基礎的機械学習手法の比較

島根大学　総合理工学部　数理･情報システム学科

計算機科学講座　田中研究室

S043013　岩脇　正浩

第1章　　序論

1.1　研究の背景

最近では人間を手助けするロボットの開発，研究が盛んに行われている．そのロボットの実現には文字認識，音声認識，画像認識といった認識の技術が必要であるが，誤った認識を行わないようにするために，機械学習が重要になってくる．

そこで，まず，機械学習手法の基本原理を理解することが重要となってくるため，複数の代表的な機械学習手法の実装を行いながら，機械学習手法について理解を深めていくことを第一の目的とする．

また，実装を行っていく上で，今回は機械学習に重点を置きながら比較を行っていく．これは，ロボットなどが周囲の環境の変化に適応するためには，素早く学習を行わなければならない．そこで，リアルタイムで学習することが求められてくる場合を想定すると，認識の誤差の少なさも重要だが，機械学習が特に重要になってくると考えたからである．

1.2　研究の概要

機械学習手法において比較的容易に実装を行える，ユークリッド距離を利用した手法として，Template Matching 法[1]と k-nn 法[1]を実装した．つぎに，ニューラルネットワークの代表的な機械学習手法であるバックプロパゲーション法[2]と，決定木学習として二分木による学習[3]を実装した．

そして，この 4 つの手法に対して，教師データとして，データ数が少なくあまり複雑でない R.A.Fisher のアヤメのデータセット[4]と比較的にデータ数が多く，複雑である WDBC のデータ[5]をあたえて，その誤り率と学習時の時間計算量の比較をおこなった．

また，データセットを教師データと試験データへと分割し，教師データで学習を行ったあとに，未分類の試験データを与え，その誤り率についても比較した．このとき，実装した機械学習手法の性能を比較するために，先行研究と比較も行った．

第2章　機械学習手法

2.1 用語説明

特徴ベクトル x_i は1行 N 列からなるベクトルとする．以後ベクトルと記す．また，ベクトル x_i が属するクラスのクラスラベルを y_i とする．$(i = 1,2\cdots,N)$

クラス c に属するベクトルとは，対応するクラスラベル $y_i = c$ であるベクトルの集合とする．

データとは，ベクトルの集合とする．

特徴空間とは，特徴ベクトルによって張られる空間のことである．

学習とは分類器が必要とするパラメータをデータから求めることである．

教師データとは，学習に利用する目標(正解)となるベクトルの集合とする．

訓練データとは，学習後に利用する未分類のベクトルの集合とする．

2.2 代表ベクトルからのユークリッド距離による分類

2.2.1 TemplateMatching 法（TM 法）

各クラスのデータが，クラスごとに局所的にまとまったものである場合には，各クラスを 1 つの代表ベクトルであらわすことができると考えられる．具体的には，クラス c に属する教師データから相加平均ベクトル μ_c を生成し，これをクラス c の代表ベクトルとする. 各クラスの代表ベクトルと入力データとのユークリッド距離 D_c を計算し, この距離 D_c がもっとも短くなるクラス c へ分類する. 距離 D_c の計算は以下の式によって計算する[1].

$$D_c(x) = \sqrt{\|x - \mu_c\|^2} = \sqrt{(x - \mu_c)(x - \mu_c)^T}$$

2.2.2 2 次元 3 群データの TM 法による分類の特徴空間　例 1(R.A.Fisher のアヤメのデータ)

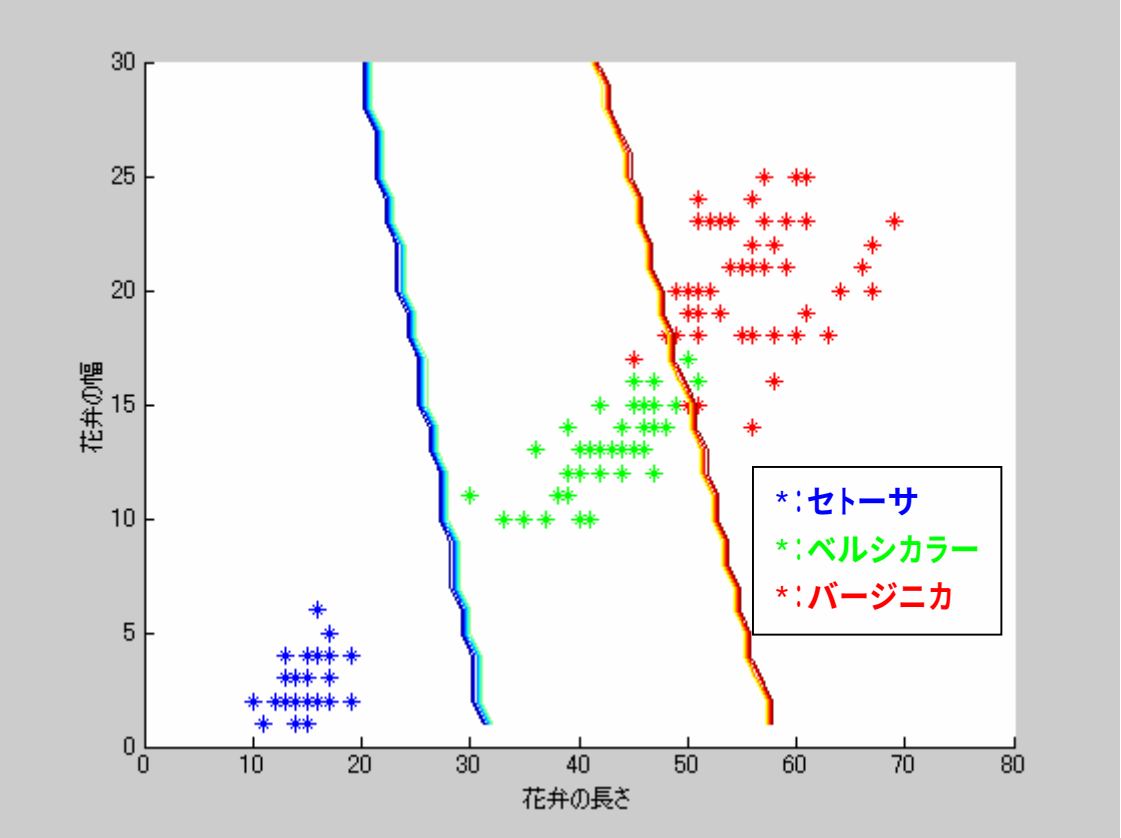


図 2.2

これは，R.A.Fisher のアヤメのデータの花弁の長さと花弁の幅のデータに対して，TM 法を行った場合の特徴空間を表している．青がセトーサ，緑がベルシカラー，赤がバージニカをそれぞれ表している．このように，TM 法は直線的な識別境界線となる．識別境界線の表示方法は 2.5 節参照.

2.2.3 *k*-NearestNeighbor 法

k-nn 法は，TM 法が各クラスの代表ベクトルが一つずつなのに対し，教師データすべてを代表ベクトルとする．具体的には，すべての代表ベクトルと入力データとのユークリッド距離 D_c をそれぞれ計算し，距離が近いものから順に k 個とる．そして，その k 個の中で多数決をとり，一番多かったクラス c へと分類する[1].

2.2.4 2 次元 3 群データの k-nn 法による分類の特徴空間　例 2(R.A.Fisher のアヤメのデータ)

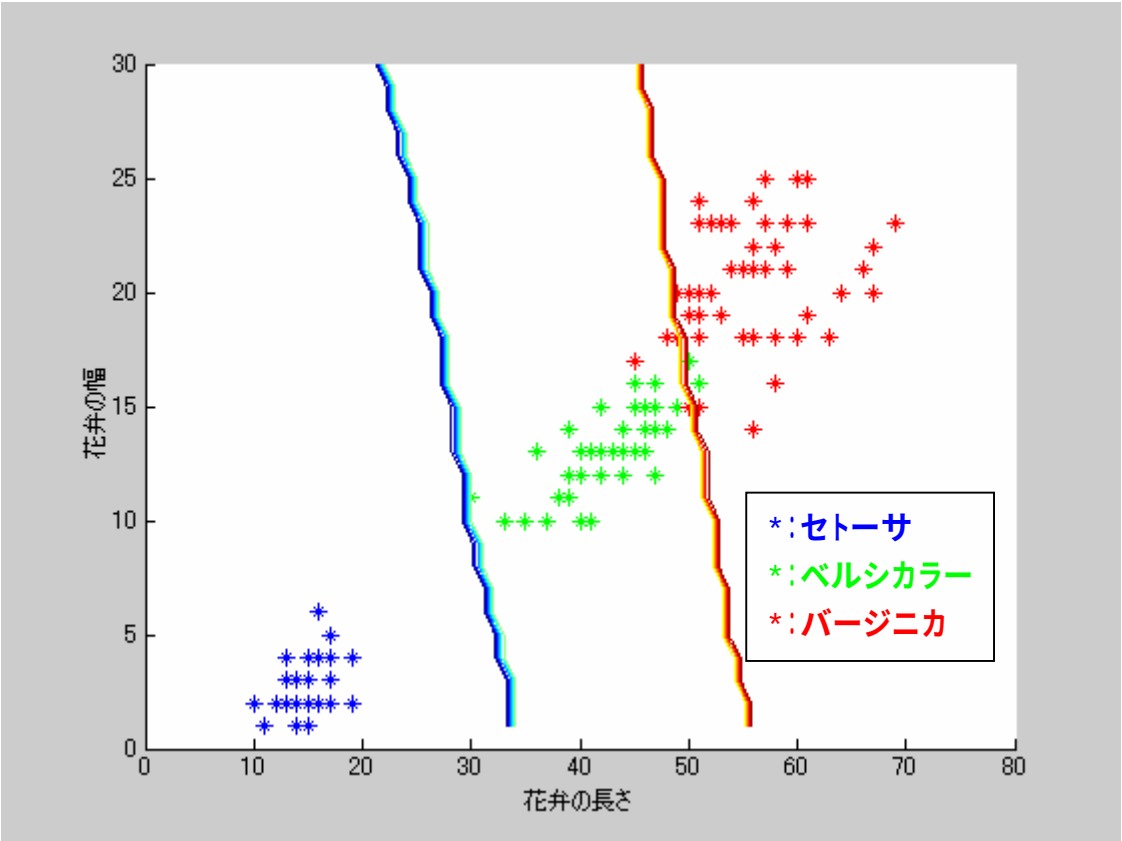


図 2.4

これは，R.A.Fisher のアヤメのデータの花弁の長さと花弁の幅のデータに対して，*k*-nn 法を行った場合の特徴空間を表している．青がセトーサ，緑がベルシカラー，赤がバージニカをそれぞれ表している．図 2.2 と同じように，直線的な識別境界線となる．

2.3 階層型ニューラルネットワーク

2.3.1 ニューラルネットワークとは

ニューラルネットワークとは，生物の脳の神経回路網を模倣した計算メカニズムの総称である．ニューラルネットワークの特徴は以下の 3 つである．

非線形システムである

学習能力を持つ

並列処理システムである

ニューラルネットの第 1 の特徴として「非線形性」が取り上げられる．従来の信号処理，画像処理では，理論的によく体系化されている「線形」の手法がもちいられてきたが．非線形信号処理は理論的に体系化されておらず，使用する対象個別に，場当たり的な工夫に頼らざるを得なかった．ニューラルネットワークは，初めて理論的に体系化され，利用されるようになった非線形近似手法である．

第 2 の特徴として「学習能力」が取り上げられる．学習能力とは，必要とされる機能を，提示される例（訓練）に基づき自動形成する能力のことである．学習能力を有するシステムにおいては，人為的に機構を設計する必要がない．

最後の特徴として「並列性」がある．生体の神経細胞は，トランジスタに比べて数桁速度が遅い演算素子である．それにも関わらず，脳は計算機が数年かけても解くことのできない人工知能の問題を一瞬にして解くことができる．その秘密は，脳の情報処理の並列性にあるといわれている．ニューラルネットは超並列計算機の実現形態の１つであり，他の超並列計算の手法に比べると構造が一様，単純であり，用途が限定されているが，その並列動作を体系的に把握する理論体系が確立されている．

中でも，最も単純なモデルは閾(しきい)素子を用いた神経細胞モデルである[1]

2.3.2 閾(しきい)素子

細胞(素子)への入力ベクトルを $x=(x_1,\cdots,x_N)$ とする時，1つの閾素子の動作は，以下の式で表される．

$$y=H\left[\sum_{i=1}^{N} w_i x_i-\theta\right]$$

w_i は入力要素 x_i にかかる結合の重み，θ は細胞(素子)の閾値と呼ばれる．H[]はしきい関数(ステップ関数)と呼ばれる関数で，[]内の値が正の時には 1，負の時には 0 の値をとる．

従って，この素子は，入力 x_i を w_i で重みをつけて足し合わせた和が，閾値 θ より小さい場合は 0 を出力し，θ を超えると 1 を出力する[2].

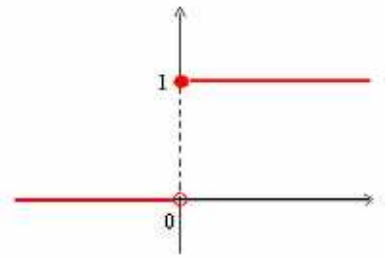
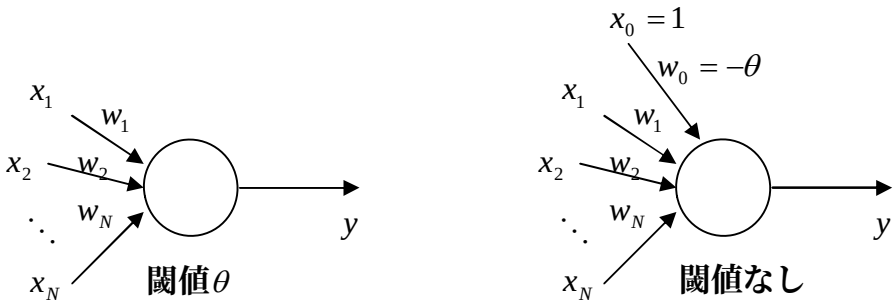


図 3.2.1

次に，しきい素子の図の書き方について説明する．一般的にしきい素子の図は，左の図のように表されるが，本研究では，閾値を結合重みの一つと考えた右図のような形で用いていく．これが等価であるので，問題はない．こうすることで，結合重みを修正するとき，同様に閾値も修正することができ，よりアルゴリズムを簡単にすることができる．



(等価)

図 3.2.2

2.3.3　パーセプトロン

パーセプトロンとは閾素子を階層的に接続したネットワークで，パターン認識を学習させることができる，そのような仕組みの総称である[2].

2.3.4　シグモイド関数

シグモイド関数は，以下の式で表される，単調増加の関数であり，ロジスティック関数とも呼ばれる.

$$sigmoid(z) = \frac{1}{1+\exp(-\alpha z)}$$

これは$z = 0$を境界として立ち上がる関数であり，閾値がθである場合には，この関数をθ右に平行移動して，次式で素子の動作を記述すればよい[1].

$$y = sigmoid(z - \theta)$$

ここで関数中のパラメータαをゲインと呼ぶ．ゲインの大小により，シグモイド関数は図のように形を変える.

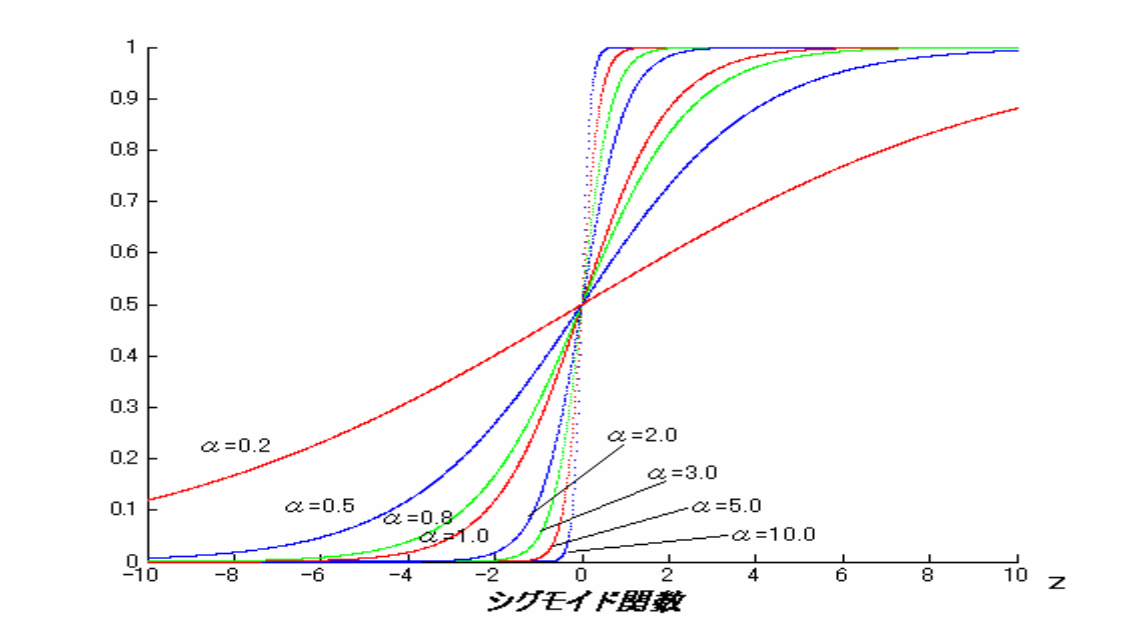


図 3.4

2.3.5 階層型ニューラルネットワーク(多層パーセプトロン)

階層型ニューラルネットワーク(多層パーセプトロン)とは，図のようなネットワーク構造をもつもののことである．

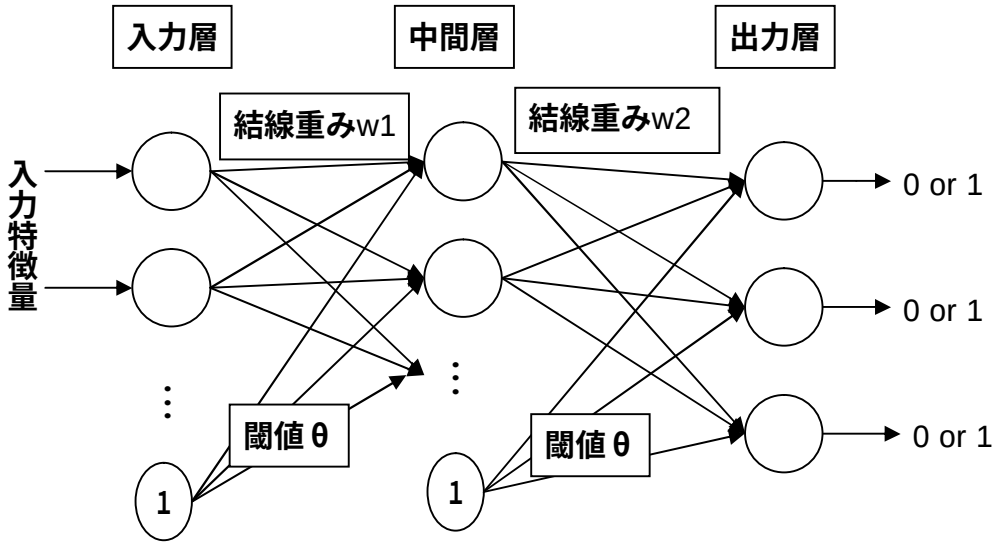


図 3.5

第 1 層は入力特徴量を受け取ってそのまま出力するだけの層であり，入力層と呼ばれる．第 2 層から最終層の 1 つ前の層までを中間層(または隠れ層)，最終層は出力層と呼ばれる．

中間層と出力層の素子では，しきい素子と同様に前層の素子出力の重みつき和を計算した後，しきい関数の代わりに，シグモイド関数を用いて出力を計算する．しきい関数の代わりにシグモイド関数を用いることで，ネットワーク全体の入出力関係を連続で微分可能なものし，最適な結合の重みの探索に，最急降下法などの連続関数の最適化技法を用いることが可能になり，パーセプトロンの学習アルゴリズムでは不可能だった中間層の素子の結合の重みの修正が行えるようになる[1]，閾関数では切捨てであった，閾値θ以下の数値も他の素子に影響を与えられるようになる．

2.3.6 バックプロパゲーション法(ANN-BP 法)

まず，バックプロパゲーション法の簡単な概要を以下に示す.

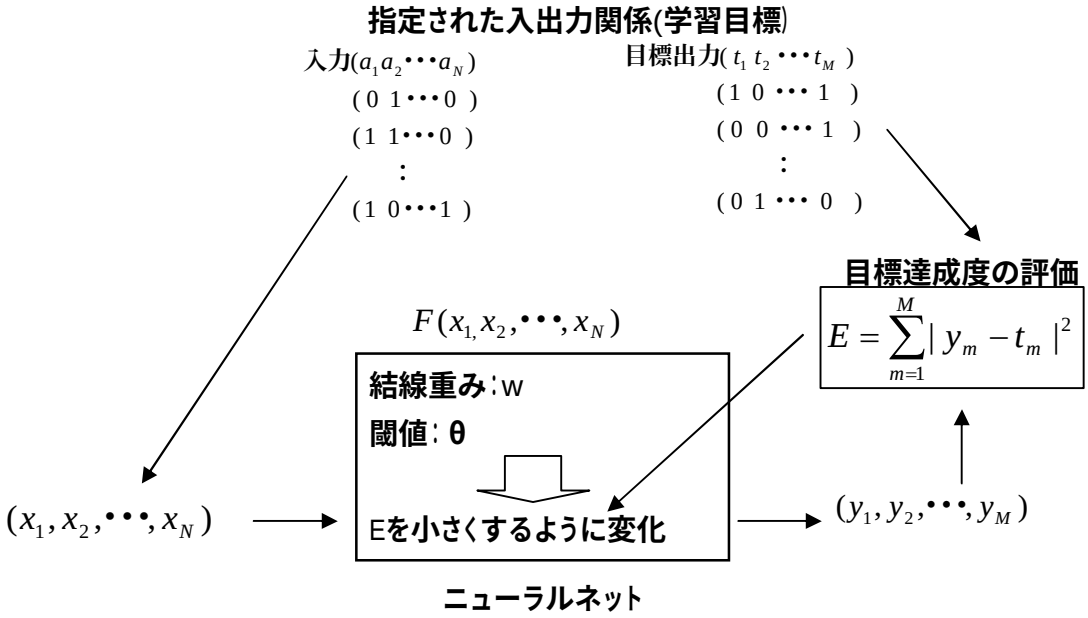


図 3.6

バックプロパゲーション法は，簡単にいえばあらかじめ決まった入出力関係を持つデータを用いて，それらを正しく判別するように，誤差を逆伝播させて，ニューラルネット内部の結線重みと閾値のパラメータを変化させていく方法である[2]. そのため，訓練されていない，入出力関係に用いられず学習させられなかったデータ(想定されていなかったようなデータ)については，うまく判別されるとはいえない（初期条件に依存する）.

ニューラルネットワークのバックプロパゲーション法のアルゴリズムを以下に示す.

(1)入力ベクトル xをネットワークの入力　層にセットし、各層のユニットの出力 yを計算して保存する
(2)出力層の各ユニットで　、入力 x、出力 y、正解 (教師信号) tから $\delta = f'(x)(y-t) = \alpha\, y(1-y)(y-t)$ を計算する
(3)δ を1つ前の層の各ユニット　に重み付きで伝播させ　る。つまり、個々の結合に沿って　、その結合の重みを　δにかけた値を前の層のユニットに戻す
(4)1つ前の層の各ユニット　で、逆伝播された値の　総和をとり，そのユニットの　$f'(x) = \alpha\, y(1-y)$ をかけて、 δ として保存する
(5)ステップ (3),(4)を繰り返して、ネット　ワークの全てのユニッ　ト (入力層は除く)に対して δ を求める
(6)各ユニットの重みベク　トルwを各ユニットの δとそのユニットへの入力ベクトル　xを使って

$$w \leftarrow w - \eta\,\delta\, x$$

のように更新する

入力から各ユニットの出力を計算する順伝播過程に対して，各ユニットのδ を計算する過程は，逆伝播(バックプロパゲーション)と呼ばれる.
学習データ 1 つ受け取るごとに重みを修正してゆくとオンライン学習になる[1].

＊行程(2)において $f'(x) = \alpha\, y(1-y)$ となるのは，以下の式でわかる.

$$s = \sum_{n=0}^{N} w_n x_n$$

$$y = sigmoid(s)$$

$$sigmoid(s) = \frac{1}{1+\exp(-\alpha s)}$$

$$sigmoid'(s) = \frac{\alpha \exp(-\alpha s)}{(1+\exp(-\alpha s))^2}$$

$$\frac{\alpha \exp(-\alpha s)}{(1+\exp(-\alpha s))^2} = \alpha \frac{1}{1+\exp(-\alpha s)}\left(1 - \frac{1}{1+\exp(-\alpha s)}\right)$$

$$= \alpha\, sigmoid(s)\,(1 - sigmoid(s))$$

$$= \alpha\, y\,(1-y)$$

2.3.7　最急降下法

関数 $f(x_1,\cdots,x_N)$ に対し，$f(x)$ を最大化(最小化)するような $x=[x_1,\cdots,x_N]$ を求める方法．最初に $x=[x_1,\cdots,x_N]$ に適当な初期値を設定し，以下のような方法で x を更新していく．

最小化 $x_i^{new} = x_i^{old} - \eta \dfrac{\partial f(x)}{\partial x_i}$

最大化 $x_i^{new} = x_i^{old} + \eta \dfrac{\partial f(x)}{\partial x_i}$

η は正の値であり，一般に $\eta = 0.01$ として用いられることが多い．なお η の値の大きさによって修正の幅が変わる．η が小さいほど，変化が小刻みになる[1].

2.3.8　バックプロパゲーション法による分類の特徴空間　例 3(R.A.Fisher のアヤメのデータ)

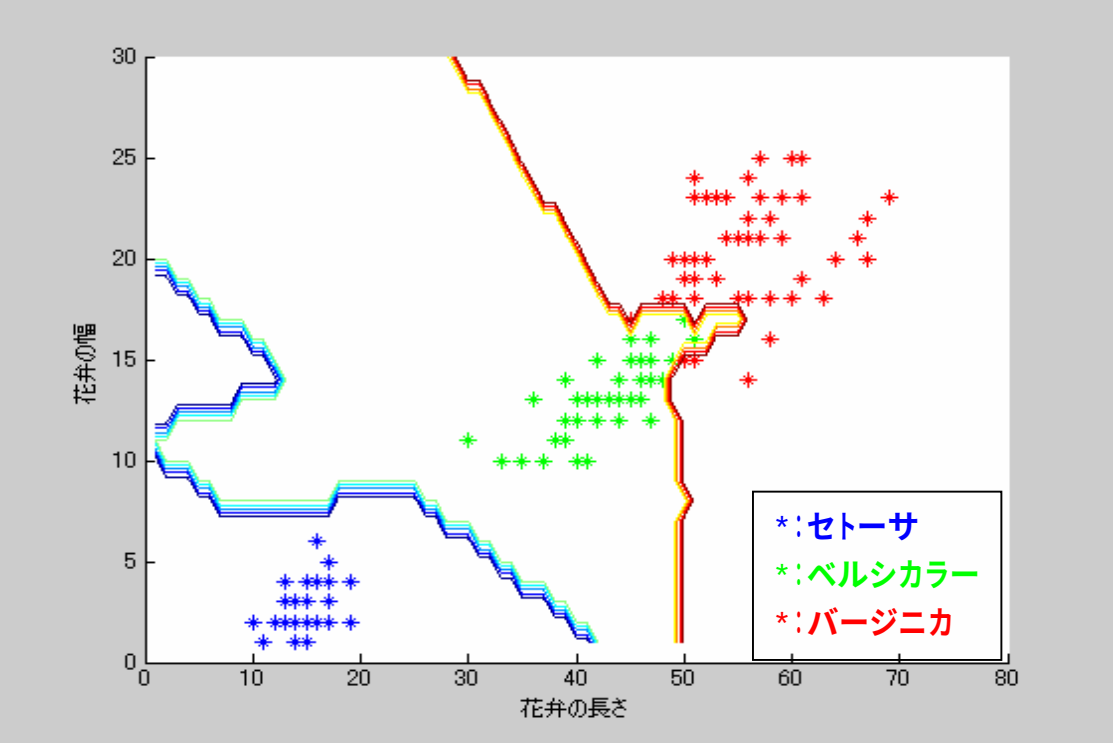


図 3.8

これは，R.A.Fisher のアヤメのデータの花弁の長さ，花弁の幅に対して，ニューラルネットワークのバックプロパゲーション法を用いた場合の特徴空間である．青がセトーサ，緑がベルシカラー，赤がバージニカをそれぞれ表している．このように, 図 2.2 や図 2.4 と比較すると, 自由で柔軟な境界線が描かれている. また，セトーサとベルシカラーの境界線については，バックプロパゲーション法の識別境界線が分布を仮定しているものではないので，教師データの存在しない範囲に関しては予測不可能な結果となった.

2.4 決定木

2.4.1 数値属性の離散化

二分木を構築する上で，まず問題となるのが数値属性の扱いである．連続値をとる数値属性をそのまま決定木学習に利用すれば，分岐数が多くなりすぎてしまい，決定木を構築するのが困難になり，また，多大な学習時間を要してしまうなどの問題が生じる．そのため，数値属性内に境界値（分割点）を定め，その境界値によって区切られた区間を離散値（記号）に置き換える処理を行わなければいけない．この処理が数値属性の離散化である．

今回の研究では，この数値属性の離散化を，最適サポートルールによって行った[3].

2.4.2 相関ルール

最適サポートルールは，相関ルールを利用している．相関ルールとはデータマイニングにおける技術の一つで，相関ルールの基本概念を説明すると，

まず，個々のアイテムをi_kとすると，全アイテムの集合は$I=\{i_1,i_2,...i_n\}$とする．

次に，各トランザクションをTとすると，Tはアイテムの集合であり，Iに含まれる$(T\subseteq I)$．そして，データベースをDとすると，Dはトランザクションの集合になる．

以上の概念を用いると，相関ルールは次のように表現できる．

（定義）相関ルール

$$A\Rightarrow B$$

ここで，$A,B\subset I$かつ$A\cap B=\phi$である．

次に，相関ルールに付随する，サポート（support,支持度）とコンフィデンス（confeden,確信度）という概念を定義する．サポートsとはデータベースDにおいてアイテム集合AとBをともに含む$(A\cup B)$トランザクションの割合である．一方，コンフィデンスcはDにおいてAを含むトランザクションにおける，Bを含むトランザクションの割合である．ここで，確率Pを用いれば，サポートとコンフィデンスは次のようにいいかえられる．

（定義）相関ルールの support と conf

$$\mathrm{support}(A\Rightarrow B)\equiv P(A\cup B)$$

$$\mathrm{conf}(A\Rightarrow B)\equiv P(B|A)$$

ここに，$P(B|A)$は条件付き確率を表す．すなわち，Aが起こったという付帯条件のもとで，Bが起こる確率である[3].

2.4.3 バケット分割

最適サポートルールを適応させるときの前処理として，バケット分割が必要となる．

与えられたリレーション R のあるタプルを t とし，そのタプルの数値属性 A の値を $t[A]$ と表記する．属性 A の定義域を，次のような交わりのないバケットの列

$$B_1, B_2, ..., B_M$$
$$(B_i = [x_i, y_i], \quad x_i \le y_i < x_{i+1})$$

に分割し，すべてのタプルの属性 A の値が必ずどれかのバケットに入るようにする．このように分割すると，任意のタプル $t \in R$ に対して，$t[A]$ を含む，あるバケット B_j がただ一つ存在することになる[3].

ここで，集合 $\{t \in R | t[A] \in B_i\}$ に入るタプル数を B_i の大きさ（size）とよび u_i と表す．また，集合 $\{t \in R | t[A] \in B_i, t$ は C を満たす$\}$ に入るタプル数を v_i とし，タプルの総数を N とすれば，ルール $(A \in [x_s, y_t]) \Rightarrow C$ の確信度は $(\sum_{i=s}^{t} v_i)/(\sum_{i=s}^{t} u_i)$ となり，$A \in [x_s, y_t]$ のサポートは $\sum_{i=s}^{t} u_i / N$ となる．

また，バケットの列 $B_1, B_2, ..., B_M$ が与えられたとき，次のような形のルール

$$(A \in [x_s, y_t]) \Rightarrow C$$

を生成することを考える．ここで，$[x_x, y_t]$ は連続するバケット $B_s, B_{s+1}, ..., B_t$ を連結したものである．任意の区間 $[x_x, y_t]$ がバケットのインデックスの対 $(s \le t)$ で特定できる．簡単のために，区間 $[x_x, y_t]$ を $[s,t]$ と表記する．また，$\mathrm{support}(A \in [x_s, y_t])$ は $\mathrm{support}(s,t)$ と，$\mathrm{conf}(A \in [x_s, y_t] \Rightarrow C)$ は $\mathrm{conf}(s,t)$ と，表記する．

2.4.4 最適サポートルール

$(A \in I) \Rightarrow C$ という形のルールに注目すると（I は数値属性 A の定義域中の区間）．確信度が，与えられた最小確信度 *minconf* より小さくない相関ルールをつくる区間 I のうち，サポート $\mathrm{support}(A \in I)$ を最大とするものを最適サポート区間とよび，それを使った相関ルールを最適サポートルールという[3].

主な流れは,

① $minconf < \mathrm{conf}(s,t)$ となる (s,t) を求める

② ①で求めた (s,t) の範囲内で，$\mathrm{support}(s,t)$ を最大とするもの（最適サポート対）を求める．

まず，①についてのアルゴリズムを説明する．

すべての $j < s$ に対して $\mathrm{conf}(j, s-1) <$ *minconf* であるようなインデックス s を有効であるとする．すると次のような補助定理が成り立つ．

（補助定理 1） もし (s,t) が最適サポート対ならば，s は有効である．

（証明）s が有効であると仮定すると，$\mathrm{conf}(j, s-1) \geq$ *minconf* とする j が存在する．(s,t) は最適サポート対だから $\mathrm{conf}(s,t) \geq minconf$ である．よって，$\mathrm{conf}(j,t) \geq minconf$. $\mathrm{support}(j,t) > \mathrm{support}(s,t)$ であるから，これは最適サポート対の定義（$\mathrm{support}(s,t)$ が最大）と矛盾する．

このことから，まず有効であるインデックスすべてを探し，そのなかから最適なものを選べばよいことがわかる．

$$w(s) = \max_{j<s}\left\{\sum_{i=j}^{s-1}(v_i - minconf \times u_i)\right\}$$

と定義すると，s が有効なら $w(s) < 0$ である．

$w(s) = v_{s-1} - minconf \times u_i + \max\{0, w(s-1)\}$ であるから，①は次のアルゴリズムで表すことができる．

（アルゴリズム 1）

1）1 は有効である；

2） $w := 0;$

3）for $s := 2,...,M\{$

4） $w := v_{s-1} - u_{s-1} \times minconf + \max\{0, w\}$ ；

5） $\mathrm{if}(w > 0)\{$

6） s は有効である；

7)　}

8) }

次に②について説明する．

$\mathrm{conf}(s,t) \geq minconf$とする最大のインデックス$t$を$\mathrm{top}(s)$と表すことにすると，後は$\sum_{i=s}^{top(s)} u_i$を最大とする$s$を求めればよいのだが，次の性質を利用するとうまく計算できる．

（補助定理 2）　s, s'がともに有効で$s < s'$ならば，$\mathrm{top}(s) \leq \mathrm{top}(s')$である．

（証明）s'が有効であるから，$\mathrm{conf}(s, s'-1) <$ *minconf*．$\mathrm{top}(s')$の定義から，$\mathrm{conf}(s, \mathrm{top}(s')) \geq$ *minconf* である．したがって，$\mathrm{conf}(s', \mathrm{top}(s)) \geq$ *minconf*．このことから，$\mathrm{top}(s) \leq \mathrm{top}(s')$となる．

この性質から，有効なインデックスのリスト$(s_1,...,s_q)$と全インデックスのリスト$(1,...,M)$を用意しておけば，これらのリストを逆方向に交互にスキャンして，$\mathrm{top}(s_i)$を見つけることができる．

②のアルゴリズムは次に示す．

（アルゴリズム 2）

1）$i := M$ ；

2）for $j := q,...,1${

3）　if $(\mathrm{conf}(s_j, i) < minconf)$

4)　　$i := i - 1$ ；

5)　else{

6)　　$\mathrm{top}(s_j) := i$ ；

7)　　$j := j - 1$ ；

8)　}

9) }

2.4.5 最適な変数の選択

決定木を構築するうえで，数値属性の離散化の次に必要となるのが，最適な変数の選択である．データを分割するためには，どの変数で分割するかを選択しなければいけないのだが，そのとき，どの変数が効率よくデータを分割できるかを事前に知ることができれば，より良い決定木を構築することができる．

2.4.6 ID3

最適な変数の選択を行うのに ID3 を利用した．

ID3 は情報量（エントロピー）を利用している．情報量は，情報の不確からしさを表しており，各変数の相互情報量（分割前の情報量－分割後の情報量）を求め，その相互情報量が一番大きい値となった変数を選択する．

情報量は次式で求められる．

$$H(X) = -\sum_{j=1}^{k} p_j \log_k p_j$$

ここで，p_j は X の k 種類ある事象 $a_1,...,a_k$ のうち，事象 $a_j (1 \le j \le k)$ の出現率

とする．ただし，$a_j \cap a_j = \phi (i \ne j)$ であるとする[3].

2.4.7 二分木

最適サポートルールと ID3 を使って二分木を構築する[3].

（アルゴリズム 3）

main( )

① 学習データ D を読み込む

② tree(D)

tree(データ集合 D)

① if 終了条件 then 終了

② 各数値属性に対し境界値を求める（最適サポートルール）

③ 最適な変数を選択する（ID3）

④ ③で選んだ変数について②で求めた境界値で，D を D_1，D_2 に分割する．

⑤ tree(D_1)

⑥ tree(D_2)

2.4.8　2 次元 3 群データの二分木による分類の特徴空間　例 4(R.A.Fisher のアヤメのデータ)

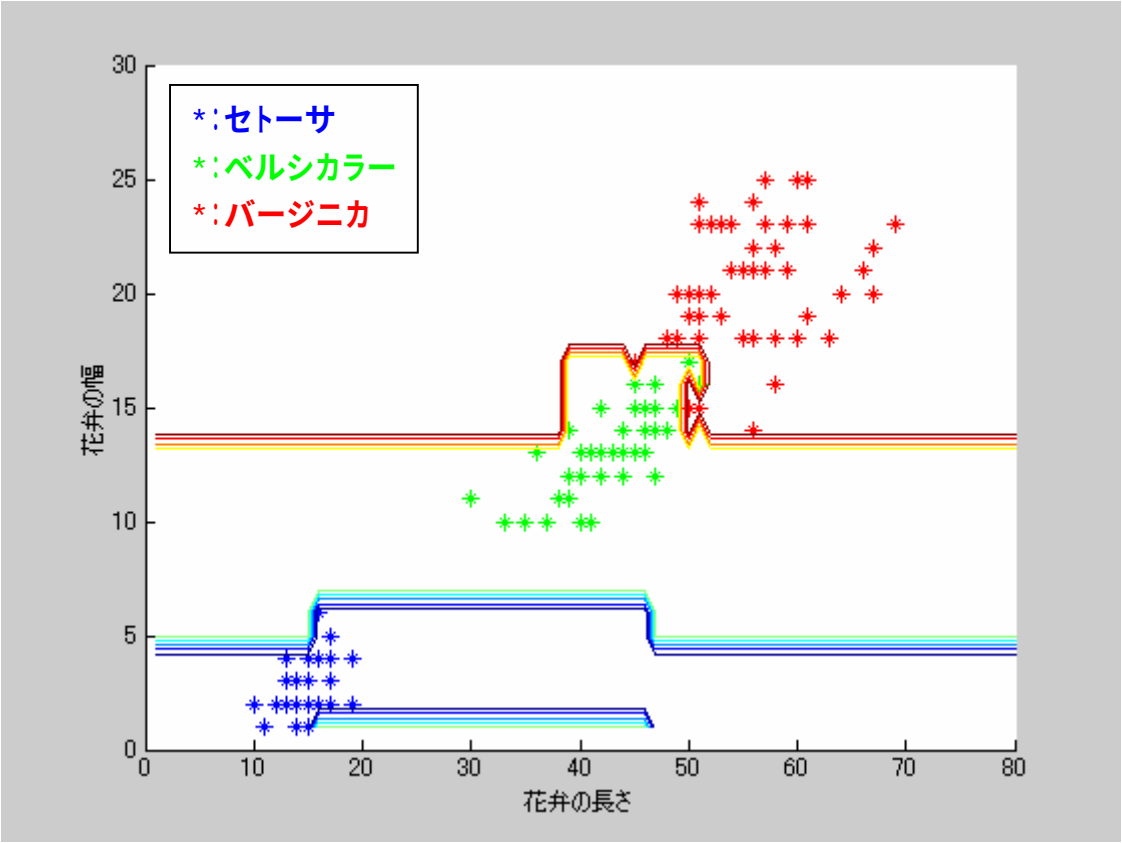


図 4.8

これは，R.A.Fisher のアヤメのデータの花弁の長さ，花弁の幅に対して，二分木を用いた場合の特徴空間である．青がセトーサ，緑がベルシカラー，赤がバージニカをそれぞれ表している．このように，図 3.8 と同様，特徴的な識別境界線となったが，バックプロパゲーション法とはまた違った識別境界線が描かれている．

2.5　分類識別境界線の表示

境界の関数型が陰関数等の場合，陽関数の場合と同じように方程式で求めることは難しい．そのため，本研究においては，特徴空間の領域全体を総当りでクラスを求め，その結果を用いて等高線で境界線を表示した．
領域全体にクラスを割り当て，等高線を引いた簡単な例を以下に図示する．

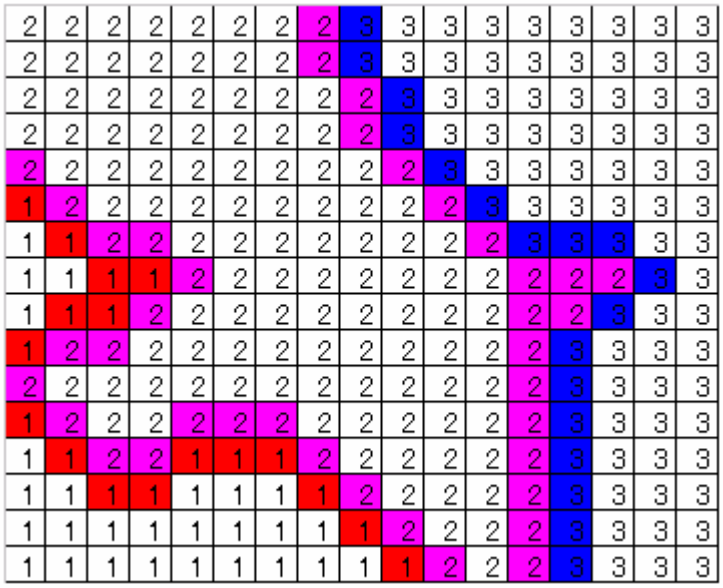


図 5

上図のように領域全体にクラスを割り当て，そのデータをもとに等高線を引く．それを 3 クラスの境界線として表示する．

第3章　実験環境

3.1 動作コンピュータ

CPU	Pentium(R)4　3.20GHz
メモリ	512MB
OS	Windows XP sp2
使用ソフト	Matlab6

3.2 Matlab について

＜発展経過＞

MATLAB は 1980 年に Cleve Moler によって開発された．Cleve Moler は，もともと行列計算の専門家で，1970 年代に米国の学者・研究者による国家的プロジェクト NATS（National Activity for Test of Software）のもとで固有値計算ライブラリ EISPACK と連立 1 次方程式ライブラリ LINPACK の開発を中心的に果たした．この当時は，ニューメキシコ大学の教授であった． EISPACK も LINPACK も Fortran 言語を用いたサブルーチン集で，コンピュータに不慣れな一般の研究者にとってはあまり使いやすいものではなかった．そこで，1980 年頃，Fortran 言語を知らない人でも行列計算が対話的にできるようにという目的で，MATLAB（Matrix Laboratory）という言語を，Fortran 言語を用いて試作した． そして，この MATLAB を何人かのエンジニアが制御関係の分野に適用して成功したため，1984 年に MATLAB を C 言語化して機能を大幅に拡張して製品化し，同時に MathWorks 社を設立し，Moler はニューメキシコ大学を辞任した．

＜特徴＞

Matlab は行列を扱うことのできるプログラム言語であり，行列やベクトルを扱う関数・計算を容易に記述できる．また行列を基にグラフの作成と表示ができる．

また，MATLAB を用いると，C 言語や FORTRAN といった従来のプログラミング言語よりも短時間で簡単に科学技術計算を行うことができる．

3.3 データセット

教師データとして,代表的なベンチマークセットの R.A.Fisher のアヤメのデータセット[4]と WDBC（Wisconsin Diagnostic Breast Cancer）のデータセット[5]を用いた.

アヤメのデータは，アヤメの花の花弁とがくの大きさをデータとしたもので，WDBC のデータはウィスコンシン州の乳癌患者の診断結果をデータとしたものである．また，30 変量については，radius，texture，perimeter，area，smoothness，compactness，concavity，concave points，symmetry，fractal dimension の 10 項目について，それぞれ平均，標準誤差，最大値の 3 種類のデータが存在しており，3×10 の 30 変量となる．データとしては，列の 3～12 が 10 項目の平均，13～22 が標準誤差，23～32 となる．また，1 は ID（今回は未使用），2 は教師信号（悪性 or 良性）となっている.

○ R.A.Fisher のアヤメのデータ

- セトーサ，ベルシカラー，バージニカの３群
- がくの長さ，幅，花弁の長さ，幅の４変量
- データ数　150

○ WDBC のデータ

- 悪性，良性の２群
- 腫瘍に関する項目　30 変量
- データ数　569

第4章　分類実験

4.1 実験方法

機械学習手法は

・TM 法

・*k*-nn 法

・ANN-BP 法

・二分木（最適サポートルール＋ID3）

を分類データ

・R.A.Fisher のアヤメのデータ

・WDBC のデータ

について行う.

＜実験時の注意事項＞

・TM 法

特になし

・*k*-nn 法

k=10 を与えて学習を行った.

・ANN-BP 法

R.A.Fisher のアヤメのデータ，WDBC のデータはともにデータの規準化をおこない，変量の変動の統一化を持って，どのようなデータにも対応できるニューラルネットワークを構成する.

結線重み w, v, 閾値 θ の初期値は，一様乱数で 0〜1 の実数を与えた．また，シグモイド関数のゲイン α，学習係数 η，中間層のユニット（素子）数の値は学習を行う上で，随時変更を重ねた.

また，ANN-BP 法は学習を重ねる（学習のループ回数を増やす）ことによって誤り率を 0 に収束させることが可能だが，局所解が存在し，学習に要する時間も大きくかかってしまうため，今回の実験ではループ回数を 10 万回こえる場合はそれ以上実験を行っていない.

・二分木（最適サポートルール＋ID3）

minconf=1.0 を与えて実験を行った.

4.2 実験結果

4.2.1 教師データについての実験結果

表 1.1，表 1.2，表 1.3，表 1.4 は R.A.Fisher のアヤメのデータ（4 変量，3 群，データ数 150）を教師データとして与え，その教師データについて，各機械学習手法別の学習結果を分割表にしたものである．

表 1.1　TM 法によるアヤメのデータの分割表

事前 / 事後	セトーサ	ベルシカラー	バージニカ	
セトーサ	50	0	0	50
ベルシカラー	0	48	4	52
バージニカ	0	2	46	48
	50	48	46	144/150

ベルシカラー，バージニカについて分類の誤りが生じた．

表 1.2　*k*-nn 法によるアヤメのデータの分割表

事前 / 事後	セトーサ	ベルシカラー	バージニカ	
セトーサ	50	0	0	50
ベルシカラー	0	47	1	48
バージニカ	0	3	49	52
	50	47	49	146/150

TM 法より分類の誤りの数は少ない．

表 1.3　ANN-BP 法によるアヤメのデータの分割表

事前 / 事後	セトーサ	ベルシカラー	バージニカ	
セトーサ	50	0	0	50
ベルシカラー	0	50	0	50
バージニカ	0	0	50	50
	50	50	50	150/150

すべて間違えることなく分類できた.

表 1.4　二分木によるアヤメのデータの分割表

事前 / 事後	セトーサ	ベルシカラー	バージニカ	
セトーサ	50	0	0	50
ベルシカラー	0	50	0	50
バージニカ	0	0	50	50
	50	50	50	150/150

ANN-BP 法と同様すべて間違えることなく分類できた.

表 1.5, 表 1.6, 表 1.7, 表 1.8 は WDBC のデータ（30 変量, 2 群, データ数 569）を教師データとして与え，その教師データについて，各機械学習手法別の学習結果を分割表にしたものである.

表 1.5　TM 法による WDBC のデータの分割表

事後＼事前	悪性	良性	
悪性	114	89	203
良性	98	268	366
	114	268	382/569

アヤメのデータと比較しても，誤った分類の数がかなり大きくなった.

表 1.6　*k*-nn 法による WDBC のデータの分割表

事後＼事前	悪性	良性	
悪性	189	12	201
良性	23	345	368
	189	345	534/569

誤った分類の数は TM 法と比べるとかなり少ない結果となったが，ANN-BP 法と二分木に比べると大きい.

表 1.7　ANN-BP 法による WDBC のデータの分割表

事前 / 事後	悪性	良性	
悪性	211	0	211
良性	1	357	358
	211	357	568/569

誤った分類の数が 1 つとなり，ほぼすべて分類できた.

表 1.8　ANN-BP 法による WDBC のデータの分割表

事前 / 事後	悪性	良性	
悪性	212	0	212
良性	0	357	357
	212	357	569/569

すべて間違えることなく分類することができた.

表 1.9 はアヤメのデータに関して，1〜4 変量を与えたときの手法別の誤り率（教師データの誤り分類数 / 教師データ数）を表している．

また，表 1.10 は WDBC のデータに関して，1〜4 変量を与えたときの手法別の誤り率（教師データの誤り分類数 / 教師データ数）を表している．

表 1.9　アヤメのデータについての手法別誤り率

手法 / データ	TM	*k*-nn (*k* =10)	ANN-BP 法	二分木
1変量 (花弁長)	0.053	0.047	0.047	0.047
2変量 (花弁長, 幅)	0.04	0.033	0.007	0.007
3変量 (がく長, 幅, 花弁長)	0.053	0.04	0	0
4変量すべて	0.04	0.04	0	0

TM 法と *k*-nn 法については，誤り率が高い結果となった．

また，ANN-BP 法と二分木については，１，２変量のとき，データの中に同じ値で重なっているものがあったため，これ以上分類を行うことができなかった．よって，ANN-BP 法と二分木については重なっているデータ以外は全て分類できたことになる．

表 1.10　WDBC のデータについての手法別誤り率

手法 / データ	TM 法	*k*-nn 法 (*k* =10)	ANN-BP 法	二分木
2 変量 (3, 4)	0.269	0.137	0.056	0
10 変量 (3〜12)	0.517	0.137	0.009	0
30 変量すべて	0.329	0.062	0.002	0

TM 法についてはかなり誤り率が高い結果となった．*k*-nn 法は TM 法ほどではないが，比較的高い結果となった．

また，ANN-BP 法はループ回数を増やせば 0 に収束させることができるが，局所解が存在しており，学習にかかる時間も大きくなってしまうため，今回の実験ではここまでの結果となった．二分木については全て分類することができた．

4.2.2 試験データについての実験結果

データセットの 75%を教師データとして与え，残りの 25%を未分類の試験データとして与えた．（表 2.1 がデータセットの分割表）

そして，その試験データについての誤り率を機会手法別に表したものが表 2.2 となる．

また，表 2.3 は先行研究の試験データについての機械手法別誤り率である．

表 2.1　データセット分割表

	全データ数	教師データ数 (75%)	試験データ数 (25%)
アヤメのデータ	150	112	38
WDBC のデータ	569	426	143

この教師データの与え方は，先行研究[6]と比較を行うために，先行研究と同じデータの与え方を行った．

表 2.2　試験データについての手法別誤り率

手法 / データ	TM 法	*k*-nn 法 (*k* =10)	ANN-BP 法	二分木
アヤメのデータ	0.026	0.053	0.053	0.079
WDBC のデータ	0.343	0.056	0.042	0.07

TM 法は，複雑でないアヤメのデータに対しては，誤り率が低いのに対して，比較的複雑な WDBC のデータについては誤り率がかなり大きくなった．

また，*k*-nn 法と ANN-BP 法が，誤り率が低くなったのに対して，二分木は比較的誤り率が高い結果となった．

表 2.3　先行研究の試験データについての手法別誤り率

手法 / データ	ECNN	ENN	MLP	PSO	KSTAR	MB	VFI
アヤメのデータ	0.032	0.026	0.026	0.026	0.053	0.079	0.079
WDBC のデータ	0.015	0.018	0.021	0.057	0.056	0.028	0.056

機械学習手法の詳細については，先行研究[6]の内容を参照.

表 2.2 と比較すると, ECNN, ENN, MLP に対しては, TM 法, *k*-nn 法, ANN-BP 法，二分木が劣っているのは明らかではあるが，それ以外の手法に対しては，TM 法以外の 3 つの手法はあまり大差がない.

4.2.3　学習時の時間計算量

表 3.1 は機械学習手法別の学習時の時間計算量を表している．このとき，ANN-BP 法については，学習時のループ回数によって学習時間が大幅に変わってくるので，*R*＝ループ回数とした.

表 3.1　手法別学習時の時間計算量

	TM 法	*k*-nn 法	ANN-BP 法	二分木
時間計算量	$O(n)$	$O(n^2)$	$R \times O(n)$	$O(n \log n)$

リアルタイムで学習することを考慮した場合，*k*-nn 法は学習に時間がかかりすぎてしまう.

ANN-BP 法については, R の値は実際では 1 万～10 万といった大きな値となるため，$R >> n$ となる. よって，ANN-BP 法も学習に時間がかかりすぎてしまう.

また, 先行研究[6]の学習時の時間計算量については, 表記はなかったが, ECNN, ENN, MLP, PSO に関してはニューラルネットワークを利用した手法であるため, 学習時間はループ回数によって大幅に変わってくる．先行研究[6]では，上記の 4 つの手法については，実験時間を約 50 分としている．よって，リアルタイムで学習することを考慮した場合では, 学習に時間がかかりすぎてしまう. また, KSTAR, MB，VFI の時間計算量は先行研究[7],[8],[9]を調べたが不明である.

第5章　考察とまとめ

5.1 考察

以上の実験結果を表にまとめたものが，表 5.1 となる．

表 5.1　まとめ

	TM 法	*k*-nn 法	ANN-BP 法	二分木
教師データ	×	△	○	○
試験データ	×	○	○	△
時間計算量	◎	△	×	○

総合的に評価すると，比較的少ない時間で学習ができ，教師データ，試験データともに誤り率も低い結果となった二分木が一番学習の効果がよいと言える．

次に，誤り率のみに注目すると，ANN-BP 法が，一番誤り率が低い結果となった．よって，事前に学習を行うことができる状況では ANN-BP 法が一番学習の効果が良いと言える．

5.2 まとめ

今回の実験では，教師データとして線形データのみを利用したため，二分木が一番学習の効果良い結果となったが，非線形データで比較を行った場合では，おそらく二分木による学習後の誤り率は高くなるだろうと考えられる．よって，非線形データでも分類が可能である ANN-BP 法がさらに良い結果になると考えられる．

よって，今後の課題としては，非線形データを教師データとして与えたときの比較を行いたい．

また，SVM 等のその他の代表的な機会学習手法についても，実装，比較を行って行きたい．

第6章　謝辞

最後に，本研究を進めるにあたり，ゼミを中心に温かく熱心にご指導，ご助言を頂ました田中章司郎教授に深く感謝の意を表すとともに心より御礼申し上げます．また，同じ学部生である木村さん，的場さん，加藤さんにもいろいろとご協力，ご助言いただいたことに御礼申し上げます．なお，本研究で作成したプログラム，発表資料等のすべての著作権を田中章司郎教授に譲渡いたします．

参考文献

[1]麻生秀樹，津田直治，村田昇：統計科学のフロンティア 6　パターン認識と学習の統計学(岩波書店 2003).
[2]熊沢逸夫　：電子情報通信工学シリーズ　学習とニューラルネットワーク(森北出版 1998).
[3]福田剛志：数値属性の最適結合ルールを発見する効率的アルゴリズム,情報処理学会論文誌, 37(6) 945-953 (1996).
[4]UCI Machine Learning Repository：Iris Data Set
http://archive.ics.uci.edu/ml/datasets/Iris.
[5]UCI Machine Learning Repository：Breast Cancer Wisconsin (Diagnostic) Data Set
http://archive.ics.uci.edu/ml/datasets/Breast Cancer Wisconsin (Diagnostic).
[6]Mauro Annunziato, Ilaria Bertini, Matteo De Felice, Stefano Pizzuti：Evolutionary Complex Neural Networks(2007).
[7]Cleary.j.G,Trigg.L.E.：K*: An Instance- based Learner Using an Entropic Distance Measure,Proceedings of the 12th International Conference on Machine learning, 108-114(2005).
[8] Webb.G. I.： MultiBoosting: a technique for combining boosting and wagging, Machine Learning, vol. 40(2), 159-196(2000).
[9] Demiroz. G, Guvenir. A.： Classification by voting feature intervals, ECML-97(1997).