

ベイズ判別法とニューラルネットワークを用いた画像分類手法の比較とその高速化

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

S023060 伊達 虎三

目次

第 1 章 序論	4
1.1 研究の背景	4
1.2 研究の概要	5
第 2 章 分類手法とその計算実現	5
2.1 用語説明	5
2.2 事後確率最大化識別(ベイズ識別)	5
2.2.1 ベイズの定理	5
2.2.2 事後確率最大化識別(ベイズ識別)	5
2.2.3 多変量正規分布	6
2.2.4 ベイズ識別の実際の計算	6
2.2.5 2次元3群のベイズ識別の特徴空間	7
2.3 階層型ニューラルネットワーク	8
2.3.1 ニューラルネットワークとは	8
2.3.2 閾(しきい)素子	9
2.3.3 パーセプトロン	9
2.3.4 シグモイド関数	10
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)	11
2.3.6 バックプロパゲーション法(誤差逆伝播アルゴリズム)	12
2.3.7 最急降下法	14
2.3.8 ニューラルネットによる分類の特徴空間 例 2 (R.A.Fisher のアヤマメデータ)	14
2.4 分類識別境界線の表示方法	15
2.4.1 境界の関数型が既知であり陽関数の場合	15
2.4.2 境界の関数型が陰関数等の場合	17
第 3 章 実験環境	18
3.1 動作コンピュータ	18
3.2 Matlab について	18
3.3 分類するテストデータ	18
第 4 章 Landsat 画像データの分類における行列を用いた計算の高速化	19
4.1 高速化する前の方法(ベイズ識別)	19
4.2 高速化した方法(ベイズ識別)	22
4.3 ニューラルネットワークにおける高速化	27
第 5 章 分類実験	29
5.1 実験方法	29
5.2 実験結果	31

第6章 考察とまとめ.....	38
6.1 考察.....	38
6.2 まとめ.....	39
第7章 謝辞.....	40
第8章 参考文献.....	40

第1章 序論

1.1 研究の背景

人間の認識能力を、コンピュータにおいて実現することは困難なことである。本研究の最終目標であるコンピュータに人間の顔表情を上手に認識させるためには、分類を上手にさせる必要がある。そのためには、分類の基本原理を理解し、数多く存在する分類手法の特徴を理解し、自分のものとすることが重要である。本研究は、自らの手で分類手法を実データを用いて実装・検証することで得た経験と知識を今後の研究のための足がかりにしていくことを目的としたものである。

代表的な分類手法として、古典的な分類手法の事後確率最大化識別(ベイズ識別)や、比較的新しい人工ニューラルネットワークのバックプロパゲーション法による方法を取り上げる。

1.2 研究の概要

分類の代表的な手法である、ベイズ識別と比較的新しい分類手法である階層型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利用した分類を実装し、その分類の正確さを比較した。また、ニューラルネットワークのバックプロパゲーション法については、初期決定する諸条件が学習時の収束速度にかなりの影響を及ぼすため、中間層のユニット数、シグモイド関数のゲイン、学習係数について変化させ、収束速度(学習回数)を比較検討した。

具体的には、Landsat の実画像データについてベイズ識別および階層型ニューラルネットワークのバックプロパゲーション法によって分類を行った。また、このときの対照のデータとしては、R.A.Fisher のアヤメのデータを用いて、実験を行った。最後に、画像分類を行う場合の、Matlab の行列計算機能を用いた画像分類の高速化アルゴリズムを実現した。

第2章 分類手法

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。 ($i = 1, 2, \dots, N$)

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、学習に利用するベクトルの集合とする。

x^T とは行ベクトル x の転置である。

2.2 事後確率最大化識別(ベイズ識別)

2.2.1 ベイズの定理

事象 A_1, \dots, A_K を互いに排反な全事象の分割とする。このとき、任意の事象 B に対して、

$$p(A_i | B) = \frac{p(B | A_i)p(A_i)}{\sum_{i=1}^K p(B | A_i)p(A_i)}$$

となる。

2.2.2 事後確率最大化識別(ベイズ識別)

$p(c)$ クラス c の事前確率

$p(c | x)$ クラス c の事後確率

$p(x | c)$ クラス c における特徴ベクトル x の確率

クラス c の個数 $|c| = K$

確率 $p(c)$ および $p(x | c)$ が既知であれば、 $p(c | x)$ はベイズの定理を用いて以下のように求めることができる。

$$p(c | x) = \frac{p(x | c)p(c)}{\sum_{c=1}^K p(x | c)p(c)}$$

事後確率 $p(c | x)$ を最大にする c を選択する判別方式を事後確率最大化識別(ベイズ識別)という[1]。

ただし、真の事前確率 $p(c)$ および $p(x|c)$ は一般に不明であり、用いるデータから推定する必要がある。次節で $p(x|c)$ の推定方法を説明する。

2.2.3 多変量正規分布

クラス c であるベクトル x がクラスごとに局所的にまとまった分布になってと考えられる実測データを用いる場合には、その分布 $p(x|c)$ を多変量正規分布によってモデル化することは適切であると考えられる。以下の式が、多変量正規分布の式である[1]

$$p(x|c) = \frac{1}{(2\pi)^{N/2} |\Sigma_c|^{1/2}} \exp\left\{-\frac{1}{2}(x - \mu_c)\Sigma_c^{-1}(x - \mu_c)^T\right\}$$

N ・・・ベクトル x の変量数

Σ_c ・・・クラス c の分散共分散行列

$|\Sigma|$ ・・・ Σ の行列式

Σ^{-1} ・・・ Σ の逆行列

2.2.4 ベイズ識別の実際の計算

$p(c|x)$ の分母の部分 $\sum_{c=1}^K p(x|c)p(c)$ はすべてのクラスに共通しているから、

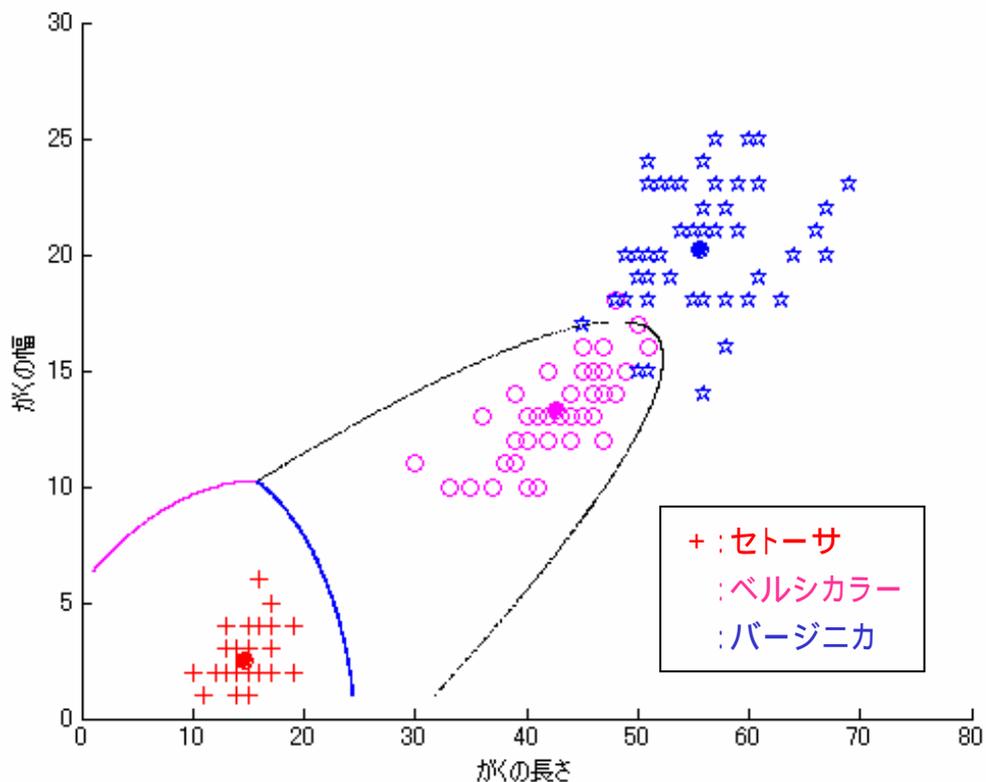
$p(c|x)$ の大小は分子部分のみを比較するだけでよい。また、分子部分の式(上記の多変量正規分布の式)の対数を取り、ベクトル x とクラスにかかわる部分のみを取り出すと、

$$-\frac{1}{2}\left\{(x - \mu_c)\Sigma_c^{-1}(x - \mu_c)^T + \log(|\Sigma_c|)\right\} + \log(p(c))$$

となる。実際の計算では、この大小を比較すればよい[1]。

2.25 2次元3群のベイズ識別の分類の特徴空間

例1(R.A.Fisherのアヤメのデータ)



これは、R.A.Fisherのアヤメのデータのがくの長さとかくの幅のデータに対して、ベイズ識別を行った場合の特徴空間を表している。赤いがセトーサ、ピンクがベルシカラー、青がバージニカをそれぞれ表している。このようにベイズ識別の識別境界線は二次曲線によって表される。

識別境界線の表示方法は2.4節参照。

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。

- (a) 非線形システムである
- (b) 学習能力を持つ
- (c) 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、場当たりの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[2]。

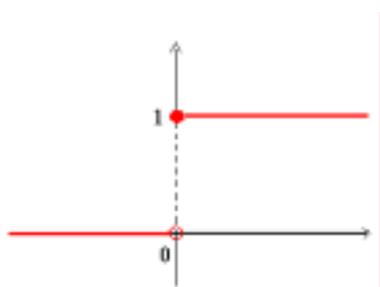
2.32 閾(しきい)素子

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

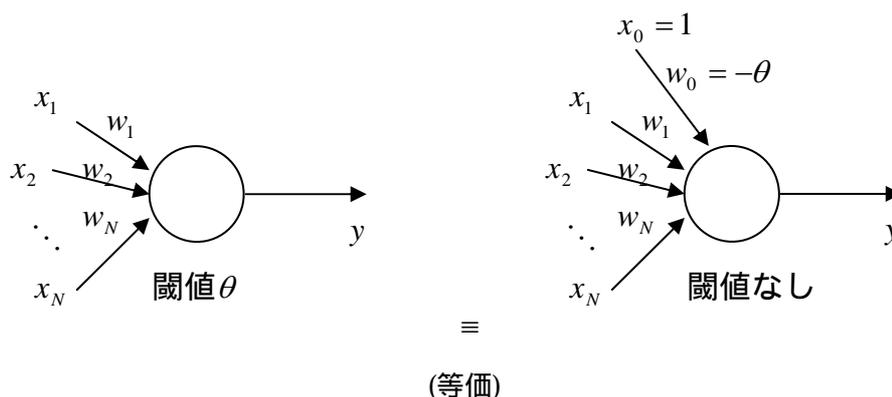
$$y = H \left[\sum_{i=1}^N w_i x_i - \theta \right]$$

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]。



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。



2.33 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]。

2.34 シグモイド関数

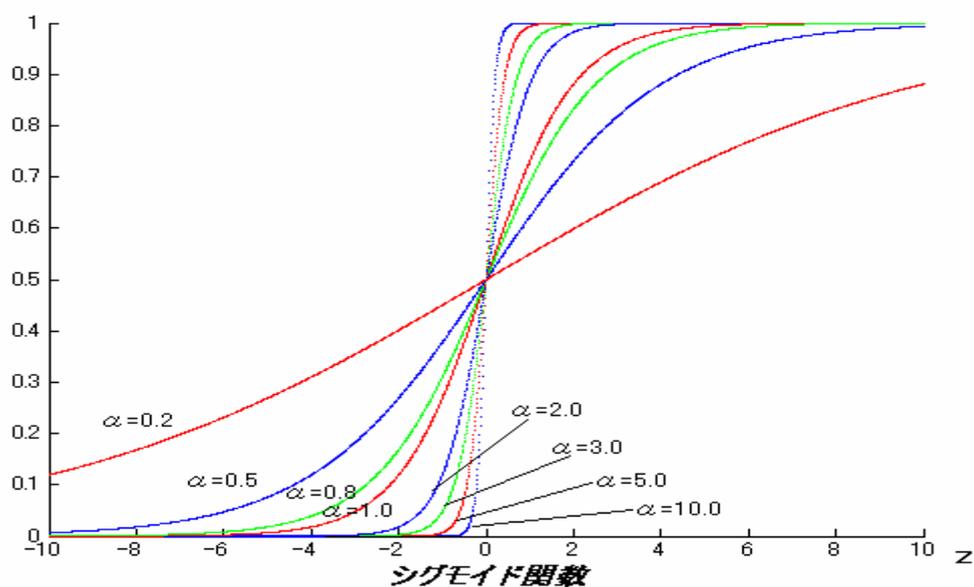
シグモイド関数は、以下の式で表される、単調増加の関数であり、ロジスティック関数とも呼ばれる。

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

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]。

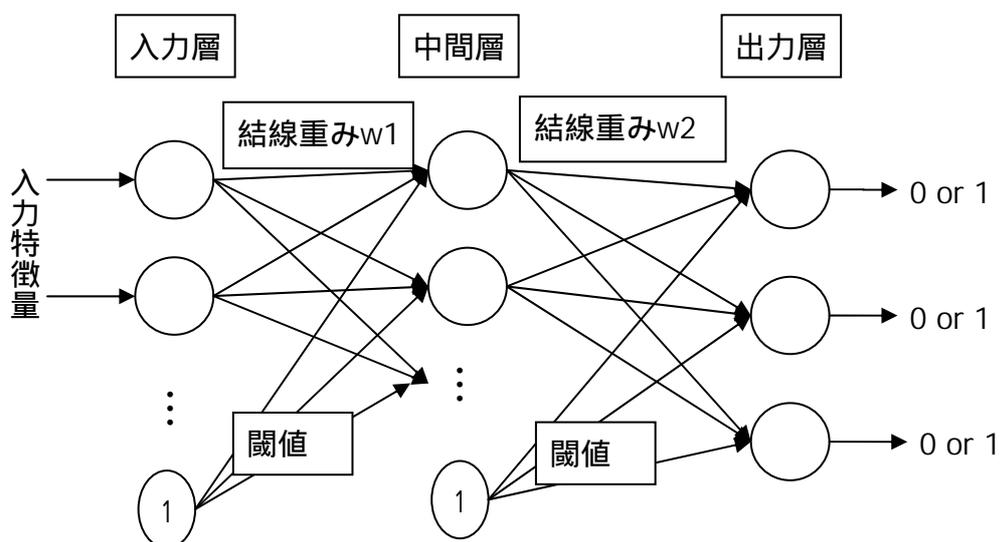
$$y = \text{sigmoid}(z - \theta)$$

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。



2.35 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。

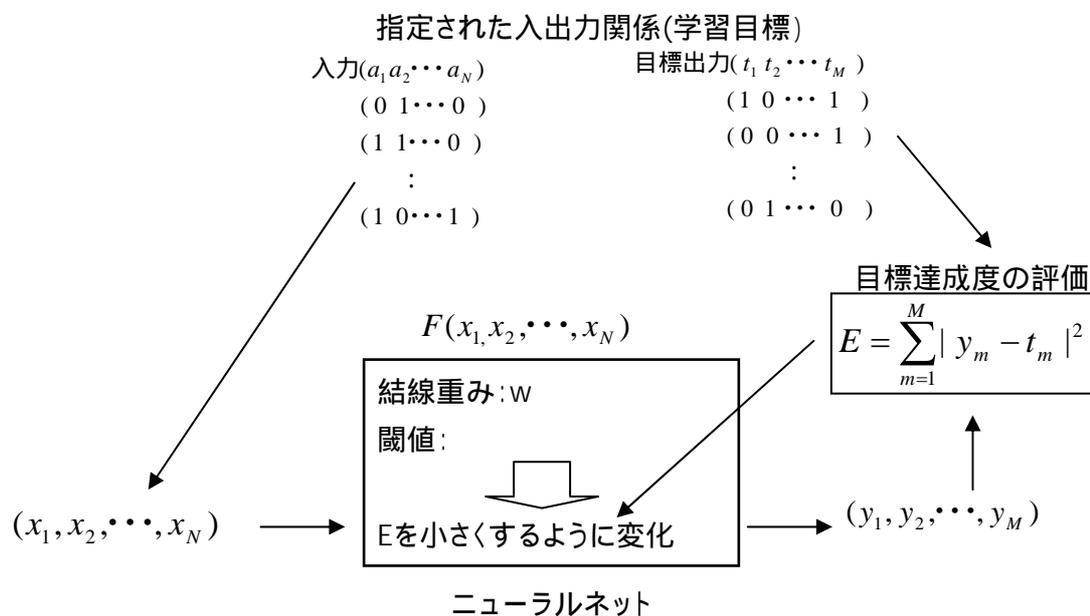


第1層は入力特徴量を受け取ってそのまま出力するだけの層であり、入力層と呼ばれる。第2層から最終層の1つ前の層までを中間層(または隠れ層)、最終層は出力層と呼ばれる。

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とし、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 以下の数値も他の素子に影響を与えられるようになる。

2.36 バックプロパゲーション法(誤差逆伝播アルゴリズム)

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。



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。

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

$$y = \text{sigmoid}(s)$$

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

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

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

$$= \alpha \text{sigmoid}(s) (1 - \text{sigmoid}(s))$$

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

2.37 最急降下法

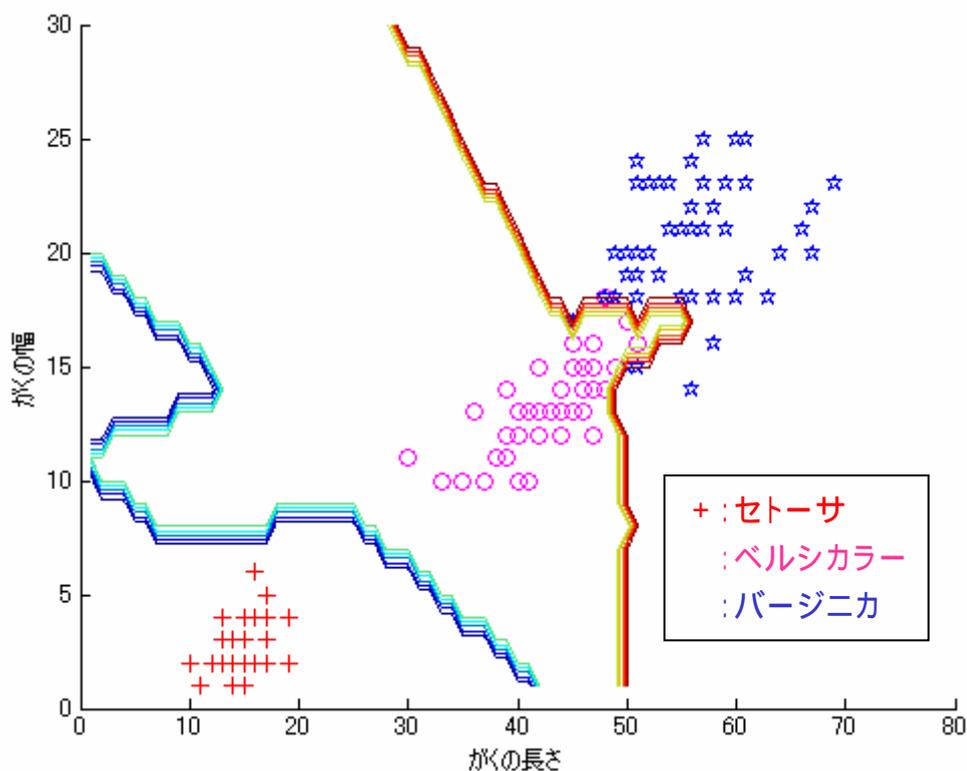
関数 $f(x_1, \dots, x_N)$ に対し、 $f(x)$ を最大化(最小化)するような $x = [x_1, \dots, x_N]$ を求める方法。最初に $x = [x_1, \dots, x_N]$ に適当な初期値を設定し、以下のような方法で x を更新していく。

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

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

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]。

2.38 ニューラルネットによる分類の特徴空間 例2 (R.A.Fisherのアヤメデータ)



これは、R.A.Fisherのアヤメのデータのがくの長さ、がくの幅に対して、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。赤がセトーサ、ピンクがベルシカラー、青がバージニカをそれぞれ表している。このように、バックプロパゲーション法の識別境界線は、ベイズ識別法と異なり、分布を仮定していないので、自由に柔軟な境界線が描かれている。

2.4 分類識別境界線の表示方法

2.4.1 境界の関数型が既知であり陽関数の場合

境界の関数型が既知であり陽関数の場合、境界線は式により求めることができる。

例：ベイズ識別(2変量)の場合

$$X = (x_1 \quad x_2), \quad \mu_k = (\mu_{k1} \quad \mu_{k2}), \quad \Sigma_k^{-1} = \begin{pmatrix} c_{k1} & c_{k2} \\ c_{k3} & c_{k4} \end{pmatrix} \quad (k: \text{クラス番号}) \text{とし、}$$

$$\{(X - \mu_i)\Sigma_i^{-1}(X - \mu_i)^T + \log(|\Sigma_i|)\} - \{(X - \mu_j)\Sigma_j^{-1}(X - \mu_j)^T + \log(|\Sigma_j|)\} = 0$$

を計算していく。

これを計算し、 $Ax_2^2 + Bx_2 + C = 0$ の形にする。この時、

$$A = c_{i4} - c_{j4}$$

$$B = -2\mu_{i2}c_{i4} + (x_1 - \mu_{i1})(c_{i2} + c_{i3}) + 2\mu_{j2}c_{j4} - (x_1 - \mu_{j1})(c_{j2} + c_{j3})$$

$$C = \{\mu_{i2}^2 c_{i4} - \mu_{i2}(x_1 - \mu_{i1})(c_{i2} + c_{i3}) + (x_1 - \mu_{i1})^2 c_{i1} + \log(|\Sigma_i|)\} \\ - \{\mu_{j2}^2 c_{j4} - \mu_{j2}(x_1 - \mu_{j1})(c_{j2} + c_{j3}) + (x_1 - \mu_{j1})^2 c_{j1} + \log(|\Sigma_j|)\}$$

これより、2次方程式の解の公式を用いて

$$x_2 = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \quad (B^2 - 4AC \geq 0)$$

を求める。これが2クラスについての境界線の方程式となる。この計算を2クラス間づつ、つまり ${}_k C_2$ 回行う。

なお、このままでは2クラスについての境界線が ${}_k C_2$ 本引かれる。このため、3クラスについての境界線を求めるには、2クラスについての境界線から選択することが必要となる。

選択方法を例を用いて説明する。

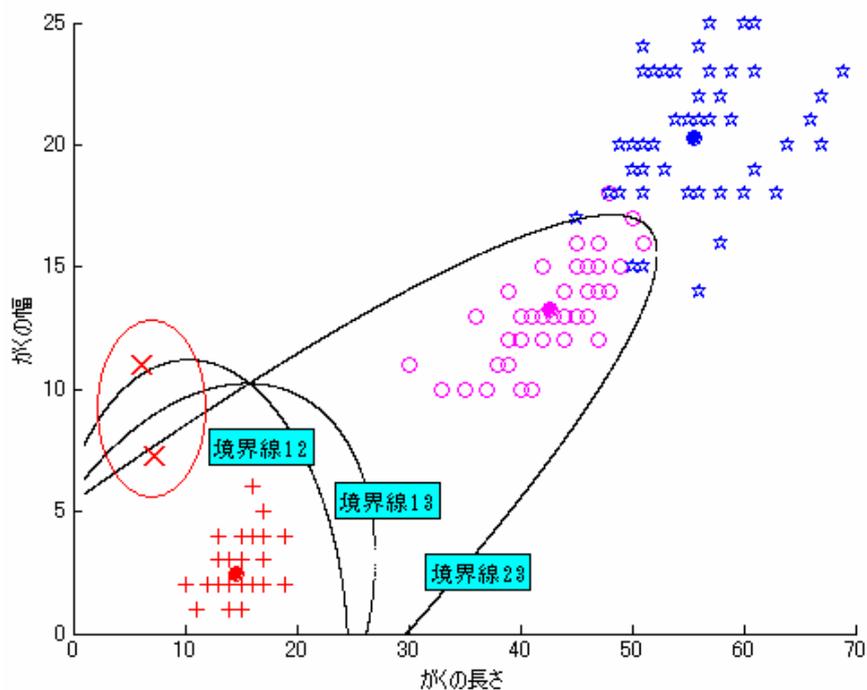
例：下図のように、赤：クラス 1、ピンク：クラス 2、青：クラス 3 の場合、この時最初に描かれる 2 クラスについての境界線は 3 本である。

- 境界線 12：クラス 1 とクラス 2 の境界線
- 境界線 13：クラス 1 とクラス 3 の境界線
- 境界線 23：クラス 2 とクラス 3 の境界線

このとき、図の赤い楕円部分には、3 本の境界線があるが、これをそれぞれの境界線を他の 2 本の境界線についてクラス(自身は 0)を判別すると

	境界線 12	境界線 13	境界線 23
境界線 12	0	3	3
境界線 13	1	0	3
境界線 23	1	1	0

となる。表の見方の例としては縦軸の境界線 12 についてみると、境界線 12 は自身では 0、境界線 12 の場所は境界線 13 のクラス分けからするとクラス 3 になり、境界線 23 のクラス分けからするとクラス 3 となる、とゆうようにこの表は見ていく。他の境界線について考えた時、クラスがきちんと分けられている(境界線の役目を果たしている)ここでは境界線 13 が求める 3 クラスについての境界線であるとわかる。



つまり、3クラスについての境界線は、例のように他の境界線について考えた時、以下の表のようなクラス分けがされるものである。

	境界線 12	境界線 13	境界線 23
境界線 12	0	1	2
境界線 13	1	0	3
境界線 23	2	3	0

このような結果となる境界線のみを表示する。

2.42 境界の関数型が陰関数等の場合

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。

2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3
2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3
2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3
2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3
2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3
1	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3
1	1	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3
1	1	1	1	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3
1	1	1	1	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3
1	1	1	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3
1	1	1	1	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3
1	1	1	1	1	1	1	1	1	1	2	2	2	2	3	3	3	3	3
1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	3	3	3	3
1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	3	3	3

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 2.0GHz
メモリ	512MB
OS	Windows2000 sp4
使用ソフト	Matlab7

3.2 Matlab について

Matlab は行列を扱うことのできるプログラムであり、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、m ファイルに関数やスクリプトを記述でき、複雑な数値計算を簡単に行うことができる。

3.3 分類するテストデータ

テストデータは 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種類について、それぞれ50個体(4変量)のデータ)と Landsat の画像データ(4320×2983)のデータ7バンドのうち、6バンドを除いた、6変量のデータを用いた。

学習には、R.A.Fisher のアヤメのデータではセトーサ、ベルシカラー、バージニカをそれぞれ50個体、計150個のデータを用いた。また、Landsat の画像データについては水部分1000画素、街部分1000画素、森林部分1000画素の計3000個のデータを学習に用いた。

第4章 Landsat 画像データの分類における行列を用いた計算の高速化

4.1 高速化する前の方法(ベイズ識別)

初期の、Landsat データ(4320×2983)を1つ(画素)づつ 4320×2983=12886560 回について処理していく方法では、膨大な計算時間を要してしまう。このときかかった計算時間は約10時間。どう鼻屑目に見ても、システムとして用いることができるものではなかった。(初期の方法)

1 データづつ計算する場合

以下に、Matlab プログラミングコマンドとともに行われた、行列操作について説明する。

計算する式

$$-\frac{1}{2} \left\{ (x - \mu_c) \Sigma_c^{-1} (x - \mu_c)^T + \log(|\Sigma_c|) \right\}$$

特徴ベクトル: x 、平均 μ_c : wmean、分散 Σ_c : wcov

Matlab コマンド

()' : ()内の行列の転置

inv() : ()内の行列の逆行列

det() : ()内の行列の行列式

log() : 対数

1. x 、wmean

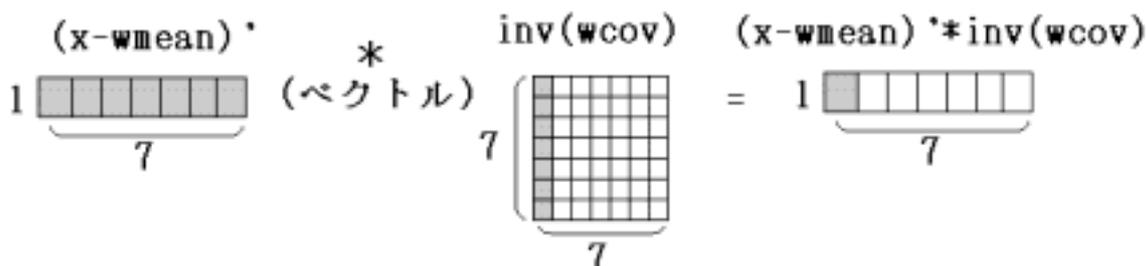
まず、 x に Landsat 画像データから1画素分のデータ(7変量)を読み出し格納する。
また、あらかじめ平均 wmean を求めておく。

$$\mathbf{x} = \begin{bmatrix} \\ \\ \\ \\ \\ \\ \end{bmatrix} \quad \mathbf{wmean} = \begin{bmatrix} \\ \\ \\ \\ \\ \\ \end{bmatrix}$$

2. $(x-wmean)' * inv(wcov)$;

次に先ほど求めた、 x と平均 $wmean$ を用いて、 $(x - wmean)$ を求めて、転置する。
 それにあらかじめ求めておいて分散 $wcov$ の逆行列 $inv(wcov)$ をかける(ここではベクトル計算の掛け算を用いる)。

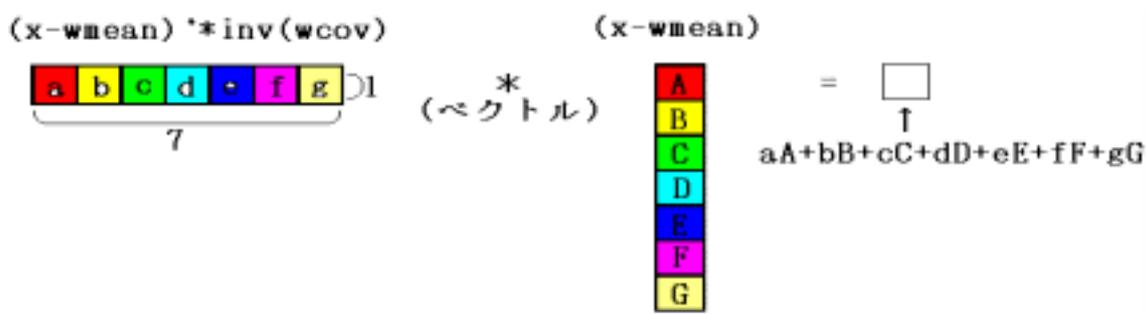
計算の進捗: $(x - \mu_c) \Sigma_c^{-1}$



3. 《式》 * $(x-wmean)$

上で求めた結果に、 $(x - wmean)$ をかける(ベクトル計算)と以下ようになる。

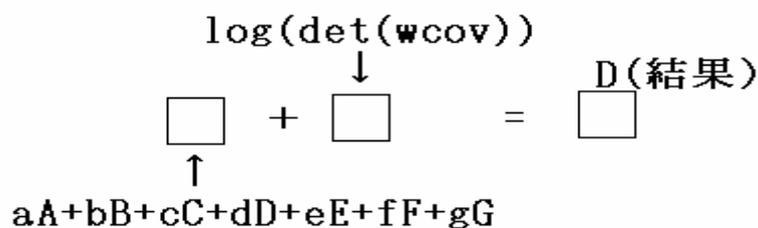
計算の進捗: $(x - \mu_c) \Sigma_c^{-1} (x - \mu_c)^T$



4. $D(結果) = (x-wmean)' * inv(wcov) * (x-wmean) + \log(\det(wcov))$;

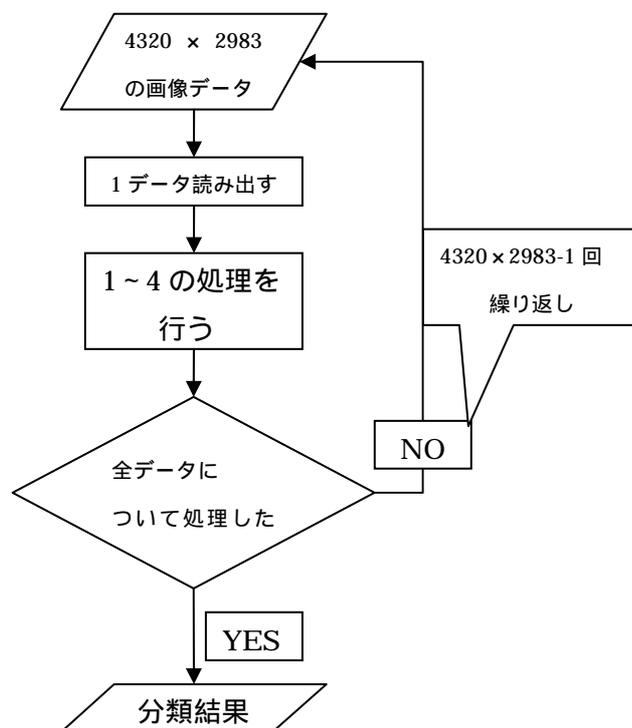
分散 $wcov$ の行列式 $\det(wcov)$ の対数 $\log(\det(wcov))$ を足して、結果 D を求める。

計算の進捗: $(x - \mu_c) \Sigma_c^{-1} (x - \mu_c)^T + \log(|\Sigma_c|)$



これを $-1/2$ することで計算式 $-\frac{1}{2} \{ (x - \mu_c) \Sigma_c^{-1} (x - \mu_c)^T + \log(|\Sigma_c|) \}$ を求めることができる。

初期の方法のフローチャートは以下のようになっていた。



この方法では、一画素づつ計算を行っているので、約 1200 万回もの繰り返し処理が必要となり、膨大な計算時間がかかってしまう。

4.2 高速化した方法(ベイズ識別)

そこで、そのとき用いていた Matlab の優れた行列計算機能を用いて、データを複数一度に計算することはできないかと考え、計算の高速化をはかり、計算時間を約 40 秒にまで短縮することを実現した。(高速化方法)

多数のデータを行列として 1 度に計算する場合

以下に、Matlab プログラミングコマンドとともに行われた、行列操作について説明する。

計算する式

$$-\frac{1}{2} \left\{ (x - \mu_c) \Sigma_c^{-1} (x - \mu_c)^T + \log(|\Sigma_c|) \right\}$$

特徴ベクトル: x 、平均 μ_c : wmean、分散 Σ_c : wcov

Matlab コマンド

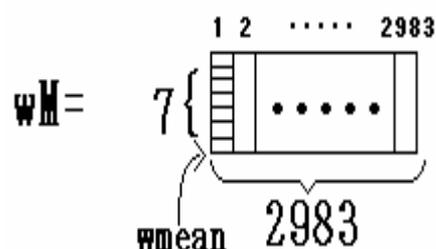
repmat(A,m,n) : 行列(データ)A を行方向に m 回、列方向に n 回繰り返して巨大な 1 つの行列にする。行列の複製に使える。

Repmat(A,m)は行方向、列方向ともに m 回複製

sum() : ()内の行列の行方向に総和をもとめる

1. wM=repmat(wmean,1,2983);

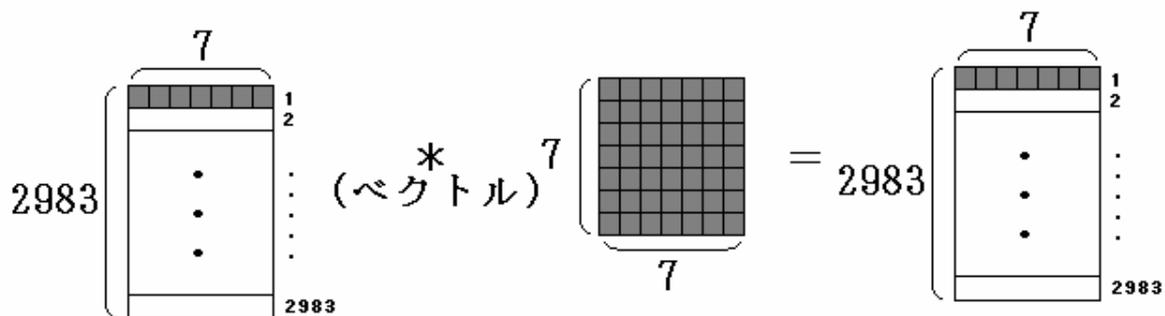
まず、平均 wmean を repmat コマンドで以下のように列方向に 2983 個複製して、(7×2983)のデータにする(2983 個一度に計算するため)。



6 . $(x-wM)' * \text{inv}(w\text{cov})$

4 で求めた結果に $\text{inv}(w\text{cov})$ をかける(ベクトル計算)。

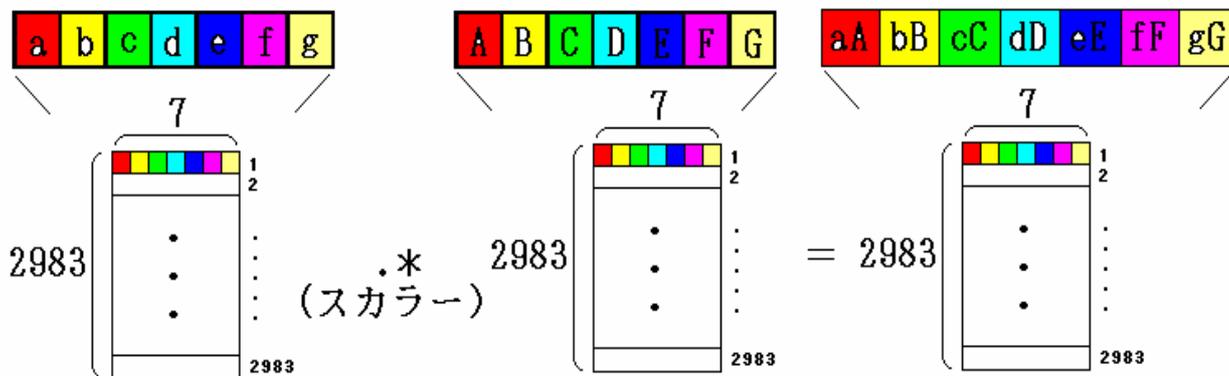
計算の進捗 : $(x - \mu_c) \Sigma_c^{-1}$



7 . 式 $\langle (x-wM)' * \text{inv}(w\text{cov}) \rangle . * (x-wM)'$

上で求めた結果に、4 で求めた $(x - w\text{mean})'$ をかける(スカラー計算)。

以下のように、同じ (i, j) 成分同士を掛け算する。 ($i=1, \dots, 2983$ $j=1, \dots, 7$)

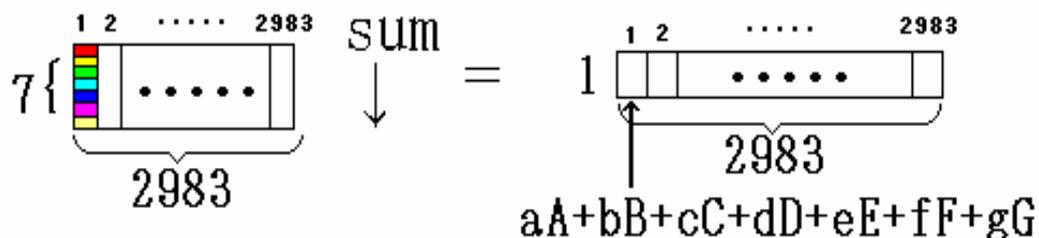


8 . sum(式の転置)

7で求めた結果を転置して、それを行方向に和を求める。

これにより求まる値は、前回の3で求めた結果が2983個一度に計算されたものである。

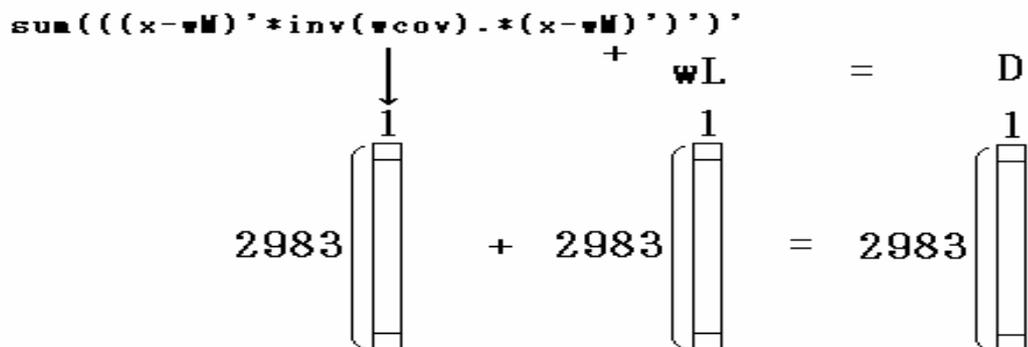
計算の進捗： $(x - \mu_c)\Sigma_c^{-1}(x - \mu_c)^T$



9 . D(結果)= 式の転置 + wL

上で求めたものに、最初に用意していた $\log(\det(wcov))$ を2983個複製したものを足すことで、結果Dを求める。

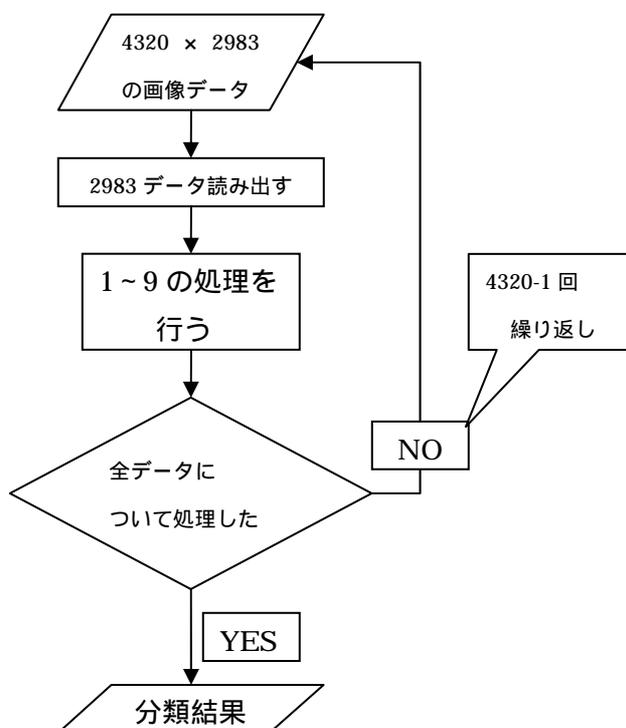
計算の進捗： $(x - \mu_c)\Sigma_c^{-1}(x - \mu_c)^T + \log(|\Sigma_c|)$



これを-1/2することで2983個分のデータについて一度に、

計算式 $-\frac{1}{2} \{ (x - \mu_c)\Sigma_c^{-1}(x - \mu_c)^T + \log(|\Sigma_c|) \}$ を求めることができる。

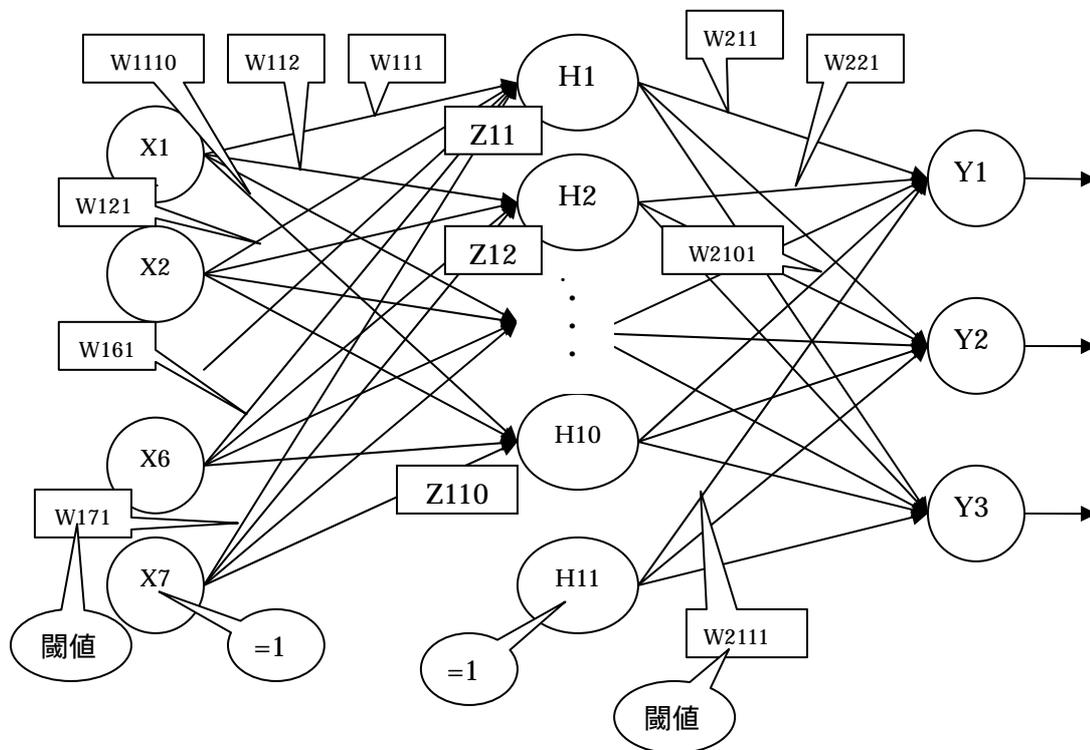
この高速化したアルゴリズムのフローチャートは以下のようになる。
前の一画素ずつ計算していくアルゴリズムと比べると繰り返しの回数が約1200万回からわずか4000回程度にまで減少している。



この高速化アルゴリズムにより、膨大な量の画像データもかなり高速に処理できるようになった。この方法は、計算式により、多少コマンドが変わるがこの行列計算による計算の並列処理はかなり有効な方法だといえる。

4.3 ニューラルネットワークにおける高速化

以下に、ニューラルネットワークについて高速化した場合の方法を示す。



この入力層：7 ユニット(1~6：入力特徴ベクトル、7：閾値用)

中間層：11 ユニット(1 1：閾値用)

出力層：3 ユニット(クラス 1, クラス 2, クラス 3)

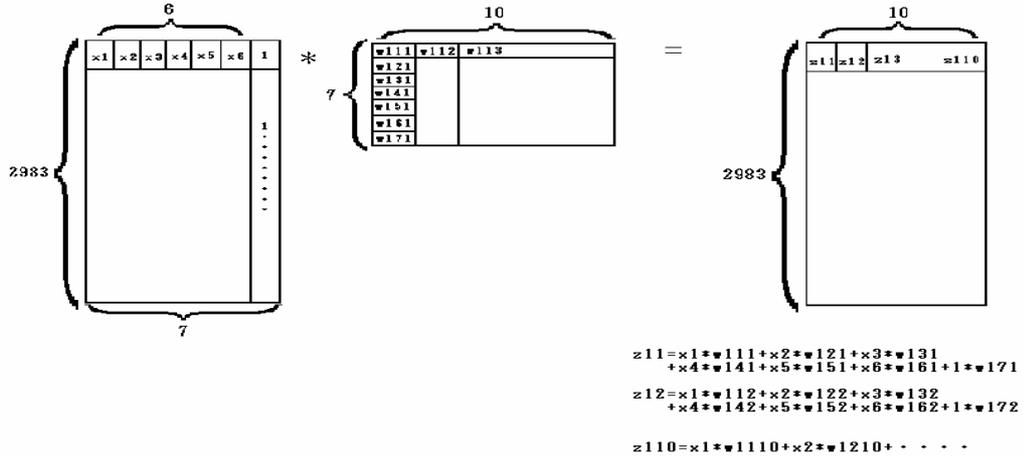
の 3 層ニューラルネットワークについて、Landsat 画像データを分類させた場合の高速化された、行列計算を次のページに示す。なお、このニューラルネットのフローチャートは前述した高速化アルゴリズムのものと同様になる。

この時、計算にかかった時間は、約 70 秒であった。前回より、シグモイド関数という式の複雑なものがあって、より計算時間がかかった。ニューラルネットワークはもともと並列処理をうたっているので、この行列計算を用いるとさらにその動作がイメージしやすいものとなった。

1. 中間層への入力 Z1 の計算

まず、中間層への入力 Z1 を計算する。ニューラルネットワークはその構造から、そのままベクトル計算していくことで、結果を求めることができる。

$$Z1 = X * W1 ;$$



2. 中間層での計算

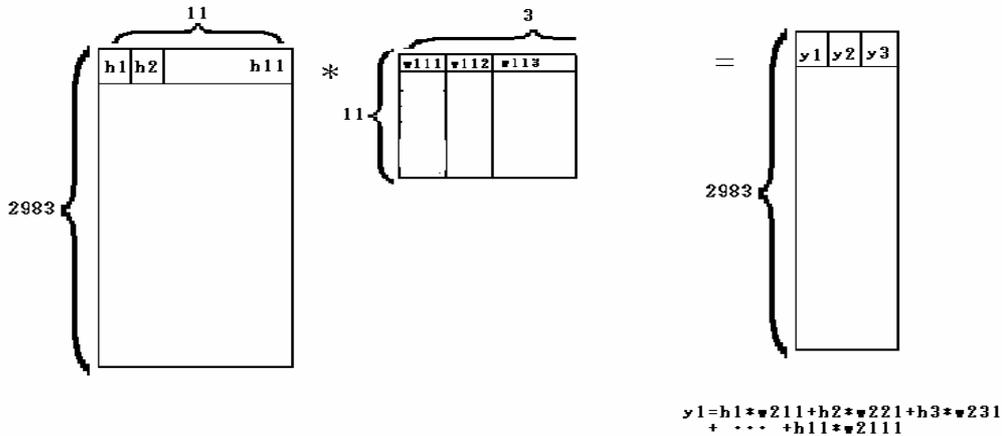
ここでは、1 で求めた Z1 をスカラー計算で、シグモイド関数を計算する。つまり、各成分ごとに以下の計算を行う。

$$H = 1 ./ (1 + \exp((-1) * Z1));$$

3. 中間層からの出力層への計算

2 で求めた H を用いて出力層への入力 Y を計算する。

$$Y = H * W2 ;$$



4. 過程 2 同様スカラー計算でシグモイドを計算してその結果の max を求める。

第5章 分類実験

5.1 実験方法

分類手法は

- ・ ベイズ識別
- ・ 3層ニューラルネットワークのバックプロパゲーション法

を分類データ

- ・ R.A.Fisher のアヤメのデータ(Landsat 画像データの対照データとして)
- ・ Landsat 画像データ

について行う。なお、3層ニューラルネットワークについては、R.A.Fisher のアヤメのデータ、Landsat 画像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。また、3層ニューラルネットワークに関して、

- ・ シグモイド関数のゲイン
- ・ 学習係数
- ・ 中間層のユニット(素子)数

を変化させて、その識別の正確さの収束速度(学習回数)の傾向について検証した。またこのとき、初めどのようにそれぞれに値を与えて始めればいいのかかわからないため、まず一度1つの変数(ここでは学習係数)を適当な値(0.2)で固定して、他の二つの値の変数(中間層数、シグモイド関数のゲイン)を変化させ、その傾向を見る。次に、変化させていた2つの変数のうちの1つ(ここではシグモイド関数のゲイン)の最も良い(収束速度が速い)値を選ぶ。そして今度はそのシグモイド関数のゲインを固定して、他の2つの変数(中間層数、学習係数)を変化させて、その傾向を見る。最後にその結果の中から最初に固定した変数(ここでは学習係数)について最も良い値を選んで、改めて他の変数について傾向を見る。

なお、今回は傾向の表を完成させスムーズに結果を得るために、また、ニューラルネットワークには局所解が存在するため、あまりに収束速度が遅いものは一定回数までで計算を終了している。

以下に完成した表の見方をまとめて示す

- ・ 表 1.1、表 1.2、表 1.3 は R.A.Fisher のアヤメのデータ(4変量、3群、データ数150)に関する分類を行った場合の学習結果の表である。終了条件は識別の正確さが99%以上になった場合、学習回数が10万回以上になった場合である。表 1.1 は初めの過程の、学習係数について適当な値0.2を入れて他の2変数について結果を得た時のものである。表 1.2 は表 1.1の結果から、シグモイド関数のゲインの最適値を求め、それに従い実験を行ったものである。表 1.3 は表 1.2から最適な学習係数を求め、それについて改めて実験した結果である。

- 表 2.1、表 2.2、表 2.3 は Landsat 画像データ (6 変量、3 群、データ数 3000) について分類を行った場合の学習結果の表である。終了条件は識別の正確さが 99.75% 以上になった場合、学習回数が 1 万回以上になった場合である。表 2.1 は先に行った R.A.Fisher のアヤメのデータに関する収束速度の良い変数の傾向から、前回同様、学習係数 を選んで、他の 2 変数に関して学習の収束速度を実験したときのものである。表 2.2 は表 2.1 の結果から、シグモイド関数のゲイン の最適値を求め、それに従い実験を行ったものである。表 2.3 は表 2.2 から最適な学習係数を求め、それについて改めて実験した結果である。

なお、用いるデータは全て、規準化したものを用いている。一般には、確率変数 X に対して

$$X^* = \frac{X - \mu}{\sigma}$$

とした確率変数を X を規準化または標準化したという[4]。

規準化することにより違う分布のデータであっても、変数の変動をそろえることができるので、データの比較をすることが可能になる。

この実験で用いたデータの詳細

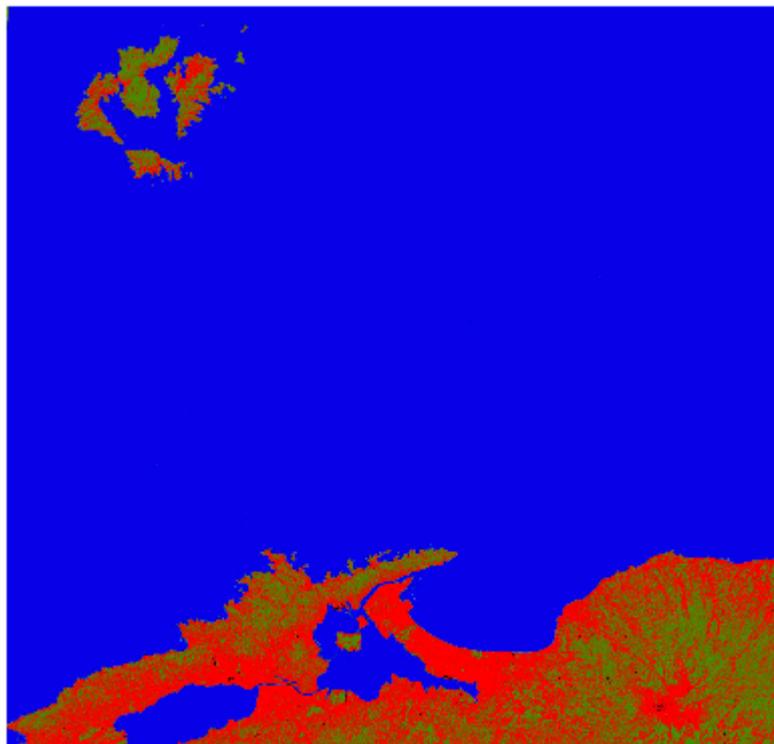
規準化後の R.A.Fisher のアヤメのデータ					規準化後の Landsat 画像データ						
以下の表の各変数名					以下の表の各変数名						
	花弁長	花弁幅	がく長	がく幅	バンド 1	バンド 2	バンド 3	バンド 4	バンド 5	バンド 7	
平均 :	1.0e-015 *				1.0e-014 *						
	0.3656	-0.1776	0.1021	-0.1628	0.4620	0.9476	-0.5608	0.0465	-0.2090	-0.3906	
単分散 :	規準化により 1.0 となる				規準化により 1.0 となる						
最大値 :	2.4837	3.0805	1.7799	1.7064	8.3144	7.9962	6.4654	3.4477	3.0887	3.5875	
最小値 :	-1.8638	-2.4258	-1.5623	-1.4422	-1.3044	-0.9549	-0.9541	-1.1351	-1.1753	-1.1475	
歪度 :	1.0e-044 *				1.0e-034 *						
	-0.0053	0.2174	0.2869	0.0073	-0.0007	-0.3062	-0.0002	0.0000	-0.0000	0.0041	
尖度 :	1.0e-058 *				1.0e-044 *						
	0.0011	0.1497	0.2166	0.0016	0.0000	0.1382	0.0000	0.0000	0.0000	0.0004	

5.2 実験結果

Landsat 画像データについて

Landsat 画像データについて、3 クラス(青 : 水部分(海、湖等) 赤 : 街部分(街、畑、平地等) 緑 : 山岳部分) に分類する実験を行った。また、2 変量 (5 バンド、7 バンド) についても実験を行い、その結果をベイズ識別とニューラルネットワークについて、それぞれ特徴空間と分類後画像を以下に示す。

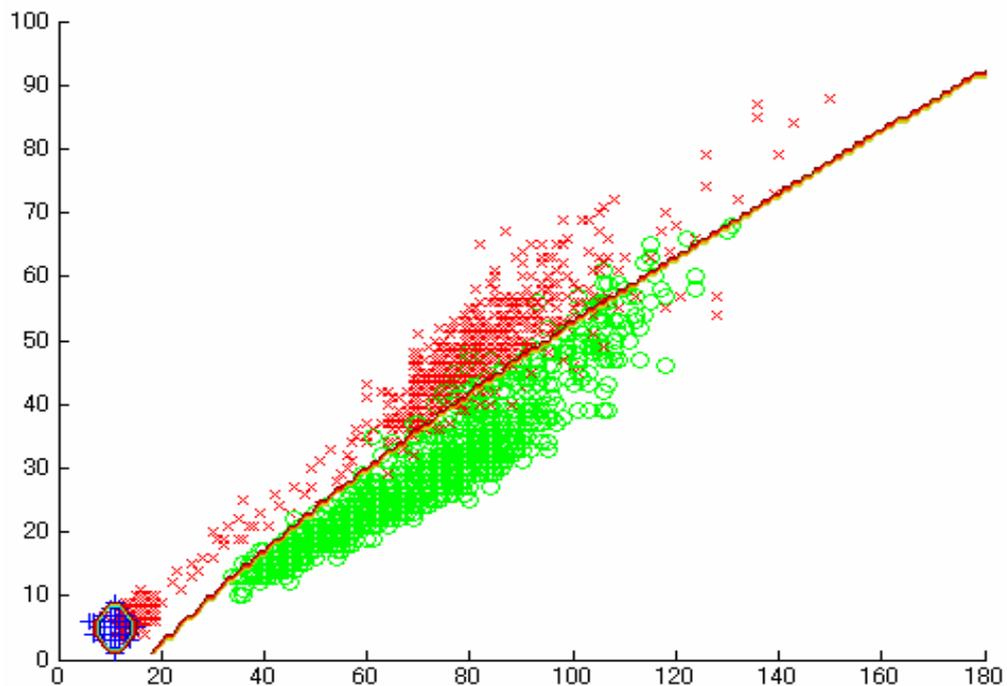
Landsat 画像データ(6 変量)にニューラルネットで行った場合(分類画像)



ベイズ識別を行った場合(特徴空間)

識別の正確さ (93.27% { 水部分 : 99%、街部分 : 86.1%、山岳部 : 94.7% })

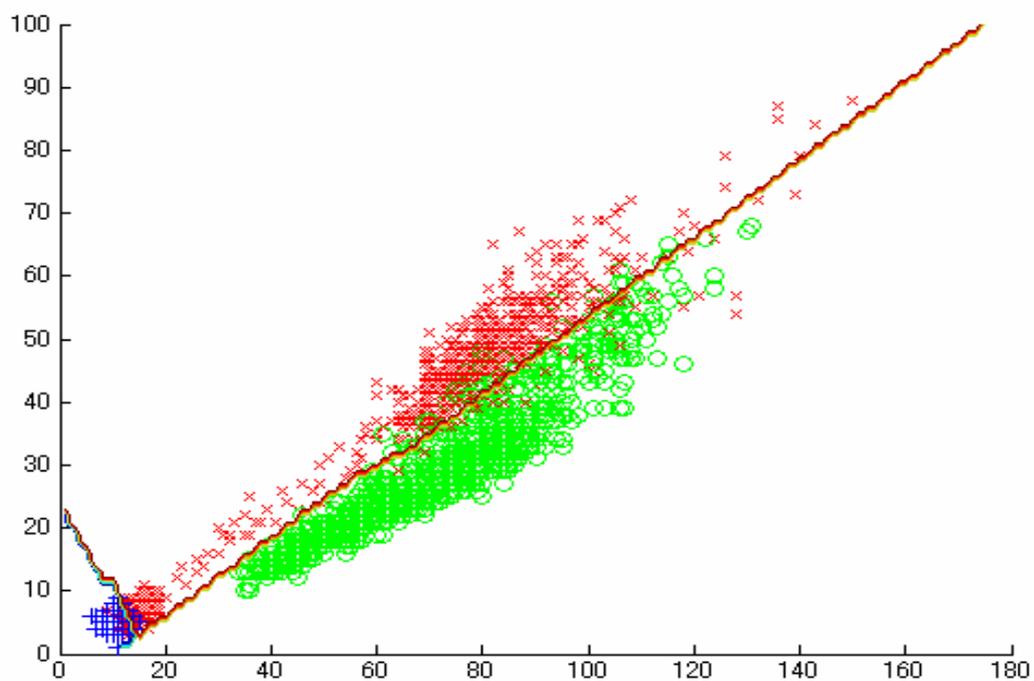
青+ : 水部分 赤× : 街部分 緑○ : 山岳部



ニューラルネットワークで分類を行った場合(特徴空間)

識別の正確さ (95.13% { 水部分 : 96.8%、街部分 : 92.7%、山岳部 : 95.9% })

青+ : 水部分 赤× : 街部分 緑○ : 山岳部



ベイズ識別を行った場合(分類画像)

識別の正確さ (93.27% { 水部分 : 99%、街部分 : 86.1%、山岳部 : 94.7% })

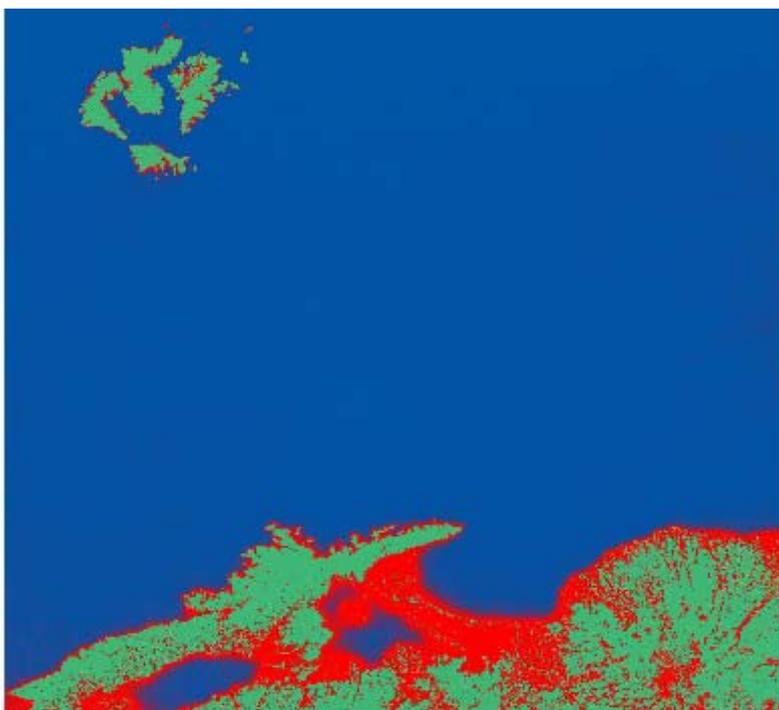
青+ : 水部分 赤× : 街部分 緑 : 山岳部



ニューラルネットワークで分類を行った場合(分類画像)

識別の正確さ (95.13% { 水部分 : 96.8%、街部分 : 92.7%、山岳部 : 95.9% })

青+ : 水部分 赤× : 街部分 緑 : 山岳部



R.A.Fisher のアヤメのデータに対する、3層ニューラルネットワークの変数の変化における学習の収束速度の傾向を検証する。

学習係数 = 0.2 終了条件：識別の正確さ 99%
 学習回数 > 10 万回

中間層数	0.5	0.8	1	2	3	4	5	6	7	10
3	100000	100000	64183	12753	6609	153	149	840	3718	100000
4	8209	2759	1729	1051	1477	1480	813	207	2956	100000
5	3695	1296	853	839	254	188	481	126	100000	100000
6	7725	2862	1878	757	652	153	203	1496	100000	100000
7	3657	1282	1117	688	265	218	1470	100000	6032	100000
8	8153	3151	2093	671	323	1534	14877	100000	100000	100000
9	8325	1341	1562	622	432	1062	100000	100000	100000	100000
10	3413	2589	1252	719	585	1884	68260	100000	100000	100000
11	8455	1426	1669	667	872	100000	100000	100000	100000	100000

表 1.1 R.A.Fisher のアヤメのデータの条件変化の結果 1 (= 0.2)

表 1.1 より、シグモイド関数のゲイン = 4 のとき最も優れた識別結果になっている。これにより、今度はシグモイド関数のゲイン = 4 について、検証を行う。

シグモイド関数のゲイン = 4.0 終了条件：識別の正確さ 99%
 学習回数 > 10 万回

中間層数	0.005	0.01	0.05	0.1	0.2	0.5	0.8
3	4946	2336	549	1467	153	100000	100000
4	5485	2432	951	1241	1448	143	100000
5	3257	1030	501	427	188	54893	15200
6	4343	2074	709	406	153	673	16366
7	7981	3700	718	342	218	100000	100000
8	9164	3994	831	603	1534	298	9646
9	14523	6503	1972	1354	1062	49818	100000
10	19450	12941	2372	1159	1884	60221	100000

表 1.2 R.A.Fisher のアヤメのデータの条件変化の結果 2 (= 4.0)

表 1.2 より、最も有効な学習係数 η の値は 0.2 であるといえる。ここで本来なら、この η の値に対して、検証を行うのだが、 $\eta = 0.2$ については、既に検証が終わっているため、ここでこの実験は終了する。

この実験の結果、R.A.Fisher のアヤメのデータ(3 クラス、4 変量、学習に用いたデータ数 150)に関しては、学習係数 $\eta = 0.2$ 、シグモイド関数のゲイン $\gamma = 4.0$ のとき最も優れた収束速度となるとわかった。

Landsat 画像データに対する、3 層ニューラルネットワークの変数の変化における学習の収束速度の傾向を検証する。

Landsat 画像データに対して、実験検証を行う。まず適当な学習係数 η の値として $\eta = 0.2$ ではじめる。

学習係数 $\eta = 0.2$ 終了条件：識別の正確さ 99.75%
学習回数 > 1 万回

中間層数	0.2	0.5	0.8	1	2	3	4	5	6	7	10
3	2610	400	370	560	150	120	290	360	200	6890	10000
4	2060	1010	830	900	280	140	150	1040	210	8560	10000
5	2490	1420	870	320	290	270	130	200	140	10000	10000
6	2280	900	670	420	270	280	180	230	190	8320	10000
7	2360	1190	730	310	220	230	190	190	270	8930	10000
8	2690	1260	740	420	300	170	250	130	370	7510	10000
9	2370	1060	640	390	310	240	140	230	10000	10000	10000
10	2260	980	630	420	440	190	100	500	160	6390	10000
11	2230	1060	500	420	410	130	140	150	4810	8580	10000

表 2.1 Landsat 画像データに対する条件変化の結果 1 ($\eta = 0.2$)

表 2.2 より、学習係数 $= 0.05$ のとき、最も良い結果となっている。
 これにより、今度は学習係数 $= 0.05$ について、検証を行う。

学習係数 $= 0.05$ 終了条件：識別の正確さ 99.75%
 学習回数 > 1 万回

中間層数	0.2	0.5	0.8	1	2	3	4	5	6	7	10
3	10000	880	2570	2710	180	180	330	420	490	180	580
4	10000	1570	1170	970	480	470	90	190	230	190	450
5	6570	1790	1170	1110	330	390	90	100	190	190	1780
6	5230	1760	1140	1060	420	280	80	60	220	200	160
7	5750	1940	1060	260	460	160	350	240	210	170	460
8	5690	1180	430	300	430	400	390	160	160	150	140
9	4980	900	400	290	410	310	370	320	340	240	430
10	4710	950	410	400	430	200	200	300	220	290	160
11	4760	970	620	370	230	210	110	250	200	110	200

表 2.3 Landsat 画像データに対する条件変化の結果 3 ($= 0.05$)

表 2.3 よりシグモイド関数のゲイン $= 5.0$ のとき最も良い結果となっている。
 この実験の結果、Landsat 画像データ (3 クラス、6 変量、学習に用いたデータ数
 3000) に関しては、学習係数 $= 0.05$ 、シグモイド関数のゲイン $= 5.0$ のとき、
 最も優れた収束速度となるとわかった。

第6章 考察とまとめ

6.1 考察

第5章で行ったニューラルネットワークとベイズ識別法の比較実験は、識別の正確さにおいては明らかにニューラルネットワークのほうが優れているとわかった。ベイズ識別法が分布に従うという限界に到達しようと、ニューラルネットワークは力づくでも、識別の正確さを100%に近づけていく。また、ニューラルネットワークは中間層の数を変えてやることで、かなり複雑な関数でも近似できるようになるので、分類の精度が上がらないようなら、条件を変えてやれば、今度はうまくいくようになるかもしれない。ニューラルネットワークは自動でよりうまく分類ができるように自身の機構を変化させていくので、こちらで適当な諸条件を与えてやると、かってにうまく分類するように学習していつてくれる。このため、分類器を設計する上で、難しいどの特徴をどのように計算していくかを考えないですむという利点がある。もちろん、よりはやく学習させていくには分類するものの前知識があったほうがいいのは確かだが、専門家でも分類することができるのはとてもうれしいことだ。しかし、識別に必要な全過程の計算時間では、ベイズ識別のほうが圧倒的に短時間ですむ。とりわけ、ニューラルネットワークのバックプロパゲーション法における学習時間の差は大きい。これより、分類手法としては、このどちらを用いるのがいいかはその場の状況（計算時間、識別の正確さ、分類するデータ）によるといえる。なお、ニューラルネットワークのバックプロパゲーション法が、計算時間がかかりすぎて実用化にふさわしくないというわけではなく、実用化に際しては、あらかじめパラメータを求めてそれをシステムに組み込めばいいので、問題はない。

第5章で行った諸条件を変えた上での検証では、ニューラルネットワークの欠点ともいべき局所解らしきものも結果として存在したが、優れた識別の正確さを出す条件はまったくのランダムというわけではなく、ある程度の傾向が見られるとわかった。

シグモイド関数のゲインは入力される特徴ベクトル数(変量数)によって、効率の良い値が変わっていつている。入力特徴ベクトル数が多くなれば、より小さな値で、少なくなれば、より大きな値で最も収束速度が速くなる。

また、学習係数は小さくなるときはその学習速度が緩やかになっていくだけだが、大きくしていく場合は、ある一定の値を超えると学習の収束速度が一気に変動(悪化)してしまう。また、このときの一定の値が最も優れた学習の収束速度を誇る値である。小さい分にはかまわないが、少しでも大きくなってしまつと、学習がうまくいかない場合があるので、は確実な値がわからない場合は、小さめの値で学習を始めさせたほうがいい。また、5章の結果の学習係数の変化により、学習係数はより複雑なデータを分類する場合ほど、小さい値として、小刻みに境界線を区切つていつたほうがいとわかる。

次に、中間層の数に関しては、5章の実験の結果では、学習係数 とシグモイド関数のゲイン により優れた値が変動し、明確な傾向が見つけられなかった。よって、前述したようにより複雑な関数（分類）をするならば中間層の数を増やしたほうがいいが、それほど困難な分類でない（カテゴリー数が少ない：分類するクラスの数が多い）場合には、最低数、分類するクラスの個数以上の数で学習をしてみて結果から少しずつ変化させるのが良い。

6.2 まとめ

今回、ベイズ識別法と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自らで実装し、ニューラルネットワークの最新の理論展開の学習とともに、その特徴を検証することで、これらの手法の原理を良く見ることができた。今後は、近年注目されているもう一つの代表的分類手法である、SVM 等について研究をしていきたい。本研究を行ったことで、新しい手法を自分の手で学んでいくやり方を、きっちり身に付けることができた。本研究で得た経験と知識は、今後の研究の足がかりとなり、基盤となってくれることは間違いない。当初予定にはなかったが、結果として実現できた多量のデータを行列で高速に処理する手法は、今後何かと用いることもあるだろう。この高速化については望んだ以上の結果が出せたので、この経験を今後の研究に生かしていきたい。

第7章 謝辞

最後に、本研究を進めるにあたり、ゼミを中心に最後まで粘り強く熱心なご指導、ご助言を頂き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、同じ学部生である島村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。

第8章 参考文献

- [1] 麻生英樹 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] 清水邦夫 “データサイエンス・シリーズ8 地球環境データ -衛星リモートセンシング- ” (共立出版 2002)
- [4] 白旗慎吾 “統計解析入門” (共立出版 1992)