbf{X} と他の点 \mathbf{Y}_i との距離 d の適当な関数 $f(d)$ を用いることにより、各サンプル点における擬似的な勾配を与える。この擬似勾配に求められる性質としては、必ずしも分布の正確な勾配ではなく、各サンプル点近傍における分布の状況、すなわち、各サンプル点の属するクラスタの中心方向と、どのクラスタに属するかを表わす属性の度合を表わすことが望まれる。また核関数 $k(\mathbf{X})$ の多くは、 $|\mathbf{X} - \mathbf{Y}_i| \rightarrow \infty$ に対し、漸近的に 0 に近づく。したがって、式(4.8), 式(4.10)における $-k_1'(\cdot), -k_2'(\cdot)$ は $|\mathbf{X} - \mathbf{Y}_i|$ が大きくなるに従い、正なる減少関数となる。したがって、擬似勾配における関数 $f(d)$ は、 d に対し、正でかつ減少関数であることが望ましい。 $f(d)$ の比較的簡単な例を、次に挙げる。

$$(i) \quad f(d) = \frac{1}{a+d} \quad (4.13)$$

$$(ii) \quad f(d) = a \left(1 - \frac{d}{r}\right) \quad (4.14)$$

ただし、 a, r は任意定数である。

上で述べた事柄を整理し、擬似勾配の特色について、以下に述べる。

- (i) サンプル数が有限の場合、局所的な r -近傍内のサンプル集合を用いて、勾配らしき値を与えている。
- (ii) 計算が容易である。得られたサンプル集合全体を考慮せずに、 r -近傍内のサンプル集合で勾配を求めるため、サンプル数が多く、また、サンプル集合全体のサイズが大きくなつたとしても、部分サンプル集合で擬似勾配値が算出できる。
- (iii) 短所として、次のことが挙げられる。サンプル点が無限に得られる極限状態において、式(4.6)の推定勾配は真の勾配値に収束する。しかし、擬似勾配は真の勾配値と一致することが保証されない。

4.3.2 クラスタ分割手法

本節では、擬似勾配を用いてクラウドを定義し、クラウドに基づきクラスタ個数の決定方法とクラスタリングの方法について述べる。

[A] クラスタ個数の決定方法

[定義 4.2]

クラウド C を、擬似勾配を用いて次のように定義する。

$$C = \{ \mathbf{X} \mid |\nabla_{\mathbf{X}}| < R_0 \} \quad (4.15)$$

ただし、 $\nabla_{\mathbf{X}}$; \mathbf{X} における擬似勾配

R_0 ; 任意定数 である。

定義されたクラウドに属する点は絶対値の小さな擬似勾配をもつ。換言すれば、クラウドに属するサンプル点の近傍において、密度関数の勾配は小さく、クラウドは密度関数の極大点および極小点を包含する集合を示している。したがって、クラウド C はクラスタの中心付近およびクラスタ間の境界付近を示している。

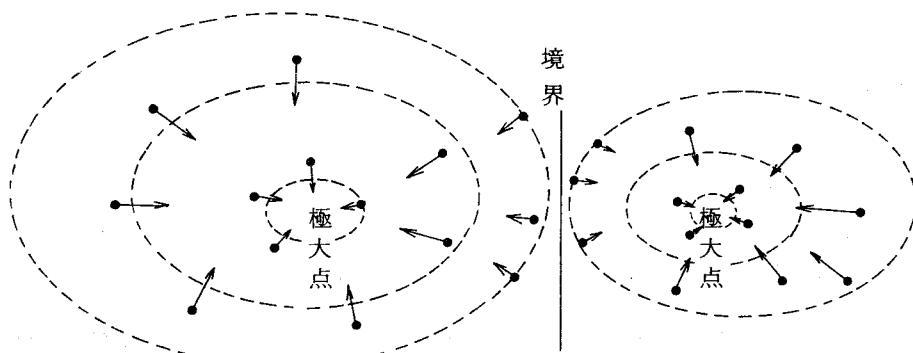


図 4.1 擬似勾配の方向

次にクラスタの中心付近のクラウドとクラスタ間の境界付近のクラウドとの分離方法について述べる。図 4.1 に示すように、密間関数の極大値付近のサンプル点 \mathbf{X} の近傍には、点 \mathbf{X} の方向を向く擬似勾配を持つサンプル点が多く存在していると考えられる。逆に、極小点付近のサンプル点の近傍には、擬似勾配がその点より遠ざかる方向を向くような点が多く存在していると考えられる。以上のことより、分布の峰および谷を表わすクラウドを次のように定義する。

[定義 4.3]

クラウド C に属する点 \mathbf{X} に対して、次の W_x を定義する。

$$W_x = \sum_{Y \in N_{xr}} V_Y^T \cdot e (\mathbf{y} - \mathbf{x}) \quad (4.16)$$

ただし、 $V^T \cdot e$ はベクトルの内積である。

式 (4.16) で定義された W_x を用いて、峰クラウド C_p を次のように定義する。

$$C_p = \{ \mathbf{X} \mid W_x \leq 0, \mathbf{X} \in C \} \quad (4.17)$$

[定義 4.4]

谷クラウド C_v を次のように定義する。

$$C_v = \{ \mathbf{X} \mid W_x > 0, \mathbf{X} \in C \} \quad (4.18)$$

定義 4.3 から、峰クラウド C_p は密度関数の極大点および鞍点を包含したサンプル集合を示し、峰クラウドの個数がクラスタ数に一致するものと思われる。また定義 4.4 より、谷クラウド C_v は密度関数の極小点を包含した集合を示すことより、クラスタ間の境界付近を示すものと考えられる。

上に述べた事柄を使用してクラスタ個数の決定方法について述べる。

〈クラスタ個数の決定手順〉

Step 1. 与えられたすべてのサンプル点における擬似勾配 \mathbf{V}_x を求める。

Step 2. $|\mathbf{V}_x| < R_0$ となる点の集合をクラウド C とする。

Step 3. クラウド C に属するサンプル点のうち、式(4.16)～(4.18)を使用して谷クラウドに属するサンプル点を除去し、峰クラウドに属するサンプル点のみを残す。

Step 4. 峰クラウド C_p に属するサンプル点の集合に対して、Minimal Spanning Tree (M.S.T.) を構成する。ただし、各点間の重みとして、サンプル点間の距離を用いる。

Step 5. M.S.T. の各枝 $e = \langle$

$a, b \rangle$ に対して、端点

a に隣接する節点 a_1, a_2

につながる枝の重み $w_{a1},$

w_{a2}, \dots, w_{ar} の平均

w_{aM} を求め、枝 e の重み

w_e が $k w_{aM}$ より大きい

時、枝 e を切断する。端点

b についても同じ操作を行

う。

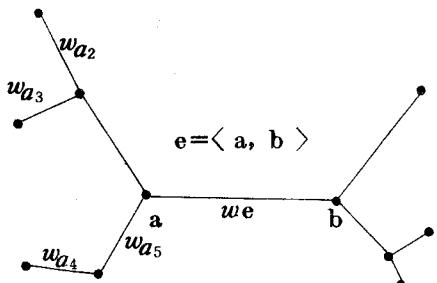


図 4.2 枝の切断

Step 6. Step 5 で生じた Sub-Tree (峰クラウドの部分集合) の個数をクラスタ数と決定する。一つの Sub-Tree を構成しているサンプル点には、同一のクラスタ名を付ける。

この方法の利点は、与えられたサンプル点に対して M.S.T. 法を直接適用するのではなく、各クラスタの中心付近を表わしていると思われる峰クラウドに

対し、M.S.T. 法を適用することにより、分離性の良い M.S.T. が構成できることにある。

[B] クラスタ分割

各サンプル点における擬似勾配が、その点の属するクラスタの中心方向およびその属性の強さを示すことより、[A] で得られたクラスタの中心（峰クラウド）を用いて、各サンプル点の属するクラスタの決定方法を述べ、図 4.3 にそのフローチャートを示す。

〈クラスタリングの手順〉

Step 1. サンプル点の集合のうち、属するクラスタの確定している点の集合を D とする。D の初期状態は [A] で決定された峰クラウドとする。

Step 2. あるサンプル点 \mathbf{X} の r -近傍内に、集合 D に属する点 \mathbf{x}_i が存在する時、次の値 q_i を計算する。近傍内に集合 D に属する点が存在しない時にはサンプル点 \mathbf{X} のクラスタの決定は保留する。

$$q_i = g(d) \{ \mathbf{V}_{\mathbf{x}}^T \cdot \mathbf{e} (\mathbf{x}_i - \mathbf{x}) \} \quad (4.19)$$

ただし、 $g(d)$ は点 \mathbf{X} 、 \mathbf{x}_i 間の距離 d に関する減少関数、
 $\mathbf{V}^T \cdot \mathbf{e}$ はベクトルの内積 である。

Step 3. 式 (4.19) より求めた q_i に対し、 $q_i > 0$ なる点 \mathbf{x}_i が存在した時、

$$q_j = \max_{\mathbf{x}_i \in N_{\mathbf{x}r}} q_i \quad (4.20)$$

を満足する点 \mathbf{x}_j を求め、サンプル点 \mathbf{X} を \mathbf{x}_j の属するクラスタ

に属させる。その他の場合はサンプル点 \mathbf{X} のクラスタの決定を保留する。

Step 4. Step 2 ~ Step 3 を繰返し実行し、すべてのサンプル点の属するクラスタが決定した時、クラスタリングは終了する。

図 4.4 の例を用いて、このクラスタリングのアルゴリズムを簡単に説明する。破線は分布の密度の等しいサンプル点を結んだものであるとする。破線で示したような分布をしたサンプル点のうち、点 $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ を例にとる。すべての点の擬似勾配を計算し、 $|\mathbf{v}_x| < R_0$ を満足する点がクラウドとなる。 $\mathbf{X}_1, \mathbf{X}_3$ がこのクラウドの集合に属する。次に式(4.16)の W_x を計算する。その結果、 \mathbf{X}_1 は $W_{x_1} < 0$ 、 \mathbf{X}_3 は $W_{x_3} > 0$ となり、 \mathbf{X}_3 は谷クラウドと決定する。

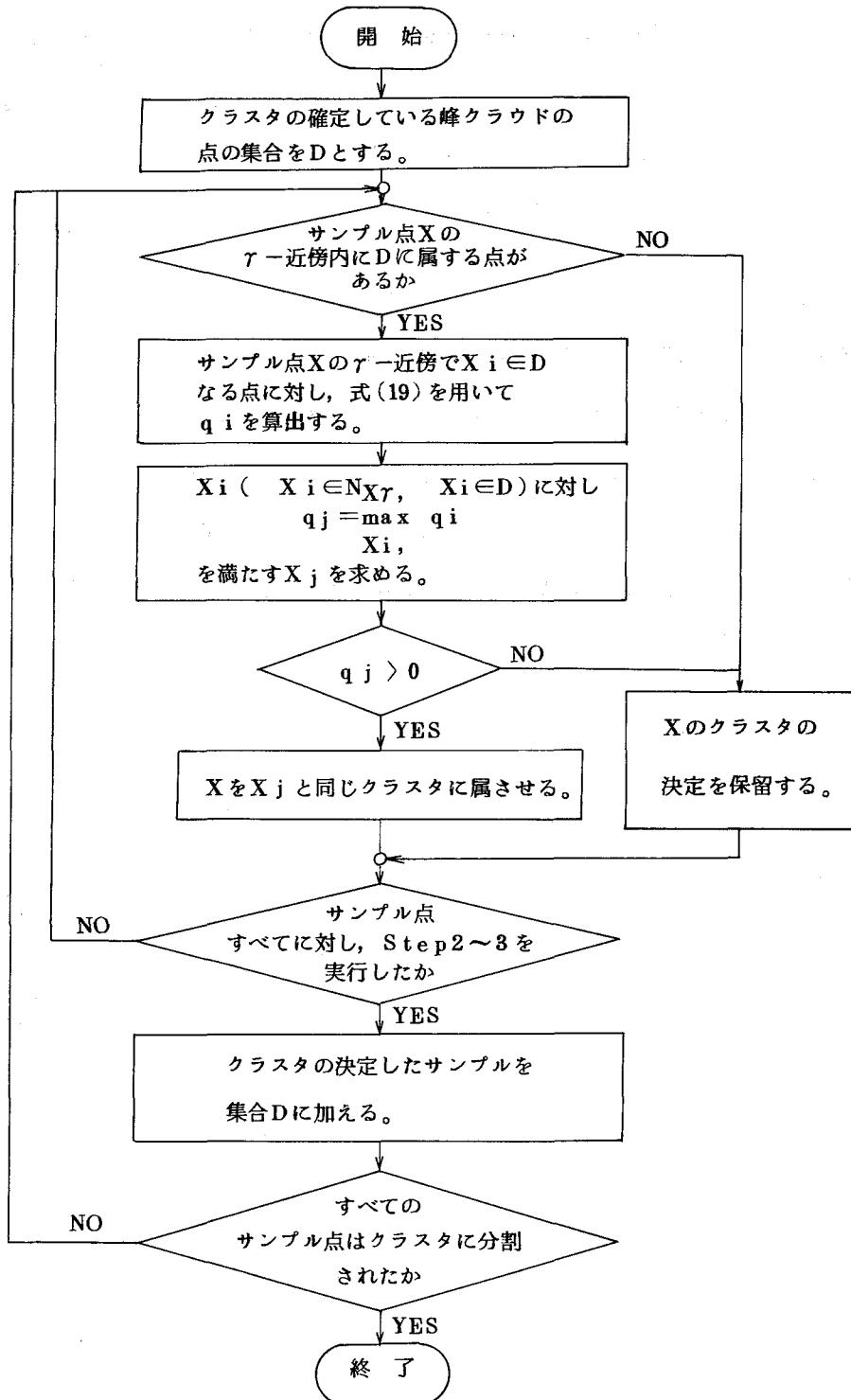


図 4.3 クラスタリングの流れ図

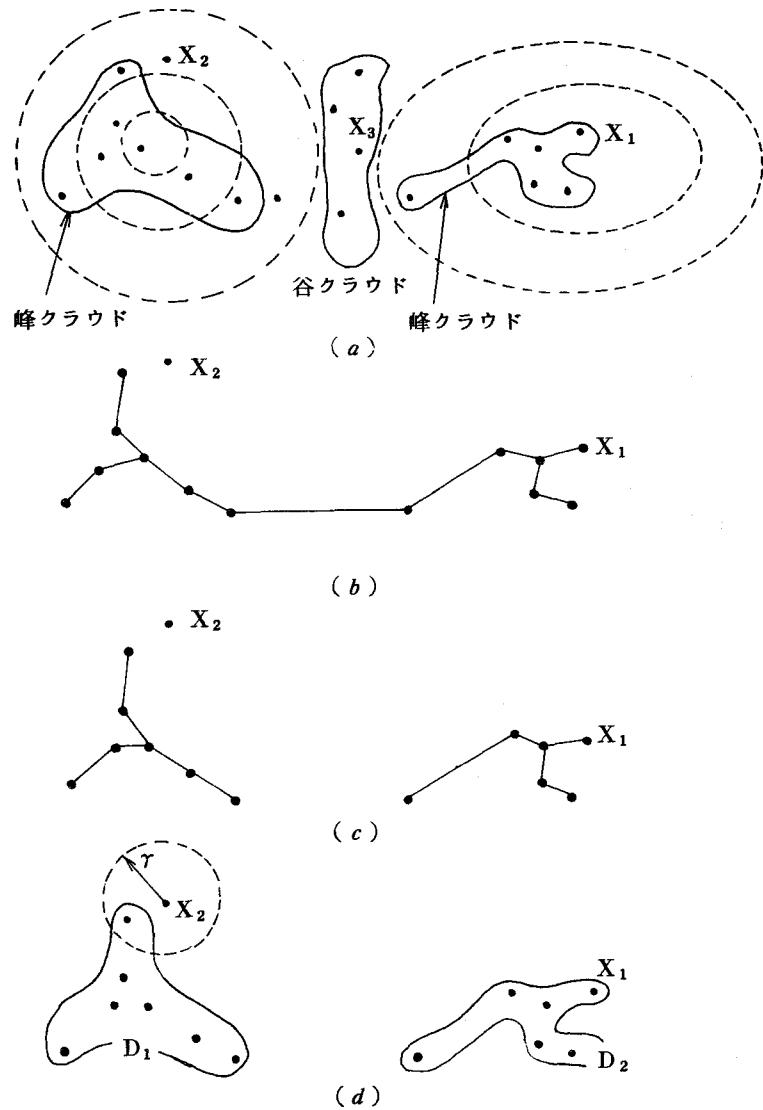


図 4.4 クラスタリングの各過程

以上の操作で残った峰クラウドの要素を用い、M.S.T. を構成しクラスタ分割を行う。その結果、クラスタ個数は 2 個となる。この後、元の集合に対し、クラスタ分割を行う。最初、峰クラウドに属する集合を集合 $D = (D_1, D_2)$ とする。（ただし、 D_1 はクラス 1 に属するクラスタで、初期状態はクラス 1 の峰クラウドとする。 D_2 も同様である。）そして、まだクラスが未定のすべての点に対しクラスタ分割を実行する。たとえば点 \mathbf{X}_2 において、その r -近傍内には、 $q_i > 0$ を満足する集合 D_1 の要素 \mathbf{X}_i が存在し、 $\mathbf{X}_j = \max \mathbf{X}_i$ を満足する \mathbf{X}_j のラベル名クラス 1 を \mathbf{X}_2 に付け、 \mathbf{X}_2 をクラス 1 に属すると決定し、この \mathbf{X}_2 を集合 D の要素に加える。上の操作を順次繰り返し、クラス名が未定のサンプルがなくなるまで行う。

4.3.3 計算機実験とその検討

提案した擬似勾配と、それを用いたクラスタリングの手法を評価するため、二次元人工データを用いて計算機による実験を行った。ここで、関数 $f(d)$ や $g(d)$ は、次のような減少関数とした。

$$f(d) = g(d) = \frac{1}{a+d}$$

ただし、 d は点間距離、パラメータ $a = 0.5$ とする。また、 $R_0 = 0.17$ とし、峰クラウドの分割方法として、峰クラウドに属する各点に対し構成した M.S.T. の各枝の重みがその枝に近接する枝の重みの平均の 3.0 倍以上の時、枝を切断するとした。

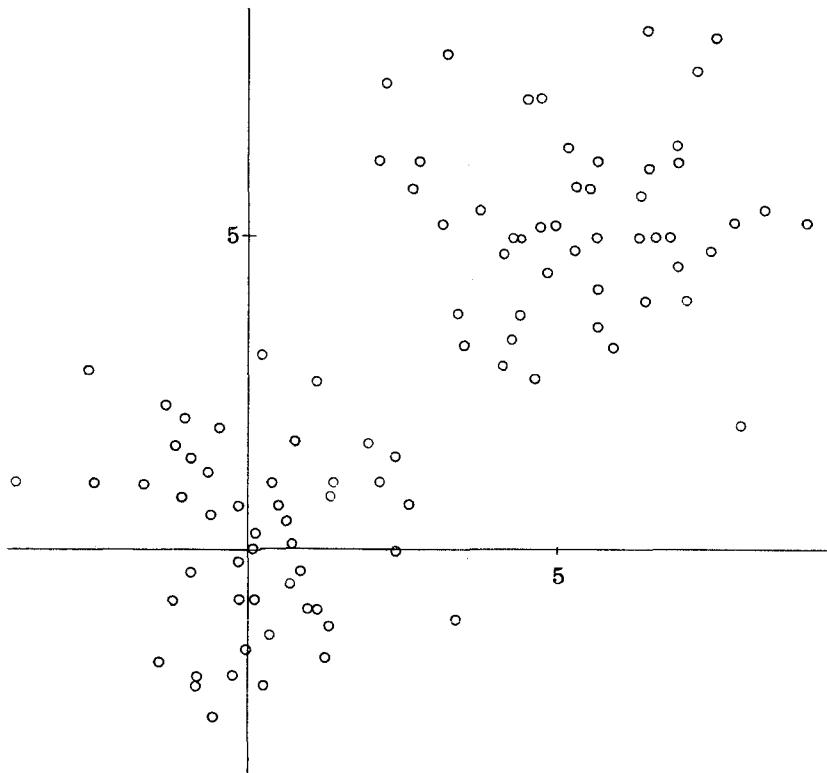
〈実験 1〉

サンプル集合として、次のような平均 μ_1, μ_2 および共分散行列 Σ_1, Σ_2 をもつ 2 つの正規分布から発生させた、合計 100 個のサンプル点を用いた。

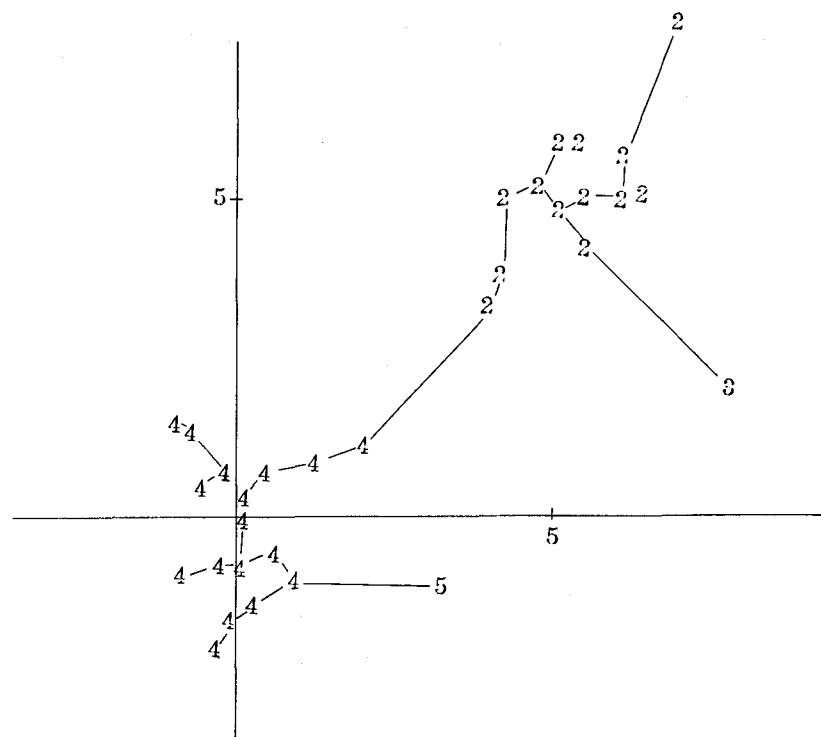
$$\mu_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \mu_2 = \begin{pmatrix} 5 \\ 5 \end{pmatrix} \quad (4.21)$$

$$\Sigma_1 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

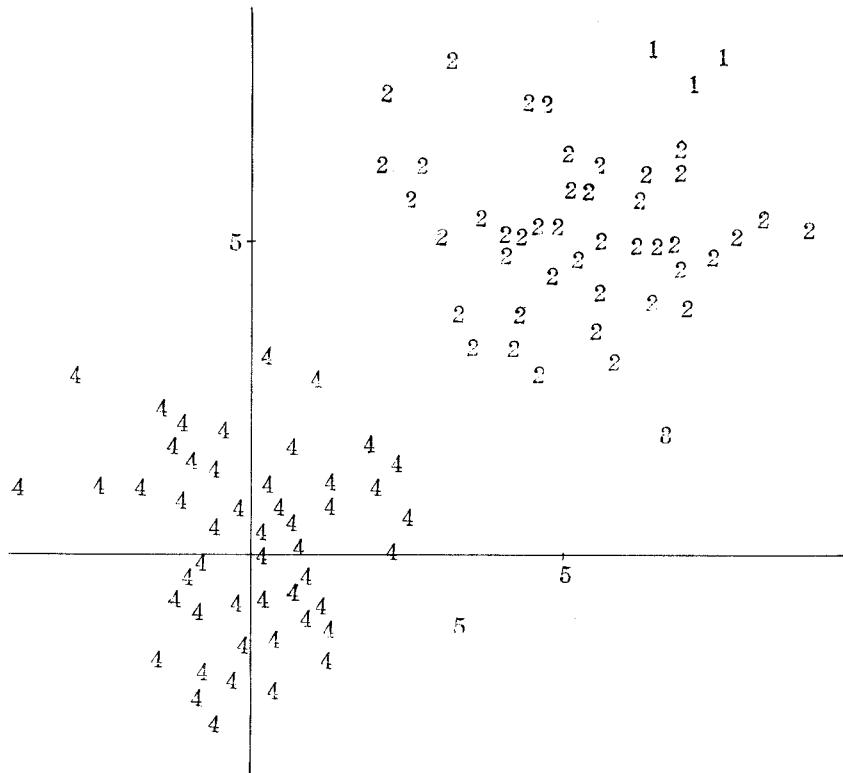
与えられたサンプル集合と、峰クラウドおよびクラスタリングの結果を図4.5(a)～(c)に示す。同一数字は一つのクラスタを示す。



(a) 2次元正規分布データ



(b) 峰クラウドとそのMinimal Spanning Tree



(c) クラスタ分割の結果

図 4.5 2次元正規分布データとそのクラスタ分割の結果

図 4.5 (b)より、峰クラウドは、ほぼクラスタの分布の中心部を示し、図 4.5 (c)より人間の視覚とほぼ一致するクラスタが得られている。ただ、要素数の少ないクラスタ、クラスタ 1、クラスタ 3、クラスタ 5、が存在している。しかし、これらを特異なクラスタとして処理するか、もしくは、これら要素の少ないクラスタに対しては、最も近いクラスタに帰属させるか、という形での処理を考えられる。

〈実験 2〉

図 4.6 に示すような粗な分布と密な分布よりなるサンプル点を人工的に与え、クラスタリングの実験を行った。その結果、峰クラウドおよびクラスタリングの結果は図 4.6 (b) となった。

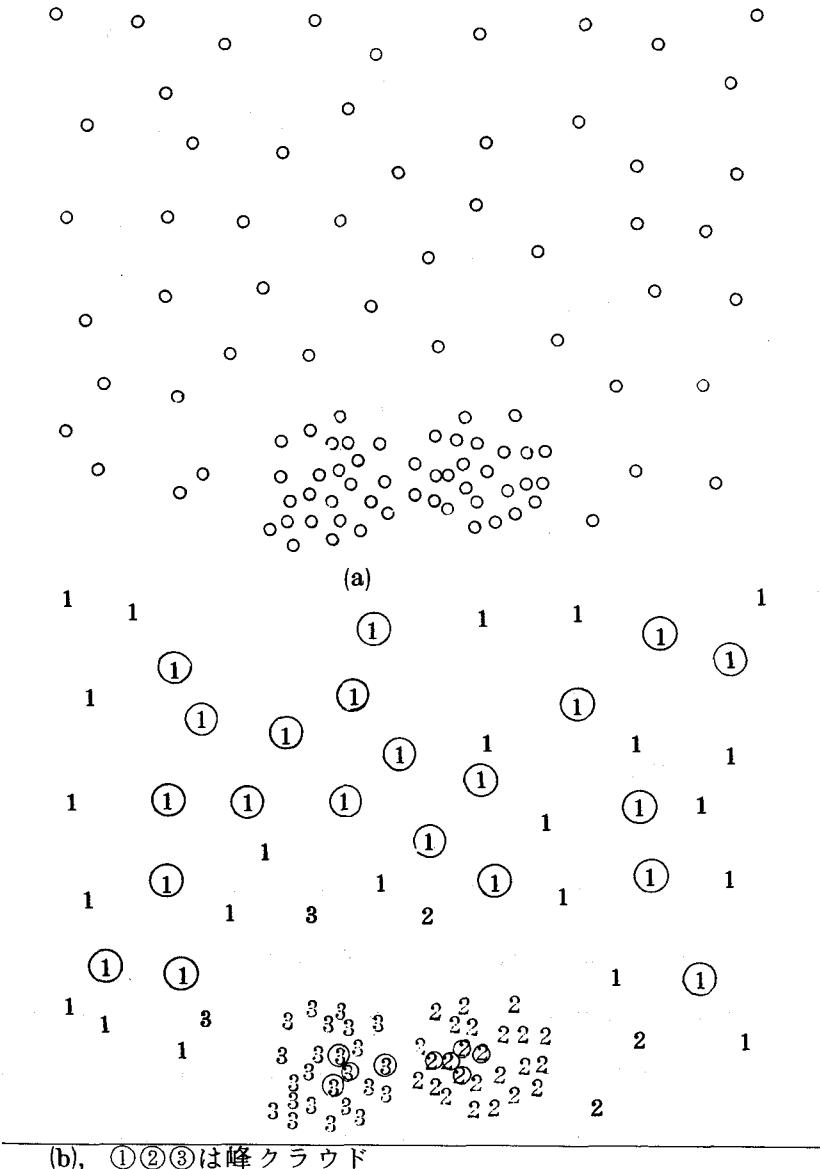
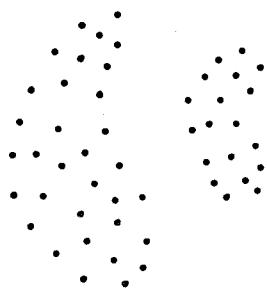
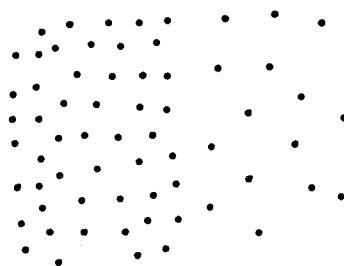


図 4.6 雜音中のクラスタ
(a) 元のデータ (b) クラスタ分割結果

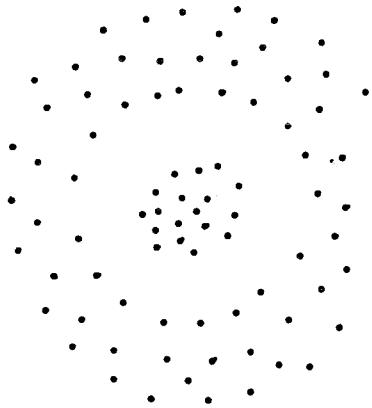
図 4.6 (b) で示されているように、峰クラウドは、粗、密なクラスタの中心部を示し、境界に位置する点を除き、人間の視覚とほぼ一致するクラスタが得られている。



(a)

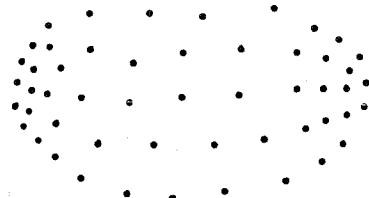


(a)



(b)

図 4.7 分離クラスタ



(b)

図 4.8 接触クラスタ

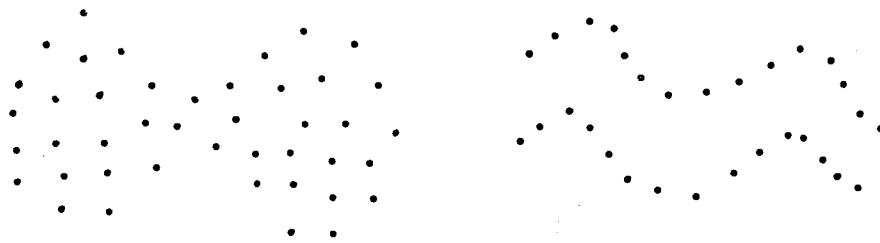


図 4.9 一様分布からなるクラスタ

図 4.10 線状クラスタ

他の 2 次元人工データにも適用した結果、図 4.7 のような分離クラスタ、図 4.8 のような粗な分布と密な分布からなる接触クラスタに対しては有効な結果が得られた。しかし図 4.9 のような一様な分布からなるクラスタに対しては、このアルゴリズムは有効に働くかない。この原因は、元のサンプル集合より峰クラウドを抽出した後、M.S.T. 法を使用したことにある。すなわち、M.S.T. 法では一様な分布に対して、辺の分離判定ができないためである。（鎖効果と呼ばれる）。この欠点は、峰クラウドに他のクラスタリング法を適用すれば解消するものと思われる。ただし、図 4.10 のような曲率の大きな部分を持つ線状クラスタに対しては、正しくクラスタ分割が行えなかった。これは曲率の大きな個所のサンプル点 X において V_x の値が $|V_x| < R_0$ を満たすべきであるのだが、2 方向のベクトル和となり満足できず、結局峰クラウドの集合の中に、曲率の大きな地点のサンプル点が残らなかったためである、と考えられる。

次に、 a 、 r の値の選択および V_x の r に対する安定性について考察する。 a の値の選択に関して、図 4.11 を参照として、定性的に次のようなことがいえる。 $0 < a \ll 1$ のとき、 $f(d) = 1/(a+d)$ の形より $f(d)$ は近似的に $f(d) \approx 1/d$ となり、 $d \approx 0$ のとき、 $f(d)$ の値が非常に大きな値をもち、

勾配値を算出するのに不適当となる。 $a \gg 1$ のとき、 $f(d) = 1/(a + d)$ の形より、 d の値が小であるサンプル点に比較し、 d の値が大であるサンプル点も、勾配 ∇_x に対し

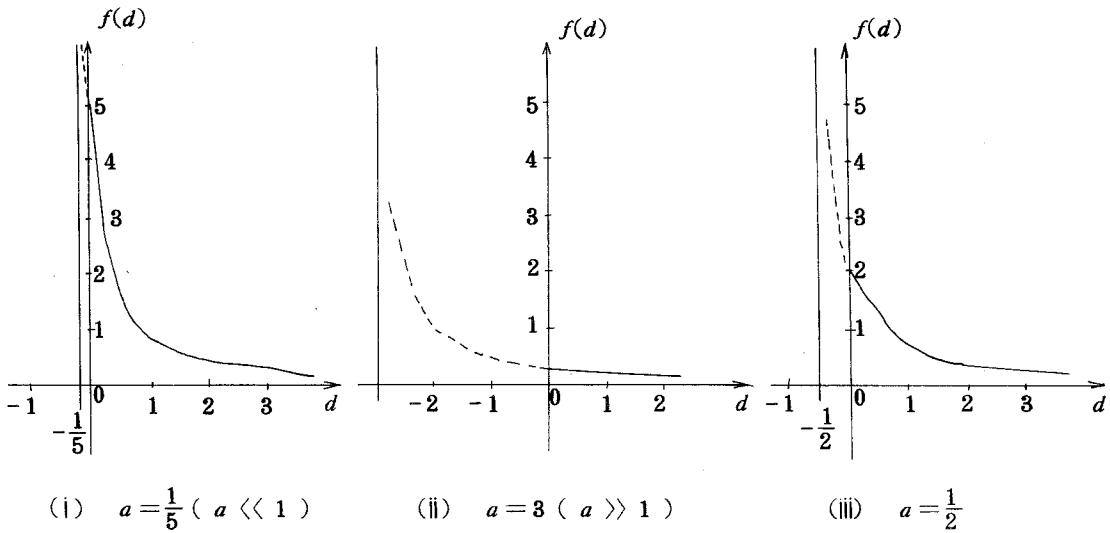


図 4.11 パラメータ a の値の選択方法

比較的大きな影響を及ぼす。 r —近傍内のサンプル数が多い時には問題はないがサンプル数が少ない場合、 r —近傍付近（境界）のサンプルの有無により、 ∇_x の方向、大きさが大きく変動し、 ∇_x の値が不安定となる。以上のことより、 a の値としては $\frac{1}{2} \sim \frac{3}{2}$ ぐらいが望ましいと考え、 $a = \frac{1}{2}$ として計算機実験を行った。 r の値の撰択に関して、次のことがいえる。 ∇_x の計算を可能にするため、 r はサンプル点の r —近傍内に別のサンプル点が少くとも 1 つ存在する値でよい。また、下に凸な減少関数である $f(d)$ の形から判断できるように ∇_x は r —近傍内の近くのサンプル点の影響を大きく受けるため、サンプ

ル点**X**より離れた地点でのサンプル数が多少変動しても、 \mathbf{V}_x の値には左程大きな影響を及ぼさない。すなわち、 r が少々変動し、 r -近傍内のサンプル数が多少増減するとしても、その増減分はサンプル点**X**より離れた地点で増減するだけであり、 \mathbf{V}_x の r に対する安定性は良いと考えられる。

以上の結果より、本方式によるクラスタリング・アルゴリズムの有効性について検討する。元のデータとして多数のサンプル点からなる集合が与えられた時、元のデータ自体にクラスタリングの手法を使用することは、記憶容量および計算時間に問題が生じてくるものと思われる。よって必然的にクラスタリングの対象サンプル点を絞る必要が生じ、本方式では擬似勾配を用いて峰クラウドを抽出し、クラスタリングの対象サンプル数を減らした。通常、ユークリッド空間を構成しているサンプル空間において、点**X**における密度関数の勾配を求めるには、その点の近傍密度とその周辺の点の近傍密度を比較して求めなければならない。しかし、クラスタリングのための対象サンプル数を減らすために、必ずしも厳密な意味をもつ勾配は必要ではなく、点**X**の近傍点とその間の距離のみを用い、その点での計算が容易な擬似勾配を求めることにより、適切なクラスタリングが可能であることが明確となった。さらに、サンプル数が多くなるほど、擬似勾配はより良く密度関数の真の勾配を反映したものとなり、したがって、峰クラウドが密度関数の極大点、鞍点をより正確に示し、クラスタ分割はより適切なものとなる。

また記憶容量に関しては、すべての点間距離を記憶する階層的手法などが、サンプル個数の2乗に比例した容量が必要であるのに対し、提案した手法は、各点の r -近傍内の点および峰クラウドに属する限られた点間の距離を記憶するだけで、サンプル個数に比例した容量のみが必要で、この意味でも、多くのサンプル点が与えられた場合、この手法は有効である。

次に多次元の問題について考察する。擬似勾配を単純なベクトルの和で算出

し、峰クラウドをベクトルの内積で抽出して、M.S.T. を構成するため、多次元クラスタ分割への拡張も容易である。

4.4 結 言

本章では、ノンパラメトリックな手法によるクラスタリングの手法を提案し、計算機実験によりその有効性を確認した。サンプル点Xにおいて、その γ -近傍内のサンプル点およびそれらの間の距離を用いて、サンプル点Xでの擬似勾配を定義し、それを用いてデータ集合から峰クラウドなる集合を抽出し、峰クラウドの集合に対しM.S.T. を構成し、枝を切断することでクラスタを検出するアルゴリズムを構成した。この手法は、サンプル空間における各点の分布の勾配を考慮したものであり、曲率の大きい線状クラスタからなる分布、一様な分布などの一部の分布を除き、種々な分布をしたデータ集合に対し適用可能であることが明らかとなった。また、この手法は、多サンプルからなるデータ集合に対し、擬似勾配が真の勾配をよりよく反映することと共に、計算時間、記憶容量の面から有効な手法であると考えられる。

ここでは、峰クラウドのクラスタ分割にM.S.T. を構成し行っているため、一様な分布に対しクラスタ分割ができないなどのM.S.T. の手法の欠点が表わされている。今後、峰クラウドを適切に分割する手法について考察する必要がある。

第 5 章 結論

第 2 章では、誤り制御機能を用いた連想方式を提案した。連想機能を考察する時、記憶装置、変換機能の 2 つの観点があり、ここでは連想の一つの変換方法について考察した。パターンを量子化すると、パターンは符号を構成しているとみなせる。よって、この方式では記憶装置内には巡回符号を生成する特性多項式を用いて、巡回符号を構成するように生成された検査パターンを、標準パターンの代りに記憶する。このことは検査パターンは標準パターンの代数的構造をとらえていることとなる。また、想起方法は検査パターンを参照として、部分的に欠損または歪んでいるパターンを復号することにより、元の標準パターンを得ている。記憶方法について次の 2 種類を提案した。一つはパターン全体に対応して検査パターンを生成し、記憶するパターン・チェック法である。しかし、文字の欠損部分が字“1”の要素の集中している行に集まつた場合を考えると、パターン・チェック法は訂正能力の高い符号を用いることが望まれる。この欠点を除去するため、パターンの各画素をランダム順序入替して得たパターンの各行ごとに検査行を生成し、検査行からなる検査パターンを記憶するライン・チェック法を考案した。ここでは、文字の構造を表わしている特徴を抽出しないで連想認識を行っているので、今後認識対象の構造をも考慮に入れた誤り制御機能による連想方式が望まれる。

第 3 章では、パターン集合を覆う被覆族間において、細分関係をもつ連想構造について考察し、被覆の具体的構成として強制学習法と Prime Event 法を用い、文字識別実験を行った。ここで、高品質手書き文字を対象とした認識では、文字の構造を考慮せずに本方式が有効であることを強調した。パターン集合に対して連想被覆を定義し、細分関係にある連想被覆族間の連想構造に

について考察した。そして有限個の観測値で表現されるパターン集合の連想構造は抽象複体さらに単体的複体を構成し、細分関係にある単体的複体間には単体写像が存在することを指摘した。また、細分化は位相の強化となり、学習過程の一つであることを明らかにした。今後、被覆を用いて連想構造をより位相的見地から論じ、複雑な手書き文字などの認識対象を識別する方法について考察する必要がある。

第4章では、ノンパラメトリックな手法によるクラスタリングの手法を提案し、計算機実験によりその有効性を確認した。高次元かつ多数のサンプルが与えられた場合、クラスタ分割にはサンプル数の平方に比例した多くの記憶量や計算時間を必要とするが、提案した手法では各点の r -近傍内に含まれるサンプル点および峰クラウドに属する限られた点間の距離を記憶するため、サンプル数に比例した記憶容量で良いことが明らかとなった。また、擬似勾配が各点の r -近傍内に含まれる点のみを用い、単純な計算より求められることから、比較的少ない計算時間に抑えられる。したがって、ここで提案した手法は、高次元かつ多数のサンプルが与えられた場合、有効であることが認められた。この手法を用いて計算機シミュレーションした結果、一部の分布を除き、種々の分布をしたデータ集合に対し適用可能であることが明らかとなった。残された課題として、M.S.T.法を改善しあらゆる分布をもつデータ集合をクラスタ分割できる手法、クラスタ分割結果の妥当性の評価法、データ表現のためのクラスタの記述方法について考察を加えなければならない。

以上、本研究で得られた成果が、図形、文字等のパターン認識装置の開発の一助となれば、筆者の最も幸いとするところである。

謝 辞

本研究の全過程を通じ、直接懇切なる御指導、御鞭撻を賜わった手塚慶一教授に心よりお礼申し上げる。

大学院前期および後期課程において、御指導、御教示賜わった笠原芳郎名誉教授、青柳健次名誉教授、加藤金正名誉教授、通信工学教室の板倉清保教授、滑川敏彦教授、熊谷信昭教授、中西義郎教授、電子工学教室の尾崎弘教授、産業科学研究所の角所収教授に対し厚くお礼申し上げる。

筆者の属する手塚研究室の真田英彦助教授、中西暉助手、打浪清一助手、奈良真彦技官をはじめ、研究室の諸氏には種々の面でお世話になった。また、福井大学工学部講師の小川秀夫氏、本学研究生の森井藤樹博士、塩野充氏、服部哲郎氏、大学院学生の蔭山恵氏、学部学生の荒木吉雄氏および本学卒業生の窪田伸一氏、塙本旨氏、安田敏彦氏には熱心な御討論をいただいた。

ここに記して、以上の方々に深く感謝の意を表する。

文 献

- (1) H.L.Dreyfus：“鍊金術と人工知能”，“情報の科学Ⅱ”（視聴覚情報研究会編，ラティス社）
- (2) W.S.McCulloch, W.Pitts：“A Logical Calculus of the Ideas Immanent in Nervous Activity ” Bulletin of Mathematical Biophysics, Vol. 5, P.115 (1943)
- (3) E.R. Caianiello：“Outline of a Theory of Thought Process and Thinking Machines ”, Journal of Theoretical Biology, Vol. 1, P.204 (1961)
- (4) F.Rosenblatt：“The Perceptron, a Theory of Statistical Separability in Cognitive Systems ”, Report VG-1196-G-1, Cornell Aeronautical Laboratory, Buffalo (1958)
- (5) H.D.Block, B.W.Knight, F.Rosenblatt：“Analysis of a Four-Layer Series-Coupled Perceptron.2 ”, Review of Modern Physics, Vol.34, No.1, p.135 (Jan., 1962)
- (6) H.D.Block：“The Perceptron:A Model for Brain Functioning ”, Reviews of Modern Physics, Vol.34, No.1, P.123 (Jan., 1962)
- (7) P.B.Post：“A Life Like Model for Association Relevance ”, Proc. International Joint Conference on Artificial Intelligence, P.271 ~ 280 (1969-05)
- (8) 中野：“アソシアトロンとその応用”，信学会インホメーション研資, IT69-27 (1969-09)
- (9) 中野, 南雲：“神経回路網モデルによる連想記憶の研究”，信学会医用電

子・生体工学研質, MBF 70-6 (1970-06)

- (10) 福島：“自己組織機能を持つ多層回路”，信学会論文誌，Vol.58-D，No.9 P.530～537 (1975-09)
- (11) 辰己，手塚：“誤り制御機能による連想”，情報処理学会 第15回大会 (昭和49)
- (12) 上坂，尾関：“連想形記憶の二，三の性質”，信学会論文誌，Vol.55-D，No.5，P.323～330 (1972-05)
- (13) 磯道：“変換機能の学習”，信学会論文誌，Vol.56-D，No.12，P.70 8～718 (1973)-12)
- (14) 市田，吉田：“ある連想モデルによる手書き文字の認識”，信学会論文誌，Vol.59-D，No.1 P.17～24 (1976-01)
- (15) W.W.Peterson: “Error-Correcting Codes”，MIT and Wiley (1961)
- (16) 宮川，岩垂，今井：“符号理論”，昭晃堂 (昭和48)
- (17) E.Prange: “Cyclic Error-Correcting Codes in Two Symbols”，AFCRC-TN-57-103, Air Force Cambridge Research Center, Cambridge, Mass. (Sept. 1957)
- (18) W.W. Peterson and D.T. Brown: “Cyclic Codes for Error Detection”，Proc. of IRE, 49, P. 228 (1961)
- (19) 辰己，手塚：“被覆による連想の一方式”，信学会 全国大会 (昭和50)
- (20) 辰己，真田，手塚：“被覆を用いた連想記述に関する一考察”，信学会 パターン認識と学習研資 PRL 75-23 (1975-7)
- (21) 辰己，真田，手塚：“有限観測可能のもとでの連想構造に関する考察”，信学会 総合全国大会 (昭和51)
- (22) 辰己，真田，手塚：“被覆による識別の一手法について”，信学会 総合

全国大会（昭和 52）

- (23) Zeemann: "The Topology of the Brain and Visual Perception", M.K.Fort 編, Topolgy of 3-Manifolds and Related Topics, Prentice, (1962)
- (24) 坂井編: "パターン認識の理論", 共立出版株式会社, 情報科学講座 E・19・1 (1967)
- (25) G.H. Ball: "Data Analysis in the Social Science: What about the Details?", Proc. FJCC, P.533 (1965)
- (26) G.Nagy: "State of the Art in Pattern Recognition", Proc. IEEE, 56, P.399 (April 1968)
- (27) C.J.Zahn: "Graph-Theoretical Method for Detecting and Describing Gestalt Clusters", IEEE Trans., C-20, No.1 P.68 (Jan. 1971)
- (28) S.C.Johnson: "Hierarchical Clustering Schemes", Psychometrika, 32, P.241 (Sept. 1967)
- (29) G.H.Ball and D.H.Hall: "ISODATA, a Novel Method of Data Analysis and Pattern Classification", Stanford Research Inst., Menlo Park, Calif., Tech.Rept.(April 1965)
- (30) H.P.Friedman and J.Rubin: "On Some Invariant Criteria for Grouping Data", Amer.Stat.Assoc.J., 62, P.1159 (Dec. 1967)
- (31) K.Fukunaga and W.L.G.Koontz: "A Criterion and an Algorithm for Grouping Data", IEEE Trans., C-19, 10,P.917 (Oct. 1970)
- (32) 坂井, 金出: "繰返しアルゴリズムによる自己適応パターン分類機械につ

- いて,”信学会論文誌, Vol.54-C, No.11, P.979 (1971-11)
- (33) 笠原：“情報理論と通信方式”共立出版株式会社(昭和40)
- (34) J.C.Stoffel：“A Classifier Design Technique for Discrete Variable Pattern Recognition Problems”, IEEE Trans. C-23, P.428 ~ 441 (1974)
- (35) 河田, 三村：“現代数学概説II”, 岩波書店(昭和40)
- (36) 近藤, 高木：“学習過程のトポロジー的考察”, 信学会論文誌, Vol.57-D, No.6, P.388 (1974-06)
- (37) 近藤：“パターン集合の位相構造と情報処理機能”, 信学会論文誌, Vol.57-D, No.9, P.551 (1974-09)
- (38) 河田編：“位相幾何学”, 現代数学演習叢書2 岩波書店(昭和40)
- (39) 近藤：“ ϵ -エントロピー理論に基づくパターン集合の近似理論”, 信学会論文誌D, Vol.56-D, No.9, P.491 (1973-09)
- (40) 永野：“学習のアルゴリズムに関する2,3の考察”, 信学会論文誌, Vol.57-D, No.12, P.661 ~ 667 (1974-12)
- (41) K.S.Fu (editor): “Digital Pattern Recognition”, Springer-Verlag Berlin, Heidelberg, New York (1976)
- (42) B.Everitt: “Cluster Analysis”, Heinemann Education Books Ltd. (1974)
- (43) 辰己, 永田, 真田, 手塚：“擬似勾配を用いたノンパラメトリック クラスタリングの一手法” 信学会論文誌D 昭和53年1月掲載予定
- (44) R.O.Duda and P.E.Hart: “Pattern Classification and Scene Analysis” John Wiley & Sons, (1973)
- (45) R.A.Jarvis and E.A.Patrick: “Clustering Using a Similarity Measure Based on Shared Near Neighbors”, IEEE Trans.,

C - 22, P. 1025 (1973)

- (46) I. Gitman and M.D.Levine : " An Algorithm for Detecting Unimodal Fuzzy Sets and Its Application as a Clustering Technique ", IEEE Trans., C-19, P. 583 (1970)
- (47) I. Gitman : " An Algorithm for Nonsupervised Pattern Classification ", IEEE Trans., SMC-3, P. 66 (1973)
- (48) K. Fukunaga and L. D. Hostetler : " The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition ", IEEE Trans., IT-21, P. 32 (1975)
- (49) W. L. G. Koontz, P. Narendra and K. Fukunaga : " A Graph-Theoretic Approach to Nonparametric Cluster Analysis ", IEEE Trans., C-25, P. 936 (1976)