

ファジィ c -平均法の正則化を導入したマージン最大化モデルによるプロトタイプ分類器の学習

清水省吾 (大阪大学) ○楠木祥文 (大阪府立大学) 巽啓司 (大阪大学)

Learning Nearest Prototype Classifiers by Maximum-Margin Model with Regularizations of Fuzzy c -Means

S. Shimizu (Osaka University), *Y. Kusunoki (Osaka Prefecture University) and K. Tatsumi (Osaka University)

Abstract— We focus on the maximum-margin model (MMM), which is an optimization-based method for learning nearest prototype classifiers. The optimization problem corresponding to MMM is a difference of convex functions (DC) program. It is solved using the DC algorithm, which can be regarded as the alternating optimization algorithm of c -means. In this paper, we propose two revised MMM by introducing the quadratic regularization and the regularization based on the largest- L norm. As a result, the proposed methods are MMM enhanced with fuzziness. The DC algorithm applied to the proposed optimization problems are reduced a fuzzy- c -means-like alternating algorithm. Their performance is evaluated in numerical experiments.

Key Words: Nearest prototype classifiers, Fuzzy c -Means, Alternating optimization, Second-order programming

1 はじめに

プロトタイプ分類器とは最近プロトタイプ分類規則に基づいたクラスラベルの推定方法である。つまり、複数のラベル付きプロトタイプを与えた上で、与えられたデータ点に最も近いプロトタイプのラベルによって推定する。プロトタイプ分類器に対し様々な学習手法が提案されてきた。本研究では、MMM (Maximum-Margin Model) ¹⁾ という手法に注目する。MMM はマージン最大化問題に基づく学習手法である。このマージン最大化問題は、最終的に DC (Difference of Convex functions) 計画問題に帰着され、DC アルゴリズムにより求解される。このアルゴリズムは、各データ点の最近プロトタイプへの割当とその割当のもとでのプロトタイプの配置最適化を交互に行っており、 c -平均法の交互最適化であると見なせる。本研究では、MMM にファジィ c -平均法の正則化のアイデアを組み合わせることで、プロトタイプへの帰属にファジィ性を導入するとともに、より汎化誤差の小さい分類器の学習を目的とする。

2 c -平均法の正則化

本節では c -平均法と 2 種類のファジィ c -平均法について説明する。(ハード) c -平均法とファジィ c -平均法を、それぞれ、HCM (hard c -means) と FCM (fuzzy c -means) と表記する。

2.1 HCM

実ベクトルで与えられた m 個のデータ $x_1, \dots, x_m \in R^n$ を考える。これを c 個のクラスタに分類する。この手法は w, u を決定変数とする問題 (1) を求解していると考えることができる。

$$\begin{aligned} \min \quad & \sum_{i=1}^m \sum_{k=1}^c u_{ik} \|x_i - w_k\|^2 \\ \text{s.t.} \quad & \sum_{k=1}^c u_{ik} = 1, i = 1, \dots, m \\ & u_{ik} \in \{0, 1\}, i = 1, \dots, m, k = 1, \dots, c \end{aligned} \quad (1)$$

ここで、 $w_1, \dots, w_c \in R^n$ は各添え字に対応するクラスタのプロトタイプである。 u_{ik} は、クラスタ k に対するデータ点 x_i の帰属度を表している。データ点 x_i がプロトタイプ w_k のクラスタに帰属するとき、 $u_{ik} = 1$ 、帰属しないとき、 $u_{ik} = 0$ となる。問題 (1) を求解するために、 u に関する最適化と w に関する最適化を交互に行う。ここで、ベクトル $t \in R^c$ に対する関数 $h(t) = \max\{t_i \mid k = 1, \dots, c\}$ を考える。これは凸関数であるが、その共役関数を h^* とする。このとき、 $t_{ik} = \|x_i - w_k\|^2$ とおくと、HCM の最適化問題は $-\sum_{i=1}^m (\sum_{k=1}^c u_{ik} t_{ik} + h^*(u_{i1}, \dots, u_{ic}))$ の最小化となる。共役関数の関係を考えて、 $\sum_{i=1}^m -h(-t_{i1}, \dots, -t_{ic})$ の最小化問題に帰着する。さらに、共役関数とその勾配の関係を考えて、上述した交互最適化アルゴリズムは、凹関数 $\sum_{i=1}^m -h(-t_{i1}, \dots, -t_{ic})$ の線形近似と、近似された問題を w_1, \dots, w_c について最適化することを交互に行っていると解釈でき、DC アルゴリズムであると思なすことができる。

HCM では、あるデータ点は必ず 1 つのプロトタイプのクラスタにしか割り当てられない。しかし、データがはっきりと分離しないクラスタ構造を持つ場合、HCM では、プロトタイプがクラスタ構造をうまくとらえられない。この問題に対し、正則化を導入し、 u_{ik} を 0 から 1 までの区間値に拡張する 2 つの手法を紹介する。

2.2 2 次正則化に基づく FCM

2 次正則化に基づく FCM (以下、2 次正則化 FCM) は次の問題 (2) を求解していると考えることができる。

$$\begin{aligned} \min \quad & \sum_{i=1}^m \sum_{k=1}^c u_{ik} \|x_i - w_k\|^2 + \frac{\nu}{2} \sum_{i=1}^m \sum_{k=1}^c u_{ik}^2 \\ \text{s.t.} \quad & \sum_{k=1}^c u_{ik} = 1, i = 1, \dots, m \\ & u_{ik} \geq 0, i = 1, \dots, m, k = 1, \dots, c \end{aligned} \quad (2)$$

ここで、 ν はファジィ度を調整するパラメータであり、 ν を小さくすると、HCM に近づく。この手法では、HCM と同様に問題 (2) に対して、 u と w に関してそれぞれ交互に最適化を行い、求解する。各 $i = 1, \dots, m$ について、決定変数 u_{i1}, \dots, u_{ic} に対する最適解 $u_{i1}^*, \dots, u_{ic}^*$ は、 $\sum_{k=1}^c u_{ik}^* = 1$ 、および、各 $k = 1, \dots, c$ について以下の条件を満たす。

$$u_{ik}^* = \max\{0, \|x_i - w_k\|^2 - \lambda\} \quad (3)$$

ただし、 λ はこの方程式を満たす唯一の実数である。また、各 $k = 1, \dots, c$ に対して、 w_k に対する最適解 w_k^* は次のようになる。

$$w_k^* = \sum_{i=1}^m \frac{u_{ik}}{\sum_{j=1}^m u_{jk}} x_i \quad (4)$$

ベクトル $t \in R^c$ の凸関数 $h_\nu(t) = \sup_{u \in R^c} \{t^T u - \frac{\nu}{2} \|u\|^2 \mid \sum_{k=1}^c u_k = 1, u \geq 0\}$ を考え、その共役関数が問題 (2) にあらわれることに注意すると、HCM と同様に、これは $\sum_{i=1}^m -h_\nu(-t_i)$ の最小化問題に帰着する。また、共役関数とその勾配に関係を考えると、2 次正則化 FCM のアルゴリズムは、 $\sum_{i=1}^m -h_\nu(-t_i)$ の線形近似と、 w_1, \dots, w_c に関する近似された問題の最適化を交互に行っており、DC アルゴリズムと一致する。

2.3 最大 L 関数を用いた正則化に基づく FCM

最大 L 関数を用いた正則化に基づく FCM (以下、最大 L 関数 FCM) を説明する。この手法は 2 次正則化 FCM と同様に次の問題 (5) を求解していると考えられることができる。

$$\begin{aligned} \min \quad & \sum_{i=1}^m \sum_{k=1}^c u_{ik} \|x_i - w_k\|^2 \\ \text{s.t.} \quad & \sum_{k=1}^c u_{ik} = 1, i = 1, \dots, m \\ & 0 \leq u_{ik} \leq \frac{1}{L}, i = 1, \dots, m, k = 1, \dots, c \end{aligned} \quad (5)$$

ここで、 $L \in \{1, \dots, c\}$ はファジィ度を調整するパラメータであり、 L が小さいほど、 c -平均法に近づき、 $L = 1$ で c -平均法と一致する。 c -平均法と同様に、問題 (5) に対して、 u と w に関してそれぞれ交互に最適化を行い、求解する。各 $i = 1, \dots, m$ に対して、 u_{i1}, \dots, u_{ic} に関する最適解を求める。 $t_{ik} = \|x_i - w_k\|^2$ とし、 t_{i1}, \dots, t_{ic} を昇順に並び替えたものを、 $t_{ik(1)} \leq t_{ik(2)} \leq \dots \leq t_{ik(c)}$ としたとき、最適解 $u_{i1}^*, \dots, u_{ic}^*$ は、 $\sum_{k=1}^c u_{ik}^* = 1$ 、および、各 $k = 1, \dots, c$ について以下の条件を満たす。

$$\begin{cases} t_{ik} > t_{ik(L)} \implies u_{ik}^* = 0 \\ t_{ik} < t_{ik(L)} \implies u_{ik}^* = 1/L \\ t_{ik} = t_{ik(L)} \implies u_{ik}^* \in [0, 1/L] \end{cases} \quad (6)$$

w_1, \dots, w_c に関する最適解は式 (4) と同様に得られる。

ベクトル $t \in R^c$ の凸関数 $h_L(t) = \frac{1}{L} \max\{t_1 + \dots + t_L \mid l_1, \dots, l_L \in \{1, \dots, c\}\}$ を考えると、2 次正則化 FCM と同様に、問題 (5) は $\sum_{i=1}^m -h_L(-t_i)$ の最小化問題に帰着する。さらに、最大 L 関数 FCM のアルゴリズムは DC アルゴリズムと一致する。

3 FCM の正則化に基づくプロトタイプ分類器の学習

本章では、MMM の求解アルゴリズムを FCM の正則化と同様のアルゴリズムに拡張した学習手法を 2 つ提案する。MMM では、識別関数に \max 関数を用いる。2 つの提案法では 2 次正則化関数と最大 L 関数をそれぞれ、識別関数に用いる。これにより、求解アルゴリズムにそれぞれ、2 次正則化 FCM と最大 L 関数 FCM による帰属度のファジィ化が導入される。また、分類規則にファジィ性が導入され、クラス内のプロトタイプの分布を反映したクラス分類ができ、汎化性能の向上が期待できる。

3.1 記号の説明

入力空間を R^n とし、クラスラベル集合を $C = \{1, \dots, c\}$ とする。教師情報として、ラベル付き訓練データ $(x_1, y_1), \dots, (x_m, y_m)$ が与えられる。ここで、 $x_i \in R^n$, $y_i \in C, i = 1, \dots, m$ である。 $M = \{1, \dots, m\}$ を訓練データの添え字集合とする。また、 $k \in C$ に対して、集合 $C_{-k} = C \setminus \{k\}$ を定義する。各クラスにそれぞれ複数のプロトタイプが与えられるが、簡単のため、プロトタイプの数は一様に d であると仮定する。集合 $D = \{1, \dots, d\}$ を定義する。クラス k の l 番目のプロトタイプを $w_{kl} \in R^n$ とし、各 kl に対してバイアス項 $b_{kl} \in R$ を与える。

3.2 2 次正則化関数を導入したプロトタイプ分類器のためのマージン最大化モデル

2 次正則化関数を導入したマージン最大化モデル (以下、2 次正則化 MMM) を説明する。各クラス k に対して識別関数 $f_k : R^n \rightarrow R$ を h_ν を用いて、次のように定義する。

$$f_k(x) = -\sqrt{-h_\nu((- \|x - w_{kl}\|^2 - b_{kl}^2)_{l \in D})} \quad (7)$$

各データ点 x に対して、識別関数 $f_1(x), \dots, f_c(x)$ が最も大きいクラスに x を割り当てる。

プロトタイプ分類器をマージン最大化により学習する。各データ点 x_i に対して、そのクラスの評価値 $f_{y_i}(x_i)$ とその他のクラスの評価値 $\max_{k \neq y_i} f_k(x_i)$ との差に注目する。これらの差の最小値をマージンとし、このマージンを最大化するようにプロトタイプの配置を決定する。よって、マージン最大化は w, b, r を決定変数とする以下の最適化問題 (8) として定式化できる。

$$\begin{aligned} \min \quad & r \\ \text{s.t.} \quad & f_{y_i}(x_i) - f_k(x_i) \geq \frac{1}{\sqrt{r}}, i \in M, k \in C_{-y_i} \end{aligned} \quad (8)$$

ただし、 $r > 0$ とする。問題 (8) では、 h_ν の代わりに $h_{2\nu/r}$ を用いる。 $t_{ikl} = \frac{\tau}{2} (\|x_i - w_{kl}\|^2 + b_{kl}^2)$ とすると、 $f_k(x_i) = -\sqrt{-h_{2\nu/r}((-2t_{ikl}/r)_{l \in D})}$ となる。

問題 (8) の最適解として、プロトタイプ (w_{kl}, b_{kl}) を求めると、プロトタイプが 1 つに求まらない。そこで、 $f_{y_i}(x_i)$ の代わりに、 $\hat{f}_{y_i}(x_i) = -\frac{\sqrt{\tau}}{2\tau} (-f_{y_i}(x_i))^2 - \frac{\tau}{2\sqrt{\tau}}$ を用いる。ここで、 $\tau > 0$ とし、 τ は 2 乗関数の影響を調整するパラメータである。 $\hat{f}_{y_i}(x_i)$ は $f_{y_i}(x_i)$ を下から近似しており、 τ が小さいと、 $\hat{f}_{y_i}(x_i)$ は $f_{y_i}(x_i)$ よりも速く減少するため、クラス y_i のプロトタイプはその

クラスに属するデータが存在する領域に留まることが期待できる。

さらに、スラック変数 $\xi_{ik}, i \in M, k \in C_{-y_i}$ を導入すると、 w, b, r, ξ を決定変数とする次の最適化問題が得られる。

$$\begin{aligned}
\min \quad & r + \mu \sum_{i \in M} \sum_{k \in C_{-y_i}} \xi_{ik} \\
\text{s.t.} \quad & \frac{1}{\tau} h_\nu((-t_{iy_i l})_{l \in D}) + \sqrt{-2h_\nu((-t_{ikl})_{l \in D})} \\
& \geq 1 + \frac{\tau}{2} - \xi_{ik}, i \in M, k \in C_{-y_i} \\
& t_{ikl} = \frac{r}{2} (\|x_i - w_{kl}\|^2 + b_{kl}^2), \\
& i \in M, k \in C, l \in D \\
& \xi_{ik} \geq 0, i \in M, k \in C_{-y_i} \\
& r \geq 0
\end{aligned} \tag{9}$$

ここで、 $\mu > 0$ は損失に対する重みである。また、第1制約式に対して、両辺に \sqrt{r} をかけて、 $h_{2\nu/r}(-2t_{ikl}/r) = (2/r)h_\nu(-t_{ikl})$ の関係を用いた変形を適用している。

第1制約式の第1項を右辺に移項すると、その両辺は非負になり、2乗することができる。さらに、Lagrange 双対問題による h_ν の別表現

$$h_\nu(t) = \inf_{s_k} \{ \max_{k=1, \dots, c} \{ t_k - \nu s_k \} + \frac{\nu}{2} \sum_{k=1}^c s_k^2 \} \tag{10}$$

を用いると、次の制約式を得る。

$$\begin{aligned}
& 2 \min_{l \in D} \{ \nu s_{ikl} + t_{ikl} \} - \nu \sum_{l \in D} s_{ikl}^2 \\
& \geq \left(1 + \frac{\tau}{2} - \frac{\nu}{\tau} h_\nu((-t_{iy_i l})_{l \in D}) - \xi_{ik} \right)^2
\end{aligned} \tag{11}$$

第2制約式を $t_{ikl} = \frac{r}{2} \|x_i\|^2 - x_i^T w_{kl} + \zeta_{kl}$ と $2r\zeta_{kl} = \|w_{kl}\|^2 + b_{kl}^2$ に分解する。ここで、 rw_{kl} と rb_{kl} を w_{kl} と b_{kl} に置き直している。さらに、変数 b_{kl} は他の制約に現れないので、一般性を失うことなく2つ目の等式を $2r\zeta_{kl} \geq \|w_{kl}\|^2$ とできる。

以上の変形を適用すると次の最適化問題を得る。

$$\begin{aligned}
\min \quad & r + \mu \sum_{i \in M} \sum_{k \in C_{-y_i}} \xi_{ik} \\
\text{s.t.} \quad & 2(\nu s_{ikl} - \nu s_{ik0} + t_{ikl}) \\
& \geq \left(1 + \frac{\tau}{2} - \frac{1}{\tau} h_\nu((-t_{iy_i l})_{l \in D}) - \xi_{ik} \right)^2, \\
& i \in M, k \in C_{-y_i}, l \in D \\
& 2s_{ik0} \geq \sum_{l \in D} s_{ikl}^2, i \in M, k \in C_{-y_i} \\
& t_{ikl} = \frac{\|x_i\|^2}{2} r - x_i^T w_{kl} + \zeta_{kl}, \\
& i \in M, k \in C, l \in D \\
& 2r\zeta_{kl} \geq \|w_{kl}\|^2, k \in C, l \in D \\
& \xi_{ik} \geq 0, i \in M, k \in C_{-y_i} \\
& r \geq 0
\end{aligned} \tag{12}$$

この問題を求解するために、DC アルゴリズムを用いる。つまり、第1制約式の h_ν の線形近似と、その近似された問題の求解を交互に行う。 $h_\nu((-t_{iy_i l})_{l \in D})$ の勾配は、 x_i のクラス y_i の d 個のプロトタイプへの帰属度 $\tilde{u}_{iy_i l}, l \in D$ と解釈される。近似された問題は2次錐計画問題となり、容易に求解できる。問題(12)に対するDC アルゴリズムは訓練データ x_i ごとにそのクラス y_i のプロトタイプへの帰属度の計算と、その状況におけるプロトタイプの配置最適化を行っている。また、帰属度の決定方法に注目すると、このDC アルゴリズムは2次正則化FCMと同様なアルゴリズムであり、交互最適化であると見なすことができる。

ν が大きくなるにつれ、アルゴリズムの結果として、異なるプロトタイプに対する訓練データの帰属度の違いが小さくなり、プロトタイプが同じ場所に集まることが考えられる。このことより、2次正則化MMMでは、あらかじめ d を大きく設定しておき、 ν を調整することで実質のプロトタイプ数を調整できると考える。

3.3 最大 L 関数を導入したプロトタイプ分類器のためのマージン最大化モデル

最大 L 関数を導入したマージン最大化モデル (以下、最大 L 関数 MMM) を説明する。識別関数を h_L を用いて、次のように定義する。

$$f_k(x) = -\sqrt{-h_L((- \|x - w_{kl}\|^2 - b_{kl}^2)_{l \in D})} \tag{13}$$

各データ点 x に対して、識別関数 $f_1(x), \dots, f_c(x)$ が最も大きいクラスに x を割り当てる。

2次正則化MMMと同様の考えに基づき、マージン最大化の問題(9)を解き、プロトタイプを求める。ただし、 h_ν は h_L に置き換えられる。

さらに、Lagrange 双対問題による h_L の別表現

$$h_L(t) = \inf_{s_k \geq 0} \{ \max_{k=1, \dots, c} \{ t_k - s_k + \frac{1}{L} \sum_{k=1}^c s_k \} \} \tag{14}$$

を用い、2次正則化MMMと同様の変形を適用すると次の最適化問題を得る。

$$\begin{aligned}
\min \quad & r + \mu \sum_{i \in M} \sum_{k \in C_{-y_i}} \xi_{ik} \\
\text{s.t.} \quad & 2(s_{ikl} + t_{ikl}) - \frac{2}{L} \sum_{l \in D} s_{ikl} \\
& \geq \left(1 + \frac{\tau}{2} - \frac{1}{\tau} h_L((-t_{iy_i l})_{l \in D}) - \xi_{ik} \right)^2, \\
& i \in M, k \in C_{-y_i}, l \in D \\
& t_{ikl} = \frac{\|x_i\|^2}{2} r - x_i^T w_{kl} + \zeta_{kl}, \\
& i \in M, k \in C, l \in D \\
& 2r\zeta_{kl} \geq \|w_{kl}\|^2, k \in C, l \in D \\
& s_{ikl} \geq 0, i \in M, k \in C_{-y_i}, l \in D \\
& \xi_{ik} \geq 0, i \in M, k \in C_{-y_i} \\
& r \geq 0
\end{aligned} \tag{15}$$

2次正則化MMM同様に、この最適化問題はDC アルゴリズムによって求解される。このとき、

$h_L((-t_{ikl})_{l \in D})$ の勾配は、最大 L 関数によってファジィ化された、 x_i のクラス y_i に属するプロトタイプへの帰属度と解釈される。

L が大きくなるにつれ、各データ点について、異なるプロトタイプに対する帰属度の違いが小さくなり、プロトタイプが同じ場所に集まることが考えられる。このことから、最大 L 関数 MMM では、あらかじめ d を大きく設定しておき、 L を調整することで、実質のプロトタイプ数を調整できると考えられる。

4 数値実験

3章で述べた 2 次正則化 MMM と最大 L 関数 MMM の性能を検証するために数値実験を行った。まず、分類境界に関する実験を述べ、次に、汎化性能の比較実験を述べる。汎化性能の実験では、比較手法として、MMM、そして、古典的な手法である LVQ^{2, 3, 4} より LVQ1 と LVQ3 を用いた。

4.1 初期プロトタイプの設定

2つの実験において、いずれの手法も初期プロトタイプを生成する必要がある。また、初期プロトタイプによって最終的に得られるプロトタイプが異なるため、適切なプロトタイプを生成することは重要である。本実験では、初期プロトタイプ生成のため、 c -平均法 ++⁵) を用いる。 c -平均法 ++ をクラスごとに設定された反復数 (1000 とした) まで適用する。

4.2 分類境界の比較

まず、MMM, 2 次正則化 MMM, 最大 L 関数 MMM の分類境界の違いを比較するため、Fig. 1(a) に示す 2 次元人工データ (データ数 4000, 次元数 2, クラス数 2) を用いた実験を行った。このデータに対して、パラメータを $d = 10, \mu = 100, \tau = 1, \nu = 10, L = 3$ に設定し、MMM, 2 次正則化 MMM, 最大 L 関数 MMM をそれぞれ適用した結果を Fig. 1(b)–(d) に示す。図中において、データ点の色がクラスラベルを示しており、プロトタイプを Δ で表現している。

Fig. 1(b) より、MMM ではクラスごとに複数のプロトタイプを生成することで区分的線形な分類境界を形成できることが分かる。一方、Figs. 1(c), 1(d) より、2 次正則化 MMM と最大 L 関数では MMM と比較して滑らかな分類境界が得られた。また、誤分類したテストデータ数を比較すると、MMM では 58, 2 次正則化 MMM では 37, 最大 L 関数 MMM では 41 となった。これは 2 つのことが起因する。1 つは MMM の求解アルゴリズムがファジィ化することにより、配置がより柔軟に修正されたプロトタイプが獲得できることである。もう 1 つはクラス分類時に最近プロトタイプだけに基いてクラス分類せず、クラスごとにプロトタイプの配置を反映したクラス分類が可能になったことである。

4.3 汎化性能の比較

Table 1 に示す UCI Machine Learning Repository⁶⁾ のベンチマークデータを用いて、2 次正則化 MMM と最大 L 関数 MMM を検証した。ベンチマークデータは、各特徴量を平均が 0, 分散が 1 となるように正規化した。性能の評価指標として、10 分割交差検定により 5 回実験を行った分類誤差の平均を採用し、これを CV 誤差と呼ぶ。

Table 1: Benchmark data sets

データ名	データ数	特徴量数	クラス数
Iris	150	4	3
Wine	178	13	3
Balance	625	4	3
Sonar	208	60	2
Hayes	132	4	3
Car	1728	6	4

実験に使用したパラメータに言及する。MMM のパラメータには 1 クラスあたりのプロトタイプ数 d , 2 乗関数のパラメータ τ , 損失の重み μ があり、文献¹⁾ を参考に以下のように設定した。

- d : 1, 2, ..., 10
- τ : 0.01, 0.02, 0.1, 0.2, ..., 10, 20
- μ : 100

2 次正則化 MMM と最大 L 関数 MMM には MMM のパラメータに加えて、それぞれ ν と L があり、以下のように設定した。

- ν : 0.1, 0.2, 1, 2, 10
- L : 1, 2, ..., 10

LVQ1 と LVQ3 のパラメータは文献⁷⁾ を参考にした。LVQ1 のパラメータには d に加えて、学習率 α があり、0.1, 0.2, ..., 0.9 とした。LVQ3 のパラメータには d, α に加えて、Window Width と呼ばれるパラメータ s と学習率に関するパラメータ ϵ がある。 ϵ, s を共に 0.1, 0.2, ..., 0.9 に設定した。

ベンチマークデータごとに、各手法を適用したときの CV 誤差の最良値とそのときのパラメータを Table 2 に示す。 $\mu = 100$ と固定したため、Table 2 では μ を省略する。Table 2 より、2 次正則化 MMM と最大 L 関数 MMM が Iris を除くデータで 1 番目、2 番目に小さい CV 誤差を示した。また、Iris では LVQ3 に次いで、2 手法が 2 番目に小さい CV 誤差を示した。

MMM と LVQ (LVQ1 と LVQ3) を比較する。Iris を除くデータで LVQ より MMM が優れた CV 誤差を示した。Iris では、MMM が LVQ1 より 0.13% 小さい CV 誤差を示した一方、LVQ3 より 0.13% 大きい CV 誤差を示した。これらのことから MMM は LVQ より汎化性能が高いと言える。LVQ はクラスラベル情報を元に決まった操作を繰り返す古典的な方法でプロトタイプを獲得するのにに対し、MMM は求解可能な最適化問題として定式化し、その解としてプロトタイプを獲得する。プロトタイプを獲得するアプローチの差異が汎化性能に影響を与えたと考えられる。

MMM と 2 次正則化 MMM を比較する。2 次正則化 MMM が Iris, Wine, Balance では、2 次正則化 MMM が MMM と同等、あるいは 0.5% 程度小さい CV 誤差を示した。Sonar で 1.63%, Hayes で 3.03%, Car で 7.91% 小さい CV 誤差を 2 次正則化 MMM が示した。MMM と最大 L 関数 MMM を比較する。Iris, Wine, Balance では、最大 L 関数 MMM が MMM と同等、あるいは 1% 程度小さい CV 誤差を示した。Sonar で 1.73%, Hayes

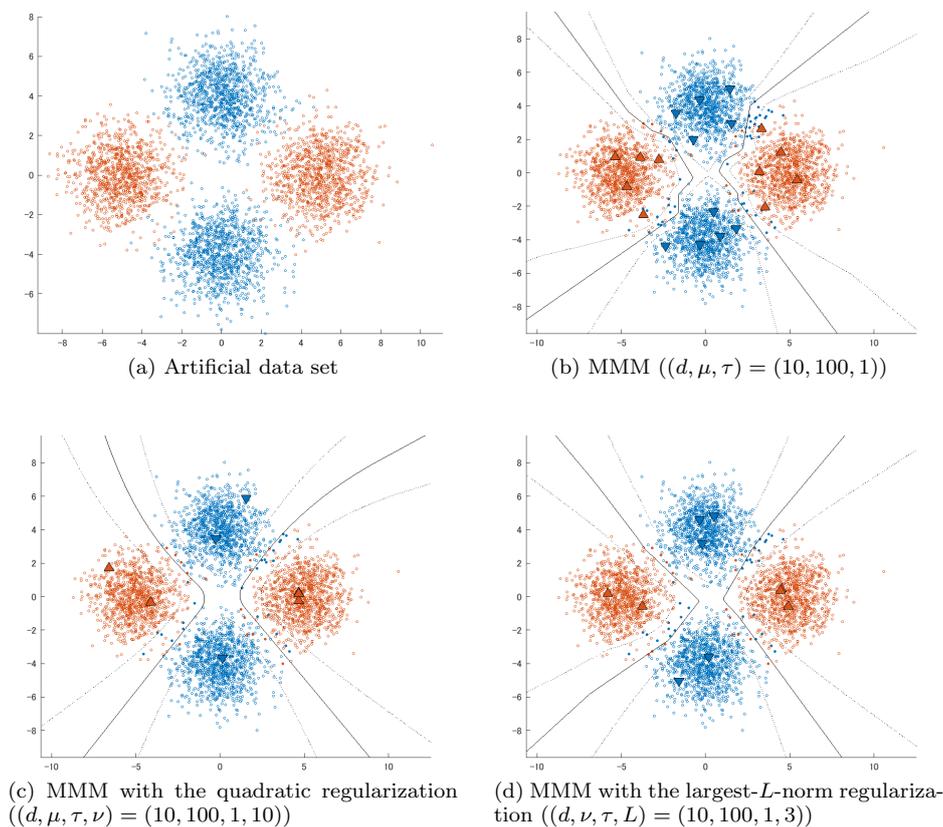


Fig. 1: Comparison of classification boundaries

Table 2: Comparison of MMM, MMM with the quadratic regularization, MMM with the largest- L -norm regularization, LVQ1 and LVQ3 by six benchmark data sets. We show the results of the best classifiers (with respect to the classification performance) among various values of hyperparameters.

データ名	2次正則化 MMM (d, τ, ν)	最大 L 関数 MMM (d, τ, L)	MMM (d, τ)	LVQ1 (d, α)	LVQ3 (d, α, s, ϵ)
Iris	2.80 ± 0.87 (1,20,0.2)	2.80 ± 0.87 (1, $\tau = 20$, 1)	2.80 ± 0.87 (1,20)	2.93 ± 0.37 (10,0.8)	2.67 ± 0.47 (4,0.1,0.3,0.3)
Wine	1.35 ± 0.50 (3,10,10)	1.57 ± 0.47 (10,2,7)	1.91 ± 0.31 (2,2)	2.47 ± 0.50 (1,0.3)	2.47 ± 0.64 (1,0.1,0.1,0.3)
Balance	6.82 ± 0.65 (2,20,10)	6.37 ± 0.41 (9,20,3)	7.33 ± 0.50 (2,20)	11.33 ± 0.98 (2,0.4)	14.34 ± 0.70 (2,0.3,0.1,0.3)
Sonar	10.00 ± 1.04 (9,10,10)	9.90 ± 1.75 (9,20,3)	11.63 ± 1.29 (8,10)	16.73 ± 1.78 (7,0.1)	13.85 ± 1.57 (10,0.2,0.1,0.6)
Hayes	24.24 ± 1.78 (3,10,10)	17.88 ± 0.86 (10,20,6)	27.27 ± 2.07 (4,10)	29.85 ± 1.82 (6,0.3)	27.73 ± 2.55 (10,0.4,0.6,0.6)
Car	1.41 ± 0.18 (5,20,10)	3.38 ± 0.21 (10,20,3)	9.32 ± 0.23 (6,20)	18.63 ± 0.55 (2,0.1)	19.66 ± 0.65 (2,0.1,0.6,0.3)

で 9.39%, Car で 5.94% 小さい CV 誤差を最大 L 関数が示した。これらのことから 2 次正則化 MMM と最大 L 関数 MMM は MMM の安定した汎化性能を引き継ぎつつ、CV 誤差を低減できたと考えられる。これは第 4.2 節で述べたように、2 次正則化 MMM と最大 L 関数 MMM では、求解アルゴリズムがファジィ c -平均法の正則化に拡張されたこと、分類規則にファジィ性が導入されたことに起因する。

5 おわりに

本研究では、プロトタイプ分類器の学習手法の 1 つである MMM に注目した。MMM にファジィ c -平均法の正則化のアイデアを導入することで、ファジィプロ

トタイプ分類器に拡張した学習手法を提案した。2次元人工データを用いた実験では、MMM よりも提案法の方が滑らかな分類境界を得られており、プロトタイプの学習と分類規則にファジィ性が導入されていることが確認できた。また、ベンチマークデータを用いた実験では、従来のプロトタイプ学習手法に加えて、MMM よりも提案法の方が汎化性能が高いことを確認した。

参考文献

- 1) Y. Kusunoki, C. Wakou and K. Tatsumi: Maximum-Margin Model for Nearest Prototype Classifiers, Journal of Advanced Computational Intelligence and Intelligent Informatics, 22(4),

565/577 (2018)

- 2) J. C. Bezdek and L. I. Kuncheva, Nearest Prototype Classifier Designs: An Experimental Study, International journal of intelligent systems, 16, 1445/1473 (2001)
- 3) J. C. Bezdek, T. R. Reichherzer, G. S. Lim, and Y. Attikiouzel, Multiple-Prototype Classifier Design, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 28(1), 67/79 (1998)
- 4) C-L. Liu and M. Nagasawa, Evaluation of prototype learning algorithms for nearest-neighbor classifier in application to handwritten character recognition, Pattern Recognition, 34, 601/615

(2001)

- 5) D. Arthur and S. Vassilvitskii, k -means++: The Advantages of Careful Seeding, Technical Report (2006)
- 6) D. Dheeru and G. Casey, UCI Machine Learning Repository <http://archive.ics.uci.edu/ml/>
- 7) I. Triguero, J. Derrac, S. Garcia, and F. Herrera, A Taxonomy and Experimental Study on Prototype Generation for Nearest Neighbor Classification, IEEE transactions on systems, man, and cybernetics part C: applications and reviews, 42, 86/100 (2012)