

線型多様体間距離に基づくパターン識別と学習

堀田 政二[†]

[†] 東京農工大学 〒184-8588 東京都小金井市中町 2-24-16

E-mail: †s-hotta@cc.tuat.ac.jp

あらまし 本稿では、線型多様体間距離を利用したパターン識別と学習について考える。はじめに、線型多様体間距離が未知データとの二乗誤差を最小にする訓練データの線型結合を利用した識別から容易に導かれることを示す。次に、距離に基づく学習の一種である一般化学習ベクトル量子化 (Generalized Learning Vector Quantization, GLVQ) を線型多様体間距離に基づくパターン識別に適用した場合の学習則を導く。手書き数字を用いた実験により、多様体間距離を利用することで低いエラー率を達成でき、学習によって識別精度が改善されることを示す。

キーワード パターン識別, 線型多様体, 一般化学習ベクトル量子化

Patten Classification and Learning Based on Distance Between Linear Manifolds

Seiji HOTTA[†]

[†] Tokyo Univ. of Agri. and Tech., 2-24-16 Naka-cho, Koganei-shi, Tokyo, 184-8588 Japan

E-mail: †s-hotta@cc.tuat.ac.jp

Abstract This report describes pattern classification and learning based on a distance between linear manifolds. First, a distance between linear manifolds is derived by expanding the concept of pattern classification that uses the linear combination of training samples as the most similar prototype for an input sample. Next, generalized learning vector quantization is applied to manifold-based classification to reduce error rates and memory requirement. Experimental results on a handwritten digit dataset show that manifold-based classification and learning can improve accuracy with small memory requirement.

Key words pattern classification, linear manifold, generalized learning vector quantization

1. はじめに

部分空間法は CLAFIC で代表される類似度型と、投影距離法のような距離型に大別できる [1]。このうち類似度型は、未知データも部分空間で表現できる場合に利用できる相互部分空間法 [2] へと拡張され、その後さらに理論拡張が進められ顔認識や物体認識において有効性が確認されている [3]。一方距離型では、投影距離法 [4] が提案されてから相互部分空間法のように線型多様体 (アフィン部分空間) 同士の距離としての拡張は行われなかったようである。しかし、文字認識において変形不変性を実現するために提案された接距離 (Tangent Distance, TD) [5] で線型多様体間の距離が利用されたのをきっかけに、joint manifold distance による動画像中の顔のクラスタリング [6] や線型多様体距離に基づく顔認識 [7] が提案された。しかし、これら以外に線型多様体間の距離を利用した研究を見つけることは難しく、研究人口も少ないようである。

これにはさまざまな理由が考えられるが、一つは投影距離法

があまり知られていないことが考えられる。投影距離法は、各クラスで線型多様体を構築し、未知データの線型多様体への垂線の長さ (残差長) が最小となるクラスへ未知データを分類する簡便な識別法である。しかし、局所部分空間法 (Local Subspace Classifier, LSC) [8] や k -nearest feature line [9] の識別則は (訓練データの選択を行う点が異なるが) 投影距離法そのものであるにも関わらず、投影距離法に関する言及はない。Oja の本 [10] でも線型多様体については述べられているが、内積や射影長に基づく識別が主題であり、距離に基づく識別については触れていない。しかし距離に基づく識別を考えることは、これまでに提案されてきた距離関数や学習を識別に取り入れられる可能性があるため有用といえる。そこで本稿では、線型多様体間距離を利用したパターン識別と学習について考える。はじめに、線型多様体間距離が未知データとの二乗誤差を最小にする訓練データの線型結合を利用した識別から容易に導かれることを示す。次に、距離に基づく学習の一種である一般化学習ベクトル量子化 (Generalized Learning Vector Quantization, GLVQ) [11] を

線型多様体間距離に基づくパターン識別に適用した場合の学習則を導く。手書き数字を用いた実験により、多様体間距離を利用した方が低いエラー率を達成でき、学習によってエラー率を低減できることを示す。

2. 二乗誤差最小パターンとの距離に基づく識別

線型多様体間距離の導出を容易にするために、はじめに未知データとの二乗誤差が最小となるような訓練データの線型結合を各クラスで作成し、最も誤差が小さくなるクラスに未知データを分類するという問題を考える (図 1 参照)。

2.1 定式化

d 次元の未知データを $\mathbf{q} = (q_1 \cdots q_d)^\top \in \mathbb{R}^d$, クラス j ($j = 1, \dots, C$) に属する n_j 個 ($n_j \leq d$) の訓練データのうち、第 i 訓練データを $\mathbf{x}_i^j = (x_{i1}^j \cdots x_{id}^j)^\top \in \mathbb{R}^d$ ($i = 1, \dots, n_j$) とする。また、 \mathbf{x}_i^j を並べた行列を $\mathbf{X}_j = (\mathbf{x}_1^j | \mathbf{x}_2^j | \cdots | \mathbf{x}_{n_j}^j) \in \mathbb{R}^{d \times n_j}$ とする。これらの訓練データは互いに独立であれば直交していても良い。なお本節では、記号の煩雑さを避けるために、クラスの添え字 j を適宜省略することにする。

まず、クラス j で未知データとの二乗誤差が最小となるような訓練データの線型結合を以下の最適化問題により求めることにする：

$$\begin{aligned} \min_{\mathbf{b}} \quad & \|\mathbf{q} - \mathbf{X}\mathbf{b}\|^2 = \left\| \mathbf{q} - \sum_{i=1}^n b_i \mathbf{x}_i \right\|^2 \\ \text{s.t.} \quad & \mathbf{b}^\top \mathbf{1}_n = \sum_{i=1}^n b_i = 1 \end{aligned} \quad (1)$$

ここで $\mathbf{1}_n = (1 \cdots 1)^\top \in \mathbb{R}^n$ は要素が全て 1 のベクトル、 $\mathbf{b} = (b_1 \cdots b_n)^\top \in \mathbb{R}^n$ は訓練データに対する係数ベクトルである。制約条件の $\mathbf{b}^\top \mathbf{1}_n = 1$ は、線型結合データが訓練データの張る線型多様体上に乗ることを保障するために必要である。これは制約条件を $b_1 = 1 - \sum_{i=2}^n b_i$ と変形して式 (1) の目的関数に代入して制約条件を消去すれば、式 (1) の最適化問題は

$$\min_{\beta} \|\mathbf{q} - (\mathbf{x}_1 + \mathbf{V}\beta)\|^2 = \left\| \mathbf{q} - (\mathbf{x}_1 + \sum_{i=2}^n \beta_i (\mathbf{x}_i - \mathbf{x}_1)) \right\|^2 \quad (2)$$

となることから確認できる。ここで \mathbf{V} は $\mathbf{x}_i - \mathbf{x}_1$ ($i = 2, \dots, n$) を並べた $d \times (n-1)$ の行列 $\mathbf{V} = (\mathbf{x}_2 - \mathbf{x}_1 | \mathbf{x}_3 - \mathbf{x}_1 | \cdots | \mathbf{x}_n - \mathbf{x}_1)$ であり $\beta = (\beta_1 \cdots \beta_{n-1})^\top \in \mathbb{R}^{n-1}$ は \mathbf{V} の各列に対応する係数ベクトルである。すなわち式 (2) は \mathbf{q} と線型多様体 $\mathcal{M} = \mathbf{x}_1 + \mathbf{V}\beta$ との二乗誤差を最小にする問題となっている。

式 (1) の解は、 \mathbf{q} を n 個並べた行列を $\mathbf{Q} = (\mathbf{q} | \cdots | \mathbf{q}) \in \mathbb{R}^{d \times n}$, 相関行列を $\mathbf{C} = (\mathbf{Q} - \mathbf{X})^\top (\mathbf{Q} - \mathbf{X}) \in \mathbb{R}^{n \times n}$ とすれば、ラグランジュ乗数法から

$$\mathbf{b} = \frac{\mathbf{C}^{-1} \mathbf{1}_n}{\mathbf{1}_n^\top \mathbf{C}^{-1} \mathbf{1}_n} \quad (3)$$

として閉じた形で求めることができる。もし $n > d$ の場合や、過学習を避けたい場合には、式 (1) の目的関数に罰則項を $\|\mathbf{q} - \mathbf{X}\mathbf{b}\| + \lambda \|\mathbf{b}\|^2$ ($\lambda > 0$) のように付け加えて正則化を施せば

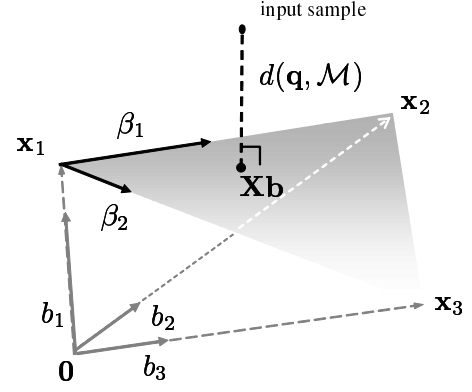


図 1 訓練データが三つの場合の未知データ \mathbf{q} と線型多様体との関係

良い。その場合にも閉じた形で解が得られ、 $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ を単位行列として \mathbf{C} の逆行列を計算する前に $\mathbf{C} + \lambda \mathbf{I}_n$ とすれば良い。識別は、 \mathbf{b} を用いてクラス j との距離を $d(\mathbf{q}, \mathcal{M}) = \|\mathbf{q} - \mathbf{X}\mathbf{b}\|^2$ と定義して、 $d(\mathbf{q}, \mathcal{M})$ が最小となるクラスへ \mathbf{q} を分類することで実現できる。なお、 \mathbf{X} として \mathbf{q} の k 近傍訓練データを用いて識別を行うものは Local Subspace Classifier (LSC) [8] と呼ばれる。

一方、式 (2) の解は、目的関数を β で偏微分して 0 と置くことにより

$$\beta = (\mathbf{V}^\top \mathbf{V})^{-1} \mathbf{V}^\top (\mathbf{q} - \mathbf{x}_1) \quad (4)$$

として求めることができる。識別は、 β を用いてクラス j との距離を $d(\mathbf{q}, \mathcal{M}) = \|\mathbf{q} - (\mathbf{x}_1 + \mathbf{V}\beta)\|^2$ と定義すれば、 $d(\mathbf{q}, \mathcal{M})$ が最小となるクラスへ \mathbf{q} を分類することで実現できる。

なお、 \mathbf{V} を $\mathbf{V}^\top \mathbf{V}$ の値の大きい固有値に対応した $n-1$ 本以下の固有ベクトルを並べた行列とすれば、 $d(\mathbf{q}, \mathcal{M})$ は $d(\mathbf{q}, \mathcal{M}) = \|\mathbf{q} - \mathbf{x}_1\|^2 - \|\mathbf{V}^\top (\mathbf{q} - \mathbf{x}_1)\|^2$ と書け、文献 [4] の投影距離法 (Projection Distance Method, PDM) と同じになる。すなわち、最小の $d(\mathbf{q}, \mathcal{M})$ は \mathbf{q} から \mathcal{M} までの垂線の長さ (残差長) で与えられることが確かめられる。また、 \mathbf{x}_1 を画像パターン、 \mathbf{V} を \mathbf{x}_1 の変形を表す接ベクトル (Tangent Vector, TV) [5] とした場合には、式 (2) は片側接距離 (one sided tangent distance, 1STD) [5] と呼ばれている。

2.2 カーネル化

二乗誤差最小パターンとの距離に基づくパターン識別は、カーネルトリック $\Phi(\mathbf{x})^\top \Phi(\mathbf{y}) = K(\mathbf{x}, \mathbf{y})$ を利用して容易にカーネル化できる ($\Phi(\cdot)$ は高次元空間への非線型写像)。カーネル化した場合、式 (1) の最適化問題は、 $\Phi(\mathbf{q})$ を n 個並べた行列を \mathbf{Q}_ϕ , $\Phi(\mathbf{x}_i)$ ($i = 1, \dots, n$) を並べた行列を \mathbf{X}_ϕ とすれば

$$\begin{aligned} \min_{\mathbf{b}} \quad & \|(\mathbf{Q}_\phi - \mathbf{X}_\phi)\mathbf{b}\|^2 \\ \text{s.t.} \quad & \mathbf{b}^\top \mathbf{1}_n = 1 \end{aligned} \quad (5)$$

と書ける。目的関数を展開し、ラグランジュ乗数法とカーネルトリックを用いれば式 (5) の解は

$$\mathbf{b} = \frac{K_{\mathbf{C}}^{-1} \mathbf{1}_n}{\mathbf{1}_n^\top K_{\mathbf{C}}^{-1} \mathbf{1}_n} \quad (6)$$

となる。ここで K_C は $n \times n$ の行列であり、その i, j 要素は

$$(K_C)_{ij} = K(\mathbf{q}, \mathbf{q}) - K(\mathbf{q}, \mathbf{x}_j) - K(\mathbf{x}_i, \mathbf{q}) + K(\mathbf{x}_i, \mathbf{x}_j) \quad (7)$$

で与えられる。正則化は $K_C + \lambda \mathbf{I}_n$ とすれば良い。カーネル投影距離 [12] と異なり、カーネル主成分分析 [13] を用いない識別ができる。

3. 線型多様体間距離に基づくパターン識別

次に、未知データも線型多様体で与えられる場合の線型多様体間の距離について考える (図 2 参照)。未知データが m 個 ($m \leq d$) の d 次元ベクトルの組で与えられたとする。そのうちの第 i ベクトルを $\mathbf{q}_i = (q_{i1} \cdots q_{id})^\top$ ($i = 1, \dots, m$) で表し、それらを並べた $d \times m$ の行列を $\mathbf{Q} = (\mathbf{q}_1 | \mathbf{q}_2 | \cdots | \mathbf{q}_m)$ とする。なお、これらは互いに独立であれば直交していなくても構わない。

3.1 定式化

\mathbf{Q} の各列に対する $m \times 1$ の係数ベクトルを $\mathbf{a} = (a_1 \cdots a_m)^\top$ とすれば、クラス j の線型多様体までの最短距離は

$$\begin{aligned} \min_{\mathbf{a}, \mathbf{b}} \quad & \|\mathbf{Q}\mathbf{a} - \mathbf{X}\mathbf{b}\|^2 \\ \text{s.t.} \quad & \mathbf{a}^\top \mathbf{1}_m = 1, \mathbf{b}^\top \mathbf{1}_n = 1 \end{aligned} \quad (8)$$

を解くことによって求めることができる。この問題もラグランジュ乗数法により閉じた形で解が求められるが、解が非常に複雑な式になってしまう。そこで二乗誤差最小パターンとの距離に基づく識別と同じように制約条件を消去することによって解を簡単に表すことを考える。

まず、一つの制約条件を消去した場合について考える。制約条件を $a_1 = 1 - \sum_{i=2}^m a_i$ として式 (8) に代入すると、式 (8) の最適化問題は

$$\begin{aligned} \min_{\boldsymbol{\alpha}, \mathbf{b}} \quad & \|\mathbf{q}_1 + \mathbf{U}\boldsymbol{\alpha} - \mathbf{X}\mathbf{b}\|^2 \\ \text{s.t.} \quad & \mathbf{b}^\top \mathbf{1}_n = 1 \end{aligned} \quad (9)$$

と書き換えることができる。ここで \mathbf{U} は $\mathbf{q}_i - \mathbf{q}_1$ ($i = 2, \dots, m$) を並べた行列 $\mathbf{U} = (\mathbf{q}_2 - \mathbf{q}_1 | \mathbf{q}_3 - \mathbf{q}_1 | \cdots | \mathbf{q}_m - \mathbf{q}_1) \in \mathbb{R}^{d \times (m-1)}$ であり $\boldsymbol{\alpha} = (\alpha_1 \cdots \alpha_{m-1})^\top \in \mathbb{R}^{m-1}$ は \mathbf{U} の各列に対応する係数ベクトルである。ここで $\mathbf{A} \in \mathbb{R}^{d \times d}$ と $\gamma \in \mathbb{R}$ を

$$\mathbf{A} = \mathbf{U}(\mathbf{U}^\top \mathbf{U})^{-1} \mathbf{U}^\top \quad (10)$$

$$\gamma = \frac{1 - \mathbf{1}_n^\top (\mathbf{X}^\top (\mathbf{I}_d - \mathbf{A}) \mathbf{X})^{-1} \mathbf{X} (\mathbf{I}_d - \mathbf{A}) \mathbf{q}_1}{\mathbf{1}_n^\top (\mathbf{X}^\top (\mathbf{I}_d - \mathbf{A}) \mathbf{X})^{-1} \mathbf{1}_n} \quad (11)$$

とおくと、式 (9) を最小にする $\boldsymbol{\alpha}$ と \mathbf{b} はラグランジュ乗数法から

$$\mathbf{b} = (\mathbf{X}^\top (\mathbf{I}_d - \mathbf{A}) \mathbf{X})^{-1} \{\mathbf{X}^\top (\mathbf{I}_d - \mathbf{A}) \mathbf{q}_1 + \gamma \mathbf{1}_n\} \quad (12)$$

$$\boldsymbol{\alpha} = (\mathbf{U}^\top \mathbf{U})^{-1} \mathbf{U}^\top (\mathbf{X}\mathbf{b} - \mathbf{q}_1) \quad (13)$$

として求めることができる。正則化が必要な場合には式 (10) と式 (13) における $\mathbf{U}^\top \mathbf{U}$ を $\mathbf{U}^\top \mathbf{U} + \lambda \mathbf{I}_{m-1}$ と変更すれば良い。この結果は後の線型多様体間距離に基づく学習で利用する。

次に二つの制約条件を消去した場合を考える。制約条件を

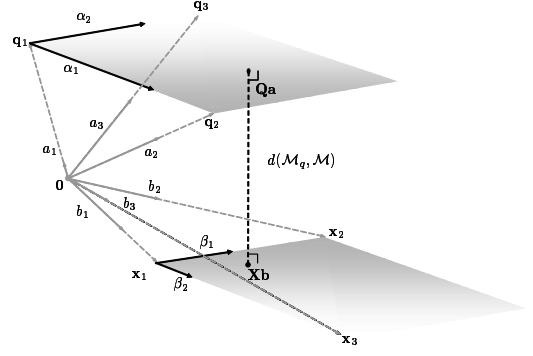


図 2 高次元空間における線型多様体間距離の概念図

$b_1 = 1 - \sum_{i=2}^n b_i$ として式 (9) に代入すると、式 (9) の最適化問題は

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \|\mathbf{q}_1 + \mathbf{U}\boldsymbol{\alpha} - (\mathbf{x}_1 + \mathbf{V}\boldsymbol{\beta})\|^2 \quad (14)$$

と書き換えることができる。式 (14) の最適解は、式 (14) をそれぞれ $\boldsymbol{\alpha}$ と $\boldsymbol{\beta}$ で偏微分して $\mathbf{0}$ において式をまとめることにより求められる。すなわち

$$\mathbf{U}_1 = \mathbf{U}^\top \mathbf{U}, \mathbf{V}_1 = \mathbf{V}^\top \mathbf{V} \quad (15)$$

$$\mathbf{U}_2 = \mathbf{U}^\top \mathbf{V}, \mathbf{V}_2 = \mathbf{V}^\top \mathbf{U} \quad (16)$$

とおけば、 $\boldsymbol{\alpha}$ と $\boldsymbol{\beta}$ は

$$\boldsymbol{\alpha} = (\mathbf{U}_1 - \mathbf{U}_2 \mathbf{V}_1^{-1} \mathbf{V}_2)^{-1} (\mathbf{U}_2 \mathbf{V}_1^{-1} \mathbf{V}^\top - \mathbf{U}^\top) (\mathbf{q}_1 - \mathbf{x}_1) \quad (17)$$

$$\boldsymbol{\beta} = (\mathbf{V}_1 - \mathbf{V}_2 \mathbf{U}_1^{-1} \mathbf{U}_2)^{-1} (\mathbf{V}^\top - \mathbf{V}_2 \mathbf{U}_1^{-1} \mathbf{U}^\top) (\mathbf{q}_1 - \mathbf{x}_1) \quad (18)$$

によって与えられる。正則化が必要な場合には式 (15) において \mathbf{U}_1 と \mathbf{V}_1 をそれぞれ $\mathbf{U}_1 + \lambda_1 \mathbf{I}_{m-1}$ と $\mathbf{V}_1 + \lambda_2 \mathbf{I}_{n-1}$ ($\lambda_1 > 0$, $\lambda_2 > 0$) に変更すれば良い。識別は、式 (9) や式 (14) の目的関数の値を線型多様体間距離 $d(\mathcal{M}_q, \mathcal{M})$ と定義し、この値が最小となるクラスを未知データのクラスとして出力することで実現できる。なお、式 (14) において未知データと訓練データの線型多様体を主成分分析で求めた場合には Inter-Subspace Distance [7], \mathbf{q}_1 と \mathbf{x}_1 を画像パターン、 \mathbf{U} と \mathbf{V} を \mathbf{q}_1 と \mathbf{x}_1 の接ベクトルとした場合、式 (14) は両側接距離 (two sided tangent distance, 2S TD) [5] と呼ばれている。

3.2 カーネル化

線型多様体間距離は二乗誤差最小パターンとの距離に基づくパターン識別と同様、容易にカーネル化できる。ここでは式 (14) を用いてカーネル化することを考える。カーネル化した場合、式 (14) の最適化問題は、 $\Phi(\mathbf{q}_i) - \Phi(\mathbf{q}_1)$ ($i = 2, \dots, m$) を並べた行列を \mathbf{U}_ϕ , $\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_1)$ ($i = 2, \dots, n$) を並べた行列を \mathbf{V}_ϕ とすれば

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \|\Phi(\mathbf{q}_1) + \mathbf{U}_\phi \boldsymbol{\alpha} - (\Phi(\mathbf{x}_1) + \mathbf{V}_\phi \boldsymbol{\beta})\|^2 \quad (19)$$

と書ける。目的関数を展開し、 $\boldsymbol{\alpha}$ と $\boldsymbol{\beta}$ でそれぞれ偏微分して $\mathbf{0}$ とおき、カーネルトリックを用いれば式 (19) の解は

$$\alpha = (K_{UU} - K_{UV}K_{VV}^{-1}K_{VU})^{-1}(K_{UV}K_{VV}^{-1}k\mathbf{q} - k\mathbf{q}) \quad (20)$$

$$\beta = (K_{VU}K_{UU}^{-1}K_{UV} - K_{VV})^{-1}(K_{VU}K_{UU}^{-1}k\mathbf{q} - k\mathbf{x}) \quad (21)$$

となる．ここで $K_{UU} \in \mathbb{R}^{m-1 \times m-1}$, $K_{VV} \in \mathbb{R}^{n-1 \times n-1}$, $K_{UV} \in \mathbb{R}^{m-1 \times n-1}$, $K_{VU} \in \mathbb{R}^{n-1 \times m-1}$, $k\mathbf{q} \in \mathbb{R}^{m-1}$, $k\mathbf{x} \in \mathbb{R}^{n-1}$ であり，その i, j 要素と第 i 行目の要素はそれぞれ

$$(K_{UU})_{ij} = K(\mathbf{q}_{i+1}, \mathbf{q}_{j+1}) - K(\mathbf{q}_{i+1}, \mathbf{q}_1) - K(\mathbf{q}_1, \mathbf{q}_{j+1}) + K(\mathbf{q}_1, \mathbf{q}_1) \quad (22)$$

$$(K_{VV})_{ij} = K(\mathbf{x}_{i+1}, \mathbf{x}_{j+1}) - K(\mathbf{x}_{i+1}, \mathbf{x}_1) - K(\mathbf{x}_1, \mathbf{x}_{j+1}) + K(\mathbf{x}_1, \mathbf{x}_1) \quad (23)$$

$$(K_{UV})_{ij} = K(\mathbf{q}_{i+1}, \mathbf{x}_{j+1}) - K(\mathbf{q}_{i+1}, \mathbf{x}_1) - K(\mathbf{q}_1, \mathbf{x}_{j+1}) + K(\mathbf{q}_1, \mathbf{x}_1) \quad (24)$$

$$(K_{VU})_{ij} = K(\mathbf{x}_{i+1}, \mathbf{q}_{j+1}) - K(\mathbf{x}_{i+1}, \mathbf{q}_1) - K(\mathbf{x}_1, \mathbf{q}_{j+1}) + K(\mathbf{x}_1, \mathbf{q}_1) \quad (25)$$

$$(k\mathbf{q})_i = K(\mathbf{q}_1, \mathbf{q}_{i+1}) - K(\mathbf{q}_1, \mathbf{q}_1) - K(\mathbf{x}_1, \mathbf{q}_{i+1}) + K(\mathbf{x}_1, \mathbf{q}_1) \quad (26)$$

$$(k\mathbf{x})_i = K(\mathbf{q}_1, \mathbf{x}_{i+1}) - K(\mathbf{q}_1, \mathbf{x}_1) - K(\mathbf{x}_1, \mathbf{x}_{i+1}) + K(\mathbf{x}_1, \mathbf{x}_1) \quad (27)$$

で与えられる．二つのパターン $\Phi(\mathbf{q}_1)$, $\Phi(\mathbf{x}_1)$ の高次元空間でのユークリッド距離の二乗は $d_{qx} = \|\Phi(\mathbf{q}_1) - \Phi(\mathbf{x}_1)\|^2 = K(\mathbf{q}_1, \mathbf{q}_1) - 2K(\mathbf{q}_1, \mathbf{x}_1) + K(\mathbf{x}_1, \mathbf{x}_1)$ で与えられるから，結局，高次元空間での未知データとクラス j の多様体間距離 $d(\mathcal{M}_q, \mathcal{M})$ は

$$\begin{aligned} d(\mathcal{M}_q, \mathcal{M}) &= \\ d_{qx} + 2k\mathbf{q}^\top \alpha - 2k\mathbf{x}^\top \beta + \alpha^\top K_{UU} \alpha \\ &\quad - \alpha^\top K_{UV} \beta - \beta^\top K_{VU} \alpha + \beta^\top K_{VV} \beta \end{aligned} \quad (28)$$

で与えられる．正則化は $K_{UU} + \lambda_1 \mathbf{I}_{m-1}$, $K_{VV} + \lambda_2 \mathbf{I}_{n-1}$ とすれば良い．カーネル相互部分空間法[14]と比べると複雑な計算が必要である．

4. 線型多様体間距離に基づく学習

ここでは線型多様体間距離に基づくパターン分類において，誤分類や記憶容量を減らすための学習について述べる．距離に基づく識別器のための学習法としては，辞書パターンを更新するベクトル量子化[15]と，距離関数を学習するメトリック学習[16]がよく用いられるが，ここではベクトル量子化に基づく手法を述べる．ベクトル量子化に基づく手法としては，一般化学習量子化 (GLVQ, Generalized Learning Vector Quantization)[11]がよく知られている．そこで，はじめに前節と同様に二乗誤差最小パターンとの距離を利用した GLVQ に基づく学習則を導出し，その後線型多様体間の距離に基づく学習を導く．なお，本節ではクラスの添え字をつけて説明をするので注意されたい．

4.1 二乗誤差最小パターンとの距離を利用した GLVQ に基づく学習則

学習に利用するクラスラベルの分かっている学習データを \mathbf{x} とする． \mathbf{x} からクラス j の線型多様体までの距離を $d_j = \|\mathbf{x} - \mathbf{X}_j \mathbf{b}_j\|^2$ とする．ただし \mathbf{b}_j は式 (3) で得られるものとする． \mathbf{X}_1 を \mathbf{x} と同じクラスに属する最近傍線型多様体を張る訓練データ， \mathbf{X}_2 を \mathbf{x} と異なるクラスに属する最近傍線型多様体を張る訓練データとし，それらを用いて計算された距離をそれぞれ d_1 , d_2 とする．GLVQ と同様に， \mathbf{x} と \mathbf{x} の属すべきクラスとの近さ μ を以下のように定義する：

$$\mu(\mathbf{x}) = \frac{d_1 - d_2}{d_1 + d_2} \quad (29)$$

この μ は全ての \mathbf{x} に対して $-1 < \mu(\mathbf{x}) < 1$ を満たしており， \mathbf{x} が正しく分類された場合には値が負となり，誤分類の場合には値が正となる．学習の目的は誤分類を減らすこと，すなわち全ての \mathbf{x} について μ が減少すればよい．したがって，学習の目的を次式の評価関数 S の最小化として定式化できる[11]：

$$S = \sum_{i=1}^N f(\mu(\mathbf{x}_i)) \quad (30)$$

ここで， N は学習に用いる学習データの総数， $f(\mu)$ は μ に対する単調増加関数でありシグモイド関数 $f(\mu, t) = 1/(1 + e^{-\mu t})$ が用いられる (t は学習ステップ)[11]．

上記の S を最小化するために \mathbf{X}_j ($j = 1, 2$) を最急降下法に基づいて修正することを考える．

$$\mathbf{X}_j \leftarrow \mathbf{X}_j - \epsilon \frac{\partial S}{\partial \mathbf{X}_j}, \quad j = 1, 2. \quad (31)$$

ここで ϵ は微小な正の実数 $0 < \epsilon < 1$ である． $\partial S / \partial \mathbf{X}_j$ は

$$\begin{aligned} \frac{\partial S}{\partial \mathbf{X}_j} &= \frac{\partial S}{\partial \mu} \frac{\partial \mu}{\partial d_j} \frac{\partial d_j}{\partial \mathbf{X}_j} \\ &= (-1)^j \epsilon \frac{\partial f}{\partial \mu} \frac{4d_{3-j}}{(d_1 + d_2)^2} (\mathbf{x} - \mathbf{X}_j \mathbf{b}_j) \mathbf{b}_j^\top \quad (j = 1, 2) \end{aligned} \quad (32)$$

として与えられるので，二乗誤差最小パターンとの距離を利用した学習は $\delta_j = (-1)^j \epsilon \frac{\partial f}{\partial \mu} \frac{d_{3-j}}{(d_1 + d_2)^2}$ ($j = 1, 2$) とおけば

$$\mathbf{X}_j \leftarrow \mathbf{X}_j - \delta_j (\mathbf{x} - \mathbf{X}_j \mathbf{b}_j) \mathbf{b}_j^\top \quad (33)$$

と書ける．したがって， \mathbf{X}_j をなんらかの方法で初期化し，学習データ \mathbf{x} に対する誤分類が少なくなるまで式 (33) の更新を行えば良い．なお，式 (32) から式 (33) にかけて $d_j / (d_1 + d_2)$ へと変更がなされているが，この変更により ϵ の決定が容易となる（収束には影響しない）[11]．また， \mathbf{X}_j の更新を \mathbf{x} の k 近傍に限定した場合には LSC のための学習則となる．なお，式 (2) を未知データと線型多様体との距離とした場合の更新式は

$$\begin{aligned} \mathbf{x}_1^j &\leftarrow \mathbf{x}_1^j - \delta_j (\mathbf{x} - (\mathbf{x}_1^j + \mathbf{V}_j \beta_j)) \\ \mathbf{V}_j &\leftarrow \mathbf{V}_j - \delta_j (\mathbf{x} - (\mathbf{x}_1^j + \mathbf{V}_j \beta_j)) \beta_j^\top \end{aligned} \quad (34)$$

となり，GLVQ を利用した学習部分空間法[17]となる．

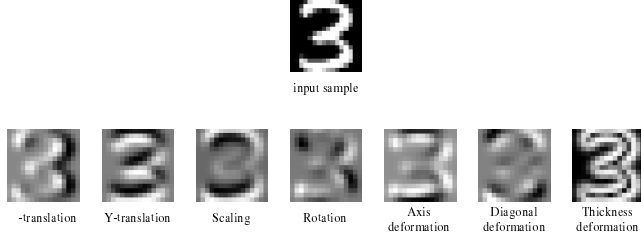


図3 上段が未知データ q ，下段が7種類の変形を表す接ベクトル U

4.2 線型多様体間距離に基づく学習則

次に、未知データも線型多様体で与えられる場合の学習則について考える。ラベルが既知の学習データが m 個のベクトルの組で与えられたとし、それらによって張られた線型多様体を $\mathcal{M}_x = \mathbf{x}_1 + \mathbf{U}\alpha$ で表す。 \mathcal{M}_x からクラス j の線型多様体までの距離を $d_j = \|\mathbf{x}_1 + \mathbf{U}\alpha - \mathbf{X}_j \mathbf{b}_j\|^2$ とする。ただし α や \mathbf{b}_j は式 (12) や式 (13) で得られるものとする。 \mathbf{X}_1 を \mathcal{M}_x と同じクラスに属する最近傍線型多様体を張る訓練データ、 \mathbf{X}_2 を \mathcal{M}_x と異なるクラスに属する最近傍線型多様体を張る訓練データとし、それらを用いて計算された距離をそれぞれ d_1, d_2 とする。二乗誤差最小パターンとの距離を利用した学習と同様にして線型多様体間距離に基づく学習を導けば、結局それは以下の更新によって実現できる：

$$\mathbf{X}_j \leftarrow \mathbf{X}_j - \delta_j (\mathbf{x}_1 + \mathbf{U}\alpha - \mathbf{X}_j \mathbf{b}_j) \mathbf{b}_j^\top \quad (35)$$

なお、式 (14) を線型多様体間距離とした場合には、更新式は

$$\begin{aligned} \mathbf{x}_1^j &\leftarrow \mathbf{x}_1^j - \delta_j (\mathbf{x}_1 + \mathbf{U}\alpha - (\mathbf{x}_1^j + \mathbf{V}_j \beta_j)) \\ \mathbf{V}_j &\leftarrow \mathbf{V}_j - \delta_j (\mathbf{x}_1 + \mathbf{U}\alpha - (\mathbf{x}_1^j + \mathbf{V}_j \beta_j)) \beta_j^\top \end{aligned} \quad (36)$$

となる。上記の更新では、学習データが与えられる毎に α や \mathbf{b}_j を最適化しなければならないため、学習に非常に時間がかかるという難点がある。

5. 実験

ここでは、公開手書き数字データ USPS [18] を用いた実験結果を示す。USPS は 7291 の訓練データと 2007 の未知データが 16×16 ピクセルのモノクロ画像で与えられている。本実験では位置合わせや正規化等の前処理を一切行っていない。実験では各手法を CPU1.86GHz、メモリ 2GB の標準 PC 上で MATLAB を用いて実装した。パラメータはランダムに選んだ 2000 個の検証データを用いて決定した。線型多様体距離を利用した実験では、図 3 に示すように未知データを 7 種類の変形を表す接ベクトル [5] を利用して線型多様体で表現した。なお、本稿ではカーネル化した識別による実験は行っていない。

5.1 線型多様体を利用した識別法のエラー率

はじめに、未知データに対するエラー率と 1 パターンあたりの識別時間をさまざまな識別法で調べた。表 1 に各手法のエラー率と識別時間を示す。比較した手法は PDM, 線型多様体間距離を利用した識別 (Manifold Distance Classifier, MDC と略す), LSC (k は近傍数), 1S TD と 2S TD を利用した最近傍決定則である。1S TD と 2S TD では計算量削減のために、予めテ

表 1 未知データに対するエラー率と識別時間

method	test error [%]	time [s]
PDM ($n_j = 30$)	5.1	0.001
MDC ($n_j = 20$)	4.2	0.01
LSC ($k=11$)	3.9	0.05
1S TD	3.3	0.1
2S TD	2.4	0.13
k NN ($k = 3$)	5.3	0.2
SVM (# SVs= 3220)	4.6	0.005

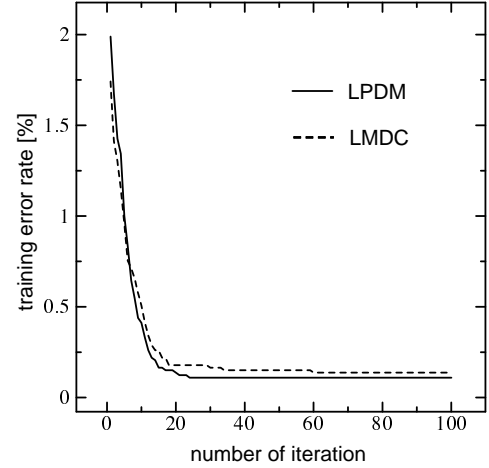


図4 学習データに対するエラー率と学習回数との関係

ストデータの 100 近傍をユークリッド距離で選択した後、選択したパターンとの接距離を計算して最近傍決定則を行った。なお、PDM と MDC では各クラスで訓練データから主成分分析により線型多様体を求め、式 (2) と式 (14) で正則化無しの分類を行った。表 1 の n_j は各クラスの線型多様体の次元数を表す。

比較のためユークリッド距離による k NN と RBF カーネルを用いた Support Vector Machine (SVM) の結果も示す。SVM は SVMLIB [19] を利用した。表 1 より PDM が最も高速であるが、エラー率は SVM よりも高い。一方、MDC の識別時間は SVM よりも 2 倍遅いがエラー率は低く、使用するメモリ量 (辞書サイズが 200) も SVM (サポートベクトルの数が 3220) よりも少ない。LSC や 1S TD, 2S TD のエラー率は低いが、全訓練データを利用しているため識別に時間がかかり、必要なメモリ量も大きい。

5.2 学習の効果

次に、4 で述べた学習の効果について実験した。実験では式 (34) と式 (36) を利用した学習 PDM (Learning PDM, LPDM) と学習 MDC (LMDC) について検証を行った。線型多様体の初期値は各クラスで主成分分析によって得られた重心と直交基底を用いた。次元数は表 1 で示したものと同じにしたので識別時間の変化は無い。学習係数は $\epsilon = 10^{-7}$ とし、正則化は行わなかった。学習では全ての訓練データを用いてバッチ型で線型多様体の更新を行った。本稿では一回の更新を一回の学習と呼ぶ。

図 4 に訓練データのエラー率と学習回数との関係を示す。実線が LPDM, 点線が LMDC を表している。図から約 30 回でエラー率に変化が見られなくなったので以後の実験では学習回数

表2 学習後の未知データに対するエラー率と学習時間

method	test error [%]	learning time [s]
LPDM ($n_j = 30$)	4.9	2789
LMDC ($n_j = 20$)	3.7	3437
GLVQ	8.7	37
SVM	4.6	34.9

を30回とした。表2にはLPDMとLMDCの学習後のテストデータに対するエラー率と学習に要した時間を示す。比較のため、ユークリッド距離に基づくGLVQとSVMの結果も示す。表2から、学習によりPDMやMDCにおいてエラー率を低減できたことがわかる。一方、SVMと比較すると学習に要する時間が非常に長く、高速化が望まれる。

6. ま と め

本稿では、線型多様体間距離が未知データとの二乗誤差を最小にする訓練データの線型結合を利用した識別から容易に導かれることを示した。また、一般化学習ベクトル量子化を線型多様体間距離に基づくパターン識別に適用した場合の学習則を導いた。手書き数字を用いた実験により、線型多様体間距離を利用することで低いエラー率を達成でき、学習によって識別精度が改善されることを示した。

本稿では、相互部分空間法との比較を行っていないが、識別精度に関しては(CLAFICと投影距離法の場合と同様に)あまり差は無いと思われる。実際、文献[7]では顔認識で比較を行っているが、ほとんど差が無いことが報告されている。線型多様体間距離を用いることの利点は、線型多様体の基底が直交しなくても計算が可能のため、距離に基づく学習の適用やカーネル化などが容易なことが挙げられる。相互部分空間法はこれまでにさまざまな拡張がなされているので、線型多様体間距離に基づく識別も拡張を行うことが今後の課題である。

文 献

- [1] 黒沢由明, “Subspace2006 部分空間法入門,” 部分空間法研究会 Subspace2006, pp. 136–143, 2006.
- [2] 前田賢一, 渡辺貞一, “局所構造を導入したパターン・マッチング法,” 信学論, vol. J68-D, no. 3, pp. 345–352 1985.
- [3] 福井和広, 山口修, “部分空間法の理論拡張と物体認識への応用,” 情処論, vol. 46, no. DIG15, pp. 21–34, 2005.
- [4] 池田正幸, 田中英彦, 元岡 達, “手書き文字認識における投影距離法,” 情処学論, vol. 24, no. 1, pp. 106–112, 1983.
- [5] P.Y. Simard, Y. LeCun, and J.S. Denker, “Efficient pattern recognition using a new transformation distance,” Proc. of NIPS, vol. 5, pp. 50–58, 1993.
- [6] A. Fitzgibbon and A. Zisserman, “Joint manifold distance: A new approach to appearance based clustering,” IEEE Int. Conf. on CVPR, 2003.
- [7] J.H. Chen, S.L. Yeh, and C.S. Chen, “Inter-subspace distance: A new method for face recognition with multiple samples,” Proc. of ICPR04, vol. 3, pp. 140–143, 2004.
- [8] J. Laaksonen, “Subspace classifiers in recognition of handwritten digits,” PhD thesis, Helsinki University of Technology, 1997.
- [9] J.-T. Chien and C.-C. Wu, “Discriminant waveletfaces and nearest feature classifiers for face recognition,” IEEE Trans. Pattern Anal. Machine Intell., vol. 24, no. 12, pp. 1644–1649, 2002.
- [10] E. Oja, “Subspace methods of pattern recognition,” Research Studies Press, 1983. (小川英光, 佐藤 誠訳, パターン認識と部分空間法, 産業図書, 1986)

- [11] A. Sato and K. Yamada, “Generalized learning vector quantization,” *Prop. of NIPS*, 7:423–429, 1995.
- [12] 前田 英作, 村瀬 洋, “カーネル非線形部分空間法によるパターン認識,” 信学論, vol. J82-D-II, no. 4, pp. 600–612, Apr. 1999.
- [13] B. Schölkopf, A.J. Smola, and K.-R. Müller, “Nonlinear component analysis as a kernel eigenvalue problem,” *Neural Computation*, vol. 10, pp. 1299–1319, 1998.
- [14] 坂野 鋭, 武川直樹, 中村太一, “核非線型相互部分空間法による物体認識,” 信学論, vol. J84-D-II, no. 8, pp. 1549–1556, 2001.
- [15] T. Kohonen, “Self-organizing map,” Springer-Verlag, 1995.
- [16] L. Yang and R. Jin, “Distance metric learning: A comprehensive survey,” Technical report, Department of Computer Science and Engineering, Michigan State University, 2006.
- [17] 佐藤 敦, 山田敬嗣, “一般学習ベクトル量子化に基づく学習部分空間法,” 信学大全, p. 236, 1997.
- [18] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural Computation*, vol.1, no.4, pp.541–551, 1989.
- [19] C.C. Chang and C.J. Lin, “LIBSVM: A library for support vector machines,” 2001.
<http://www.csie.ntu.edu.tw/~cjlin/libsvm>