このように、対象とするパターン情報（文字情報）が多次元の数値データへと変換されればパターン認識の実行は可能となり、後に述べる様々な解析を行う事が出来る。
パターン認識による解析では解析対象とするものを数値データ（多次元）に変換する過程も重要な問題となる。　変換された数値データには、目的とする解析を成功させるに必要充分な情報を含んだものである事が必要であり、意味のないデータ変換は実行されるパターン認識手法がいかに強力なものであっても失敗におわってしまう。

## 9．画像認識から化学の世界へ！！！

□　パターン認識の化学への展開時に必要な情報と出力結果

パターン認識を化学の分野に適用するには、解析目的事項と解析に利用される入力データ関の相関や因果関係を吟味する事が必要である。　解析目的を最も的確に説明する情報を含むデータを見出す事が、パターン認識による解析を成功させるポイントである。
パターン認識では先にも述べたように利用データと結果だけわかっていれば、その間に存在する何らかのルール（ブラックボックスとみなされる）を知らなくても解析可能である。　これがパターン認識の本質であり、ルールや式等で厳密に表現する事の出来ない世界を解析する最後の手段でもある。　化学分野への応用においてもこの本質はそのまま維持される。
図４にはこの関係を示し、同時にこの関係において利用される入力データの種類／形態及び、その入力データを用いた時考えられる出力結果の形態を記してある。　ここで示されている入力データと出力結果との関係は化学と限らないすべての分野において利用される基本的なものである。

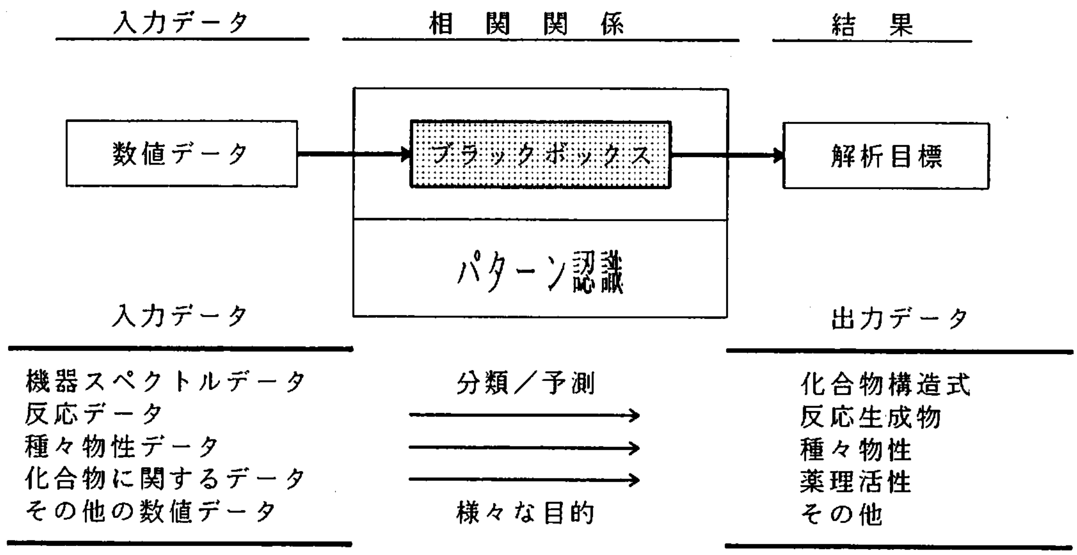



図４．化学関連分野におけるパターン認識で利用される数値データと期待される解析結果

□　解析に用いるデータについて

パターン認識による解析を行う時、解析用データを集めるにあたり注意しなければならない事は以下の２点である。
①　自分が解決しようとする問題に対し、最適な数値データを利用する
②　数値データを多数揃える事（事例を可能な限り多く集める）
最初の項目は具体的に言うならば、利用する数値データ中に解析の最終目的を達成するのに必要となる情報を多く含んでいる事が必要であるという事である。　入力データと解析目的とに何の関係も無い時は、例え結果が出たとしても意味の無い／信頼性の低い結果しか得られない。　例えば、分子量を求めるのにＩＲスペクトルを入力データとして用いても結果は期待出来ないし、その結果は何の意味も持たない無意味なものである。
このように、入力データとして用いる数値データの情報と解析結果をつなぐ何らかの関係は、パターン認識による解析結果を意味あるものとし、結果の信頼性を高めるという点で重要な要素である。　前記の例は極端なものでデータと解析目的との相関はすぐわかるが、実際の解析ではパラメータの種類が多く、数値データへの変換過程と解析目的との関係が明確でない時が多く、ノイズデータと重要なデータとの識別が困難である等の様々な要因から、最適数値データを見出す事は重要な問題となる。
２番目の項目は解析結果の信頼性の向上という点で重要な項目となる。　パターン認識

に限らず統計等多数のパターン（サンプル）を扱う解析手法では、解析に利用されるパターン数が多いほど解析結果の信頼性が向上する。　従って、解析前に充分な数、且つ良質なサンプルを収集する事が大切である。

## １０．第１章の簡単なまとめ

- パターン認識によるアプローチは、従来手法では解決困難な問題に対して解を与える強力な手段である。
- 解決する為の手段や原理の解明はひとまずパターン認識にまかせておき、当面必要となる解析結果を手にいれる為の便利な手法である。
- 従来の論理的なアプローチと手法的に重なる所はなく、むしろ相互に補いあうものである。
- 従来手法とパターン認識の２種類のアプローチを排他的に用いるのでなく、相互に長所を補いあう形で利用する事で、高度な作業の実行が可能である。

## 2. パターン認識の定義

### 1. ニューラルネットワークによるパターン認識の実行について

ニューラルネットワークではデータ中に潜む人間では解読不可能な情報を“学習”でつかみとり、ネットワークの太さの情報として蓄積し、その蓄積された結果を利用して未知データの解析を行う。 従って、ニューラルネットワークでは“学習”が重要で、人間と同様に学習の出来／不出来に従ってパターン認識の結果の信頼性がきまってしまう。

### 2. 学習内容についての考察

□ ニューラルネットワークによる学習（パターン学習）と
人間による学習（ルール学習）の違い

ニューラルネットワークを用いたパターン認識では“学習”と称される過程を経る。この学習は人間が行う学習とは内容が異なっている。 つまり、パターン認識の学習に与えられるデータは様々な事象を起こすに必要なデータとその結果を示すデータである。一方、人間の学習はむしろ物事を解決するのに必要なルールを学ぶものであり、学習する内容は問題解決の為のルールであり、知識の習得である。 ニューラルネットワークで行う学習は、人間が行う時にはむしろ、データ整理および法則の発見であり学習というよりもむしろ解析過程と考えられる。 従ってパターン認識の学習と人間の学習は言葉は同じでも、その内容／目的は全く異なっている。

### 3. パターン認識におけるニューラルネットワークと多変量解析との関係

パターン認識を行う基本手法としてニューラルネットワークと多変量解析とがあることは既にのべた。 これら2手法の共通点は、共に多次元データを扱うという点である。パターン認識ではこのニューラルネットワークと多変量解析は混同して利用される事が多いが、歴史的にみると2つの手法は全く異なる起源と展開を経てきた。

ニューラルネットワークは人間の優れた認識能力に着目し、その認識の中核をなすと思われる神経細胞の動きをシミュレートすることで展開されたもので、生物／生物物理／生理学という分野から発生してきた学問である。 その後、この学問は2つの流れに別れてゆく。 ひとつは生物関連の研究者が中心となって展開されるバイオニクス、サイバネティクス等の学問。 もうひとつは工学関係の研究者が中心となって展開されるパターン認識関係の学問である。 この流れはパーセプトロン／アソシアトロン／バックプロパゲーションと受け継がれ、現在では単に認識するという過程を経て、理解をするというパターン理解の学問へむけての展開が始まっている。

一方、多変量解析は数学の分野で展開されて来たものであり、生体細胞との関連性は何もない。 これは計算機の発達に伴い、従来は人間の計算力では扱う事が困難であった多次元データを簡単に扱えるようになった事から急速に展開された手法であり、統計等の手法とも一部オーバーラップしてくる。 従って、多変量解析をパターン認識手法として利用することは多変量解析の一利用例にしかすぎない。

ニューラルネットワーク

| 起源／ルーツ | 様々な展開手法 | 主たる適用分野 |
|---|---|---|
| 生体細胞の<br>シミュレーション | パーセプトロン<br>アソシアトロン<br>バックプロパゲーション | 画像／音声認識<br>学習／連想 |

多変量解析

| 起源／ルーツ | 様々な展開手法 | 主たる適用分野 |
|---|---|---|
| 計算機の発展<br>多次元データ | クラス分類手法<br>クラスタリング<br>重回帰、その他 | 様々な多変量<br>データ解析分野 |

図1．ニューラルネットワークと多変量解析の全体概要

共通点は先にも述べたように多次元データを扱う点であるが、解析目的はニューラルネットワークはクラス分類が主目標で他の解析目標には利用されないことが多い。　多変量解析でもクラス分類が主目的ではあるが、これ以外の様々な目的にも適用される。
ニューラルネットワークは画像認識、音声認識という面で大いに利用されており、独自の発展を続けている。　多変量解析も最近の計算機の発展に伴い様々な手法が展開されつつある。　以下には、パターン認識を支えるこれら２種類の手法についてのべる。
＊化学の分野で“パターン認識”という時、ニューラルネットワークと多変量解析とを特に区別して扱う事は少ない。　これは、化学の分野では両手法とも単なる解析手法の一つとして考えている為である。　ここではこの伝統的な考えに従い、特に２つを区別して扱う事はしない。　従って、単に“パターン認識”という時はその要素技術としてニューラルネットワークと多変量解析を含むものとする。　特に区別が必要な時は、そのたび明記する。

## ４．ニューラルネットワークと多変量解析手法の分類

□　既知クラス情報の有無を基本とした分類

現在パターン認識を行う為の解析手法としては様々な手法が存在する。　ここではこれらの手法を分類し、分類された手法がどのような特徴を持つかについて考察する。
ニューラルネットワークや多変量解析を問わず、総ての手法は解析時に用いるパターン（母集団）が有するクラス情報を解析時に用いるか否かにより、２グループ（教師付き学習、教師無し学習）に分類する事が出来る。
この中で特にクラス分類問題に関しては、分類に用いられる判別関数の求め方の差異により、さらに２つのグループ（パラメトリック、ノンパラメトリック）に分類出来る。
表３にはパターン認識に現在利用されている代表的手法を分類したものが示されている。

表３．パターン認識に用いられる様々な手法の分類

| 解析手法 | 学習手法 | | 統計指標の利用 | |
|---|---|---|---|---|
| | 教師付き | 教師無し | ノンパラメトリック | パラメトリック |
| パーセプトロン | ○ | | ○ | |
| ＢＡＹＥＳ判別分析 | ○ | | | ○ |
| Ｋ－ＮＮ（最近隣法） | ○ | | ○ | |
| クラスタリング | | ○ | ○ | |
| 主成分分析 | | ○ | ○ | |
| 因子分析 | | ○ | ○ | |

□　分類基準の詳細説明

①　教師付き学習（SUPERVISED LEARNING）

前提条件：　既知クラス情報（クラス分けを行う基準となるもので教師データとして利用される）が存在する事（図１）

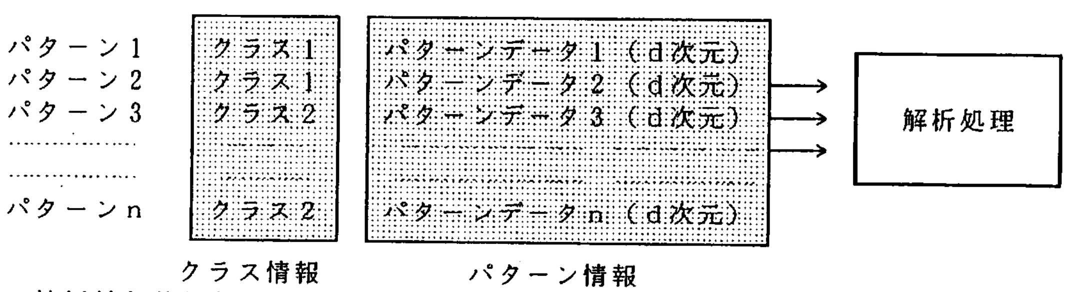



図１．教師付き学習を行う時のデータ構成及び解析の流れ

この手法はクラス帰属に関する情報があり、その情報を利用しながら解析を行う手法に関するものである。　このクラス帰属情報とは、例えば構造活性相関であるならばサンプルとして取り出された個々の化合物の薬理活性（効果の有無／強さ等）の情報であり、文字情報であるならば、その入力された情報が何の文字であるかについての情報である。
多くのパターン認識手法は教師データを用いた“学習”の過程を経て、入力された総て

のデータが正しく解析出来た時点の結果（学習成果）を用いて、クラス未知パターンの解析を行う。
この分類には神経細胞をシミュレートする事で始まったニューラルネットワーク手法の総てが含まれている。

例）パーセプトロン（線型学習機械法）
この手法では、線型判別式を求める“学習（エラーフィードバックトレイニング）”過程で教師データが利用　　される。

例）ＢＡＹＥＳ判別分析（多変量解析）
この手法では、まず先に既知クラス情報を用いてクラス毎の確立密度を求めるが、この過程で教師データが利用される。

② <u>教師無し学習 (UNSUPERVISED LEARNING)</u>

前提条件：　既知クラス情報を必要としない手法（図２）

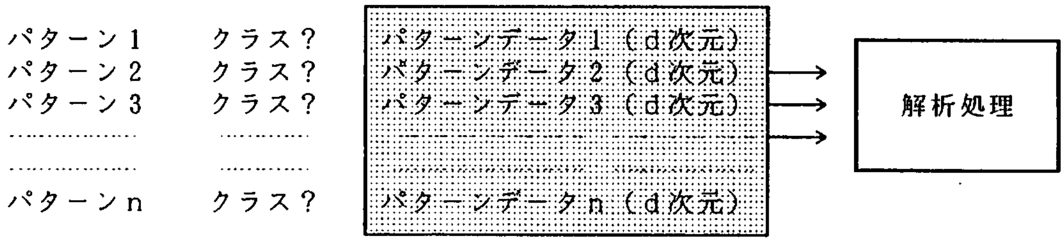



図２．教師無し学習を行う時のデータ構成及び解析の流れ

この手法では、その解析過程でクラス情報を必要としない手法である。　むしろ、新たにクラス帰属に関する情報をまとめたり、全体的な分布状態などを俯瞰する目的で利用される事が多い。　手法としては、クラスタリング、マッピングやその他の手法がこの分類に入る。

例）クラスタリング
・パターンの類似関係を基準にグループ化についての情報を与える手法。

例）ＮＬＭ　( NON-LINEAR MAPPING)（非線形写像）
・Ｎ－次元空間上でのパターン間の相互位置関係を、人間が把握しうる２次元上でのマップとして表示する。

例）主成分分析
・Ｎ－次元空間上でのパターンの分布状態を、最も分散（広がり／分布）の大きな軸を基準とした新たなＮ－次元空間上での分布にうつしかえる。

## ５．パラメトリック及びノンパラメトリック手法

パラメトリック及びノンパラメトリック手法との基本的区別は、解析時に統計的な確率分布等のデータを用いて解析するか否かにある。
パラメトリック手法の基本は、予め総てのパターンに関するクラス分布データが存在する事が前提である。　従って、この基準が適用出来る手法は総て教師付き学習手法に含まれる。　但し、このクラス帰属データの扱いがパラメトリック手法とノンパラメトリック手法とで異なる事になる。　以下にパラメトリック及びノンパラメトリック手法について簡単にまとめる。

①<u>パラメトリック手法</u>：　確率分布やパターン分布に関する基本データを求め、そのデータを用いて様々な解析をする

例）Ｂａｙｅｓ　線型／非線型判別分析：
Ｂａｙｅｓ法ではクラス毎の分布の重心を求め、その重心間に垂直２等分線を引きクラス分類を行う。

②<u>ノンパラメトリック手法</u>：　パターン分布データを用いない解析手法

例）パーセプトロン（線型学習機械法）：
パーセプトロンで用いる判別関数は個々のパターンの学習過程（エラーフィードバック学習）で獲得される。

## 3.　パターン認識における代表的手法：ニューラルネットワーク

### 1．ニューラルネットワークの基本的なネットワーク構造

先にパターン認識を実行する為の要素技術としてニューラルネットワークと多変量解析の２種類存在する事を述べた。　ここではパターン認識を代表する手法として伝統的に利用されてきたニューラルネットワークの手法を中心として述べる。　尚、ここで展開するニューラルネットワークは現在華々しく展開されている第２世代のニューラルネットワークとは異なり、パターン認識の原点ともなる第１世代のニューラルネットワークである。この第１及び第２世代の歴史的な違いは第一章第　節で述べたが、これらの機能的な違いは第１世代のニューラルネットワークが線型問題にしか適用できないのにたいし、第２世代のニューラルネットワークは非線型問題にも適用可能である事である。　第２世代のニューラルネットワークは、第　章にて詳しく紹介するが、パターン認識を行う手法としての基本的な考えは第１／２世代とも同じである。

現在まで利用されてきた第１世代ニューラルネットワークの最も代表的な手法はパーセプトロンである。　この手法は生体内部における神経細胞の働き、特に生体が得意とする認識問題を細胞レベルで追求し、同時に最も高尚な機能の一つである生物の“学習”機能をシミュレートしようとするものである。

従ってこの手法の基本的考えは、“学習”（フィードバックトレイニング）によりネットワーク（神経細胞間のモデル）相互の結合強度が変化し、この結合強度の変化が学習の“成果”そのものであり、新たな問題の解決（主として分類問題）は学習で獲得されたネットワークを用いて行なうというものである。

この神経細胞をシミュレートする為のネットワークモデルとして様々な形のネットワークが提唱された。　最も一般的で広く使われているネットワーク構造としては階層的構造を持つパーセプトロンが有名である。　このパーセプトロンの学習アルゴリズムを実践する一つの手法として線型学習機械法(LINEAR LEARNING MACHINE) とよばれる手法である。ここではこの線型学習機械法を中心にのべてゆく。

図１は、このパーセプトロンのネットワーク構造と“学習”の流れを示す。

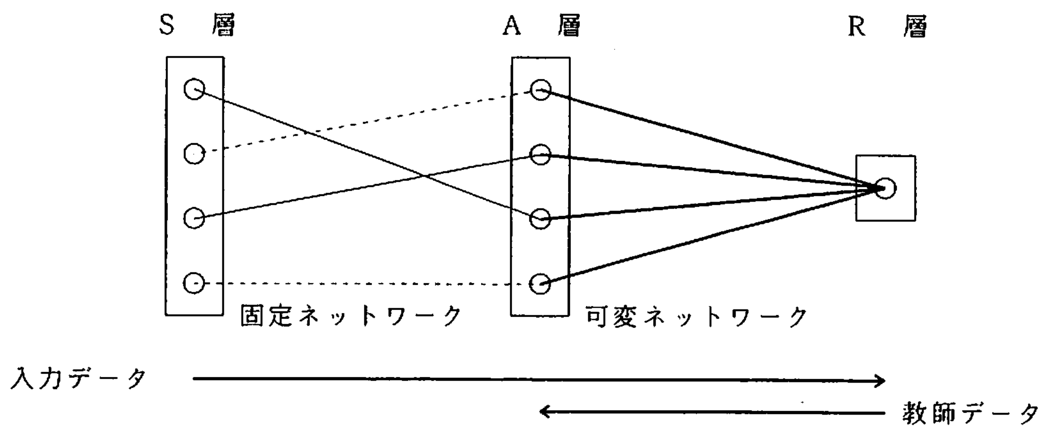



図１．パーセプトロンの基本ネットワーク構造

パーセプトロンは３層構造を持つニューラルネットワークである。　データはＳ層から入力され、順にＡ層、そしてＲ層へと各層内で処理されながら流れて行く。　このネットワーク構造で、Ｓ層とＡ層間に張られたネットワークは固定されているが、Ａ層とＲ層間に張られたネットワークは強度が変化する可変ネットワークである。　このネットワークを変化させるのに、一般的にフィードバックトレイニングと呼ばれる学習アルゴリズムを用いている。

Ｓ層とＡ層間のネットワークは固定（値があらかじめ決められており変化しない）されているので、Ｓ層から入力されてくるデータに一定の処理をする単なる前処理過程と考えることができる。　ニューラルネットワークとして最も特徴的な“学習”はＡ層とＲ層との間に張られたネットワーク強度（値）を変化させる事で実現される。

この“学習”を行う具体的な動きを図２に示す。　この図では図１中のＡ層とＲ層間を拡大し、学習を行う為のデータ等を含めて図示したものである。

図中Ｘ１～４は入力データ（従って４次元データ）、Ｗ１～４は学習で変化するネットワーク強度の値、Ｄは入力データＸ１～４とネットワーク強度Ｗ１～４とで算出される値で、各パターンの出力データとなる。　なお、このＤを求める式（１）は線形判別関数と呼ばれているものとおなじである。

この図に従ってパーセプトロンの学習におけるデータの動きを簡単に追ってみる。 ①入力パターンＰｉに関する４次元入力データＸ１～４が与えられたならば、この入力データとネットワークＷ１～４の積和を取り、Ｄｉ値を求める。 このＤｉは入力されたｉ番目のパターンに関し、現在のネットワークの値を用いて求められたものである。 ②各パターンの分類は、Ｄｉ値の正／負の判定を行うことで行なわれる。 ③そのＤｉ値が教師データＴｉと比較される。

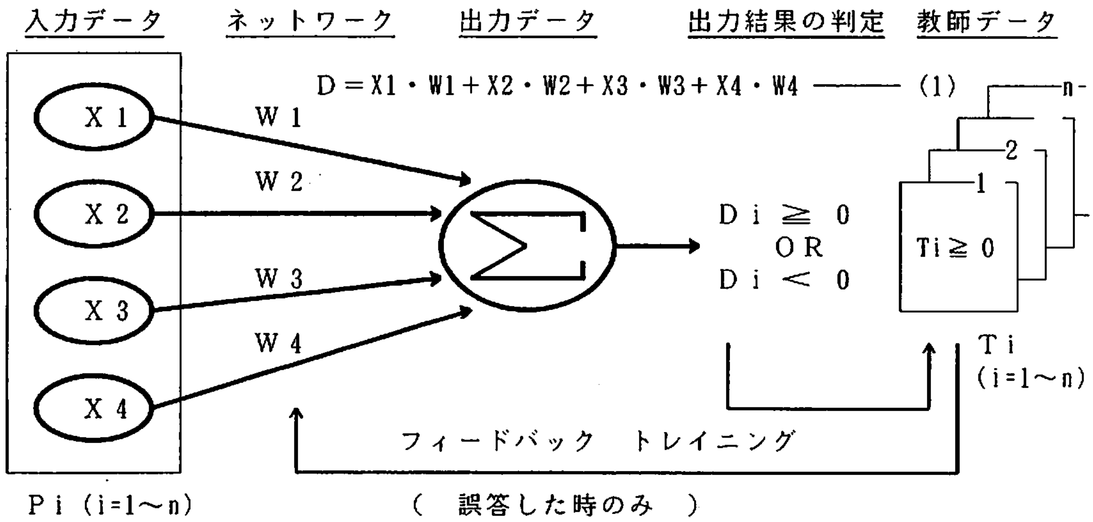



図２．パーセプトロンの“学習”の流れ

④Ｄｉの正／負が教師データＴｉの情報と一致したならば、ネットワーク強度Ｗ１～４の値を変更する事なく次パターンの分類を行う。 一致しない時は、ある定められた式に従いクラス分類が正しく行われるようにネットワーク強度Ｗ１～４の変更が実行される。 この①～④の過程が“学習”であり、フィードバックトレイニングと呼ばれている。 この学習を繰り返すことで、分類が成功するようなネットワーク強度（Ｗ１～Ｗ４）が求められる。 学習は、全パターンが正しく分類された時点で終了する。 クラス未知パターンの予測は、学習が終了した時点のネットワークを用いて実行される。

＊　Ｘ１～４は形式上パターンベクトル、Ｗ１～４はウエイトベクトルと呼ばれる。

## ２．線型学習機械法

前節ではパーセプトロンのネットワーク構造を中心とし、学習がいかにして行われるかという点について述べた。 ここでは多次元空間におけるクラス分類機という観点を中心とし、線型学習機械法と呼ばれる分類手法について考える。

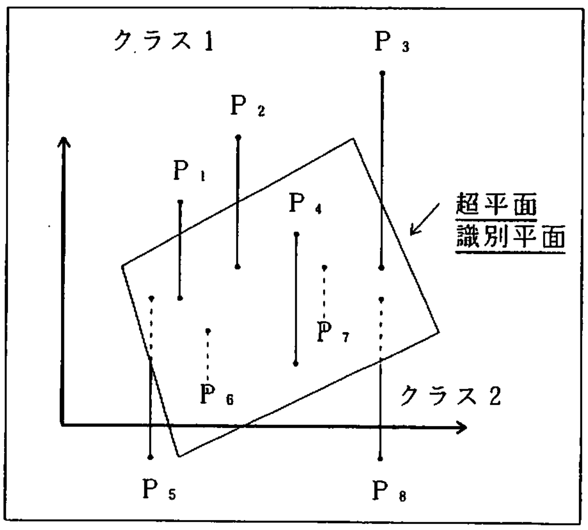



図３．パーセプトロンによる多次元データの分類

図３には線型学習機械法の多次元空間における分類の状態が示されている。 この線型

学習機械法の幾何学な意味は、N－次元パターン空間中に存在するパターンを超平面（HYPER PLANE)／識別平面（DECISION PLANE）（線型判別関数で示される）で分類するものである。 線型学習機械法の学習ではこの分類に用いる超平面を求めていることになる。

クラス決定に用いる面が平面である事から推測できるが、この線型学習機械法は文字通り“線型”に分割できる問題にのみ有効な手法である。 この分割面が非線型である解析手法はニューラルネットワークや多変量解析を通しても数すくないが、最近展開されてきた第2世代のニューラルネットワークは非線型の識別面を用いて分類する手法である。

図3において、パターン空間上のパターン$P_1$～$P_8$は識別平面（超平面）の上にあるか、下にあるかで2クラスに分類される。 超平面の上にあるパターンをクラス1、下にあるものをクラス2とするならば、超平面により$P_1$～$P_4$はクラス1に、$P_5$～$P_8$はクラス2に帰属される。

＊超平面は多次元空間のパターン分布等を述べる時によく利用されるが、線型学習機械法を論じる時は識別平面と呼ばれる事が多い。 以下、識別平面と記述する。

### □ 識別平面によるクラス分類とd＋1次元との関係

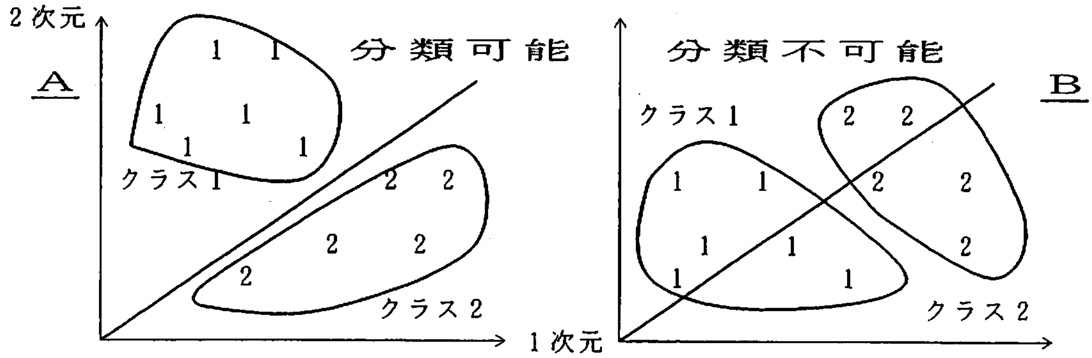



図3．パターン分布と識別平面によるクラス分類（2次元の時）

前節では個々のパターンは識別平面の上にあるか、下にあるか（数学的には判別関数の値Dの正／負に従う）でクラスが決定される事を示した。

一般的に、パターンの分布状態により分類が可能な時と不可能な時との2つのケースが存在する。 即ち、図3のAのようにパターンが分布している時は2次元データによる決定線（2次元の為決定線とした）によりクラス決定可能であるが、図3のBのように分布する時はクラス分類不可能である。

このような分布パターンを持つデータを分類可能とする為、一般的には次元数を1次元増やして分類を行う。 即ち1次元増やす事で、この2次元パターンの分類問題は3次元分類問題へと拡張され、クラス決定に用いる識別線は1次元増えた3次元平面となる。この結果図4で示されるように2次元では分類不可能であったデータのクラス分類が可能となる。 この関係は2、3次元から多次元（d及びd＋1次元）へと拡張できる。

従って、線型学習機械法ではパターンベクトルとウエイトベクトルそれぞれを1次元増やしたデータを用いて解析するのが一般的である。 この増やされたデータは、通常d＋1次元として扱われる。

図4のような特殊な分布をするパターンを分類する時、個々のパターンは2次元データで表現されるが、分類の為の識別平面を形成するには、3次元データが必要となる。 従って、線形学習機械法では一般的にd＋1個目のデータが必要となる。

従って、線型学習機械法を実行する時に用いられる個々のパターンベクトルX及びウエイトベクトルWはそれぞれ、式1、及び式2のようになる。

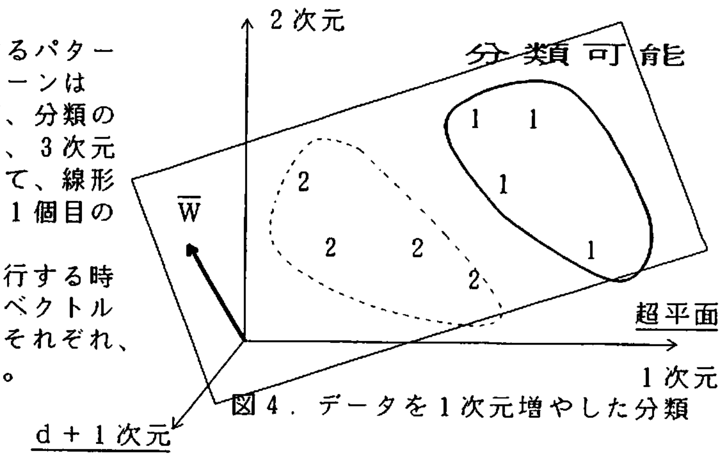



図4．データを1次元増やした分類

$$\overline{X} = (X_1, X_2, \cdots\cdots X_{N-1}, X_N, \underline{X_{d+1}}) \quad (1)$$

$$\overline{W} = (W_1, W_2, \cdots\cdots W_{N-1}, W_N, \underline{W_{d+1}}) \quad (2)$$

・ $\underline{X_{d+1}}$ 及び $\underline{W_{d+1}}$ 項がデータに追加されたｄ＋１次元である。

□　クラス分類について

線型学習機械法によるクラス分類は図４からもわかるように、パターンベクトルＸとその原点からのびるウエイトベクトルＷとの内積（スカラープロダクト）を計算することで行われる。　すなわち、パターンのクラス決定（分類）はこの内積値の正／負に従って決定される。

| | |
|---|---|
| $D = \overline{W} \cdot \overline{X} = \|W\| \cdot \|X\| \cos\theta$ | $D \geqq 0$　の時、クラス１<br>$D < 0$　の時、クラス２ |

パターンが形成するベクトルとウエイトベクトルが識別平面に対し同じ方向に向いている時、ＣＯＳθは正となる。　従って、内積Ｄは正をとる。　一方、２つのベクトルが互いに反対の方向をむいている時、ＣＯＳθは負となり、内積Ｄは負となる。

このように各パターンのクラスは識別平面にたいするパターンベクトルとウエイトベクトルの内積をとることで容易に決定される。

□　判別関数について

クラス決定はパターンベクトルとウエイトベクトルの内積を求め、その内積値の正／負によりクラスを判定する。　実際の計算は以下の式に従って行われる。

$$D = \overline{W} \cdot \overline{X} = W1\ X1 + W2\ X2 + \cdots\cdots + Wn\text{-}1\ Xn\text{-}1 + Wn\ Xn + Wd\text{+}1\ Xd\text{+}1$$

この判別関数を用いて各パターンのクラスを決定する。

□　パーセプトロンと線型学習機械法との関係

線型学習機械法はパーセプトロンモデルを数学モデルとして実現したものである。　線形学習機械法における数学的手続きは、パターンベクトルとウエイトベクトルの内積をとり、得られたスカラー値の符号を教師データと比較することでクラス決定を行うものである。　この時用いられるウエイトベクトルはパーセプトロンにおけるネットワーク構造で考えるならば、ネットワークの太さ（値の大きさ）と対応させられる。　従って、このウエイトベクトルはさきにも述べたように“学習”により変化させることでニューラルネットワークを実現することになる。

## ３．学習（LEARNING: ERROR FEED-BACK TRAINING）について

線型学習機械法での“学習”とは、誤分類パターンがあった時、その誤分類パターンが正しく分類出来るように識別平面の位置を変更する事を意味する。　この識別平面の位置を変更する事は、判別関数のウエイトベクトル値を変更する事で達成される。

この識別平面の位置を求める手法については様々なアプローチがあるが、ここでは識別平面を誤分類したパターンの対称位置に反転させる手法についてのべる。

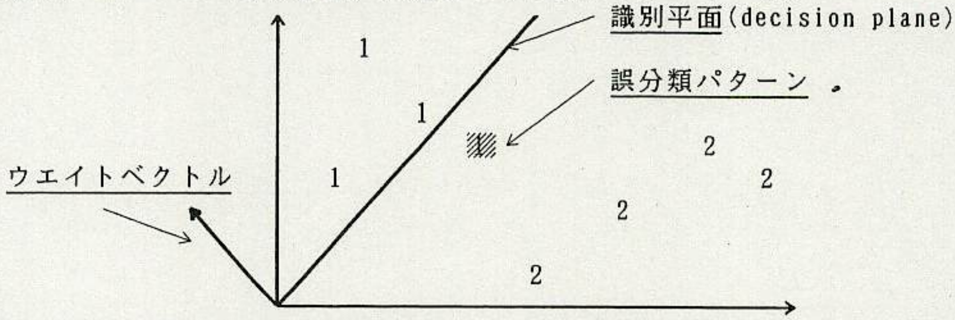



図５．識別平面によるパターンの分類

上図ではクラス１パターンの一つが誤分類されている。　この誤分類したパターン１と原点を結ぶ線を基準として識別平面を反対側に反転させると下図のようになる。　この操作により、誤分類されていたパターン１が正しく分類される。　このように、誤分類パターンが正しく分類されるようにウエイトベクトルを変化させる過程が“学習”そのものである。　この学習（識別平面の反転）は、すべてのパターンが正しく分類されるまで繰り返し実行される。

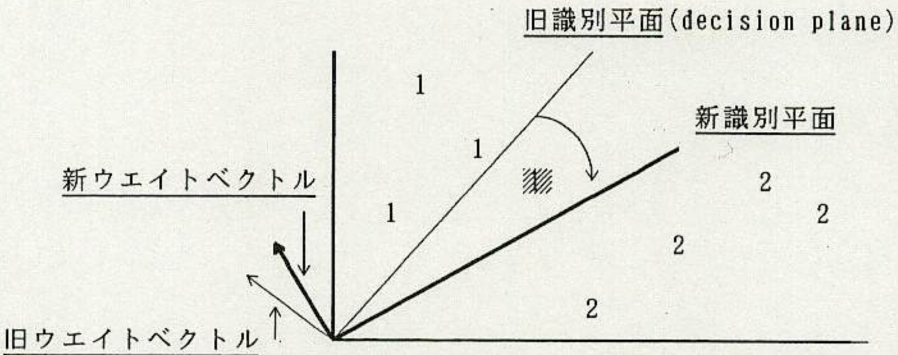



図６．ウエイトベクトルの反転による誤分類パターンの分類修正

□　学習過程におけるウエイトベクトルの修正について

既に何回ものべたが、線型学習機械法で学習（フィードバックトレイニング）を行うということは、ウエイトベクトルを修正することを意味する。　以下にこのウエイトベクトルの修正について順を追って述べる。

識別平面を誤分類パターンに対し反転させると、古いウエイトベクトル**Wo**は新たなウエイトベクトル**Wn**へと変換され、パターンベクトル**Xp**とから得られるスカラープロダクトＤは正から負または負から正へと変化して、誤分類パターンは正しく分類される。

（学習前）　Ｄ　→→→　－Ｄ　（学習後）

この時スカラープロダクトＤの符号は変わるが、その絶対値の大きさに変化は無い。従って、学習前のスカラープロダクトをＤとし、学習後のスカラープロダクトをＤ’とすると以下の関係が成り立っている。

$$D' = -D \qquad (3)$$

従って、

$$\mathbf{Wn} \cdot \mathbf{Xp} = -\mathbf{Wo} \cdot \mathbf{Xp} \qquad (4)$$

またウエイトベクトルを変更して新たなウエイトベクトルにする為に必要な修正項をＣとし、パターンベクトル**Xp**との内積をとると、

$$Wn = Wo + C \cdot Xp \qquad (5)$$

となる。

ここで、（５）式のWnを（４）式に代入し、Cについて変換すると、

$$C = \frac{2(-Wo \cdot Xp)}{Xp \cdot Xp} \qquad (6)$$

（３）と（４）式から　$D' = -Wo \cdot Xp$　従って、

$$C = \frac{2D'}{Xp \cdot Xp} = -\frac{2D}{Xp \cdot Xp} \qquad (7)$$

となる。

学習は、誤分類が発生したならば（７）式に従ってCの値が計算され、次に（５）の式に従って新たなウエイトベクトルWnが求められる。　この修正されたウエイトベクトルを用いた判別関数により、誤分類パターンは正しく分類される。

## ４．ウエイトベクトルの初期値の決定

学習を行う前に判別関数はパターン空間上のどこかに位置していなければならない。この判別関数の位置を決める要素としてパターンベクトルのｄ＋１次元目の値とウエイトベクトルとがある。　ここではこの２つについて考察する。

### □　ｄ＋１次元の値の初期値

線型学習機械法を行うには線型分割性を保証する為にｄ＋１次元目のウエイトベクトル値が必要である。　このｄ＋１次元値は０以外の任意の値を取ることが可能であるが、通常初期値としては１０００をとることが多い。

### □　ウエイトベクトルの初期値

一般にウエイトベクトルの初期値として、通常はｄ＋１次元目を含む総てのウエイトベクトルを＋１又は－１に設定する事が普通である。　この時、初期ウエイトベクトルは用いるデータの種類によらずｎ次元パターン空間中で、常にある定められた位置に設定される。　この為、真に理想とする場所とは程遠い場所に位置する可能性も高く、結果として学習の収束（完了）に時間がかかったりすることになる。

$$\text{初期判別関数} = +1X_1 + 1X_2 + \cdots\cdots + 1X_{n-1} + 1X_n + 1X_{d+1}$$

この他、多変量解析手法（例えばＢＡＹＥＳ判別分析）等で得られた判別関数のウエイトベクトルや、クラスパターン分布の重心を通る垂直２等分線等も初期ベクトルとして利用される事が多い。　このようなウエイトベクトルを用いる時、これらのウエイトベクトルは既にある程度の分類能を示すものとなっているので、理想に近い位置にウエイトベクトルが落ち着いている可能性が高く、学習時間（繰り返し回数）の大幅な短縮が期待される。

## ５．分類率や信頼性向上の為の工夫

入力データさえあれば、線型学習機械法を単に実行するだけで結果をうる事は可能である。　ここで考えなければならない問題として、得られる解析結果（分類／予測率）の向上や結果の信頼性等の問題がある。　解析結果の信頼性を守る為の基準等に関する議論は第　章で行う。　ここではこれらの問題を避ける、あるいは出来るだけ影響を少なくするという方法についてのべる。

### □　デッドゾーン（ＤＥＡＤ　ＺＯＮＥ：不感領域）設定による分類精度の向上

通常、識別平面は厚みを持たない。　しかし、特別な目的でこの識別平面に厚みを持たせることが行われる。　識別平面の両側に設けられたこの厚みの部分は"デッドゾーン"と呼ばれる。

このデッドゾーンが設定された線型学習機械法では、デッドゾーン内に落ち込んだパターンをその外部に押し出すようにして学習が行われる。　さらに、パターンのクラス予測時には、このデッドゾーン内に落ち込んだパターンは分類／予測の対象にはならない。このデッドゾーン設定の主目的は二つある。

| |
|---|
| (1)　クラス識別平面がクラス間の理想的な位置に正しく設定されるようにする。<br>(2)　予測率の向上及びその信頼性の向上を目指す。 |

以下にこれらの詳細内容について順を追って説明する。

(1)　適切な判別関数（識別平面）の発見
デッドゾーンを設けて学習を行う事で、クラス分類を行うためのクラス識別平面が偏らない、理想的な位置（クラス間の中心の位置）に設定されるようにする。

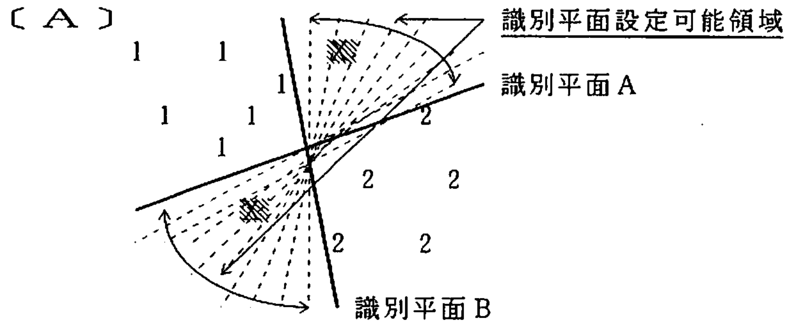



図７．デッドゾーンが無い時の識別平面設定可能領域図

線型学習機械法ではクラスター（グループ）間に識別平面が引かれる事でクラス分類が行われる。　この時、２つのクラスター間にギャップ（パターンの存在しない空間）が存在する時、このクラスター間に引かれる識別平面は多数存在する。
図７からもわかるように、識別平面によるクラス分類では識別平面ＡとＢの間の空間の任意の位置をとる事が可能であり、いずれの場合でも１００％正しく分類できる。　この時の問題点はクラス未知パターンＸが存在した時、そのクラス予測は引かれた識別平面の位置に依存する為、全く同じデータを用いても識別平面の場所によりクラス予測が全くことなる結果になる事である。
理想的な識別平面は、クラスター間の真ん中に垂直２等分線を引く事である。　しかしながら線型学習機械法では、識別平面はクラスター間の接線方向（図７中Ａ及びＢ）に落ちつき易いという特性を持つ。　識別平面Ａ、Ｂのようにクラスター間で偏った位置にある識別平面を用いた時は、予測率の大幅な落ち込みと信頼性の低下をともなう。
この問題はデッドゾーンを設定する事で解決可能である。　即ち、識別平面に厚み（デッドゾーン）を与える事で、識別平面をより理想的な位置（クラスター間の中心）に近づける事が可能となるのである。

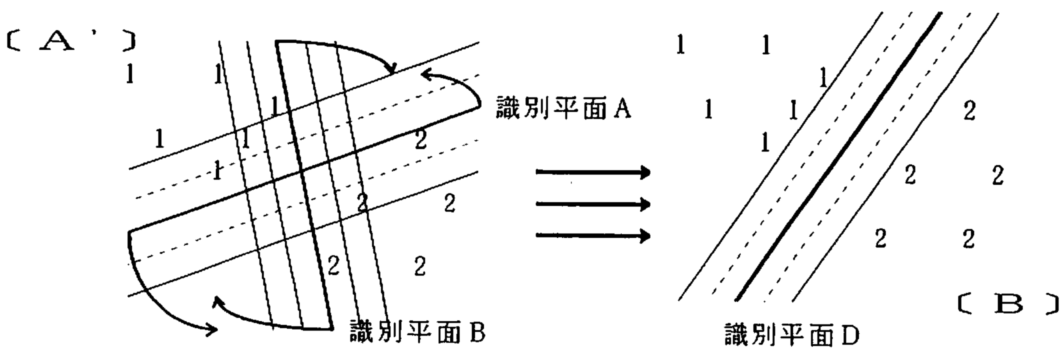



図８．デッドゾーン設定による識別平面の改良

図８はデッドゾーンを設けない時と設けた時との識別平面の状態を示している。
図７の〔A〕の識別平面ＡとＢとにデッドゾーンを設定（図８〔A'〕）して学習を行わせると、学習の結果得られる識別平面は図８の〔B〕のように、クラスの中央に近い位置に設定される事がわかる。
これは、先にも述べたように識別平面を設定する為の学習過程で、デッドゾーン内部に落ち込んだパターンをそのデッドゾーンの外に押し出す形で新たに識別平面を設定するという、特殊な学習を行うからである。　この学習を繰り返し、完全にクラス分類が出来る（収束した事を意味する）まで学習を行う。　この学習で得られた最終識別平面は、その

デッドゾーン内部にパターンを含まないものであり、図Bのようにそのデッドゾーンの値（厚さ）が適切であれば、得られた識別平面は理想位置に近いものとなっている事が期待される。　即ち、予測に関する信頼性の高い識別平面が得られた事を意味する。

なお、デッドゾーンの厚さが不充分な時は最終的に得られた識別平面の信頼度は低くなる。（例：図８のＢ中、…で示されたデッドゾーン等で、一般的には識別平面の傾きも増加する）　従って、より信頼度の高い識別平面を得るには、適切な厚さのデッドゾーンを捜し出す事が必要である。

デッドゾーンを設定したとき、学習の収束はデッドゾーンを設定しない時とくらべて困難となる。　当然ながらこの傾向はデッドゾーンの厚さが大きくなるほど顕著となるが、厚いデッドゾーンを設定して得られた判別関数の信頼度は極めて高いものとなる。　現時点では、残念ながらこの厚さを自動的にきめるスマートな方法は存在しないので、試行錯誤的に行うことが必要となる。

(2)　予測率および予測精度の向上

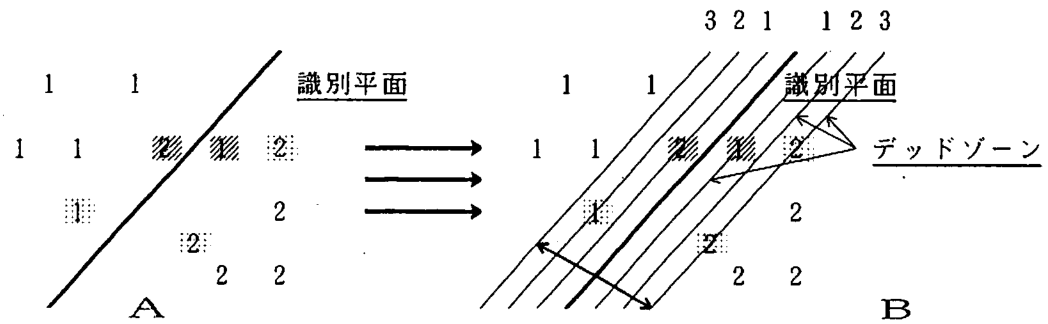



図９．デッドゾーン設定による境界パターンの排除によるクラス分類／予測

デッドゾーンを設けて学習した結果得られた識別平面を用いて予測する時、予測時にこのデッドゾーン上に落ち込んだパターンは予測範囲外として予測の対象からはずされる。

例えば、図９のＡ図を例に取るならば、デッドゾーンが無い時には網かけ（図　中▨の網かけ）されたパターンは誤分類されることになり、この場合の予測率は８３％（１０／１２）となる。　しかし、デッドゾーンを設けた状態（図Ｂ）で分類を行うならば先に誤分類されたパターンがデッドゾーン内に落ち込むために分類対象から除かれる。　この結果、分類率は１００％に向上し、境界領域付近に位置するパターンが除かれるため、分類の信頼度も向上する。

このようにデッドゾーンを設定した分類では、境界領域付近に存在するパターンが多いデータセットを扱う時には高い効果が期待される。

## ６．その他の線型学習機械法特有の性質

線型学習機械法は実用上極めて有効な手法であるが、線型学習機械法そのものの特性や問題点も幾つか存在する。　解析問題に線型学習機械法を利用するには、このような特性や問題点を正しく理解して利用する事が必要となる。　ここでは線型学習機械法の特性や問題点について簡単にまとめる。

□　学習収束の問題

学習が「収束した」ということは、解析母集団の全パターンについてクラス分類が正しく行われた状態を意味する。　学習過程の最終目的はこの収束にあるが、パターンの分布状態により収束する事が困難／不可能な場合がある。

例えば解析母集団中に、相手のクラス中に落ち込んだインライヤー（IN-LIER)パターンが存在する時、学習の収束は不可能となる。　これは線型な識別平面を用いている限り、学習回数をふやしてもけっして正しく分類される事のないパターンである。　このような線形学習機械法を用いて学習収束（完全分類）が実現されなかった時に取るべき手続きは２通り存在する。　一つは、完全分類が実現するように何らかの手段をこうじる、他は完全分類しなくとも最終的に得られた判別関数を用いて結果を導き出すものである。

線形学習機械法の特性（学習が収束しなかった時に得られる判別関数の信頼性は極端に低下する）を考える時、学習が収束しなかった時は学習が収束するように何らかの手続きをとることが必要である。　以下に学習が収束しなかった時に取るべき手続きを簡単にまとめる。

学習が収束しない時の対策：

(1) インライヤー（IN-LIER)パターンをデータセット中から取り出す。
(2) 新しい数値データ（記述子）を追加してパターンの分離性を向上させる。
(3) 線型でなく、非線型問題に適用可能な分類手法を用いる。
(4) 例え完全に分類出来なくとも、ある程度分類がよく出来る段階の結果（判別関数）を用いる*1,2。（学習途中での分類率のチェックが必要である）
(5) AVERAGE SUBSET LEARNING MACHINE *3に従って新たな判別関数を求める。
(6) 他の線形判別分析手法*4の適用を試みる。

＊1．収束途中に得られた判別関数を用いて分類予測等を行う事は、線型学習機械法ではあまり望ましくないアプローチである。 これは線型学習機械法の特性上、その判別関数はパターン群間の中央部分に位置する事が少なく、パターン群の接線方向に引かれる可能性が高いからである。

＊2．学習が収束しない時に得られる判別関数がパターン間の中央に位置し易い手法としては、判別関数をシンプレックスや最小二乗アルゴリズムに従って最適化する手法がある。 従って、線型学習機械法で収束しない時、このような手法で得られた判別関数を用いて分類／予測等を行うのが望ましい。

＊3．この問題に対しRitter,Lowry,Isenhour,Wilkins ら（２４２、２４３）がひとつの試みを行っている。（AVERAGE SUBSET LEARNING MACHINE） このアプローチは、先ずデータセットを分割可能な幾つかのグループに分割する。 この時得られた複数の判別関数の平均値を新たな判別関数として採用する手法である。

＊4．線型学習機械法と同じ２クラス分類機でも、後に述べる最小二乗アルゴリズムによる線型判別分析やシンプレックスアルゴリズムを用いた線型判別分析は線型学習機械法と比較して、学習が収束しないときに得られる判別関数の信頼度は高い。

### □ 最終的に得られる判別関数の変動要因

線型学習機械法では学習時の初期値の差異や学習方法等の差異により、最終的に求められる判別関数のウエイトベクトル値が変化する。 この最終判別関数の値を変える大きな要因としては、以下に示す３つが主たる要因となる。

(1)学習時に用いるパターンの順番

学習時には判別関数の修正が誤分類パターンについてのみ行われる（アプローチにもよる）。 この為、算出される判別関数の値は学習時に用いられるパターンの順番に影響される。 特に最終判別関数の値は、学習過程の最後に誤分類されたパターンに大きく影響される。

(2)ウエイトベクトルの初期値の差

ウエイトベクトルの初期値は次に述べるｄ＋１項と同じく、学習時の判別関数の出発点を決めるものである。 従ってこのウエイトベクトルの初期値が異なると、最終的に得られる判別関数の値も変わってくる。 通常はこのウエイトベクトルの初期値として、全てに＋１、－１を与えて学習を開始する事が多い。

(3)ｄ＋１項の値の差

ｄ＋１項の初期値も学習時のウエイトベクトルのスタート地点を決める大きな要素である。 このｄ＋１項の値の変化により判別関数の収束地点も変化してくる。

このように線形学習機械法により得られる判別関数は、例え同じ解析母集団を用いたとしても様々な要因の影響をうけ、簡単に変化するものであることがわかる。 しかし、変化するとはいっても同じ解析母集団を用いて解析する限りにおいてはその値が大きく変化することはない。

このように判別関数の値が変化するという特性を利用することで、後に述べるノイズデータの除去（特徴抽出）（第４章）を行うことが出来る。 このような手続きの代表的な手法としてはウエイトサイン法やバリアンス法といった手法が存在する。 これらの特徴抽出手法については第　章にてくわしく説明する。

### □ その他の判別関数に関する問題

最終的に得られる識別平面（判別関数）には、線型学習機械法の特性に由来する独特の癖／傾向が存在する。 従って、これらの点に留意して解析する事が必要である。

この特性として幾つか知られているが、この中で解析時にある程度意識しなければならないものとして以下に３つ程明記する。

(1) 識別平面付近に存在するパターンによる影響

識別平面はパターンの分布状態により大きな影響を受ける。 特に、相手クラスに隣接したパターンに大きく影響をうける。 これらのパターンはインライヤー（相手クラスのクラスター中に存在し、正しく分類する事が不可能なパターン）に近いパターンが多い。この種のパターンを正しく分類しようとすると、識別平面を偏在させる結果となり、他の

正常な分布をするパターンのクラス分類／予測に悪影響を及ぼす事になる。
【解決手法】
① パターンベクトルに新たなデータを加え、パターン分布状態そのものを変える
② インライヤー（IN-LIER)パターンを取り除く

(2) **最終識別平面の落ち込む場所の癖について**
最終的に得られる識別平面はパターン群の中心部分でなく、パターン群の接線方向に落ち着くという特性がある。 これは学習過程のアルゴリズムに依存する特性であり、誤分類するパターンは境界領域に近いものが多いことと、新たに設定される識別平面は誤分類パターンを中心とした点対象反転で行われる事に由来する。
この問題に対する解決手法としては第２章の第５節で述べたデッドゾーンを設けることで達成される。

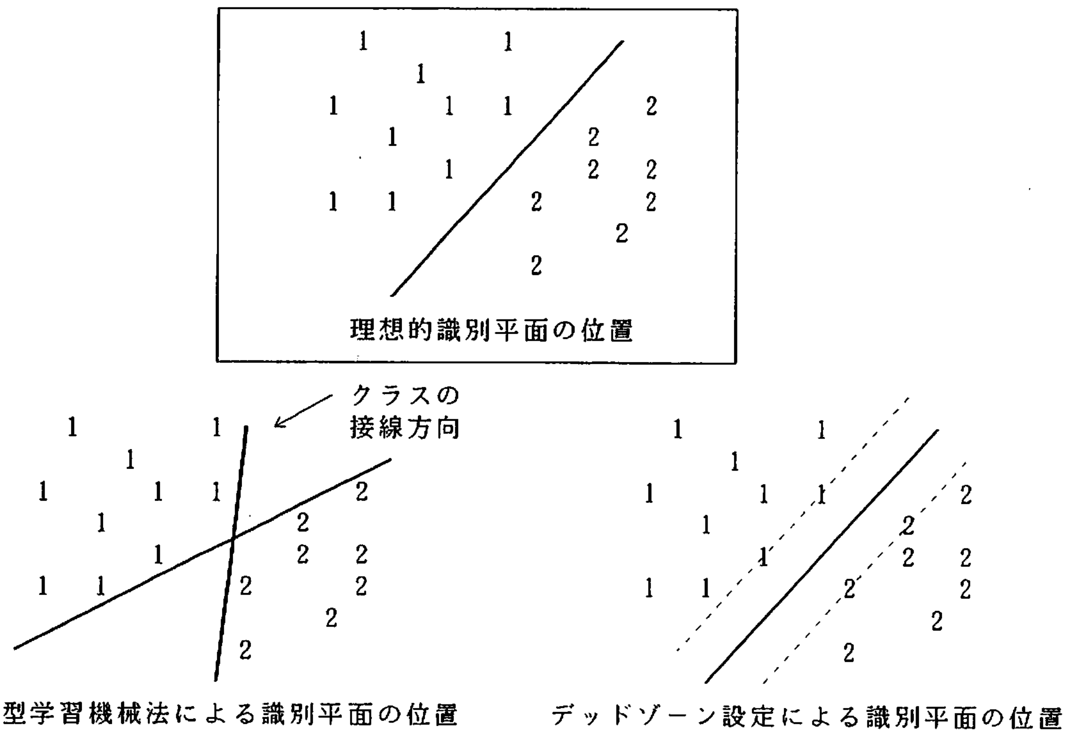
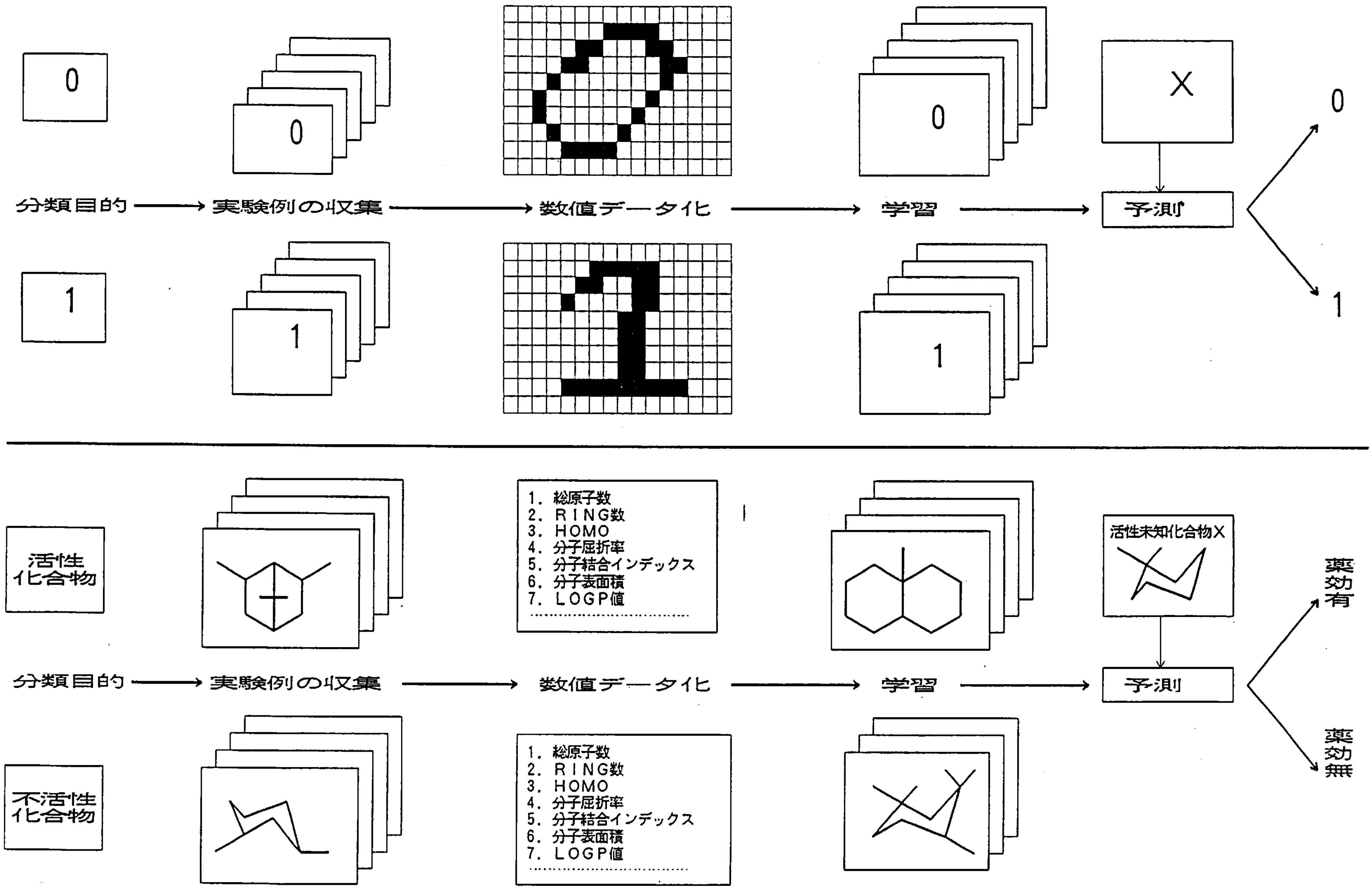



図１０．デッドゾーン設定による識別平面の改良



0
1
0
1
X
0
1
分類目的
実験例の収集
数値データ化
学習
予測
活性化合物
不活性化合物
1. 総原子数
2. RING数
3. HOMO
4. 分子屈折率
5. 分子結合インデックス
6. 分子表面積
7. LOGP値
活性未知化合物X
薬効有
薬効無

## 4.　線型／非線型判別分析（パラメトリック及びノンパラメトリック手法）

前節では線型学習機械法を伝統的なパターン認識手法として扱った。　この線型学習機械法は、多次元パターン分類の観点からみると線型判別分析手法の一種と考えられる。線型学習機械法では識別平面の求め方が“学習（フィードバックトレイニング）”という形で行われていたが、これ以外にも識別平面を求めるアプローチが存在する。　ここでは線型学習機械法とは異なる識別平面の求め方についてのべる。

### □1．線型および非線型判別分析に関する基本概念

以下に示される識別平面を構成する判別関数（1）のウエイトベクトル（W1～Wn+1）を変化させ、分類に最適な値を求める。　最終的に求められたウエイトベクトル（目的の分類に適したものとなっている）を用いた判別関数により、データの分類／予測を行うのが判別関数による分類／予測の基本である。

今、n次元データを用いる時、

$$D = W_1X_1 + W_2X_2 + \cdots\cdots + W_nX_n + W_{n+1} \qquad (1)$$

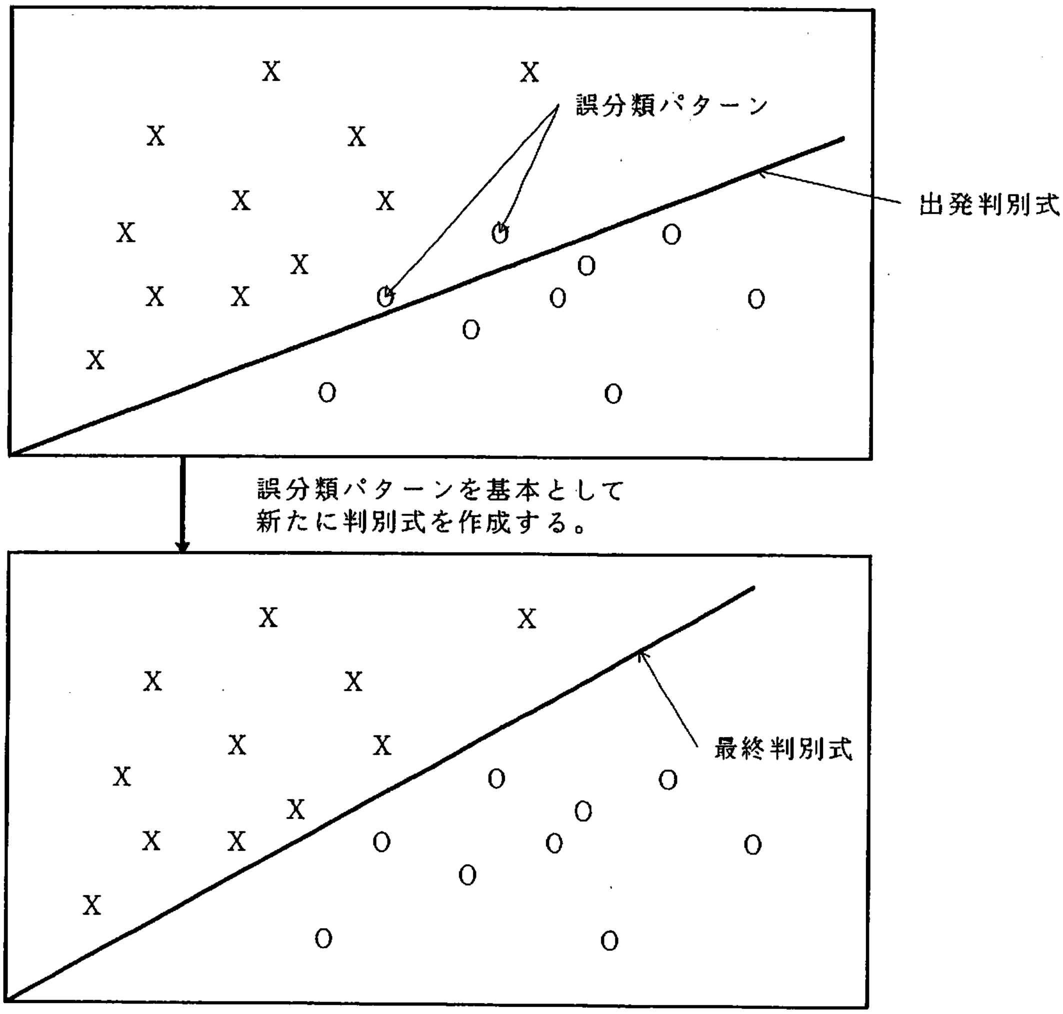



図1．判別関数を用いたクラス分類

(1)　出発判別関数

$$D = W_1X_1 + W_2X_2 + \cdots\cdots + W_nX_n + W_{n+1}$$

この判別式は1次関数の線形結合で示されており、図1で示される様な線形な識別面で分類される事を示している。　この線型識別関数の係数（W1～Wn+1）を変える（この変換手法に従い種々の手法が存在する）情報は、誤分類されたパターンデータを用いる。