G-09

モデルサイズから見たカーネル最小分類誤り学習法の有用性の検証
Evaluation of Kernel Minimum Classification Error Training in Terms of Model Size

谷 陵真 †　渡辺 秀行 ‡　片桐 滋 †　大崎 美穂 †
Ryoma Tani　Hideyuki Watanabe　Shigeru Katagiri　Miho Osaki

1　はじめに

幾何マージンやカーネルの効力を活用するパターン分類器の学習法に，多クラスサポートベクターマシン（Multi-Class Support Vector Machine: MSVM）[1] やカーネル最小分類誤り（Kernel Minimum Classification Error: KMCE）学習法 [2] がある．MSVM は，多クラス分類問題を扱うことを前提に設計された SVM であり，非線形写像による高い分類能力と幾何マージン最大化による未知標本耐性をもつ，優れた学習法の 1 つである．KMCE 学習法は，カーネルに基づく高次元空間において，大幾何マージン最小分類誤り（Large Geometric Margin-MCE: LGM-MCE）学習法 [3] を行う学習法である．LGM-MCE 学習法は，分類誤り数の最小化と幾何マージンの増大化を，同時かつ直接的に目指す．

両手法は，カーネル写像を伴う線形識別関数を用いる点，そして，幾何マージンを大きくすることで未知標本耐性の向上を目指す点においては同じである．一方で，学習に用いる損失関数や，学習対象とするパラメータなどから両手法を区別することができる．

これまでにも，KMCE 法の有用性の検証が行われてきたが，MSVM 法と KMCE 法の直接的な比較は行われていなかった．そこで，本稿では MSVM 法と KMCE 法の比較実験に基づいて，KMCE 法の有用性を検証する．

2　多クラスサポートベクターマシン

2.1　分類課題

与えられた固定次元ベクトルパターン $\boldsymbol{x}$（$\in \mathcal{X}$）を，次の分類規則に従って，J 個のクラス $\{C_j\}_{j=1}^{J}$ のうちの 1 つに分類する課題を考える．ここで $\mathcal{X}$ は $\boldsymbol{x}$ が属するパターン空間である．

$$C(\boldsymbol{x}) = C_k \quad \text{iff} \quad k = \arg\max_j g_j(\boldsymbol{x};\Lambda), \tag{1}$$

ここで $g_j(\boldsymbol{x};\Lambda)$ は C_j に対する識別関数であり，$\boldsymbol{x}$ が C_j に帰属する程度を示す．また，Λ は学習対象の分類器パラメータ集合であり，識別関数 $g_j(\boldsymbol{x};\Lambda)$ は，学習に勾配法などの利用を可能とするため Λ に関して微分可能であるとする．

2.2　定式化

与えられた $\boldsymbol{x}$ を分類するために，MSVM は次式の線形識別関数を用いる．

$$g_j(\boldsymbol{x};\Lambda) = \boldsymbol{w}_j \cdot \phi(\boldsymbol{x}), \tag{2}$$

ここで，$\{\boldsymbol{w}_j\}_{j=1}^{J}$ は係数ベクトルであり，$\phi(\cdot)$ はカーネル（後述参照）により陰に定義される非線形写像である．最適化の対象となる分類器パラメータ Λ は，この係数ベクトル $\{\boldsymbol{w}_j\}_{j=1}^{J}$ とカーネルの特性を決定する制御パラメータ（例：ガウスカーネルのカーネル幅）である．

係数ベクトル $\{\boldsymbol{w}_j\}_{j=1}^{J}$ の最適状態は，N 個の学習用標本対 $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N}$ 上で行われる次式の制約付き最小化問題の解として求められる（y_n は学習標本 $\boldsymbol{x}$ が属するクラスの指標）．

$$\begin{aligned} &\underset{\{\boldsymbol{w}_j\},\{\xi_n\}}{\text{minimize}} \quad \left\{ \frac{\beta}{2} \sum_{j=1}^{J} \|\boldsymbol{w}_j\|^2 + \sum_{n=1}^{N} \xi_n \right\}, \\ &\text{subject to}: \ \forall n, j \\ &\qquad \boldsymbol{w}_{y_n} \cdot \phi(\boldsymbol{x}_n) + \delta_{y_n,j} - \boldsymbol{w}_j \cdot \phi(\boldsymbol{x}_n) \geq 1 - \xi_n, \end{aligned} \tag{3}$$

ここで，$\delta_{y_n,j}$ は，$y_n = j$ のとき 1 に，$y_n \neq j$ のとき 0 になる，Kronecker のデルタ関数である．また，$\{\xi_n\}_{n=1}^{N}$ はスラック変数，β は正の正則化定数である．一方，カーネルの特性を制御する，カーネル幅などのパラメータは，学習標本群と別に用意する評価標本群などを用いて，実験的あるいは試行錯誤的に求められる．

学習標本が線形分離可能である場合，式 (3) におけるスラック変数は $\{\xi_n = 0\}_{n=1}^{N}$ とすることができる．その時，式 (3) の制約は，学習標本 $\boldsymbol{x}_n$ が所属するクラス，すなわち正解クラス C_{y_n} の識別関数値が，その他のすべてのクラスの識別関数値より 1 以上大きくなることを要求していることがわかる．これは，すべての学習標本が正分類されていることに加え，関数マージン値，即ち，正解クラスの識別関数値とそれ以外のクラスの識別関数値との差の最小値が 1 よ

† 同志社大学，Doshisha University
‡ 株式会社 国際電気通信基礎技術研究所，ATR

り小さくなる領域には，学習標本が存在しないような境界を引くことを意味している．

しかし，実際の分類問題におけるパターン標本が線形分離可能であることは稀である．そこで本手法の定式化は，線形分離可能ではない問題も扱えるようにスラック変数を導入している．学習標本 $\boldsymbol{x}_n$ に対するスラック変数 ξ_n の値の分だけペナルティ，即ち損失を与える代わりに，関数マージン値が 1 より小さくなる領域に $\boldsymbol{x}_n$ が存在することを許していることになる．式 (3) から，ξ_n は，次式のような $\boldsymbol{w}_{i_n}\cdot\phi(\boldsymbol{x}_n)-\boldsymbol{w}_{y_n}\cdot\phi(\boldsymbol{x}_n)$ に関するヒンジ関数であることがわかる．

$$\xi_n = \max\{0, 1+\boldsymbol{w}_{i_n}\cdot\phi(\boldsymbol{x}_n)-\boldsymbol{w}_{y_n}\cdot\phi(\boldsymbol{x}_n)\}, \quad (4)$$

ここで，C_{i_n} は $\boldsymbol{x}_n$（$\in C_{y_n}$）の最も誤り易いクラスである．

こうして，MSVM は，一般的な（線形分離可能とは限らない）分類課題において，式 (3) の第 1 項の最小化を通して写像特徴空間における幾何マージンを大きくすることを目指し，同時に，その第 2 項の最小化を通してヒンジ損失を小さくすることを目指すことになる．学習全体に対する第 1 項と第 2 項の影響度は正則化定数 β によって制御する．

2.3 学習手続き

式 (3) の制約付き最小化，即ち MSVM の学習は，以下のように双対問題に置き換えられた上で行われる．

まず，ラグランジュ乗数 $\eta_{n,j}$ を導入し，次式のラグランジュ関数を得る．

$$\mathcal{L} = \frac{\beta}{2}\sum_{j=1}^{J}\|\mathbf{w}_j\|^2 + \sum_{n=1}^{N}\xi_n + \sum_{n=1}^{N}\sum_{j=1}^{J}\Big[\eta_{n,j}\times \{\boldsymbol{w}_j\cdot\phi(\boldsymbol{x}_n)-\boldsymbol{w}_{y_n}\cdot\phi(\boldsymbol{x}_n)-\delta_{y_n,j}+1-\xi_n\}\Big], \quad (5)$$

$$\text{subject to}: \quad \forall n, j \quad \eta_{n,j}\geq 0.$$

続いて，ラグランジュ関数の勾配を求めて，次のようにカーネル写像関数に関する重みパラメータの表現を得る．

$$\frac{\partial\mathcal{L}}{\partial\xi_n} = 1-\sum_{j=1}^{J}\eta_{n,j}=0 \;\Rightarrow\; \sum_{j=1}^{J}\eta_{n,j}=1, \quad (6)$$

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\boldsymbol{w}_j} &= \sum_{n=1}^{N}\eta_{n,j}\phi(\boldsymbol{x}_n) - \sum_{n,y_n=j}\sum_{q=1}^{J}\eta_{n,q}\phi(\boldsymbol{x}_n)+\beta\boldsymbol{w}_j \\ &= \sum_{n=1}^{N}\eta_{n,j}\phi(\boldsymbol{x}_n) - \sum_{n=1}^{N}\delta_{y_n,j}\phi(\boldsymbol{x}_n)+\beta\boldsymbol{w}_j = 0, \\ &\Rightarrow \boldsymbol{w}_j = \beta^{-1}\sum_{n=1}^{N}(\delta_{y_n,j}-\eta_{n,j})\phi(\boldsymbol{x}_n).\end{aligned} \quad (7)$$

次に，$\tau_{n,j}=\delta_{y_n,j}-\eta_{n,j}$ の変数置き換えを経て，得られた $\boldsymbol{w}_j$ を

$$\boldsymbol{w}_j = \beta^{-1}\sum_{n=1}^{N}\tau_{n,j}\phi(\boldsymbol{x}_n), \quad (8)$$

と改め，さらに得られた式 (8) を式 (5) に代入する．こうして，式 (3) の最小化問題（主問題）は，次の双対問題，即ち $\tau_{n,j}$ に関する最大化問題に置き換えられる．

$$\begin{aligned}\underset{\{\boldsymbol{\tau}_n\}_{n=1}^N}{\text{maximize}}\Big\{ &-\frac{1}{2}\sum_{n=1}^{N}\sum_{l=1}^{N}K(\boldsymbol{x}_n,\boldsymbol{x}_l)(\boldsymbol{\tau}_n\cdot\boldsymbol{\tau}_l) \\ &+\beta\sum_{n=1}^{N}\boldsymbol{\tau}_n\cdot\mathbf{1}_{y_n}\Big\} \\ \text{subject to}: \;\forall n \quad &\boldsymbol{\tau}_n\leq\mathbf{1}_{y_n} \quad\text{and}\quad \boldsymbol{\tau}_{\mathrm{n}}\cdot 1=0,\end{aligned} \quad (9)$$

ここで，$\mathbf{1}_{y_n}$ は y_n 番目の要素のみが 1 でその他の要素が 0 の J 次元ベクトル，$\mathbf{1}$ はすべての要素が 1 の J 次元ベクトル，$\boldsymbol{\tau}_n=(\tau_{n,1},\tau_{n,2},\ldots,\tau_{n,J})^{\mathrm{T}}$ であり（T は転置），また $K(\boldsymbol{x}_n,\boldsymbol{x}_l)$ は写像された学習標本 $\phi(\boldsymbol{x}_n)$ と $\phi(\boldsymbol{x}_l)$ との内積，即ちカーネル関数である．

この双対問題から得た $\{\boldsymbol{\tau}_n\}_{n=1}^N$ を式 (8) に代入し，さらにその式 (8) を式 (2) に代入し，学習結果として，次の識別関数を求めることができる．

$$g_j(\boldsymbol{x};\Lambda) = \beta^{-1}\sum_{n=1}^{N}\tau_{n,j}K(\boldsymbol{x},\boldsymbol{x}_n). \quad (10)$$

ここで，係数 β^{-1} は定数倍であるため識別結果に影響しない．こうして，最終的な学習結果として，β^{-1} を省略した次の識別関数に至る．

$$g_j(\boldsymbol{x};\Lambda) = \sum_{n=1}^{N}\tau_{n,j}K(\boldsymbol{x},\boldsymbol{x}_n). \quad (11)$$

式 (11) では N 個の全ての学習標本に関するカーネルの和が用いられている．しかし，学習を通して，$\boldsymbol{\tau}_n=\mathbf{0}$ となる学習標本，即ち分類境界の構成に関与しない学習標本が自動的に選択される．その結果，識別関数は $\boldsymbol{\tau}_n\neq\mathbf{0}$ となる $\boldsymbol{x}_n$，即ちサポートベクターのみによって構成される．

選択されるサポートベクターの数は，基本的に対応する分類課題の難しさに依る．しかし，一般にその数は大きく，膨大な標本から成る大規模課題に適用することが困難となるスケーラビリティの問題や，ハードウェアの制約が大きい組み込みシステムに搭載することが困難となる実装困難性の問題等につながりやすい．

3 カーネル最小分類誤り学習法

3.1 識別関数と学習対象パラメータ

2.1 と同様に，分類規則 (1) を用いて，J クラスの固定次元ベクトルパターン $\boldsymbol{x}$ を分類する課題を考える．

この時，KMCE 法は，次のカーネル写像を伴う線形識別関数を識別関数に採用する．

$$g_j(\boldsymbol{x};\Lambda)=\sum_{m=1}^{M}\tau_{m,j}K(\boldsymbol{x},\boldsymbol{p}_m), \tag{12}$$

ここで，$\{\tau_{m,j}\}_{m=1}^{M}$ はカーネルの線形和に用いられる係数であり，$\{\boldsymbol{p}_m\}_{m=1}^{M}$ は学習標本群 $\{\boldsymbol{x}_n\}_{n=1}^{N}$ を集約したプロトタイプ群である．学習対象となる分類器パラメータ Λ は，$\{\tau_{m,j}\}_{m=1,j=1}^{M\ \ J}$ と，カーネルを決定する $\{\boldsymbol{p}_m\}_{m=1}^{M}$ およびカーネル制御パラメータ（例 : カーネル幅）である．なお，一般に，$M<N$ とするが，全ての学習標本をプロトタイプとして用いても構わない．その時，式 (12) は，形式上，MSVM 法の識別関数 (11) に等しくなる．

式 (12) の識別関数は，カーネルによって陰に定義される非線形写像 $\phi(\cdot)$ を用いて，

$$g_j(\boldsymbol{x};\Lambda)=\sum_{m=1}^{M}\tau_{m,j}\phi(\boldsymbol{x})\cdot\phi(\boldsymbol{p}_m), \tag{13}$$

に書き換えられる．さらに，

$$\boldsymbol{w}_j(\hat{\boldsymbol{\tau}}_j)=\sum_{m=1}^{M}\tau_{m,j}\phi(\boldsymbol{p}_m), \tag{14}$$

と置き換えると，

$$g_j(\boldsymbol{x};\Lambda)=\boldsymbol{w}_j(\hat{\boldsymbol{\tau}}_j)\cdot\phi(\boldsymbol{x}), \tag{15}$$

となる．ここで，$\hat{\boldsymbol{\tau}}_j=(\tau_{1,j},\ldots,\tau_{M,j})^{\mathrm{T}}$ である．

式 (15) は，式 (12) で定義した KMCE 法の識別関数が，カーネルに付随する写像関数によって写像される（無限次元をも含む）高次元特徴空間における線形識別関数に他ならないことを示している．また，そこで用いられる係数ベクトル $\boldsymbol{w}_j(\hat{\boldsymbol{\tau}}_j)$ は，カーネルにかかる重み $\{\tau_{m,j}\}_{m=1}^{M}$ によって決まる．即ち，高次元特徴空間における係数ベクトル $\boldsymbol{w}_j(\hat{\boldsymbol{\tau}}_j)$ の学習が，高々M 次元の重みベクトル $\hat{\boldsymbol{\tau}}_j$ の調整によって行われることがわかる．

3.2 学習手続きの概要

KMCE 学習法は，その定形化の基礎を，LGM-MCE 学習法，ひいてはその基盤となる MCE 学習法（区別のため，関数マージン最小分類誤り（FM-MCE: Functional Margin Minimum Classification Error）学習法）に置く．

一連の MCE 学習法に共通する特徴の一つに，次式に示す誤分類尺度の利用がある．C_y に属する学習標本 $\boldsymbol{x}$ に対して，

$$d_y(\boldsymbol{x};\Lambda)=-g_y(\boldsymbol{x};\Lambda)+\log\left[\frac{1}{J-1}\sum_{j,j\neq y}\exp(\psi g_j(\boldsymbol{x};\Lambda))\right]^{\frac{1}{\psi}}\ (\psi>0), \tag{16}$$

と定義される．本尺度は，学習標本が属するクラスの識別関数 $g_y(\boldsymbol{x};\Lambda)$ とそれ以外のクラスの識別関数の“平均”との差であり，その値が負のとき $\boldsymbol{x}$ が正しく分類されたことを，正のとき誤って分類されたことを示している．ここで $\psi\to\infty$ とすると，式 (16) は

$$d_y(\boldsymbol{x};\Lambda)=-g_y(\boldsymbol{x};\Lambda)+\max_{j,j\neq y}g_j(\boldsymbol{x};\Lambda), \tag{17}$$

となり，上記の尺度が持つ，学習標本に対する分類の正誤を尺度の符号によって表す意味がより明確になる．尺度の元々の定義 (16) の第 2 項に max オペレータでなく L_ψ ノルムが使われているのは，先に識別関数 $g_j(\boldsymbol{x};\Lambda)$ を Λ に関して微分可能であると要請したのと同じ理由による．ただし，実際の学習手続きにおいては，計算の便利を優先し，式 (17) の誤分類尺度が用いられることが多い．

式 (17) の $d_y(\boldsymbol{x};\Lambda)$ は，$\boldsymbol{x}$ が所属するクラスとそれが最も誤り易いクラスとの 2 つの識別関数の差異，即ち関数マージンであることがわかる．これが，この誤分類尺度を用いる MCE 学習法を FM-MCE 学習法と呼ぶ由来である．

3.3 平滑な分類誤り損失の導入

分類器パラメータ学習の究極の目標は，最小分類誤り確率状態，即ちベイズ誤りの状態に対応するパタメータ状態を見出すことにある．MCE 学習は，学習標本に対する分類誤り数の最小化を通して，この目標の達成を目指す．

誤分類尺度の符号が分類の正誤を表すことから，次の 0-1 損失を用いることで分類誤り数を求めることができる．

$$\ell(\boldsymbol{x};\Lambda)=\begin{cases}1 & \text{if}\quad d_y(\boldsymbol{x};\Lambda)>0\\ 0 & \text{if}\quad d_y(\boldsymbol{x};\Lambda)\leq 0.\end{cases} \tag{18}$$

しかし，この 0-1 損失関数は，明らかに勾配法による最適化に適さない．そこで，MCE 学習法は，0-1 損失関数の代わりに以下のシグモイド関数などを用いて平滑化分類誤り数損失を定義する．

$$\ell(d_y(\boldsymbol{x};\Lambda))=\frac{1}{1+\exp(-\alpha d_y(\boldsymbol{x};\Lambda))}\quad(\alpha>0), \tag{19}$$

ここで α は損失関数の平滑度を制御するハイパーパラメータである．

本来，Λ の最適化は無限個の標本からなる $\mathcal{X}$ において行われるべきである．しかし，無限個の標本を入手することは，事実上，不可能である．そのため MCE 学習は，有限の N 個の標本とそのクラス帰属指標からなる学習標本対 $\{(\boldsymbol{x}_n,\, y_n)\}_{n=1}^{N}$ を用いて以下の経験的平均損失を求め，この経験的平均損失の最小化を通して学習パラメータ集合 Λ の最適化を目指す．

$$L(\Lambda) = \frac{1}{N}\sum_{n=1}^{N} \ell(d_{y_n}(\boldsymbol{x}_n;\Lambda)). \tag{20}$$

多くの場合，$L(\Lambda)$ の最小化には，最急降下法や確率的降下法などの勾配法に基づく手法を採用する．確率的降下法を用いる場合，次式のパラメータ更新式の繰り返しにより Λ の最適解を求める．

$$\Lambda^{(t+1)} = \Lambda^{(t)} - \mu_t \nabla_\Lambda \ell(d_{y_n}(\boldsymbol{x}_n;\Lambda^{(t)})) \quad (\mu_t > 0), \tag{21}$$

ここで t は学習の繰り返しにおけるステップ回数を，∇_Λ は Λ に関する偏微分を表す．μ_t は学習係数であり，パラメータの更新量を調整する．なお，パラメータの初期値 $\Lambda^{(0)}$ は何らかの方法で初期化されているものとする．

3.4 幾何マージン型誤分類尺度の導入

学習が達成するクラス境界は，学習標本を正しく分類できるだけではなく，学習には登場しない（無限個の）試験標本を，ベイズ誤りに対応する意味で“正しく”分類できなければならない．この理想的なクラス境界の達成を，有限個の標本しか利用できない現実において目指すとき，クラス境界とその最近傍パターン標本との間の幾何学的（ユークリッド）距離，即ち幾何マージンをできるだけ大きくなるようにクラス境界，あるいはそれをもたらす分類器パラメータの推定を行う学習アプローチを考えることができる．

こうした推定の実現を目指し，MCE 学習は，式 (17) の誤分類尺度を，次式で与えられる，幾何マージンに基づく誤分類尺度によって置き換える．

$$D_y(\boldsymbol{x};\Lambda) = \frac{d_y(\boldsymbol{x};\Lambda)}{\|\nabla_{\boldsymbol{x}} d_y(\boldsymbol{x};\Lambda)\|} \approx -r, \tag{22}$$

ここで r は，$\boldsymbol{x}^\dagger$ をクラス決定境界に最も近い正分類の学習標本とし，かつ $\boldsymbol{x}^*$ を $\boldsymbol{x}^\dagger$ から境界に下ろした垂線の足として，以下で定義される幾何マージンである．

$$r = \frac{|d_y(\boldsymbol{x}^\dagger;\Lambda) + o(r)|}{\|\nabla_{\boldsymbol{x}} d_y(\boldsymbol{x}^*;\Lambda)\|}. \tag{23}$$

式 (22) にあるように，この新しい誤分類尺度 $D_y(\boldsymbol{x};\Lambda)$ は，一般の識別関数（あるいはそれを決定する Λ）に関しては，(符号が反転された）幾何マージンの近似となる．しかし，プロトタイプとのユークリッド距離で識別関数を構成する距離型の識別関数や，線形識別関数においては，両者は正確に一致し，

$$D_y(\boldsymbol{x};\Lambda) = -r = \frac{d_y(\boldsymbol{x}^\dagger;\Lambda)}{\|\nabla_{\boldsymbol{x}} d_y(\boldsymbol{x}^\dagger;\Lambda)\|}. \tag{24}$$

となる．また，(一般に高次元の）パターン標本空間内における幾何マージンと 1 次元の誤分類尺度空間におけるそれとが一致することも明らかにされている．

こうして，$D_y(\boldsymbol{x};\Lambda)$ を誤分類尺度とする MCE 学習は，$D_y(\boldsymbol{x};\Lambda)$ 上で定義する平滑な分類誤り数損失を，N 個の学習標本対上で経験的平均損失に統合し，その最小化を通して，分類誤り数の最小化と幾何マージンの最大化の双方を同時に目指す．LGM-MCE 学習法の名は，この 2 つの目標の同時最適化に因んでいる．

LGM-MCE 学習法で用いられる平滑化分類誤り数損失の平滑度（式 (20) の α に相当）は，幾何マージン型誤分類尺度上の（学習標本に対応する）分類判断の周辺に仮想的に分類判断を生成する “仮想標本領域” の大きさを制御することが明らかにされている．この仮想標本どうしの重なりは，誤分類尺度上のある点において異なるクラスの仮想標本どうしが重なること，言い換えればゼロではないベイズ誤りの存在を模倣する．従って，適切に設定された平滑度をもつ平滑化分類誤り数損失を用いる LGM-MCE 学習は，分類誤り数の最小化と幾何マージンの最大化を通して，ベイズ誤り状態を優れて近似する分類器パラメータの状態を見出すことが期待できる．

3.5 学習手続き

KMCE 学習法は，上述の LGM-MCE 学習の手続きに式 (15) の識別関数を適用することで定義される．

$\boldsymbol{x}$ が所属するクラスを C_y とし，それが最も誤り易いクラスを C_i とすると，カーネル写像関数で写像された特徴空間 H における幾何マージン r_H は，式 (17) と式 (24)，式 (15) より，

$$r_H = \frac{(\boldsymbol{w}_y(\hat{\boldsymbol{\tau}}_y) - \boldsymbol{w}_i(\hat{\boldsymbol{\tau}}_i)) \cdot \phi(\boldsymbol{x})}{\|\boldsymbol{w}_y(\hat{\boldsymbol{\tau}}_y) - \boldsymbol{w}_i(\hat{\boldsymbol{\tau}}_i)\|}, \tag{25}$$

となる．このとき，線形識別関数を用いていることに拠って，幾何マージン r_H は近似ではなく正確に式 (25) の右辺に等しい [3]．さらに式 (14) を代入し，内積部分をカーネルに置き換えると，

$$r_H = \frac{\hat{\boldsymbol{\tau}}_y^{\mathrm{T}} \boldsymbol{k}(\boldsymbol{x}) - \hat{\boldsymbol{\tau}}_i^{\mathrm{T}} \boldsymbol{k}(\boldsymbol{x})}{\sqrt{(\hat{\boldsymbol{\tau}}_y - \hat{\boldsymbol{\tau}}_i)^{\mathrm{T}} \boldsymbol{K} (\hat{\boldsymbol{\tau}}_y - \hat{\boldsymbol{\tau}}_i)}}, \tag{26}$$

となる．ここで $\boldsymbol{k}(\boldsymbol{x})$ は経験的カーネルマップと，$\boldsymbol{K}$ はグラム行列と呼ばれ，次式で表される．

$$\boldsymbol{k}(\boldsymbol{x}) = \begin{bmatrix} K(\boldsymbol{x}, \boldsymbol{p}_1) \\ K(\boldsymbol{x}, \boldsymbol{p}_2) \\ \vdots \\ K(\boldsymbol{x}, \boldsymbol{p}_M) \end{bmatrix}, \tag{27}$$

$$\boldsymbol{K} = \begin{bmatrix} K(\boldsymbol{p}_1, \boldsymbol{p}_1) & K(\boldsymbol{p}_1, \boldsymbol{p}_2) & \cdots & K(\boldsymbol{p}_1, \boldsymbol{p}_M) \\ K(\boldsymbol{p}_2, \boldsymbol{p}_1) & K(\boldsymbol{p}_2, \boldsymbol{p}_2) & \cdots & K(\boldsymbol{p}_2, \boldsymbol{p}_M) \\ \vdots & \vdots & \ddots & \vdots \\ K(\boldsymbol{p}_M, \boldsymbol{p}_1) & K(\boldsymbol{p}_M, \boldsymbol{p}_2) & \cdots & K(\boldsymbol{p}_M, \boldsymbol{p}_M) \end{bmatrix}. \tag{28}$$

これまでと同様に，得られた幾何マージンから次式の誤分類尺度を得ることができる．

$$D_y(\boldsymbol{x}; \Lambda) = -\frac{\hat{\boldsymbol{\tau}}_y^{\mathrm{T}} \boldsymbol{k}(\boldsymbol{x}) - \hat{\boldsymbol{\tau}}_i^{\mathrm{T}} \boldsymbol{k}(\boldsymbol{x})}{\sqrt{(\hat{\boldsymbol{\tau}}_y - \hat{\boldsymbol{\tau}}_i)^{\mathrm{T}} \boldsymbol{K} (\hat{\boldsymbol{\tau}}_y - \hat{\boldsymbol{\tau}}_i)}}. \tag{29}$$

以上を踏まえ，KMCE 学習は，学習標本群 $\{\boldsymbol{x}_n\}_{n=1}^N$ 上において，この新しい誤分類尺度を平滑化分類誤り数損失を用いて評価し，学習対象パラメータ $\{\hat{\boldsymbol{\tau}}_j\}_{j=1}^J$ と $\{\boldsymbol{p}_m\}_{m=1}^M$ との更新を通して，写像された特徴空間 H における幾何マージンを大きくしつつ分類誤り数を小さくできるパラメータ状態を探索する．

4 評価実験

4.1 概要

MSVM 法と KMCE 法を比較するため，まず MSVM 法によってカーネル写像を伴う線形識別関数型分類器を作成し，次にその分類器を初期状態として KMCE 学習法を行い，KMCE 法に基づくカーネル写像を伴う線形識別関数型分類器を作成した．また，先に述べたように，MSVM 学習法は，サポートベクター数が膨大になる欠点がしばしば指摘されてきた．学習を通して自動的に選択される MSVM 学習のサポートベクターと異なり，KMCE 学習法においてはプロトタイプ数をその学習手続きとは独立に定めることが出来る．しかも，そのプロトタイプを，分類力を高めるべく最適化することも出来る．その結果，KMCE 学習は，サポートベクターよりも少数のプロトタイプによって高い分類力を達成し得ることが期待できる．この点を検証するため，MSVM 学習法によって得られたサポートベクターの一部を削除することによって得た小規模な分類器を対象に，再び MSVM 学習と KMCE 学習を適用し，その結果の比較も行った．なお，この再学習において，KMCE 学習法は，通常のその実行と同様に，一部が削除されたサポートベクターをプロトタイプとして，そのプロトタイプ自身とそれに対応する重みの双方を，学習標本上で再学習した．一方，MSVM 法は，サポートベクターを選択するその手続きの原理上，削減されたサポートベクターから成る分類器の再学習に，元々の学習標本を用いることは困難である．その再学習は，残されたサポートベクターを学習標本として，そこから新たにサポートベクターを選び直すことで行わざるを得ない．本実験でも，MSVM 法の再学習は残されたサポートベクターを学習標本として行った．

サポートベクターの削減は，MSVM 法で得られた各サポートベクターに対応する重みベクトルのノルムの大きさを基準として行った．即ち，ノルムが大きい重みをもつサポートベクターは識別関数値の決定に対する寄与が大きく，ノルムが小さな重みをもつサポートベクターの寄与は小さいと考えた．具体的には，正の定数である閾値 T を制御して，サポートベクター $\boldsymbol{x}_n$ に対応する重みベクトル $\boldsymbol{\tau}_n$（$= (\tau_{n,1}, \tau_{n,2}, \ldots, \tau_{n,J})^{\mathrm{T}}$）が $\|\boldsymbol{\tau}_n\| \geq T$ となる場合は，そのサポートベクターと重みを残し，それ以外の場合は削除した．なお，式 (9) の制約式より，$\|\boldsymbol{\tau}_n\|$ の取り得る値の範囲は $0 \leq \|\boldsymbol{\tau}_n\| \leq \sqrt{2}$ となる．

サポートベクターが削減された分類器を再学習する際，KMCE 学習法では残された $\boldsymbol{x}_n$ と $\boldsymbol{\tau}_n$ を初期化に用いて学習を行った．一方，MSVM 法では，残された $\boldsymbol{x}_n$ を学習標本として再度学習を行った．

なお，MSVM 法と KMCE 学習法のいずれにおいても，カーネルには次式のガウスカーネルを用いた．

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{\sigma^2}\right) \quad (\sigma > 0), \tag{30}$$

ここで σ はカーネル幅を示す．

4.2 データおよびその利用法

実験には，UCI Machine Learning Repository が提供する Letter Recognition データセット（以下 LR データセット）と Abalone データセットを使用した．LR データは英語アルファベットのフォント文字画像から特徴抽出されたデータ 20000 個で構成される，16 次元，26 クラスのデータセットである．Abalone データセットは，アワビの測定値データ 4177 個で構成される，7 次元，3 クラスのデータセットである．

MSVM 法も KMCE 法も，学習対象パラメータを学習するために用いる学習標本群に加えて，カーネル幅などのハイパーパラメータを適切に設定するために用いる評価標本群が必要である．実験では，それぞれのデータセットを，以下のように，学習用と評価用，(学習にも評価にも登場しない未知標本に相当するものとして）試験用とに 3 分割した．LR データセットでは，学習用標本を 1000 個に，評価用標本を 9500

個に，試験用標本を 9500 個とした．また，Abalone データセットでは，学習用標本を 1045 個に，評価用標本を 1566 個に，試験用標本を 1566 個とした．

4.3 結果と考察

LR データセットに対する MSVM 法の結果を表 1 に，KMCE 学習法の結果を表 2 に示す．表中，左端の T は重みベクトルを削減する際に用いた閾値を，サイズは残されたサポートベクター（プロトタイプ）の数を示している．また，学習用と評価用，試験用の欄にある数字は，それぞれの標本群に対する分類率（%）である．なお，本稿中の全ての分類率は，評価用標本に対して最も高い分類率を記録したときの分類器（分類器パラメータの状態）によって得られたものである．各表において，最上段の “縮小なし” は，重みベクトルのノルムを用いた削減を行わずに，MSVM 法によって得られたサポートベクターを全て分類器パラメータとして残したことを意味している．

まず，表 1 と表 2 の双方における最上段の結果を比較する．両学習法ともに，試験用標本に対して 82.5% 前後の高い分類精度を達成していること，および，その達成のために，初期化手続きに相当する MSVM 法が 1000 個の学習標本の 80%に及ぶ 807 個の学習標本をサポートベクターとして残していることがわかる．2 つの学習法を比べると，若干ではあるものの，KMCE 法の分類率は MSVM 法のそれを上回っている．これは，用いる損失の違いや，MSVM 法では固定されていたサポートベクターが，KMCE 法では重みベクトルとともに学習の更新対象となっていたことに起因するものと考えられる．

次に，サイズを小さくした場合の結果を比較する．まず，両学習法ともに，サイズが小さくなるにつれてほぼ単調に分類率は低下している．しかし，両学習法の間で比較すると，MSVM 法の分類率がサイズ縮小と共に著しく低下しているのに対し，KMCE 学習法のそれの低下は優れて抑制されている．サイズが小さな場合における KMCE 学習法の優位性は，明らかに，重みとともにプロトタイプも学習できるその学習の特徴に拠るものと考えられる．また興味深いことに，サイズを MSVM 法によるそれの 3/4 程度にまで小さくしても，KMCE 学習は MSVM 法を上回る分類精度を達成していることがわかる．ここでも再び，KMCE 学習法がもつ，より直接的にベイズ誤りの推定を目指す特質が影響したものと考えられる．

Abalone データセットに対する MSVM の結果を表 3 に，KMCE 学習法の結果を表 4 に示す．LR データセットを使用した場合ほど顕著ではないものの，やはりサイズが小さくなるにつれて，両手法の分類率の差は大きくなっている．このデータにおいても，特にプロトタイプ数が小さな分類器の学習に対する，KMCE 学習法の MSVM 法に対する優位性は明らかであった．

表 1　MSVM の結果（LR データセット）．

T	サイズ	学習用	評価用	試験用
縮小なし	807	99.8%	83.26%	82.44%
0.1	696	99.8%	83.22%	82.36%
0.2	591	96.9%	80.48%	79.96%
0.3	485	87.9%	72.49%	71.69%
0.4	399	78.7%	65.20%	64.26%
0.5	328	68.7%	55.88%	55.66%
0.6	276	61.8%	50.58%	50.66%

表 2　KMCE 学習法の結果（LR データセット）．

T	サイズ	学習用	評価用	試験用
縮小なし	807	100%	83.27%	82.68%
0.1	696	99.9%	82.87%	82.82%
0.2	591	99.9%	82.94%	82.67%
0.3	485	99.5%	82.55%	82.08%
0.4	399	99.1%	82.48%	81.96%
0.5	328	99.6%	81.91%	81.48%
0.6	276	99.3%	81.77%	81.24%

表 3　MSVM の結果（Abalone データセット）．

T	サイズ	学習用	評価用	試験用
縮小なし	825	65.7%	65.6%	64.9%
0.2	821	65.5%	65.5%	64.8%
0.4	800	66.3%	65.2%	65.4%
0.6	790	59.2%	57.6%	57.2%
0.8	784	57.4%	57.0%	56.6%
1.0	778	56.1%	54.8%	54.7%
1.2	776	56.0%	54.6%	55.2%
1.4	670	45.5%	42.4%	46.7%

表 4　KMCE 学習法の結果（Abalone データセット）．

T	サイズ	学習用	評価用	試験用
縮小なし	825	65.3%	65.6%	64.8%
0.2	821	66.0%	65.5%	65.4%
0.4	800	65.8%	63.8%	64.1%
0.6	790	65.5%	63.6%	63.9%
0.8	784	65.7%	63.8%	64.1%
1.0	778	64.5%	63.0%	63.4%
1.2	776	62.8%	62.0%	61.6%
1.4	670	64.5%	62.6%	62.0%

上述したように，MSVM 法の再学習は，削減後に残されたサポートベクターを学習標本として行った．表中の，サイズが小さな場合の MSVM 法の分類精度が著しく低い原因が，この限られた標本のみを用い

る再学習そのものにある可能性を考えることもできる．この点を確かめるため，再学習を行う前の，サポートベクターを削減した直後の MSVM 法の分類精度を表 5 と表 6 にまとめた．

表 5　再学習を伴わない MSVM の結果（LR データセット）．

T	サイズ	学習用	評価用	試験用
縮小なし	807	99.8%	83.26%	82.44%
0.1	696	99.8%	81.66%	81.16%
0.2	591	79.9%	60.21%	60.27%
0.3	485	54.8%	41.63%	40.97%
0.4	399	35.8%	29.01%	28.14%
0.5	328	26.9%	21.69%	22.51%
0.6	276	17.9%	16.21%	16.28%

表 6　再学習を伴わない MSVM の結果（Abalone データセット）．

T	サイズ	学習用	評価用	試験用
縮小なし	825	65.3%	65.6%	64.8%
0.2	821	46.5%	45.3%	47.0%
0.4	800	31.7%	31.7%	31.7%
0.6	790	31.7%	31.7%	31.7%
0.8	784	31.7%	31.7%	31.7%
1.0	778	31.7%	31.7%	31.7%
1.2	776	31.7%	31.7%	31.7%
1.4	670	30.8%	31.7%	30.4%

表 1 と表 5 の比較，および表 3 と表 6 の比較は，再学習前の分類精度が再学習後のそれよりも明らかに低いことを示している．再学習は，サポートベクターが削減された MSVM 法の分類力の回復に，一定の効果を持つことを確認することができる．

5　まとめ

カーネルに基づく入力ベクトルパターンの特徴写像を伴う，2 つの分類器学習法，多クラスサポートベクターマシン（MSVM）法とカーネル最小分類誤り（KMCE）学習法の，実験的比較を行った．Letter Recognition データセットと Abalone データセットとの 2 種のデータを用いた実験の結果，プロトタイプとそれに対する重みの双方を学習できる KMCE 法が，特に少数のプロトタイプからなる分類器の場合に，MSVM 法に対する明確な優位性を示すことが明らかとなった．KMCE 学習法は，MSVM 法に伴うスケーラビリティや実装困難性の問題を解決する，分類器学習の有望な選択肢になり得ることが期待できる．

謝辞

本研究の一部は，科研費（番号 : 26280063）及び私学研究基盤形成支援事業「ドライバ・イン・ザ・ループ」の支援を受けて行われた．

参考文献

1) Koby Crammer and Yoram Singer: "On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines", Journal of Machine Learning Research, vol. 2, pp. 265-292 (2001).

2) 渡辺秀行, 片桐滋, 足立守, 大崎美穂: "カーネルに基づく高次元空間における大幾何マージン最小分類誤り学習の提案," PRMU, Dec.2010.

3) 渡辺秀行, 片桐滋, 山田幸太, マクダーモットエリック, 中村篤, 渡部晋治, 大崎美穂: "幾何マージンに基づく誤分類尺度を用いた最小分類誤り学習法", 電子情報通信学会論文誌. D, 情報・システム vol. J94-D(10), pp. 1664-1675 (2011).

4) Biing-Hwang Juang and Shigeru Katagiri: "Discriminative Learning for Minimum Error Classification," IEEE Trans. Signal Processing, vol. SP-40, no. 12, pp. 3043-3054 (1992).