情報処理学会論文誌 Vol. 50 No. 8 1789–1797 (Aug. 2009)

# 二次判別分析の計算時間の短縮

鈴木道孝[†1] 伊藤彰義[†2]

二次判別分析では，入力された特徴量ベクトル $\boldsymbol{x}$ は，条件付き確率 $P(c\,|\,\boldsymbol{x})$ が最大になるようなカテゴリ $c$ に分類される．ただし，前もって学習データとして，特徴ベクトルとカテゴリの組の集合 $\{\boldsymbol{x}_i, c_i\}$ が与えられている．単純な計算方法は，すべてのカテゴリに対して $P(c\,|\,\boldsymbol{x})$ を計算し最大値を得ることであるが，カテゴリ数が大きい場合には，特徴量の次元数も大きくしなければならず，計算時間が膨大になる．$P(c\,|\,\boldsymbol{x})$ の具体的な表式を吟味すると，この単純な計算方法には冗長な部分があることが分かり，その部分を除くことによって，効率の良いアルゴリズムを得ることができる．本アルゴリズムを日本語手書き文字の認識率測定実験に適用し，単純な計算方法と比べて計算時間を 4%までに削減することができた．

# Reduction of Processing Time for Quadratic Discriminant Analysis

Michitaka Suzuki[†1] and Akiyoshi Itoh[†2]

In quadratic discriminant analysis, the input feature vector, $\boldsymbol{x}$, is classified to the category, $c$, whose conditional probability $P(c\,|\,\boldsymbol{x})$ is the maximum, given a training data set of feature vectors and category labels $\{\boldsymbol{x}_i, c_i\}$. The simple and straightforward way of the calculation is to calculate the probabilities for all the categories to determine the maximum, but when the number of the categories is large and so is the dimension of the feature vector accordingly, the processing time becomes enormous. When we examine the specific expression of $P(c\,|\,\boldsymbol{x})$, we can spot redundant parts in the straightforward calculation, and we can obtain a fast algorithm by avoiding these redundant calculations. Applying this algorithm to the experiment to measure the recognition rates of Japanese handwritten characters, we could reduce the processing time to 4% of that by the straightforward calculation.

## 1. まえがき

二次判別分析では，入力された特徴量のベクトル $\boldsymbol{x}$ に対しそれが属するカテゴリ $c$ を，その確率 $P(c\,|\,\boldsymbol{x})$ が最大になるという条件によって決定する．ただし，前もって学習データとして，カテゴリのラベルの付いた特徴量の集合 $\{\boldsymbol{x}_i, c_i\}$ が与えられている．最も単純で明快な計算方法は，与えられた入力に対して，すべてのカテゴリに対する確率を計算し，最大値を決定することである．この場合の計算量は (カテゴリ数) $\times$ (次元数)$^2$ に比例する．カテゴリ数が大きいときは，必然的に特徴ベクトルの次元数も大きくしなければならない．たとえば，日本語の手書き文字認識の場合では，カテゴリ数が 3,000 程度で次元数が 200 程度は必要になる[1)]．このような問題では単純な計算方法による計算時間は膨大になる．

計算時間の削減のために近似計算がしばしば用いられてきた．たとえば，二次判別分析から分散共分散の効果を無視すれば，ユークリッド距離法が得られる．この方法は，精度は良くないが計算量が少なくて済むので，大分類にしばしば用いられる．すなわち，全カテゴリから複数の上位候補を選び，絞られた上位のカテゴリに対してだけ精度の良い二次判別分析を適用するというものである．

成分間の共分散を保持した近似には，部分空間法や修正ベイズ近似[1)] などがある．これらは，認識率の低下は小さいので，前述の大分類と組み合わせて，詳細分類に使われるが，大きな計算時間の削減をもたらすものではない．

本研究では，二次判別分析の範囲で近似を使わずに，かつ，従来の計算方法より格段に効率の良いアルゴリズムを示す．2 章で，二次判別分析について説明する．3 章で，効率の良い識別アルゴリズムを説明する．4 章で，手書き文字の特徴量およびその他のデータについて，本アルゴリズムを適用しその効果を調べる．5 章で結論を述べる．

## 2. 二次判別分析

入力 $\boldsymbol{x}$ がカテゴリ $c$ に属する確率は，ベイズの定理を使って，次のように表される．

$$P(c\,|\,\boldsymbol{x}) = \frac{p_c(\boldsymbol{x})P_c}{\sum_{c'} p_{c'}(\boldsymbol{x})P_{c'}} \tag{1}$$

†1 日本大学理工学部理工学研究所
Research Institute of Science & Technology, Nihon University
†2 日本大学理工学部
College of Science and Technology, Nihon University

ここで，$p_c(\boldsymbol{x})$ は，カテゴリ $c$ の条件付き確率分布であり，$P_c$ は事前確率である．式 (1) の分母は $c$ に関係しないので，確率の大小だけを問題にするときは無視してよい．

特徴量はカテゴリごとに正規分布すると仮定する．訓練データが十分得られるときは，標本平均ベクトル，標本共分散行列の正規分布で条件付き確率分布を近似できる．しかし，訓練データの個数が特徴ベクトルの次元数 $d$ 以下のときは標本共分散行列は正則でなくなり，事後確率は計算できなくなる．また，$d$ と比べて十分大きくないときも，良い認識精度は得られない．

このような状況でも，階層的ベイズ推定[2] は比較的に良い結果をもたらす．それによると，共分散行列は次のように与えられる．

$$\Sigma_c = \frac{n_c S_c + \nu_1 \mathrm{diag} S + \nu_2 S}{n_c + \nu_1 + \nu_2} \tag{2}$$

ここで，$S_c$ はカテゴリ $c$ の標本共分散行列

$$S_c = \frac{1}{n_c} \sum_{i=1}^{n_c} (\boldsymbol{x}_i - \boldsymbol{m_c})(\boldsymbol{x}_i - \boldsymbol{m_c})^{\mathrm{t}} \tag{3}$$

である．$n_c$ はカテゴリ $c$ の訓練データの数である．標本平均ベクトルは特に特異性を持たないので，平均ベクトルの推定値として，標本平均ベクトルとする：$\boldsymbol{m_c} = n_c^{-1} \sum_{i=1}^{n_c} \boldsymbol{x}_i$．$S$ は $S_c$ をすべてのカテゴリで平均したものである．

$$S = \frac{1}{n} \sum_c n_c S_c \tag{4}$$

ここで，$n$ は訓練データの総数である：$n = \sum_c n_c$．また，式 (2) の $\mathrm{diag} S$ は $S$ の非対角要素を零にした行列であり，係数 $\nu_1$，$\nu_2$ は，信頼度定数と呼ばれる，正のパラメータである．共分散行列のこのような形は，正則化判別分析[3),4)] からも予測されていた形である．

式 (2) を使い，条件付き確率分布を

$$p_c(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma_c|^{1/2}} \times \exp\left[ -\frac{1}{2} (\boldsymbol{x} - \boldsymbol{m}_c)^{\mathrm{t}} \Sigma_c^{-1} (\boldsymbol{x} - \boldsymbol{m}_c) \right] \tag{5}$$

で近似し，式 (1) の対数をとって，次のように識別関数 $f(\boldsymbol{x}, c)$ を定義する．

$$f(\boldsymbol{x}, c) \equiv -2 \ln P(c \,|\, \boldsymbol{x}) = a_c + (\boldsymbol{x} - \boldsymbol{m}_c)^{\mathrm{t}} \Sigma_c^{-1} (\boldsymbol{x} - \boldsymbol{m}_c) \tag{6}$$

$$a_c = -2 \ln P_c + \ln |\Sigma_c| \tag{7}$$

ここで，$c$ によらない項は省略している．式 (6) は，通常の二次判別分析の識別関数の形をしているが，その共分散行列，式 (2)，は訓練データ数が少ない場合にも頑健である．事前確率 $P_c$ は，訓練データから次のようにおくのが妥当である．

$$P_c = n_c / n \tag{8}$$

入力 $\boldsymbol{x}$ に対する最終的な識別結果は，

$$c(\boldsymbol{x}) = \underset{c}{\mathrm{argmin}}\ f(\boldsymbol{x}, c) \tag{9}$$

となる．

## 3. 効率的な識別アルゴリズム

計算の高速化を達成するには，冗長な計算をしないことが肝要である．今，識別関数値に収束し，添え字 $k$ に関して単調増加の数列 $f_k(\boldsymbol{x}, c)$ が与えられていて，$f_k(\boldsymbol{x}, c) > f(\boldsymbol{x}, c_1)$ なる $c_1$ が存在することが分かっているとき，$f(\boldsymbol{x}, c) > f(\boldsymbol{x}, c_1)$ なので，$c$ が最終的な識別結果になることはない．したがって，$c$ に関する $k+1$ 以降の計算は冗長である．このように，識別関数値に収束する単調増加数列を求めれば，冗長な計算を省くことができる．そこで，まず，そのような単調増加数列を求め，その後に，それを使った効率の良いアルゴリズムについて述べることにする．

### 3.1 単調増加数列

式 (6) の識別関数を共分散行列の固有値 $e_i$ と固有ベクトル $\boldsymbol{v}_i$ を使って，次のように書き直すことができる．

$$f(\boldsymbol{x}, c) = a_c + \sum_{i=1}^{d} \frac{[\boldsymbol{v}_{ci}^{\mathrm{t}} (\boldsymbol{x} - \boldsymbol{m}_c)]^2}{e_{ci}} \tag{10}$$

ここで，添え字 $i$ は固有値が降順に並ぶようにふられている．式 (10) の和記号の第 $k$ 項までをとった部分和を $f'_k(\boldsymbol{x}, c)$ とする（$0 \le k \le d$）．

$$f'_k(\boldsymbol{x}, c) \equiv a_c + \sum_{i=1}^{k} \frac{[\boldsymbol{v}_{ci}^{\mathrm{t}} (\boldsymbol{x} - \boldsymbol{m}_c)]^2}{e_{ci}} \tag{11}$$

ただし，$f'_0(\boldsymbol{x}, c) = a_c$ である．共分散行列の固有値 $e_{ci}$ はすべて正なので，数列 $f'_k(\boldsymbol{x}, c)$ は単調増加で識別関数値に近づき，$k = d$ で識別関数値になる．これは，我々が求めている数列の条件を満たしているが，これよりも収束の速い単調増加数列があるので，それについて説明する．

式 (11) は $k+1$ 以降の項を無視しているが，ここを改め，$k+1$ 以降の固有値が一定値 $\varepsilon$

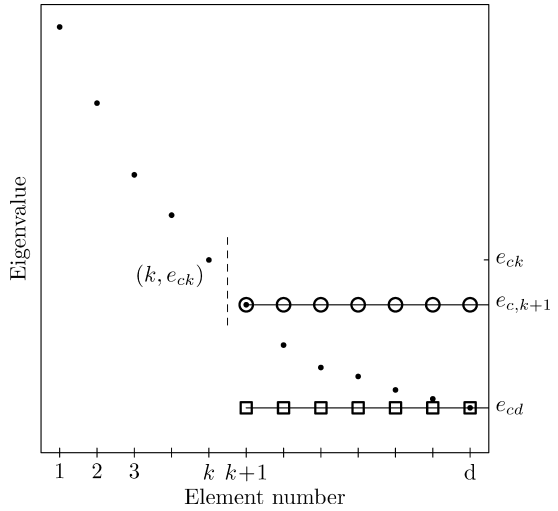



**図 1**　$\varepsilon$ の選択
Fig. 1　Choices of $\varepsilon$.

であるとした近似量 $f_k(\boldsymbol{x}, c, \varepsilon)$ を考える．

$$f_k(\boldsymbol{x}, c, \varepsilon) \equiv f'_k(\boldsymbol{x}, c) + g_k(\boldsymbol{x}, c)/\varepsilon \tag{12}$$

$$g_k(\boldsymbol{x}, c) \equiv \sum_{i=k+1}^{d} [\boldsymbol{v}_{ci}^{\mathrm{t}}(\boldsymbol{x} - \boldsymbol{m}_c)]^2 = |\boldsymbol{x} - \boldsymbol{m}_c|^2 - \sum_{i=1}^{k} [\boldsymbol{v}_{ci}^{\mathrm{t}}(\boldsymbol{x} - \boldsymbol{m}_c)]^2 \tag{13}$$

式 (13) の変形において，固有ベクトルの規格直交性を利用して，和の範囲を式 (11) と同じになるようにしている．したがって，$f_k(\boldsymbol{x}, c, \varepsilon)$ は，$f'_k(\boldsymbol{x}, c)$ と $g_k(\boldsymbol{x}, c)$ を使って逐次的に計算することができ，そのさい，$\boldsymbol{v}_{ci}^{\mathrm{t}}(\boldsymbol{x} - \boldsymbol{m}_c)$ の計算は両者に共通で 1 回だけで済むことに注意しておこう．

数列 $f'_k(\boldsymbol{x}, c)$ は，$f_k(\boldsymbol{x}, c, \varepsilon)$ の $\varepsilon$ を $\infty$ とおいた特別な場合である．$\varepsilon = \varepsilon(k) = e_{k+1}$ としたときに，最速の単調増加数列が得られ，それを $f_k(\boldsymbol{x}, c)$ と定義する：

$$f_k(\boldsymbol{x}, c) \equiv f'_k(\boldsymbol{x}, c) + g_k(\boldsymbol{x}, c)/e_{c,k+1} \tag{14}$$

この $\varepsilon$ のとり方を**図 1** の○で模式的に示す．明らかに次の不等式が成り立つ．

$$f_k(\boldsymbol{x}, c) \geq f'_k(\boldsymbol{x}, c) \tag{15}$$

また，$f_k(\boldsymbol{x}, c) \leq f(\boldsymbol{x}, c)$ であり，$k \to d-1$ のとき $f(\boldsymbol{x}, c)$ に収束するので，次の関係がある．

$$f_0(\boldsymbol{x}, c) \leq f_1(\boldsymbol{x}, c) \leq \cdots \leq f_{d-1}(\boldsymbol{x}, c) = f(\boldsymbol{x}, c) \tag{16}$$

これの厳密な証明は付録 A.1 で与えられている．式 (15) と式 (16) から，$f_k(\boldsymbol{x}, c)$ は，$f'_k(\boldsymbol{x}, c)$ よりも速やかに識別関数値に収束する単調増加数列であることが分かる．

我々の目的は，近似なしで最速の二次判別分析を行うことであるが，式 (14) は，$k < d$ の $f_k(\boldsymbol{x}, c)$ の値をもって識別関数値とする近似に使われている[5]．より広く使われる修正ベイズ近似[1] は，$\varepsilon = e_{cd}$ とおくことで得られる（図 1 の□）．この場合の数列

$$f''_k(\boldsymbol{x}, c) \equiv f'_k(\boldsymbol{x}, c) + g_k(\boldsymbol{x}, c)/e_{cd} \tag{17}$$

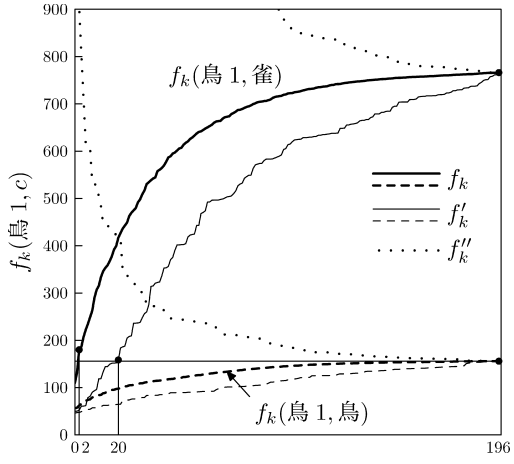
は，単調減少であることが同様に証明できる．

$f_k(\boldsymbol{x}, c)$ と $f'_k(\boldsymbol{x}, c)$ と $f''_k(\boldsymbol{x}, c)$ の収束の様子を**図 2** に 3 通り示す．これらは，実際に ETL9G[6] の文字画像から抽出された稜線特徴量[7] を使って計算したものである．図 2 の (a)，(b) ともに，カテゴリ「鳥」の 1 番目のサンプル「鳥 1」の特徴量についての収束数列を表している（$f_k(\boldsymbol{x}_{\text{鳥}1}, \text{雀})$ を簡単に $f_k(\text{鳥}\,1, \text{雀})$ と記している）．縦軸の原点は，(a) と (b) で共通の適当な値をとっている．図 2 (a) には，数列 $f_k(\text{鳥}\,1, c)$ が，$c =$ 雀（太い実線）と $c =$ 鳥（太い破線）に対して描かれている．細線は $f'_k(\text{鳥}\,1, c)$ を表し，点線は $f''_k(\text{鳥}\,1, c)$ を表している．図 (b) は，同じサンプルに対して，カテゴリ $c =$ 烏! と $c =$ 鳥 の場合が描かれている（「鳥」と「烏」の違いを明確にするために「烏」には「!」を付して「烏!」と記している）．いずれの場合においても，太線の $f_k(\boldsymbol{x}, c)$ は細線の $f'_k(\boldsymbol{x}, c)$ よりも早く収束していることが分かる．

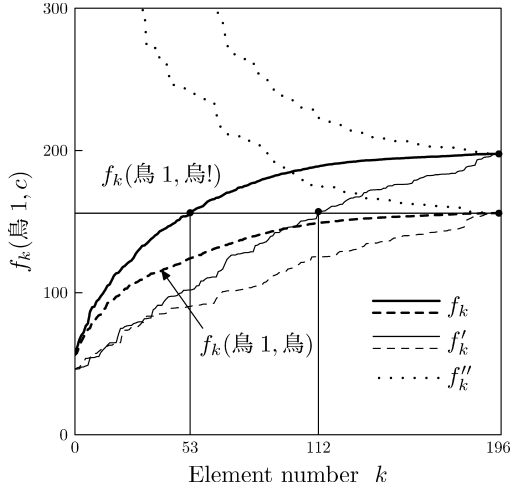
### 3.2　アルゴリズム

式 (14)，ならびに式 (11)，(13) の関係を利用して，識別計算は**図 3** のように効率的に行うことができる．ステップ 10 の固有ベクトルと変位ベクトルの内積を計算するところが最も時間がかかるところである．ステップ 14 で計算途中の識別関数値 $f$ が仮の最小値 $f_{\min}$ よりも上回った時点で $c'$ に関する計算を打ち切っている．これにより計算時間を大幅に減らすことができる．

この様子を再び図 2 で具体的に見てみる．図 2 (a) は一般的な例として，カテゴリ「鳥」と「雀」の数列を表している．識別関数値 $f(\text{鳥}\,1, \text{鳥})$ がすでに求まっているとする．数列 $f_k(\text{鳥}\,1, \text{雀})$ は $k = 2$ において，$f(\text{鳥}\,1, \text{鳥})$ を超えているので，この時点で $f(\text{鳥}\,1, \text{雀}) > f(\text{鳥}\,1, \text{鳥})$ と結論できる．同じ結論を得るために，$f'_k(\text{鳥}\,1, \text{雀})$ を使うと，$k = 20$ まで計算する必要がある．さらに，$f''_k(\text{鳥}\,1, \text{雀})$ では，$k = d - 1 = 195$ まで，すなわち，すべての項を計算する必要がある．




(a) $c =$ 鳥　vs.　$c =$ 雀




(b) $c =$ 鳥　vs.　$c =$ 烏!

**図 2**　識別関数値への収束数列
Fig. 2　Sequences converging to discriminant functions.

$c$: category number
$\boldsymbol{m}_c$: mean vector
$e_{ci}$: eigenvalue of the covariance matrix
$\boldsymbol{v}_{ci}$: eigenvector of the covariance matrix
$i$: element number of the i-th biggest eigenvalue
$d$: dimensionality of feature vector

```
1.    input x
2.    c = first category
3.    f_min = f(x, c)
4.      c' = next category
5.      IF c' = "end," GOTO 20.
6.      f' = −2 ln P_c' + ln |Σ_c'|
7.      g = |x − m_c'|^2
8.      f = f' + g/e_c'1
9.      i = 1
10.       w = [v^t_c'i (x − m_c')]^2
11.       f' = f' + w/e_c'i
12.       g = g − w
13.       f = f' + g/e_c',i+1
14.       IF f > f_min, GOTO 4.
15.       i = i + 1
16.       IF i < d, GOTO 10.
17.     c = c'
18.     f_min = f
19.     GOTO 4.
20.   output c
```

**図 3**　効率的な識別アルゴリズム
Fig. 3　Fast classification algorithm.

図 2 (b) は特別な例として，類似文字対の「鳥」と「烏!」についての数列を表している．図 2 (b) の縦軸のスケールが図 2 (a) の 3 倍になっていることに注意されたい．破線は図 2 (a) と同じものであり，$f_k$(鳥 1, 雀) の代わりに $f_k$(鳥 1, 烏!) が太い実線で描かれている．識別関数値 $f$(鳥 1, 烏!) が $f$(鳥 1, 鳥) を上回っているので，このサンプル 鳥 1 が「烏!」に誤認識されることはないが，その差はわずかである．このような繊細な場合でも，$f$(鳥 1, 鳥) が先に求まっているとき，$f_k$(鳥 1, 烏!) を使えば $k = 53$ まで計算したところで，$f$(鳥 1, 烏!) $> f$(鳥 1, 鳥) と結論付けられる．同じ結論を得るために，$f'_k$(鳥 1, 烏!) では $k = 112$ まで計算する必要がある．

$f$(鳥 1, 鳥) が求まるっているとき，計算を打ち切ってよい成分番号を $k^*$ とし，その頻度分布を**図 4** に示す．実線で結ばれた大きな点，および破線で結ばれた小さな点は，それぞ

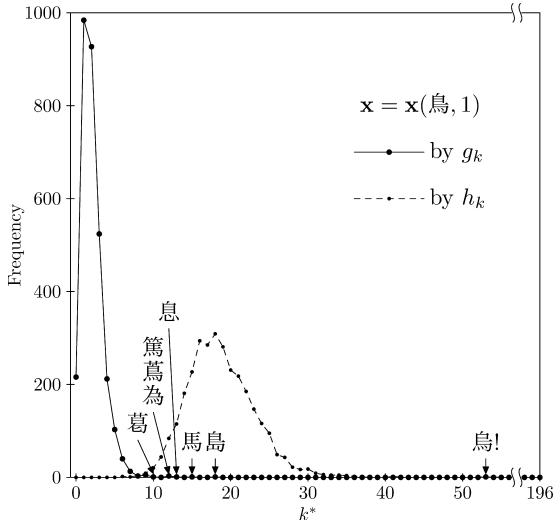



**図 4**　$k^*$ の分布
Fig. 4　Distribution of $k^*$.

れ，$f_k$，$f'_k$ による $k^*$ の頻度数を表している．頻度数の合計は 3,035 である．$f_k$ による分布は，ポアソン分布に似ていて，その平均は 2.1 である．$k^*$ の大きな領域の頻度数はほとんど 0 であるが，ところどころに「鳥」に似た字種のカテゴリが生じる．それらの字種を矢印を付けて表している．$f'_k$ による $k^*$ の平均は，18.8 である．

図 2 の 2 つの例で，$f$(鳥 1, 鳥) が先に求まっていると仮定したが，調べる順序が $f$(鳥 1, 雀) → $f$(鳥 1, 鳥) と逆になっていたとすると，両方の識別関数値を計算しなければならなくなる．このように，計算時間は調べるカテゴリの順序に依存する．最悪の場合は，識別関数の大きい順に候補カテゴリを調べていった場合で，このときには全カテゴリに対して識別関数値を計算しなければならなくなる．こういう事態を避けるため，ステップ 2 の初期値は，なるべく識別関数値が小さいものを選ぶべきである．認識率を測定する実験においては，真のカテゴリが分かっているので，それを $c$ の初期値として設定すればよい．さらに，更新が起こった時点で誤認識が起こったことが確定するので，その入力に対する計算をそこで打ち切ってよい．このようにして時間短縮を図ることは，特徴量の研究開発者にとっては有益であるが，真のカテゴリが未知である実際の場面では使えない．

実際の応用の場面では，真のカテゴリは分からないので，次の節で述べるような近似によって，小さな識別関数値のカテゴリを選ぶ方法が考えられる．主記憶が全辞書のデータを収容できるなら，付録 A.2 のようなアルゴリズムも考えられる．日本語の文字認識の場合には，辞書データは 10 G バイトになるので，現時点で手軽に手に入るハードウェアで，このアルゴリズムを効率的に適用することは難しい．

### 3.3　対角近似による初期値設定

小さい識別関数値を持つカテゴリを初期値として設定する最も簡単な方法は，事前確率によるものであろう．すなわち，$P_c$ の値が一番大きいカテゴリを初期値とする．この設定法は，カテゴリの出現分布に偏りが大きい場合に有効であり，わずかな計算量で計算できる．

カテゴリの出現分布に大きな偏りがないとき，事前確率による推定は有効でなくなる．そのような場合のために，それほど計算量が大きくない近似として，対角近似を考える．すなわち，共分散行列の非対角要素を無視する方法である．この近似で使われる変数の肩に D をつけてこれまでの変数と区別して書くと，式 (10) と (7) に対応する式は次のようになる．

$$f^{\mathrm{D}}(\boldsymbol{x}, c) = a_c^{\mathrm{D}} + \sum_{i=1}^{d} \left( \frac{x_i - m_{ci}}{\sigma_{ci}} \right)^2 \tag{18}$$

$$a_c^{\mathrm{D}} = -2 \ln P_c + \ln \prod_{i=1}^{d} \sigma_{ci} \tag{19}$$

ここで，$\sigma_{ci}$ は，共分散行列 (2) の対角成分の平方根であり，成分の添え字 $i$ は，$\sigma_{ci}$ が降順に並ぶようにふられている．式 (18) の右辺第 2 項は，重み付きユークリッド距離である．

$f^{\mathrm{D}}(\boldsymbol{x}, c)$ の最小値を求める計算には前述の効率の良いアルゴリズムがそのまま使える．そのときの初期値としては事前確率を使えばよい．ただし，対角近似の場合には，$\epsilon = \sigma^2_{c,k+1}$ とおくよりも $\epsilon = \infty$ とおいた $f'_k$ 型の数列を使った方が効率が良い．なぜなら，前者ではノルム $|\boldsymbol{x} - \boldsymbol{m}_c|^2$ を最初に計算する必要があり，対角近似の場合にはそれが計算の半分近くを占めてしまうからである．対角近似により，次元数の 2 乗に比例していた計算量が次元数の 1 乗に比例するようになる．したがって，対角近似は，次元数が大きいときに相対的に有利になる．

## 4. 実 験 結 果

いくつかのデータを使って認識実験に行い，その計算時間を測定し，本アルゴリズムの効果を調べた．用いたシステムの環境は，動作周波数 1.80 GHz の 2 つの CPU，2 G バイト

**表 1**　実験結果
Table 1　Experimental results.

(a) データ仕様

| 項目 \ データ名 | Etl9g 3036 | Etl9g 300 | Isolet | Letter | Landsat | Musk 2 |
|---|---|---|---|---|---|---|
| 1. カテゴリ数 | 3,036 | 300 | 26 | 26 | 6 | 2 |
| 2. 次元数 | 196 | 196 | 617 | 16 | 36 | 166 |
| 3. サンプル数 | 606,270 | 59,971 | 7,797 | 20,000 | 6,425 | 6,598 |
| 4. カテゴリの分布 | ほぼ均一 | ほぼ均一 | ほぼ均一 | やや不均一 | 不均一 | 大いに不均一 |

(b) 認識率

| 項目 \ データ名 | Etl9g 3036 | Etl9g 300 | Isolet | Letter | Landsat | Musk 2 |
|---|---|---|---|---|---|---|
| 1. パラメータ $\nu_1, \nu_2$ | 90, 65 | 70, 60 | 400, 140 | 0.2, 0.9 | 20, 10 | 0, 0 |
| 2. 認識率，二次判別 | 99.55 | 99.93 | 96.87 | 88.53 | 84.09 | 90.91 |
| 3. 認識率，対角近似 | 97.94 | 99.51 | 89.69 | 64.13 | 78.77 | 79.58 |

(c) 認識率測定実験における計算時間（入力サンプルあたり）

| データ名 | Etl9g 3036 | | Etl9g 300 | | Isolet | | Letter | | Landsat | | Musk 2 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 識別法 \ 単位 | msec | (%) | msec | (%) | msec | (%) | msec | (%) | msec | (%) | msec | (%) |
| 1. 全識別関数 | 484 | (100) | 46.6 | (100) | 41.2 | (100) | 0.041 | (100) | 0.037 | (100) | 0.24 | (100) |
| 2. 識別関数 | 483 | (100) | 46.6 | (100) | 40.6 | ( 98) | 0.039 | ( 94) | 0.036 | ( 97) | 0.23 | (100) |
| 3. $f'_k(\boldsymbol{x}, c)$ | 65 | ( 12) | 6.2 | ( 13) | 8.8 | ( 21) | 0.023 | ( 55) | 0.017 | ( 45) | 0.17 | ( 73) |
| 4. $f_k(\boldsymbol{x}, c)$ | 19 | ( **4**) | 1.8 | ( **4**) | 3.9 | ( **10**) | 0.017 | ( **42**) | 0.015 | ( **40**) | 0.16 | ( **68**) |

(d) 実用時における初期値設定法と計算時間（入力サンプルあたり）

| データ名 | Etl9g 3036 | | Etl9g 300 | | Isolet | | Letter | | Landsat | | Musk 2 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 初期値設定法 \ 単位 | msec | (%) | msec | (%) | msec | (%) | msec | (%) | msec | (%) | msec | (%) |
| 1. 無作為 | 44 | (100) | 5.8 | (100) | 10.5 | (100) | 0.028 | (100) | 0.023 | (100) | 0.20 | (100) |
| 2. 事前確率 | 44 | (101) | 5.9 | (101) | 10.5 | (100) | 0.028 | (100) | 0.024 | (105) | 0.18 | (**90**) |
| 3. 対角近似 | 28 | ( **63**) | 2.5 | (**43**) | 4.3 | (**41**) | 0.028 | ( **97**) | 0.019 | ( **83**) | 0.19 | ( 91) |
| 3.1. 初期値設定 | 6 | ( 15) | 0.5 | ( 8) | 0.1 | ( 1) | 0.007 | ( 26) | 0.004 | ( 16) | 0.01 | ( 2) |
| 3.2. 本識別 | 22 | ( 48) | 2.0 | ( 35) | 4.2 | ( 40) | 0.021 | ( 71) | 0.015 | ( 67) | 0.18 | ( 89) |

の主記憶を持つパソコンの Java 仮想マシン環境である．

### 4.1　データ仕様

**表 1** (a) に実験に用いる特徴量データの仕様を示す．Etl9g 3036 および Etl9g 300 は，いずれも文字画像データベース ETL9G [6] の文字画像から抽出した稜線特徴量[7] である．ETL9G は，3,036 字種，200 サンプル/字種の多値の文字画像データからなる．ここでは，そのうち誤記などの不適切なサンプル 930 件を除外してある．Etl9g 3036 はその全字種のデータであり，Etl9g 300 は字種番号 1001 から 1300 までのデータである．そのほかの 4 つのデータセット Isolet，Letter，Landsat，Musk2 は，UCI 機械学習リポジトリ[8] のものである．数あるデータセットの中からこれらを選んだ理由は，計算時間が正確に測定できるように大きなデータセットであること，いろいろなカテゴリ数と次元数を含んでいること，の 2 つである．表 1 (a) の行 4 は，カテゴリごとのサンプル数が均一になっているか否かを定性的に表している．

### 4.2　認　識　率

認識実験を行うために，データセット内のサンプルを 10 個のグループに分ける．このさい，カテゴリのサンプル数がグループ間でなるべく同じになるようにする．10 個のグループのうちの 1 つのグループを未知のテストデータとして使い，残りを訓練データとして使う．テストデータと訓練データの役割を代えて，計 10 回の認識実験を行った．表 1 (b) は，10 回の認識実験で得られた認識率の平均である．二次判別のほかに対角近似についても同

じように認識率を求めた．パラメータは二次判別による認識率が最高になるように決めた．Etl9g 3036 において除外した 930 件のデータをすべて誤認識として，認識率を計算し直すと，99.40%となり，他の報告[1),9)] とほぼ一致する．

### 4.3　認識率測定実験における計算時間

認識率測定実験における各アルゴリズムの識別処理にかかる計算時間を入力サンプルあたりの時間として表 1 (c) に示す．これらの計算時間には，識別計算だけが含まれ，辞書データの作成や入出力に関する時間は含まれていない．初期値はサンプルの真のカテゴリとしている．行 1 と行 2 は，識別関数値の計算を処理単位とし，識別関数値の計算中にそれ以降の計算の要不要を判断しない，従来の計算方法である．行 1 は，すべてのカテゴリに対して識別関数値を求めた計算時間である．この計算時間は，(カテゴリ数) $\times$ (次元数)$^2$ に比例する．この時間を基準として，() 内の百分率が計算されている．行 2 は，誤認識が起こったとき，その入力に関する以降の計算を省いた場合の計算時間である．行 3 と行 4 は，識別関数値を計算している最中でも，それ以降の計算が不要と判断されるときは，そのカテゴリに関する計算を打ち切る方法である．行 3 は，数列 $f'_k(\boldsymbol{x}, c)$ を使った場合で，行 4 は，数列 $f_k(\boldsymbol{x}, c)$ を使った，我々が提案する効率の良いアルゴリズムによる計算時間である．

すべてのデータセットについて，行 4 の提案方法の計算時間が最短である．特に，カテゴリ数と次元数が大きい場合に，大きな計算時間の削減が得られる．Etl9g 3036 の場合には 4% に短縮されている．

全識別関数の計算量は，カテゴリ数を $C$ として，$Cd^2$ に比例するが，本アルゴリズムで必要となる実質的な次元数は，図 2 と図 4 で見たように，$d$ よりもかなり小さくて済むので，計算時間の短縮が可能になる．

### 4.4　実用時における初期値設定法と計算時間

表 1 (d) は，初期値のとり方の影響を示したものである．行 1 は入力ごとに乱数を使って無作為に初期値のカテゴリを設定したものである．この場合の計算時間を基準として，他の方法の計算時間の () 内の百分率を計算している．行 2 は事前確率（式 (8)）が最大のカテゴリを初期値として設定した場合である．同率 1 位のカテゴリが複数存在するときには，その中から入力ごとに乱数を使って無作為に選んだ．行 3 は対角近似による識別関数値が最小のものを初期値とした場合である．行 3.1 と行 3.2 はその内訳で，それぞれ，対角近似自体にかかる時間と本識別にかかる時間である．行 1 と行 2 の初期値設定法での計算時間はわずかであり，計算時間に入れていない．本識別では，すべて，表 1 (c) 行 4 と同様に数列 $f_k$ を使って計算している．

表 1 (d) から，行 2 の事前確率による初期値設定はあまり効果がなく，しばしば逆効果になることが分かる．唯一効果が認められるのは，Musk2 の場合である．Musk2 では，サンプルが 85 : 15 で 2 つのカテゴリに分かれていて，事前確率の相違が大きくなっているからである．

行 3 の対角近似による初期値設定は，つねに計算時間の削減をもたらしている．特に次元数（表 1 (a) の行 2）の大きい Isolet では最も大きい時間短縮になっている．Etl9g 300 が Etl9g 3036 より時間短縮が大きいのは，認識率（表 1 (b) の行 3）が高いことによる．Letter，Landsat，Musk2 での時間短縮が小さい理由としては，次元数が小さいこと，対角近似の認識率が低いこと，の 2 つがあげられる．次元数が小さいときは，初期値設定自体にかかる時間も相対的に無視できなくなってくる．本識別にかかる時間だけを問題にするなら，行 3.2 から分かるように，対角近似がつねに最短である．

表 1 (c) の行 4 と表 1 (d) の行 3.2 を比べてみると，前者の計算時間の方が 1 割ほど少なくなっている．これは，前者の場合には，実用時には知りえない最良の初期カテゴリとなる真のカテゴリの情報を使っていることと，正認識か誤認識かを調べているだけで 1 位候補を計算しているわけではないこと，の 2 つの理由による．

なお，これら 3 つの初期値設定法は，あくまでも仮のカテゴリを決めているだけであり，計算時間には影響するが，最終的な認識率に影響しない．認識率は，どの設定法でも表 1 (b) 行 2 の値になる．

## 5. 結　　論

二次判別分析において，識別関数値に単調増加で収束する数列を逐次的に計算していく過程において，それ以降の計算が不要であると判断できる場合がある．不要な計算を行わないことで，効率の良い計算が可能になる．最も収束の速い単調増加数列 $f_k$ は，共分散行列の $k+1$ 番目に大きい固有値で，それより小さい固有値を置き換えることにより得られる．本アルゴリズムは，ほとんどの場合に有効であるが，特にカテゴリ数が大きい識別問題で大きな時間短縮をもたらす．手書き文字認識の特徴量データを使った認識率測定実験において，従来法と比べ 4% に計算時間が削減された．実用時においては，初期値カテゴリを対角近似によって適切に設定することが計算時間の短縮に効果がある．

**謝辞**　付録 A.2 のアルゴリズムに関して示唆をいただきました，京都大学数理解析研究所の藤重悟教授に深く感謝いたします．正則化判別分析と階層ベイズ推定について，有益な議論をしていただいた林千里さんと渡辺秀人さんに感謝します．

## 参 考 文 献


1) Kimura, F., Wakabayashi, T., Tsuruoka, S. and Miyake, Y.: Improvement of handwritten Japanese character recognition using weighted direction code histogram, *Pattern Recognition*, Vol.30, No.8, pp.1329–1337 (1997).
2) Brown, P.J., Fearn, T. and Haque, M.S.: Discrimination with many variables, *J. Am. Stat. Asoc.*, Vol.94, No.448, pp.1320–1329 (1999).
3) Friedman, J.H.: Regularized Discriminant Analysis, *J. Am. Stat. Asoc.*, Vol.84, pp.165–175 (1989).
4) Hoffbeck, J.P. and Landgrebe, D.A.: Covariance matrix estimation and classification with limited training data, *IEEE Trans. PAMI*, Vol.18, No.7, pp.763–767 (1996).
5) 木村文隆，高階健治，鶴岡信治，三宅康二：2 次識別関数のピーキング現象とその防止に関する考察，信学論（D），Vol.J69-D, No.9, pp.1328–1334 (1986).
6) 斉藤泰一，山田博三，山本和彦：JIS 第 1 水準手書漢字データベース ETL9 とその解析，信学論（D），Vol.J68-D, No.4, pp.757–764 (1985).
7) 鈴木道孝，林　千里，伊藤彰義：稜線特徴量により多値手書き文字の認識，*PRMU*, Vol.106, No.606, pp.85–90 (2007).
8) Asuncion, A. and Newman, D.J.: UCI Machine Learning Repository [http://www.ics.uci.edu/˜mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science (2007).
9) Kato, N., Suzuki, M., Omachi, S., Aso, H. and Nemoto, Y.: A handwritten character recognition system using directional element feature and asymmetric Mahalanobis distance, *IEEE Trans. PAMI*, Vol.21, No.3, pp.258–262 (1999).


## 付　　録

### A.1　式 (16) の証明

まず，$0 \leq k \leq d-2$ に対して：

$$
\begin{aligned}
f_{k+1}(\boldsymbol{x},c) - f_k(\boldsymbol{x},c) &= f'_{k+1}(\boldsymbol{x},c) - f'_k(\boldsymbol{x},c) + \frac{g_{k+1}(\boldsymbol{x},c)}{e_{c,k+2}} - \frac{g_k(\boldsymbol{x},c)}{e_{c,k+1}} \\
&= \frac{[\boldsymbol{v}_{c,k+1}^{\mathrm{t}}(\boldsymbol{x}-\boldsymbol{m}_c)]^2}{e_{c,k+1}} \\
&\quad + \frac{1}{e_{c,k+2}}\left(|\boldsymbol{x}-\boldsymbol{m}_c|^2 - \sum_{i=1}^{k+1}[\boldsymbol{v}_{ci}^{\mathrm{t}}(\boldsymbol{x}-\boldsymbol{m}_c)]^2\right) \\
&\quad - \frac{1}{e_{c,k+1}}\left(|\boldsymbol{x}-\boldsymbol{m}_c|^2 - \sum_{i=1}^{k}[\boldsymbol{v}_{ci}^{\mathrm{t}}(\boldsymbol{x}-\boldsymbol{m}_c)]^2\right) \\
&= \left(\frac{1}{e_{c,k+2}} - \frac{1}{e_{c,k+1}}\right) \times \left(|\boldsymbol{x}-\boldsymbol{m}_c|^2 - \sum_{i=1}^{k+1}[\boldsymbol{v}_{ci}^{\mathrm{t}}(\boldsymbol{x}-\boldsymbol{m}_c)]^2\right) \\
&= \frac{e_{c,k+1}-e_{c,k+2}}{e_{c,k+1}e_{c,k+2}} \sum_{i=k+2}^{d}[\boldsymbol{v}_{ci}^{\mathrm{t}}(\boldsymbol{x}-\boldsymbol{m}_c)]^2 \geq 0 \qquad (20)
\end{aligned}
$$

また，$k = d-1$ に対して：

---

$c$: category number
$\boldsymbol{m}_c$: mean vector
$e_{ci}$: eigenvalue of the covariance matrix
$\boldsymbol{v}_{ci}$: eigenvector of the covariance matrix
$i$: element number of the i-th biggest eigenvalue
$Q$: priority queue of data in ascending order of first arguments
$d$: dimensionality of feature vector

1. input $\boldsymbol{x}$
2. FOR each c
3. $f' = -2\ln P_c + \ln|\Sigma_c|$
4. $g = |\boldsymbol{x}-\boldsymbol{m}_c|^2$
5. $f = f' + g/e_{c1}$
6. $Q$.add$(f, f', g, c, 0)$
7. END-FOR
8. WHILE
9. $(f, f', g, c, i) = Q$.getFirst()
10. $i = i + 1$
11. IF $i = d$, GOTO 18.
12. $w = [\boldsymbol{v}_{ci}^{\mathrm{t}}(\boldsymbol{x}-\boldsymbol{m}_c)]^2$
13. $f' = f' + w/e_{ci}$
14. $g = g - w$
15. $f = f' + g/e_{c,i+1}$
16. $Q$.add$(f, f', g, c, i)$
17. END-WHILE
18. output $c$

---

**図 5**　高速識別アルゴリズム 2
Fig. 5　Fast classification algorithm 2.

$$
\begin{aligned}
f_{d-1}(\boldsymbol{x},c) &= f'_{d-1}(\boldsymbol{x},c) + \frac{g_{d-1}(\boldsymbol{x},c)}{e_{cd}} \\
&= a_c + \sum_{i=1}^{d-1} \frac{[\boldsymbol{v}_{ci}^{\mathrm{t}}(\boldsymbol{x}-\boldsymbol{m}_c)]^2}{e_{ci}} + \frac{1}{e_{cd}} \left( |\boldsymbol{x}-\boldsymbol{m}_c|^2 - \sum_{i=1}^{d-1} [\boldsymbol{v}_{ci}^{\mathrm{t}}(\boldsymbol{x}-\boldsymbol{m}_c)]^2 \right) \\
&= a_c + \sum_{i=1}^{d-1} \frac{[\boldsymbol{v}_{ci}^{\mathrm{t}}(\boldsymbol{x}-\boldsymbol{m}_c)]^2}{e_{ci}} + \frac{[\boldsymbol{v}_{cd}^{\mathrm{t}}(\boldsymbol{x}-\boldsymbol{m}_c)]^2}{e_{cd}} \\
&= f(\boldsymbol{x},c) \qquad (21)
\end{aligned}
$$

ゆえに，式 (16) が証明された．

### A.2　初期カテゴリを計算しない方法

主記憶にすべての辞書データが収容できる場合には，**図 5** のアルゴリズムも有力である．これは，すべてのカテゴリを対象に，数列値の低いものから順に更新していくものである．これによると，内積の計算の回数は必要最小限で済むが，最小値を探すために余計な計算が必要となる．これに対して，本文のアルゴリズムは，初期値選定や仮の最小値の更新に関する余計な計算がかかるが，キャッシュメモリにロードされたデータをバッチ処理で有効に使うなどの，融通性に勝る．今後ハードウェアが進歩したときに，どちらのアルゴリズムが総合的に勝るかは興味のあるところである．

(平成 20 年 9 月 25 日受付)
(平成 21 年 5 月 13 日採録)

**鈴木　道孝**（正会員）

昭和 50 年千葉大学理学部物理学科卒業．昭和 52 年東北大学大学院博士前期課程修了．昭和 58 年同大学院博士後期課程修了．昭和 60 年（学）岩崎学園入社．平成 3 年（株）エヌ・ケー・エクサ入社．平成 20 年日本大学理工学研究所研究員．理学博士．

**伊藤　彰義**

昭和 41 年日本大学理工学部電気工学科卒業．昭和 43 年同大学大学院修士課程修了．昭和 46 年同大学大学院博士課程修了．同年同大学理工学部助手．昭 62〜63 年 CMU 客員助教授．平成元年日本大学電子工学科教授．平成 7〜9 年応用物理学会常務理事．日本応用磁気学会論文賞，同学会業績賞受賞．矢崎学術賞受賞．手書き文字認識・超高密度情報記録・薄膜の研究に従事．工学博士．