

最小分類誤り学習法におけるプロトタイプ数の自動的最適化手法の実験的評価

Experimental Evaluation of Automatic Optimization Methods for the Number of Prototypes in Minimum Classification Error Training

荻野将大† 渡辺秀行‡ 片桐滋† 大崎美穂† 堀智織‡
Masahiro Ogino Hideyuki Watanabe Shigeru Katagiri Miho Ohsaki Chiori Hori

1. はじめに

パターン認識における学習方法の一つに最小分類誤り (MCE: Minimum Classification Error) 学習法がある [1]. MCE 学習法は, パターン標本の分類の正誤の程度を示す誤分類尺度と, 誤分類尺度をもとに損失を与える平滑な 0-1 損失関数を定義することで, 誤分類数を損失として評価する. 損失関数の平滑度は, 分類器パラメータを最適化するために勾配法を適用可能にする. また, 損失関数平滑度は未知標本に対する耐性を高めるための重要な要素であり, 与えられたデータに対して適切に設定される必要がある.

MCE 学習法の有用性は音声認識などの様々な利用において実証されてきたが, 未知標本耐性の向上につながる幾何マージン増加に関しては, それを保証する機能を持っていなかった. そのため, この問題を軽減することを目指して MCE 学習法に幾何マージンの概念を導入した大幾何マージン最小分類誤り (LGM-MCE: Large Geometric Margin Minimum Classification Error) 学習法が提案された [2]. 幾何マージンとは正しく分類されたパターン標本と決定境界との最小距離であり, 幾何マージンを大きくすることで決定境界付近の誤りやすい未知標本に対しても正しく分類できる可能性が高まる. LGM-MCE 学習法は幾何マージンが大きくなるように学習するので, MCE 学習法に比べて未知標本に対する耐性が強化されており, 成果をあげてきた. しかし, LGM-MCE 学習法をプロトタイプ型分類器 [3] で用いる場合, 各クラスを代表する特徴ベクトルであるプロトタイプの数に各クラスに同数割り当てている. 各クラスに同数のプロトタイプを割り当てると, クラスによってプロトタイプ数が過剰, または不足するクラスが出現する可能性がある. プロトタイプが過剰に存在すると学習標本に適合しすぎてしまい, 過学習状態になってしまう. 逆にプロトタイプが不足すると学習標本を表現することが困難になる. 各クラスのプロトタイプを適切に表現できないと複雑なクラス境界をもつデータを分類する問題において分類精度の低下を招く恐れがある.

そこで各クラスのプロトタイプ数をあらかじめ一律に決定してしまうのではなく, 各クラスの最適なプロトタイプ数を自動的決定する手法が考案された [4][5]. その手法とは, まず初期化の段階で各クラスに 1 つずつプロトタイプを求める. そして LGM-MCE 学習法を行うたびに分類率の最も低いクラスにプロトタイプを逐次追加して分類精度の向上を図りつつ, 過学習を起ささないようにプロトタイプの追加終了条件を定めて, その条件を満たすと追加を終了する. この手法を用いることで, 最適なプロトタイプ数

を自動で求めて各クラスを適切に表現することができる. 先行研究において幾何マージンに基づくプロトタイプの追加法が考案され, 実験が行われてきたがこれ以外にもプロトタイプの追加法に関しては様々な方法を考えることができる [6]. そこで本稿では学習標本の分類率を用いた追加法を提案する. この追加法は追加候補の学習標本のそれぞれに対して一旦プロトタイプとして仮追加し, 学習標本に対する分類率を算出する. そして分類率の最も良かった時に仮追加していた学習標本を最終的に追加するプロトタイプとして選出する.

本稿の目的として幾何マージンに注目した追加法と学習標本の分類率に注目した追加法との比較実験を行い, 追加法を検討する. また従来の LGM-MCE 学習法を用いたプロトタイプ型分類器との比較評価も行う. 今回の一部の実験条件下で提案手法と幾何マージンに基づく追加法を未知標本の分類率の観点から比較すると, 幾何マージン追加法のほうがより良い分類率となった. また, これらのプロトタイプ逐次追加法と従来法を比較すると, プロトタイプ逐次追加法は従来よりも高い分類精度を示し, 有効な手法であることが明らかになった.

2. 大幾何マージン最小分類誤り学習法

2.1 最小分類誤り学習法の定式化

入力パターン空間 \mathcal{X} 上に存在する入力標本集合 $\{(x_n, y_n)\}_{n=1}^N$ の 1 つを x_n とし, この標本 x_n を J 個のクラス $\{C_j\}_{j=1}^J$ のいずれか 1 つに分類する問題を考える. MCE 学習では識別関数を用いて, 分類決定則を以下のように定義する.

$$C(x_n) = C_k \quad \text{iff} \quad k = \underset{j}{\operatorname{argmax}} g_j(x_n; \Lambda) \quad (2.1)$$

このとき $g_j(x_n; \Lambda)$ は入力パターン x_n がクラス C_j に帰属する程度を示す識別関数である. Λ は学習パラメータ集合であり, $g_j(x_n; \Lambda)$ は Λ に関して微分可能であるとする. 次に, 識別関数の出力値を用いて分類の誤り度合を表す誤分類尺度を以下のように定義する.

$$d(x_n; \Lambda) = -g_y(x_n; \Lambda) + \ln \left[\frac{1}{J-1} \sum_{j:j \neq y} \exp(\eta g_j(x_n; \Lambda)) \right]^{\frac{1}{\eta}} \quad (2.2)$$

ここで, y は x_n が実際に属するクラスラベルであり, η は正の数である. 上で示した誤分類尺度の式 (2.2) において $\eta \rightarrow \infty$ のとき, 誤分類尺度は次式となる.

†同志社大学, Doshisha University

‡ 情報通信研究機構, NICT

$$d(x_n; \Lambda) = -g_y(x_n; \Lambda) + \max_{j, j \neq y} g_j(x_n; \Lambda) \quad (2.3)$$

式(2.3)は標本 x_n の誤分類の割合をあらわしている。 $d(x_n; \Lambda) > 0$ のとき誤った識別をしており、 $d(x_n; \Lambda) < 0$ のとき正しい識別をしている。そのため、分類誤り数を求めるには以下の0-1関数を用いればよい。

$$\ell(x_n; \Lambda) = \begin{cases} 1 & \text{if } d(x_n; \Lambda) > 0 \\ 0 & \text{if } d(x_n; \Lambda) \leq 0 \end{cases} \quad (2.4)$$

しかし、式(2.4)は学習パラメータ集合 Λ について微分不可能であり、勾配法による学習に適さない。そのため、微分可能であるシグモイド関数で0-1損失関数を近似することで、以下の平滑化分類誤り数損失関数を定義する。

$$\ell(d(x_n; \Lambda)) = \frac{1}{1 + \exp(-\alpha d(x_n; \Lambda))} \quad (2.5)$$

ここで α は正の数であり、損失関数の滑らかさを示す。 α の値が大きいかほど0-1損失関数の近似となり、小さいほどより平滑な関数となる。そして、次式に示す全サンプルの損失を平均した経験的平均損失の最小化を目指す学習を行う。

$$L(\Lambda) = \frac{1}{N} \sum_{n=1}^N \ell(d(x_n; \Lambda)) \quad (2.6)$$

この経験的平均損失の最小化には、最急降下法を用いることができる。 $L(\Lambda)$ を最小化する Λ は次式の繰り返しにより求まる。

$$\Lambda^{(t+1)} = \Lambda^{(t)} - \varepsilon_t \frac{1}{N} \sum_{n=1}^N \nabla_{\Lambda} \ell(d(x_n; \Lambda^{(t)})) \quad (2.7)$$

ここで、 ε_t は学習レートであり、正の数である。また、 t は学習の繰り返しにおけるステップ回数を意味している。 ∇_{Λ} は Λ の偏微分を示し、 $\Lambda^{(t)}$ は t 回目の学習によって求められたパラメータである。初期値 $\Lambda(0)$ は初期化されているものとする。式(2.7)より、パラメータ Λ の一回の更新において全ての学習標本を用いているので、バッチ的な手法である。これに対して、1つの学習標本を得るごとに Λ を調整する逐次的な手法も存在する。この手法は確率的降下法といい、更新式を次式に示す。

$$\Lambda^{(t+1)} = \Lambda^{(t)} - \varepsilon_t \nabla_{\Lambda} \ell(d(x_n; \Lambda^{(t)})) \quad (2.8)$$

ここでの ε_t も正の数である。経験的平均損失 $L(\Lambda)$ が局所的な最小解になる条件は ε_t が以下の2つの条件を満たしているときである。

$$\sum_{t=0}^{\infty} \varepsilon_t = \infty \quad (2.9)$$

$$\sum_{t=0}^{\infty} \varepsilon_t^2 < \infty \quad (2.10)$$

2.2 大幾何マージン最小分類誤り学習法の定式化

分類問題において学習標本を用いて決定境界を定めるとき様々な決定境界を引くことができるが、未知標本に対し

ても正分類を行うことができる可能性を高める決定境界を引くためには幾何マージンに注目する。幾何マージンとは決定境界に最も近い正分類された学習標本と決定境界とのユークリッド距離である。幾何マージンは次式で近似的に表現される。

$$r(x_n; \Lambda) \approx -\frac{d(x_n; \Lambda)}{\|\nabla d(x_n; \Lambda)\|} \quad (2.11)$$

$r(x_n; \Lambda) > 0$ のとき正分類となり、 $r(x_n; \Lambda) < 0$ のとき誤分類となる。 $r(x_n; \Lambda)$ が大きければ幾何マージンが増大し、決定境界付近の誤分類しやすい未知標本に対しても正分類できる可能性が高まる。従来のMCE学習法も $d(x_n; \Lambda)$ を負の方向に大きくするように学習を行うので式(2.11)の分子については問題ないのだが、分母の $\|\nabla_{\mathbf{x}} d(x_n; \Lambda)\|$ の値が増大してしまうと幾何マージンの増加に繋がらない。そこでMCE学習法に幾何マージンの概念を取り入れたLGM-MCE学習法は式(2.11)を正負反対にしたものを新しい誤分類尺度として定義する。

$$D(x_n; \Lambda) = \frac{d(x_n; \Lambda)}{\|\nabla d(x_n; \Lambda)\|} \approx -r(x_n; \Lambda) \quad (2.12)$$

MCE学習法の誤分類尺度 $d(x_n; \Lambda)$ と同様にLGM-MCE学習法の誤分類尺度 $D(x_n; \Lambda)$ も値が負であれば正分類を表し、値が正であれば誤分類を表す。 $D(x_n; \Lambda)$ が負の方向に大きくなることは、幾何マージンを増大することになり、MCE学習法と比較して未知標本に対する識別性能の向上が見込める。MCE学習法に対してLGM-MCE学習法の相違点は誤分類尺度を $d(x_n; \Lambda)$ から $D(x_n; \Lambda)$ に定義した点である。

2.3 プロトタイプ型分類器を用いた実装

プロトタイプとはそのクラスを代表とするパターンであり、各クラスに設定する。ある学習標本ベクトル x_n がどのクラスに属するか判断する基準として各プロトタイプとのユークリッド距離を計算し、最も距離の近いプロトタイプが属するクラスを x_n の識別結果とする。クラス C_i に対する識別関数を次式に示す。

$$g_i(x_n; \Lambda) = -\|x_n - p_i\|^2 \quad (2.13)$$

ここで p_i はクラス C_i のプロトタイプ特徴ベクトルであり、クラス C_i に複数のプロトタイプがある場合、 x_n に最も近いものとする。識別決定則としては式(2.1)と同様である。ある学習標本 x_n が C_y に属しており、式(2.2)において $\eta \rightarrow \infty$ としたときの $d(x_n; \Lambda)$ は次式となる。

$$d(x_n; \Lambda) = \left\| x_n - p_y \right\|^2 - \left\| x_n - p_j \right\|^2 \quad (2.14)$$

$$j = \underset{i, i \neq y}{\operatorname{argmax}} g_i(x_n; \Lambda)$$

ここでクラス C_j は x_n が最も誤りやすいクラスであり p_i はクラス C_j の x_n に最も近いプロトタイプ特徴ベクトルである。LGM-MCE学習法における誤分類尺度 $D(x_n; \Lambda)$ は

$$D(x_n; \Lambda) = \frac{\|x_n - p_y\|^2 - \|x_n - p_j\|^2}{2\|p_y - p_j\|} \quad (2.15)$$

となる。新しい誤分類尺度 $D(x_n; \Lambda)$ を定義したので次に損失関数の式(2.5)、経験的平均損失式(2.6)のそれぞれの

$d(x_n; \Lambda)$ に対して $D(x_n; \Lambda)$ で置き換える。こうすることで経験的平均損失を求めることができ、LGM-MCE 学習法は幾何マージンを増大させ、経験的平均損失の最小化を目指す学習を行う。

3. クラスモデルサイズの自動的最適化

3.1 プロトタイプ数の自動的最適化手法

クラスを表現するモデルの一例としてプロトタイプモデルに焦点を当てる。プロトタイプモデルを用いた場合、クラスモデルサイズを最適化することは各クラスのプロトタイプ数を最適化することと同等である。従来のプロトタイプを用いたLGM-MCE 学習法は全クラスに対して同じ数のプロトタイプを設定していた。しかし、全クラスに同数のプロトタイプを設定してしまうと、あるクラスにはプロトタイプ数が過剰な状態で設定されていたり、またあるクラスにはプロトタイプ数が不足した状態で設定されてしまう可能性がある。プロトタイプ数が過剰なクラスは学習標本に過度に適合して、未知標本に対する分類性能が低下する過学習という現象が起こる。またプロトタイプ数が不足しているクラスは表現能力が低いため分類精度が低下する。この問題を解決するために学習の過程でクラスごとの分類性能を計測し、それに基づいて各クラスのプロトタイプ数を自動的に最適化する手法を提案する。この提案手法は初期化の際に各クラスに1つプロトタイプを設定して、学習の過程のなかでクラスごとの学習標本に対する分類精度を計算し、最も分類精度の低いクラスにプロトタイプを1つ追加していく。その提案手法の流れの概要を図3.1に示す。

初期化: K平均法を用いて各クラスに1つずつプロトタイプを設定する
 終了条件を満たすまで1,2,3の手順をループ

1. LGM-MCE学習を行う
2. 検証用標本に対する認識率を算出する
3. プロトタイプを追加する

図3.1 プロトタイプ数最適化の手順

まず初期化の段階で K 平均法を用いて各クラスにプロトタイプを1つ設定する。ここでは $K=1$ なので各クラスの重心がプロトタイプとなる。そして終了条件を満たすまで1,2,3の手順を繰り返し行うのだが終了条件については3.3節にて後述する。手順1では学習標本を用いてLGM-MCE 学習を行い、プロトタイプを更新していく。手順2で学習後のプロトタイプを用いて検証用標本の分類率を算出し、未知標本に対する分類率の指標とする。そして手順3で学習標本においても分類率を算出し、分類率の最も低かったクラスにプロトタイプを1つ追加する。分類率の最も低いクラスにプロトタイプを追加することで、学習標本に対する分類精度の向上を目指すことになるが、学習標本の分類精度のより一層の向上を目指したり、未知標本に対する耐性を考慮するなど、追加するプロトタイプの初期値によって様々な方針がある。その中の一例として先行研究の追加法である幾何マージンに着目した追加法と本稿で提案する学習標本の分類率に着目した追加法を3.2節にて後述する。以上の3つの手順を終了条件を満たすまで繰り返し行うことで各クラスのプロトタイプ数の自動的最適化を行う。

3.2 プロトタイプの逐次的生成手順

図3.1の手順3におけるプロトタイプを逐次的に生成し追加する手順の詳細を示す。逐次追加方法は以下の4つに分かれる。

- (1) 学習標本の分類結果からConfusion Matrix (CM) 作成
 LGM-MCE 学習法を行った後に学習標本 $\{(x_n, y_n)\}_{n=1}^N$ に対して分類を行い、CMを作成する。CMの概念を図3.2に示す。

		識別結果		
		C_1	\dots	C_j
所属クラス	C_1	s_1^1	\dots	s_j^1
	\vdots	\vdots	\ddots	\vdots
	C_i	s_1^i	\dots	s_j^i

図3.2 Confusion Matrix の概念。

CMとはクラスが J 個存在するとき $J \times J$ 行列となり、各要素は $C_i (i = 1, \dots, J)$ に属し $C_j (j = 1, \dots, J)$ が分類結果である学習標本の総数となる。CMの (i, j) の要素を s_j^i としたとき s_j^i は次式で求めることができる。

$$s_j^i = \sum_{n=1}^N 1(x_n \in C_i) 1(C(x_n) = C_j) \quad (3.1)$$

s_j^i を $i = 1, \dots, J, j = 1, \dots, J$ のそれぞれについて計算してCMを作成する。

- (2) 分類精度が最も悪いクラスを算出
 (1)で作成したCMを用いて各クラスの分類率を算出し、最も分類率の低かったクラスをworstクラス C_w とする。worstクラスを求める式を以下に示す。

$$\text{worst class} = \underset{i}{\operatorname{argmin}} \frac{s_i^i}{\sum_j s_j^i} \quad (3.2)$$

- (3) worstクラスが最も誤分類したクラスを算出
 次に、worstクラスに属する学習標本が最も誤りやすいクラスを導出する。この最も誤りやすいクラスをmost-confusableクラス C_{mc} と定義し、以下の式で算出する。

$$\text{most-confusable class} = \underset{j, j \neq \text{worst}}{\operatorname{argmax}} s_j^{\text{worst}} \quad (3.3)$$

- (4) worstクラスのプロトタイプを新たに1つ生成
 新しいプロトタイプの生成に関して説明する。学習標本に対する分類精度に悪影響を及ぼす原因の1つに先ほど求めた C_w と C_{mc} の間の決定境界が適切に引かれていないことが挙げられる。したがって分類精度を向上するために誤って C_{mc} として分類されてしまった C_w の学習標本を新たなプロトタイプとして追加する。プロトタイプの追加方法には様々な方法が考えられるが、本稿では C_w に属しながらも C_{mc} に誤って分類されてしまった学習標本 $\{x_m^{w,mc}\}_{m=1}^M$ から1つを選出して、 C_w のプロトタイプとして追加する。 $\{x_m^{w,mc}\}_{m=1}^M$ から1つを選択する基準は様々なものが存在するなか、そのうちの2つを説明する。

1つ目の選択基準として幾何マージンに注目した追加方法を図3.3に示す。

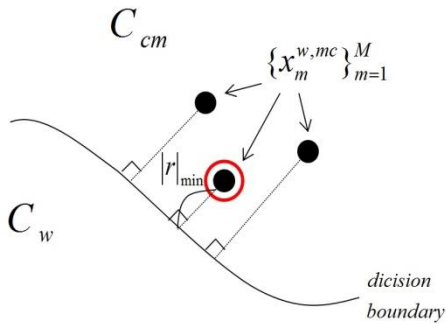


図3.3 幾何マージンに注目した追加するプロトタイプの初期値。

$\{x_m^{w,mc}\}_{m=1}^M$ のそれぞれに対して幾何マージンの絶対値を算出し、その絶対値が最小となる学習標本の特徴量を追加するプロトタイプの初期値とする。つまり $\{x_m^{w,mc}\}_{m=1}^M$ のうち、 C_w との決定境界に最も近いものを C_w のプロトタイプとして追加する。この追加方法を用いることで、 C_w と C_{cm} の間で適切に表現できていない決定境界付近にプロトタイプを追加し決定境界の改善を図る。

2つ目の選択基準として仮追加後の学習標本に対する分類率に注目した追加方法を図3.4に示す。

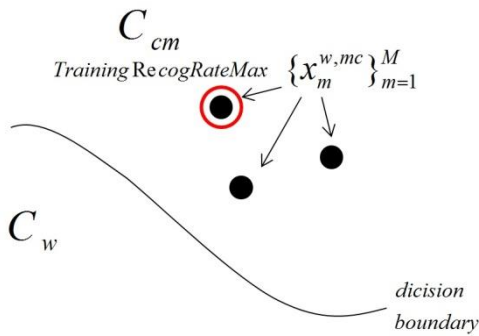


図3.4 仮追加後の学習標本に対する分類率に注目したプロトタイプの初期値。

$\{x_m^{w,mc}\}_{m=1}^M$ のそれぞれに対して一旦プロトタイプとして仮追加し、そのあと学習標本に対する分類率を算出する。そして仮追加した中で、最も分類率が高くなったときの $\{x_m^{w,mc}\}_{m=1}^M$ を最終的に追加するプロトタイプとして決定する。この選択基準を採用することで、学習標本に対してより適したプロトタイプを初期値としてあたえることができる。

ここまで最も分類率の低いクラスに属しながらも、最も誤りやすいクラスに誤って分類されてしまった学習標本である $\{x_m^{w,mc}\}_{m=1}^M$ が存在するものとしてプロトタイプの追加方法を説明してきたが、 $\{x_m^{w,mc}\}_{m=1}^M$ が存在しない場合、つまり学習標本に対する分類率が100%の場合も考えることができ、 C_w を算出することができない。そこで学習標本に対する分類率が100%の場合のプロトタイプ追加法について説明する。プロトタイプの追加候補として全ての学習標本に注目する。全学習標本のそれぞれに対して幾何マージンを算出し、その絶対値が最小となる学習標本を求め、幾何マージンの絶対値が最小となる学習標本をプロトタイプとして追加する。このようにすることで決定境界付近の誤分

類しやすい未知標本に対する耐性を向上することができる。

3.3 プロトタイプの逐次的生成手順

プロトタイプを逐次追加していくのだが、どの段階のプロトタイプ数を最適なプロトタイプ数と判断し学習を終了するかについて述べる。プロトタイプを追加し、学習することを繰り返し行うことで学習標本に対する分類率は向上していくが、一定以上のプロトタイプ数を追加すると過学習をおこして未知標本に対する分類率が低下してしまう。よって章の図1に示した通り、プロトタイプを追加することによってLGM-MCE学習を行ったあと学習に用いない検証用標本に対する分類率を算出してその分類率を最適なプロトタイプ数を決定するための指標とする。検証用標本に対する分類率の推移として図3.5のようなグラフが得られたとする。

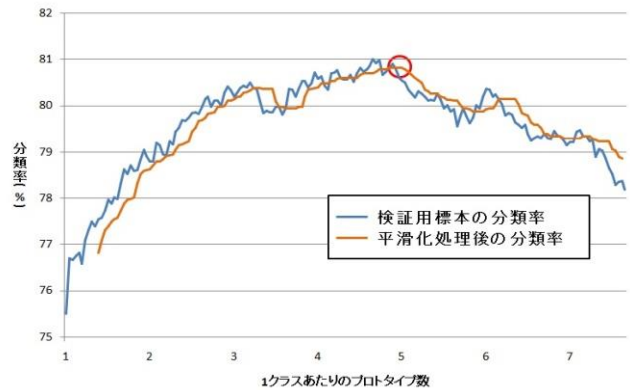


図3.5 プロトタイプ追加ごとの検証用標本の分類率の推移。

このグラフを見る限り、プロトタイプを追加していくと分類性能の向上により分類率が上昇する範囲と過学習を起こしてしまい分類率が低下していく範囲が存在する。よって両範囲の中間周辺のプロトタイプ数を最適なモデルと考えるのが妥当である。しかしプロトタイプを追加するたびに検証用標本の分類率を算出した値を見てみると、値が変動していることがわかる。値の大きな変動具合を排除した安定した推移に基づいて最適なプロトタイプ数を決定するため、分類率の推移にメディアンフィルターを適用し平滑化を行い、その平滑化曲線上で最も高い分類率を出すプロトタイプ数が最適であると決定する。図3.5の場合、1クラスあたりの平均プロトタイプ数が5付近の状態が最適なプロトタイプ数状態として選択される。次に学習の終了条件について述べる。プロトタイプを追加していく上で、最適なプロトタイプ数状態を過ぎると、各クラスに過剰にプロトタイプを追加することになり、過学習の現象が確認できる。そのため、平滑化後の分類率のピークを逐一記憶しておき、ピーク時から閾値よりも低い分類率が算出された場合には過学習と判定し、学習を終了する。

4. 評価実験

4.1 目的と条件

プロトタイプを逐次的に増やしプロトタイプ数の最適化を行う手法において3章の(4)で説明した幾何マージンに基づく追加法と仮追加後の学習標本に対する認識率に基づく追加法をそれぞれ幾何マージン法、仮追加法と呼ぶ。これら両追加法の比較実験を行い検証する。また、プロトタイプ数を一律に設定していた従来のLGM-MCE学習法においても比較実験を行う。今回の実験で使用するデータセッ

トとしてUCI Machine Learning Repository が提供するLetter Recognition データセット(以下Letter データ) と Abalone データセット(以下Abalone データ) を用意した. Letter データは, 英語アルファベットのフォント文字画像から抽出された20,000 個のデータで構成される, 26 クラス, 16 次元のデータセットである. Abalone データはアワビ測定値から抽出された4177 個のデータであり, 若年層, 中年層, 高年層の3 クラスに分けた7 次元のデータセットである. 識別性能の評価方法としてデータセットを学習標本, 検証用標本, 未知標本に分割するHoldOut 法を用いる. 今回の実験においてLetter データ20000 個のうち1000 個を学習標本, 9500 個を検証標本, 9500 個を未知標本とする. また, Abalone データ4177 個のうち1045 個を学習標本, 1566 個を検証標本, 1566 個を未知標本とする. 以下, 学習用標本集合による評価, Validation 標本集合による評価, Open 標本集合による評価をそれぞれClosed Test, Validation Test, Open Test と呼ぶ. 従来のLGM-MCE法においてプロトタイプ型分類器を用い, プロトタイプ数は各クラス共通, 1~10 の10 通りで実験を行った. 一方, プロトタイプ数を自動的に最適化する方法は, 3節で紹介したプロトタイプを逐次的に生成し, プロトタイプの追加法として幾何マージン法と仮追加法でそれぞれ実験を行った. その際, LGM-MCE 法で用いるあらかじめ設定しなければならない値 (ハイパーパラメータ) である学習係数と損失平滑度は両追加法統一して同じ値を使用した. プロトタイプの自動的最適化手法における最適なプロトタイプ数の決定則で用いるメディアンフィルターのフレーム数を前後5 フレーム, 7 フレームに設定した.

4.2 結果と考察

まずLetter データの結果について述べる. 幾何マージン法, 仮追加法のそれぞれに対してValidation Test の結果へメディアンフィルターの前後5フレームによる平滑化処理を行った場合に, 最大の分類率となった状態の各クラスのプロトタイプ数を図4.1, 図4.2 に示す.

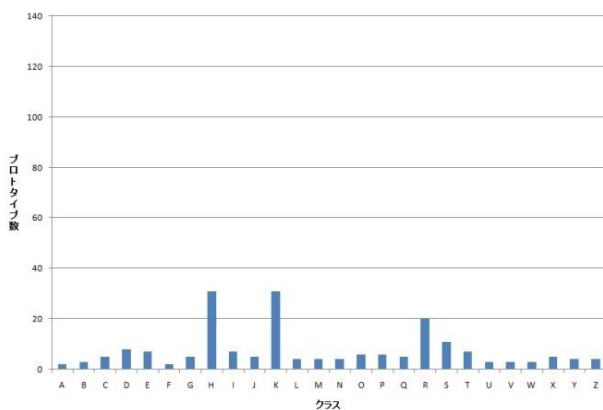


図4.1 各クラスのプロトタイプ数 (Letter データ, 幾何マージン法) .

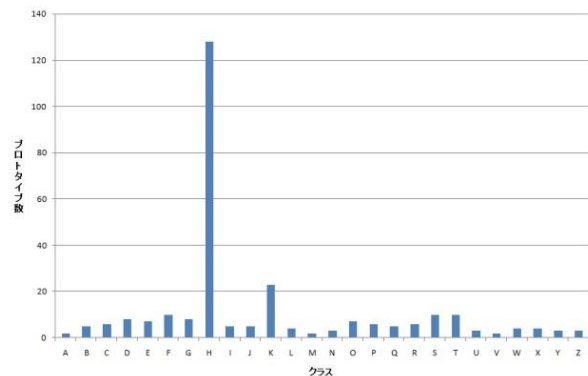


図4.2 各クラスのプロトタイプ数 (Letter データ, 仮追加法) .

このときの1 クラス当たりのプロトタイプ数の平均はそれぞれ7.5 個, 10.73 個であった. 図4.1, 図4.2 より, 両追加法において各クラスのプロトタイプ数は異なっている. このことからプロトタイプを追加し, 学習を行っていくなかで分類率の最も低いクラスである C_w が追加法によって変化することがわかる. 特に仮追加法においてクラスH は多数のプロトタイプが追加されており, C_w になる回数が多く, クラスH の分類精度の向上を目指していることがわかる. 幾何マージン法においてクラスH, K の2 クラスが C_w になる回数も多く, 同数のプロトタイプが追加されている. 次にLetter データに対する, 両追加法と全クラス共通のプロトタイプ数を用いた従来のLGM-MCE 法の分類率の結果を表1 に示す. 表中の従来型LGM-MCE 法の値は, Validation Test において最も高い分類率を出した分類器によって得られた分類率である.

表4.1 LGM-MCE 法と両追加法における実験結果 (Letter データ) .

プロトタイプ数	Closed Test	Validation Test	Open Test
1	86.4%	76.08%	75.57%
2	93.3%	76.80%	77.06%
3	96.4%	78.17%	77.29%
4	97.9%	78.83%	78.87%
5	98.7%	78.31%	77.87%
6	98.9%	78.71%	79.11%
7	99.7%	79.22%	79.14%
8	99.5%	78.31%	78.37%
9	99.4%	78.35%	77.93%
10	99.9%	78.59%	78.85%
7.5(幾何マージン)	100%	82.12%	82.17%
10.73(仮追加)	100%	82.33%	82.03%

プロトタイプの追加法の検討であるが, 表4.1 より幾何マージン法, 仮追加法とも同程度の分類率を算出している. しかし幾何マージン法は仮追加法よりもモデルサイズが小さく, 小規模なモデルサイズにもかかわらず, 仮追加法と同程度の分類率を出している. この点で今回の実験条件下では幾何マージン法の方が優位であった. 次に, 幾何マージン法と仮追加法の両追加法と各クラスに一律にプロトタイプを設定していた従来法を比較してみると, 両追加法が従来法よりも高い分類率を算出しており, プロトタイプ数の自動的最適化手法の有効性を示した. 以上よりLetter データに関しては従来法よりもプロトタイプ数の自動的最適化手法の方が優れた分類精度を有しており, さらに2 種類の追加法に関してはモデルサイズの観点から幾何マージン

法が有効であることが確認できた。続いてAbalone データの結果について述べる。Letterデータの場合と同様に、メディアンフィルターを前後5フレームに設定しプロトタイプ数の最適化を行った際の各クラスのプロトタイプ数を両追加法それぞれに対して図4.3, 図4.4 に示す。

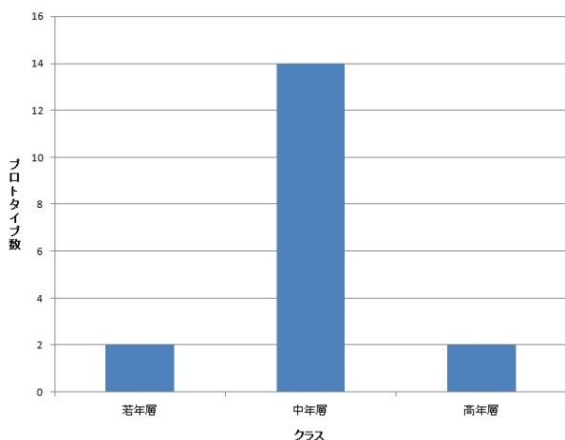


図4.3 各クラスのプロトタイプ数(Abaloneデータ,幾何マージン法)

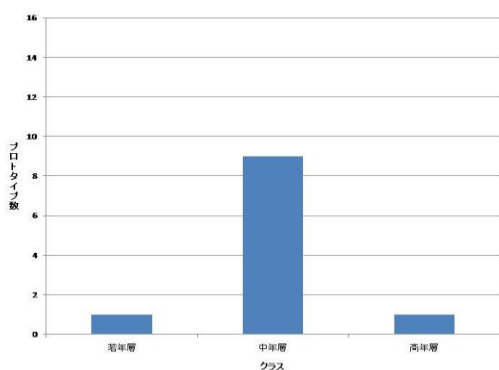


図4.4 各クラスのプロトタイプ数(Abaloneデータ,仮追加法)

このときの1クラス当たりのプロトタイプ数の平均はそれぞれ6個, 3.66個であった。図4.3, 図4.4より幾何マージン法において若年層, 高年層にそれぞれ1つずつプロトタイプを追加している以外は全て中年層のクラスにプロトタイプを追加している。さらなる調査から, C_w クラスは中年層で C_{mc} クラスは高年層になる回数が多かった。このことから, 中年層と高年層間の決定境界が適切に引けておらず, 決定境界付近には高年層クラスの領域に中年層クラスの学習標本が混入している複雑な分布のデータであることが示唆される。

Abalone データに対する提案手法と従来法の分類結果を表4.2に示す。表中の従来型LGM-MCE法の値は, Validation Testにおいて最も高い分類率を出した分類器によって得られた分類率である。

表4.2 LGM-MCE法と両追加法における実験結果 (Abaloneデータ)

プロトタイプ数	Closed Test	Validation Test	Open Test
1	67.75%	64.18%	63.60%
2	69.09%	64.50%	64.81%
3	69.76%	65.39%	65.01%

4	68.71%	65.64%	65.26%
5	72.25%	64.69%	64.37%
6	71.29%	65.39%	65.56%
7	73.40%	63.99%	64.50%
8	73.68%	64.05%	63.28%
9	76.08%	64.11%	63.54%
10	75.31%	64.05%	63.79%
6(幾何マージン)	68.04%	66.03%	65.01%
3.66(仮追加)	65.74%	64.30%	63.92%

プロトタイプ数の追加法の検討であるが, 表4.2より幾何マージン法の方が仮追加法よりも高い分類率を算出している。さらに仮追加法は学習標本の分類率がより改善するようにプロトタイプを追加する手法であるがClosed Testの分類率を見ても幾何マージン法に劣っている。今回の実験条件下ではAbaloneデータに関して幾何マージン法が仮追加法よりも有効な手法であることが確認できる。次に, 従来法と仮追加法を比較してみると, Validation Test, Open Testにおいて従来法とほぼ同程度の分類率を算出している。よって, 仮追加法は従来法と同等の分類精度を達成できていることがわかる。一方, 幾何マージン法と従来法を比較してみるとValidation Testにおいては従来法よりも高い分類率を算出しており, Open Testにおいては従来法と同程度の分類率を算出している。以上よりAbaloneデータに関しては, 従来型LGM-MCE法の最高分類率を上回ることができなかったが, ほぼ同等の分類率を達成できていることがわかる。

次にメディアンフィルターの前後フレーム数の違いについて検証するために5フレーム, 7フレームに設定し, そのときの幾何マージン法と仮追加法の分類率を示す。使用データはLetterデータである。Closed Testに関してはどちらの手法も100%を達成していたので省略し, Validation Test, Open Testを図4.5, 図4.6に示す。

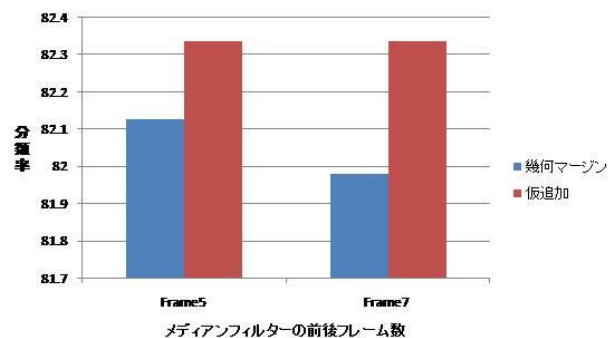


図4.5 前後フレーム数ごとのValidation Test

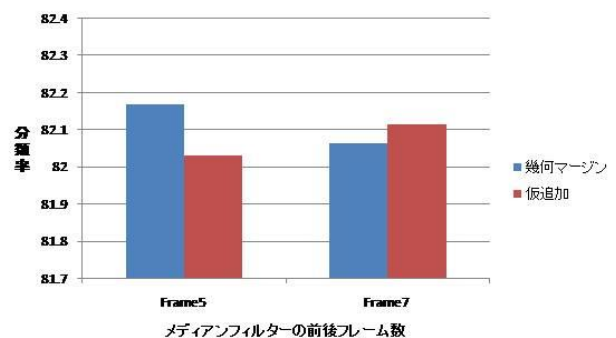


図4.6 前後フレーム数ごとのOpen Test

図4.5, 図4.6より前後フレーム数を増やすことで幾何マ

ジンはValidation Test, Open Testともに分類率が低下しており、仮追加法は各Testにおいて同等、または向上していることがわかる。メディアンフィルターの前後フレーム数を増やすことでValidationの分類率に対してより平滑化処理をかけており、値の大きな変動を排除する。幾何マージン法はValidation Testの分類率の推移に比較的大きな変動があり、前後フレーム数を増やすことによってメディアンフィルターをかけた後の分類率のピークが変化したことが示唆される。これを検証するために、幾何マージン法と仮追加法に関してプロトタイプの追加にともなうValidation Testの分類率の推移とその分類率にメディアンの前後フレーム数5フレーム、7フレームを適用したときの分類率の推移を図4.7、図4.8に示す。横軸のスケールが違う理由として幾何マージン法は3.3節で述べた終了条件を満たしたためプロトタイプの追加を終了したからである。

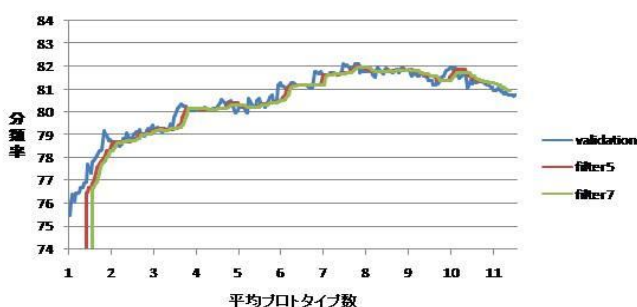


図4.7 Validation Testと各フレーム数の分類率の推移(幾何マージン)

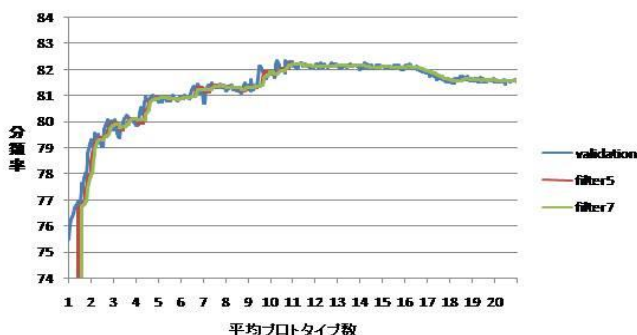


図4.8 Validation Testと各フレーム数の分類率の推移(仮追加)

図4.7、図4.8より幾何マージン法のValidation Testの分類率の変動具合は比較的大きく、フレーム数を増やすことによって分類率のピークが変化していることがわかる。仮追加法に関してはValidation Testの分類率の変動具合が小さく、前後フレーム数にかかわらず、安定した推移を示している。今回の実験条件ではメディアンフィルターをかけたときのピーク時の分類率と逐次に算出されるメディアンフィルターをかけたときの分類率との差が1を超えたら学習を終了してしまうため、幾何マージン法は平均プロトタイプ数が約11個までしか計測していない。Validation Testの分類率の低下具合をどれほど許容するかについて注意しながら、プロトタイプをさらに追加したときのValidation Testの分類率の変動具合を計測する必要がある。

今回の実験を通して、プロトタイプ数の自動的最適化手法は従来のプロトタイプを各クラス一律に設定していた従来法より有効であることが確認できた。プロトタイプ

追加に関して、メディアンフィルターの前後フレーム数を5に統一したときの分類率とクラスモデルサイズの実験結果から、今回提案した仮追加法よりも幾何マージン法がより優れた追加法であることが確認できた。しかし、幾何マージン法のValidation Testの分類率の推移は比較的大きな変動しており安定性について今後検証する必要がある。

5. まとめ

本稿では、学習途上におけるクラスごとの分類精度に基づいて、プロトタイプを逐次追加していき、クラスモデルサイズを最適化する手法の中でプロトタイプの追加法に観点を置いて実験を行った。実験を行うことで提案した学習標本の分類率による追加法は全クラス共通のプロトタイプ数を経験的に設定して得られた分類器での最も高い分類精度と同程度、またはそれ以上の分類性能を達成することができた。しかし、幾何マージンに基づいた追加法と比較するとデータによって同程度、またはそれ以下の分類性能を示した。Letter データで同程度の分類性能を示したことについて、両追加法は学習標本に対する分類率が100%に到達すると3節で述べたように幾何マージンに注目した追加法に切り替わる。追加行程の早期に学習標本に対する分類率が100%に到達すると追加行程の初期段階でしか異なった追加法を行っておらず、両追加法とも幾何マージン法を用いることになるため、結果的に同程度の分類性能を示したと考えられる。ただし、クラスごとのプロトタイプ数が両追加法で異なっていたので、最も分類率の低いクラスは変わっており、プロトタイプの追加法はプロトタイプの自動的最適化手法において考慮すべき重要な要因の1つであることがわかる。Abalone データでは、最も分類率の低いクラスに中年層が最も多く選ばれており、追加法によってあまり差が出なかった。ただし、最適なプロトタイプ数は両追加法で異なっており、幾何マージン法が仮追加法よりもモデルサイズが大きく、高い分類性能を示した。メディアンフィルターの前後フレーム数を増やしたとき、幾何マージン法の分類率は低下しておりValidation Testの分類率に比較的大きな変動があることが計測できた。

今後の展望として実験を行う前に設定しなければならない学習係数などのハイパーパラメータによって実験結果は大きく変わってくることを考慮すると、今回の実験で設定したハイパーパラメータの範囲を拡大して調査する必要がある。安定性を確保するための枠組みのメディアンフィルターの前後フレーム数についても調査を行い、フィルターをかけたあとのピーク時と追加途中の分類率の差に関して考察する必要がある。プロトタイプを逐次追加していく手法のさらなる改善を目指すためにプロトタイプの追加法や最適なプロトタイプ数の決定則について様々な手法を検討し、実験を行う。

[謝辞] 本研究の一部は、平成26年度科学研究費助成事業・基盤研究 (B) 「高識別的特徴空間とその探索法の最小分類誤り基準に基づく統一的実現」に支援して行われたものである。

5. 参考文献

[1] B.-H. Juang and S. Katagiri: "Discriminative Learning for Minimum Error Classification", IEEE Trans. Signal Processing, vol. SP-40, no.12, pp. 3043-3054, Dec., 1992.

[2] 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, vol. J94-D, no. 10, pp. 1664-1675, Oct., 2011.

[3] A. Sato and K. Yamada: "Generalized learning vector quantization", Advances in Neural Information Processing Systems, 8, pp. 423-429, MIT Press, 1996.

[4] Y.Linde, A.Buzo, and R.M.Gray: "An Algorithm for Vector Quantizer Design", IEEE Trans. on Communications, vol. 28, no. 1, pp.84-95, 1980.

[5] 神谷祐樹; 申富饒; 長谷川修: "自己増殖型ニューラルネットワークを用いたプロトタイプ生成による高速最近傍識別器の構成手法", 電子情報通信学会論文誌 D-II, vol. 90, no. 11, pp. 3000-3013, 2007.

[6] 高山雄史, 渡辺秀行, 片桐滋, 大崎美穂, 松田繁樹, 堀智織: "プロトタイプ数の自動的最適化を伴う大幾何マージン最小分類誤り学習法", 電子情報通信学会 パターン認識とメディア理解研究会, 信学技法PRMU2013-94, pp.19-24, 2014.