

電気通信大学大学院 情報理工学研究科 情報・通信工学専攻
電子情報システムコース

線形計画MAP識別器へのマルチカーネル学習適用法に
関する研究

学籍番号 1431052

佐藤 光

指導教員 鷺沢嘉一 助教

2016年1月31日

目次

第 1 章 序論	1
1.1 研究の背景と位置づけ	1
1.2 本論文の構成	2
1.3 本論文で用いる記号表現	2
第 2 章 パターン認識の基本概念と既存手法	3
2.1 パターン認識系の構成	3
2.2 機械学習	4
2.3 パラメトリック法	4
2.3.1 最尤推定法	5
2.3.2 ベイズ推定法	6
2.3.3 最大事後確率推定法	8
2.4 ノンパラメトリック法	9
2.4.1 カーネル密度推定法	9
2.5 カーネル法	10
2.5.1 Support Vector Machine	11
2.5.2 マルチカーネル学習	13
2.6 線形計画 MAP 識別法	14
第 3 章 提案手法	18
3.1 確率密度推定を用いた拡張手法	18
3.1.1 カーネル密度関数を用いた拡張手法	19
3.1.2 ガウス関数を用いた拡張手法	22
3.1.3 ガウス差分関数を用いた拡張手法	24
3.1.4 制約追加法	26
3.2 マルチカーネル学習	28
3.2.1 カーネル密度関数を用いた拡張手法への適用法	28
3.2.2 ガウス関数を用いた拡張手法への適用法	30
3.2.3 ガウス差分関数を用いた拡張手法への適用法	31
3.2.4 制約追加法	32
第 4 章 計算機実験	35
4.1 オープンデータベースを用いた 2 クラス識別問題	35
4.1.1 実験環境	35
4.1.2 実験結果	35
4.1.3 考察	36

第 5 章	まとめ	46
5.1	総括	46
5.2	今後の課題	46
	謝辞	48

目次

2.1	パターン認識系の構成	3
2.2	識別関数法による識別	4
2.3	尤度方程式は最尤推定量の十分条件ではない例	6
2.4	ベイズ推定法と最尤推定法の直感的違い	8
2.5	2次元空間におけるハードマージン SVM	13
2.6	2次元空間におけるソフトマージン SVM	14
2.7	事後確率による2クラス分類	15
2.8	MAPLP で求める代理関数と真の事後確率との関係	16
3.1	ガウス差分関数とガウス関数の波形 (平均 0, 分散 1)	25
4.1	訓練標本数の違いによる識別率の変化 (制約追加法を適用した確率密度推定による拡張手法)	43
4.2	訓練標本数の違いによる識別率の変化 (制約追加法を適用した確率密度推定による拡張手法)	44
4.3	訓練標本数の違いによる識別率の変化 (制約追加法を適用した確率密度推定による拡張手法)	45

表 目 次

4.1	識別問題の属性	36
4.2	実験 1：確率密度推定を適用した拡張手法の比較実験における最適パラメータ	38
4.3	実験 1：確率密度推定を適用した拡張手法の平均識別率 [%] と標準偏差	39
4.4	実験 1：平均識別率に対する Nemenyi 検定 (有意水準 $\alpha = 0.05$) の結果	39
4.5	実験 2：マルチカーネル学習を適用した拡張手法における最適パラメータ	40
4.6	実験 2：マルチカーネル学習を適用した拡張手法の平均識別率 [%] と標準偏差	40
4.7	実験 2：平均識別率に対する Nemenyi 検定 (有意水準 $\alpha = 0.05$) の結果	41
4.8	確率密度推定適用手法 (実験 1) とマルチカーネル学習適用手法 (実験 2) の識別率の差	41
4.9	初期値の違いによるマルチカーネル学習適用手法 (制約追加法適用無し) の平均識別率 [%] と標準偏差の比較	42

第1章 序論

1.1 研究の背景と位置づけ

現在、私たちの生活は多様な科学技術によって支えられている。その中でも、計算機の基盤技術の一つであり、20世紀最大の発明とも称されている集積回路は、トランジスタ数がムーアの法則に従うように指数関数的に増加することに伴い、その性能を向上させてきた。これらの進化に伴い、インターネット等のネットワーク技術、データベースなどのストレージ技術も急速に進歩している。めまぐるしい計算機技術の発達は、情報処理に関わる様々な分野を開拓することに貢献し、産業や文化、社会の構造的な変革をもたらすまでに至った。しかしながら、私たちが夢に描くような「人間のように外界を知覚し、自発的に働きかけ、思考する」といった認識・学習機能を有するコンピュータは未だ存在しない。

古来より、人類は高度な情報処理機能を備えている脳に関心を抱いてきた。脳機能の中でも、認知・認識の機能は最も基本的で重要な役割を果たしている。認知機能の工学的な研究は、認知科学や人工知能として広く知られている。人工知能 (AI; Artificial Intelligence) とは、1956年に開催されたダートマス会議において John McCarthy より提唱された用語であるが、明確な定義がある訳ではない。今日では「知能の記述を主体とする情報処理や研究でのアプローチ」という意味あいでも使われることが多く、人工知能に関わる技術は、パターン認識、機械学習、自然言語処理、コンピュータゲーム、人工生命など多岐にわたる。パターン認識とは、与えられた標本からパターン (規則性) を抽出し、予め任意に定められた概念に対応づける機能のことである [1]。1943年に、McCulloch と Pitts らによって提案されたパーセプトロンはパターン認識器のプロトタイプであり、ヒトの脳神経細胞の機能をモデル化したものである。パーセプトロンを発端とし、部分空間法や誤差逆伝播法による多層ニューラルネットなど様々な識別手法が現在に至るまで数多く提案された。その中でも、Vapnik らにより提案された Support vector machine (SVM) は、カーネル法の適用と高度な計算機能力の恩恵により高い汎化能力が示され、脚光を浴びることとなった。

線形計画 MAP 識別器 (Maximum A Posteriori based kernel classifier trained by Linear Programming; MAPLP) は Nopriadi と Yamashita によって 2010 年に提案された、最大事後確率 (Maximum A Posteriori; MAP) 推定に基づく識別手法である。MAPLP では事後確率を直接モデル化・計算する代わりに、ベイズ決定則を満たす代理関数を定義することで、識別器を線形計画法により容易に学習できる。また、MAPLP の定式化は SVM におけるヒンジ損失基準と類似しながらも、SVM で用いられる誤識別最小化とマージン最大化のトレードオフを調整するマージンパラメータを設定する必要がないといった長所が存在する [2]。しかしながら、MAPLP では入力データに関する条件付き密度を考慮せず、標本点のみに適合する学習を行っているため、学習標本数が少ない場合に過学習し、未知データに対する汎化能力が低下する恐れがある。さらに、カーネル法の拡張手法として知られるマルチカーネル学習 (Multiple Kernel Learning; MKL) を直接適用することができないという欠点も存在する。

本研究では、線形計画 MAP 識別器に対して、各クラスにおける入力データの条件付き確率密度関数の推定による拡張手法とマルチカーネル学習適用による拡張手法の提案する。入力データの条

件付き確率密度関数を予めモデル化することで、学習標本数が少ない場合でも過学習を緩和できると考えられる。さらに、マルチカーネル学習を適用することにより、カーネル関数を複数の重み付きカーネルの線形和により表現をすることが可能になる。これにより、単カーネルを用いた場合の最適パラメータ推定が不要になり、汎化能力の高い識別器の学習が期待できる。2クラス識別実験では、入力データの条件付き確率密度のカーネル密度推定による拡張手法が既存手法より高い精度を示した。マルチカーネル学習適用による拡張手法では、初期値を一定値とした場合、多くの識別問題で識別性能が低下した。一方で、初期値を単カーネルにおける最適値を用いて調整した場合は、従来手法と比較して同程度以上の識別性能をもつことを確認した。この結果から、マルチカーネル学習適用法の識別性能はカーネル関数の重みパラメータの初期値に大きく依存している、ということが分かった。

本論文では、線形計画 MAP 識別器に対して「各クラスにおける入力データの条件付き確率密度関数の推定による拡張手法」と「マルチカーネル学習適用による MAPLP 拡張手法」を定義し、最適化アルゴリズムを示す。また、計算機実験により性能を評価し、既存の手法との比較・検討を行う。

1.2 本論文の構成

本論文は、全5章で構成されている。

第1章では、本研究の背景と位置づけ、構成を述べる。

第2章では、パターン認識に関する基礎事項と関連研究、既存手法について述べる。

第3章では、確率密度推定による拡張手法、マルチカーネル学習による拡張手法を定義する。

第4章では、計算機実験による既存手法と提案手法の性能評価を提示し、考察を述べる。

第5章では、本研究における成果をまとめ、今後の課題を提示する。

1.3 本論文で用いる記号表現

本論文では、数字とスカラ定数はローマン体、スカラ変数はイタリック体、ベクトルと行列はボードイタリック体で表す。

第2章 パターン認識の基本概念と既存手法

2.1 パターン認識系の構成

パターン認識とは、与えられた標本からパターン（規則性）を抽出し、予め任意に定められた概念に対応づける機能のことである [1]。この「概念」をクラス (class) もしくはカテゴリ (category) と呼ぶ。例えば、平仮名の文字認識であれば、入力パターンを 50 個のクラスいずれかに対応される処理となる。また、音声のような時系列信号を処理して五十音や単語、個人の声紋と対応させる機能をもつ音声認識もパターン認識の一つである。さらに、心電図や脳波などを利用して身体の異常の有無を判定する処理や BCI (Brain Computer Interface) の研究においても、パターン認識は重要な役割を担っている [3, 4]。このようにパターン認識で扱う対象は幅広い。

機械を用いたパターン認識系の構成を図 2.1 に示す [5, 6, 7]。観測とは、画像、音、その他の物理的入力をコンピュータが扱える離散信号に変換する処理のことである。例えば、画像や文書のスキャン、音声の録音、動画の録画、A/D 変換などが挙げられる。前処理では、ノイズ除去、正規化などの処理を行う。特徴抽出は、元パターンから識別に必要な本質的な特徴のみを抽出することを目的として行う。特徴抽出は認識系の中でも極めて重要な処理であるが、有効な特徴を抽出するための系統的な手段は未だ存在しない。そのため、抽出方法は対称となる問題や分野に依存し、分野の知識が必要となることが多い。例えば、画像認識では、線抽出、領域抽出、エッジ検出、テクスチャ抽出などがこれにあたる。識別では、抽出された特徴ベクトルと予め用意された識別辞書とを照合し、入力パターンのクラスを決定する。本研究では、この識別器を主として扱う。

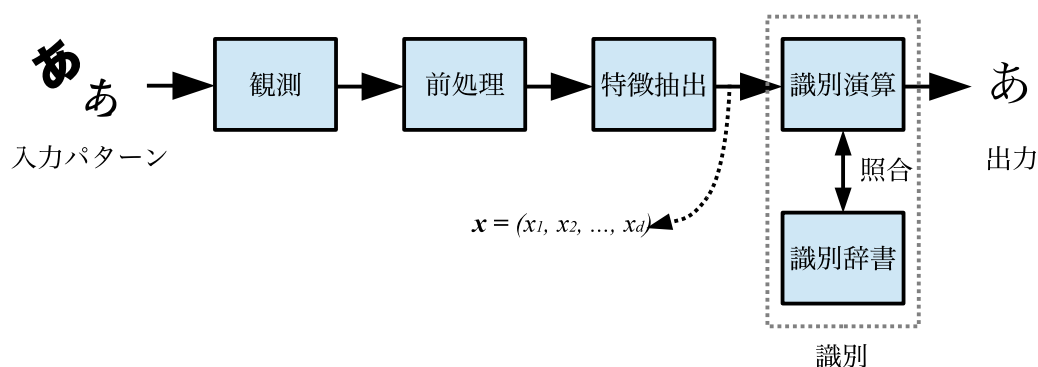


図 2.1: パターン認識系の構成

パターン認識では、一般的に各クラスに対し識別関数 $g_i(\mathbf{x})$ ($i = 1, \dots, c$) を用意し、 $g_i(\mathbf{x})$ の値の大小関係によってパターン \mathbf{x} の属するクラスを判別する。この方法は識別関数法と呼ばれている。ここで c はクラス数を表す。図 2.2 は、識別関数法によるクラス識別のブロック図を表したも

のである。クラス i に属する要素の集合を Ω_i とすると、識別法は

$$g_i : \mathbb{R}^N \rightarrow \mathbb{R} \quad (2.1)$$

$$k = \underset{i \in \{1, \dots, c\}}{\operatorname{argmax}} g_i(\mathbf{x}) \Rightarrow \mathbf{x} \in \Omega_k \quad (2.2)$$

で表される。識別関数 $g_i(\mathbf{x})$ としては様々な形態が考えられるが、パターン \mathbf{x} に対して重み係数 \mathbf{w}_i とバイアス項 b_i を用いて次式のように表現できる場合を線形識別関数と呼ぶ。

$$g_i(\mathbf{x}) = \langle \mathbf{w}_i, \mathbf{x} \rangle + b_i \quad (i = 1, \dots, c) \quad (2.3)$$

ここで $\langle \cdot, \cdot \rangle$ は内積を表す。線形識別関数以外にも、識別関数が \mathbf{x} に関して2次形式で表される2次識別関数や、非線形関数として表現されるものも存在する。

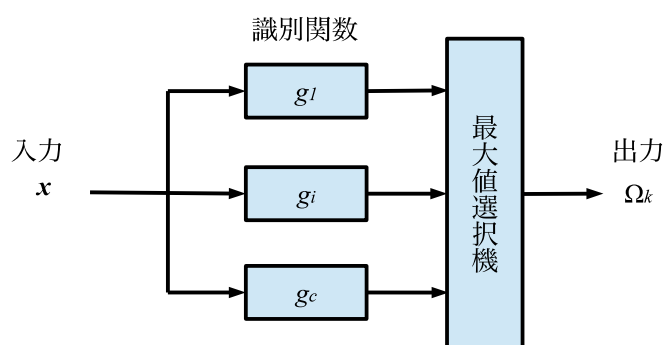


図 2.2: 識別関数法による識別

2.2 機械学習

機械学習とは、有限の標本から抽出されたパターン（規則性）を用いてパラメータを推定し、新たな未知パターンに対しても真の出力を得られるようにする処理のことである。機械学習は大きく分けて、教師あり学習、教師なし学習、強化学習の3種類がある。教師あり学習は、入力パターンに対して、出力もしくは所属クラスを表すラベルを利用する学習である。教師なし学習は、標本に対する出力やクラスラベルは無く、入力パターンの自然な組分けを行う。ここで自然な組分けとは、システム毎に定義されるものである [7]。強化学習とは、学習結果に対して報酬をフィードバックしながら学習を行うものである。

もう一つの学習の分類法として、パラメトリックな学習とノンパラメトリックな学習がある。パラメトリックな学習とは標本の確率密度関数を仮定あるいは既知とし、分布パラメータの推定を行うものである。識別は、推定された確率密度関数を用いたベイズ決定則を実現することにより行われる。ノンパラメトリックな学習とは、確率密度関数を想定することなく、観測標本より直接識別関数の設計を行う方法であり、パーセプトロンやニューラルネットワークで用いられている。

2.3 パラメトリック法

有限個のパラメータを用いて定義された確率密度関数の集合をパラメトリックモデルと呼ぶ。本節では、確率密度関数のパラメータ推定を題材とする。最尤推定法、ベイズ推定法、最大事後確率

推定法について定義を示し、パターン認識への適用法を示す。

2.3.1 最尤推定法

最尤推定法 (maximum likelihood estimation) は 20 世紀初頭に Fisher によって考案された手法であり、訓練標本が最も生起しやすいようにパラメータを決定する。

訓練標本を $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ とする。ここで、 $\mathbf{x}_i \in \mathbb{R}^d$ は d 次元からなる標本とし、確率密度関数 $p(\mathbf{x}; \boldsymbol{\theta})$ に従い独立に観測されたものとする。 $\boldsymbol{\theta}$ はパラメータであり、 N は標本数である。セミコロンの (;) は、その前後で確率変数とパラメータを区別するために用いた。訓練標本 \mathcal{X} がモデル $p(\mathbf{x}; \boldsymbol{\theta})$ から生起する確率は、 \mathcal{X} が独立同一分布に従うという仮定より

$$L(\boldsymbol{\theta}) = \prod_{i=1}^N p(\mathbf{x}_i; \boldsymbol{\theta}) \quad (2.4)$$

で与えられる。ここで、式 (2.4) を $\boldsymbol{\theta}$ の関数と見なすとき、 $L(\boldsymbol{\theta})$ は尤度と呼ばれる。最尤推定法では、この尤度を最大化するようにパラメータ $\boldsymbol{\theta}$ の値を決定する。このようにして求まる値を最尤推定量と呼び、

$$\hat{\boldsymbol{\theta}}_{ML} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} L(\boldsymbol{\theta}) \quad (2.5)$$

で表す。最尤推定法によって得られたパラメータ $\hat{\boldsymbol{\theta}}_{ML}$ を用いて、確率密度関数は次式で推定される。

$$\hat{p}_{ML}(\mathbf{x}) = p(\mathbf{x}; \hat{\boldsymbol{\theta}}_{ML}) \quad (2.6)$$

確率密度関数 $P(\mathbf{x}; \boldsymbol{\theta})$ が $\boldsymbol{\theta}$ に関して微分可能である時、最尤推定量 $\hat{\boldsymbol{\theta}}_{ML}$ は

$$\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{ML}} = \mathbf{0}_t \quad (2.7)$$

を満たす。 $\mathbf{0}_t$ は t 次元の零ベクトルであり、 $\nabla_{\boldsymbol{\theta}}$ は $\boldsymbol{\theta}$ に関する勾配を表す。式 (2.7) は尤度方程式と呼ばれ、最尤推定解の必要条件となる。すなわち、最尤推定解は尤度方程式を満たす。しかし、尤度方程式は一般に最尤推定解の十分条件ではないことに注意する必要がある。図 2.3 に示すように、局所解が求まり最尤推定解が求まらない場合もある [8]。 $p(\mathbf{x}|\boldsymbol{\theta})$ が 1 より極めて小さい値をとるとき、式 (2.4) は非常に小さい値となり数値計算が困難になる [9]。そこで、一般的に式 (2.5) の左辺の対数をとった対数尤度

$$\hat{\boldsymbol{\theta}}_{ML} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log L(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left[\sum_{i=1}^N \log p(\mathbf{x}_i; \boldsymbol{\theta}) \right] \quad (2.8)$$

を最大化する手段を用いる場合が多い。対数尤度に対する尤度方程式は

$$\nabla_{\boldsymbol{\theta}} \log L(\boldsymbol{\theta}) \Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{ML}} = \mathbf{0}_t \quad (2.9)$$

で与えられる。

ガウスモデル ガウスモデルに対する最尤推定の例を以下に示す。ガウスモデルとは多変量ガウス分布をパラメトリックモデルとして用いたものである。 d 次元の入力パターン $\mathbf{x} \in \mathbb{R}^d$ に対するガウスモデルは

$$p(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2}} \exp \left(-\frac{1}{2} \langle \mathbf{x} - \boldsymbol{\mu}, \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \rangle \right) \quad (2.10)$$

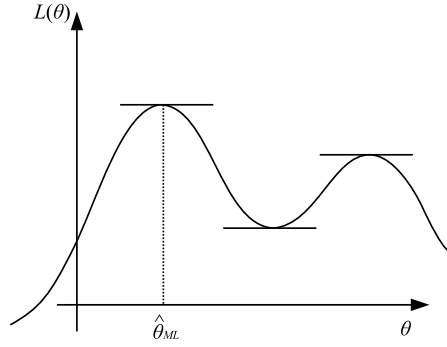


図 2.3: 尤度方程式は最尤推定量の十分条件ではない例

で表現される。ここで、 $\boldsymbol{\mu} \in \mathbb{R}^d$, $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ はガウスパラメータである。また、 $\boldsymbol{\Sigma}$ は正定値行列である必要がある [8]。すなわち、

$$\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^T \quad (2.11)$$

であり、任意の非ゼロベクトル \mathbf{a} に対して

$$\langle \mathbf{a}, \boldsymbol{\Sigma} \mathbf{a} \rangle > 0 \quad (2.12)$$

を満たす必要がある。

ガウスモデルでは、パラメータ $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ がそれぞれ分布の期待値と分散共分散行列に対応している。つまり

$$\boldsymbol{\mu} = \mathbb{E}_{\mathbf{x}}[\mathbf{x}] = \int \mathbf{x} p(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) d\mathbf{x} \quad (2.13)$$

$$\boldsymbol{\Sigma} = \mathbb{V}_{\mathbf{x}}[\mathbf{x}] = \int (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T p(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) d\mathbf{x} \quad (2.14)$$

が成り立つ。ここで、式 (2.9) よりガウスモデルの対数尤度方程式は

$$\begin{cases} \left. \frac{\partial}{\partial \boldsymbol{\mu}} \log L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \right|_{\boldsymbol{\mu}=\hat{\boldsymbol{\mu}}_{ML}} = \mathbf{0}_d \\ \left. \frac{\partial}{\partial \boldsymbol{\Sigma}} \log L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \right|_{\boldsymbol{\Sigma}=\hat{\boldsymbol{\Sigma}}_{ML}} = \mathbf{0}_{d \times d} \end{cases} \quad (2.15)$$

で与えられる。この誘導方程式を解くと、パラメータ $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ の最尤推定量 $\hat{\boldsymbol{\mu}}_{ML}$, $\hat{\boldsymbol{\Sigma}}_{ML}$ は

$$\begin{cases} \hat{\boldsymbol{\mu}}_{ML} = \frac{1}{n} \sum_{i=1}^N \mathbf{x}_i \\ \hat{\boldsymbol{\Sigma}}_{ML} = \frac{1}{n} \sum_{i=1}^N (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_{ML})(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_{ML})^T \end{cases} \quad (2.16)$$

と計算できる。

2.3.2 ベイズ推定法

2.3.1 節で求めた最尤推定量は、一致性と漸近不偏性をもつ推定量であり、訓練標本数が無限に多い場合にその妥当性が保証される。更に、最尤推定量は漸近有効性ももつため、訓練標本数が十分に大きくとれるとき、分散が小さく信頼性の高い推定量であることが知られている [8]。しかし、

実際の問題では、十分に訓練標本数が観測できるとは限らず、最尤推定量が良い推定量であるという保証はない。

ベイズ推定とは、モデルの未知パラメータ θ を確率的な変数とみなして、確率密度関数を推定する手法である。ここで、 θ に関する確率を以下に定義する。

$$\begin{aligned} \text{事前確率} & : p(\theta) \\ \text{尤度} & : p(\mathcal{X}|\theta) \\ \text{事後確率} & : p(\theta|\mathcal{X}) \\ \text{同時確率} & : p(\mathcal{X}, \theta) \end{aligned} \tag{2.17}$$

事前確率 $p(\theta)$ は標本 \mathcal{X} を得る前の θ の生起確率密度関数であり、事後確率 $p(\theta|\mathcal{X})$ は訓練標本 \mathcal{X} を得た後の θ の生起確率密度関数である。 $p(\mathcal{X}|\theta)$ は θ が得られた後の \mathcal{X} の生起確率密度関数であるが、これは θ の関数と見なして尤度とも呼ばれ

$$p(\mathcal{X}|\theta) = \prod_{i=1}^N p(\mathbf{x}_i|\theta) \tag{2.18}$$

で与えられる。尤度と事前確率を用いれば、同時確率は

$$p(\mathcal{X}, \theta) = p(\mathcal{X}|\theta)p(\theta) \tag{2.19}$$

と表すことができる。

ベイズ推定とは、モデル $p(\mathbf{x}|\theta)$ を事後確率 $p(\theta|\mathcal{X})$ に関して周辺化することにより、 $p(\mathbf{x})$ を推定する手法である。

$$\hat{p}_{Bayes}(\mathbf{x}) = \int p(\mathbf{x}|\theta)p(\theta|\mathcal{X})d\theta \tag{2.20}$$

これは事後予測分布と呼ばれる。ここで、ベイズの定理を用いれば、事後確率 $p(\theta|\mathcal{X})$ は事前確率 $p(\theta)$ と尤度 $p(\mathcal{X}|\theta)$ を用いて

$$p(\theta|\mathcal{X}) = \frac{p(\mathcal{X}|\theta)p(\theta)}{p(\mathcal{X})} = \frac{p(\mathcal{X}|\theta)p(\theta)}{\int p(\mathcal{X}|\theta')p(\theta')d\theta'} \tag{2.21}$$

と表せる。したがって、式 (2.18), (2.21) より、事後予測分布は

$$p_{Bayes}(\mathbf{x}) = \frac{\int p(\mathbf{x}|\theta) \prod_{i=1}^N p(\mathbf{x}_i|\theta)p(\theta)d\theta}{\int \prod_{i=1}^N p(\mathbf{x}_i|\theta')p(\theta')d\theta'} \tag{2.22}$$

で求められる。つまり、ベイズ推定では最適化などの数値計算を行うことなく、事前確率 $p(\theta)$ とパラメトリックモデル $p(\mathbf{x}|\theta)$ を定めれば、上式より確率分布を推定できる。

ここで、ベイズ推定法と最尤推定法の違いについて考察する。2.3.1 節で示したように、最尤推定法では決定論的なパラメータ $\hat{\theta}_{ML}$ を用いて確率密度関数 $\hat{p}_{ML}(\mathbf{x})$ を近似した。一方で、ベイズ推定法では、パラメータ θ を確率的な変数とみなし、モデルを事後確率 $p(\theta|\mathcal{X})$ により周辺化することにより、確率密度関数 $\hat{p}_{Bayes}(\mathbf{x})$ を近似する。この違いを直観的に理解するために、図 2.4 を示す。最尤推定法では、モデル $p(\mathbf{x}|\theta)$ の中から尤度を最大にする確率密度関数を選ぶため、パラメトリックモデルの関数系から外れることはない。しかし、ベイズ推定法ではモデルを事後確率 $p(\theta|\mathcal{X})$ により平均化しているため、図のようにモデルが歪んでいる場合、事後確率分布 $p_{Bayes}(\mathbf{x})$ はモデルの外にはみ出し、結果として、真の確率密度関数に近いものが得られる。

更に、ベイズ推定法が最尤推定法と異なる点は、パラメータ θ に関する事前確率 $p(\theta)$ の知見を含めることができる点である。式 (2.22) によれば、ベイズ推定の推定結果は事前確率 $p(\theta)$ の選び

方に依存するため、これを適当に選ぶことで真の確率密度関数に近いものが得られる。このように、事前確率 $p(\theta)$ について主観的な知見を含めた推定を行うことをベイズ主義と呼ぶ。一方で、最尤推定法のような事実（観測結果）からのみ推定を行うことを頻度主義と呼ぶ。本研究では、ベイズ主義のアプローチによって識別器を構成する手法を提案する。

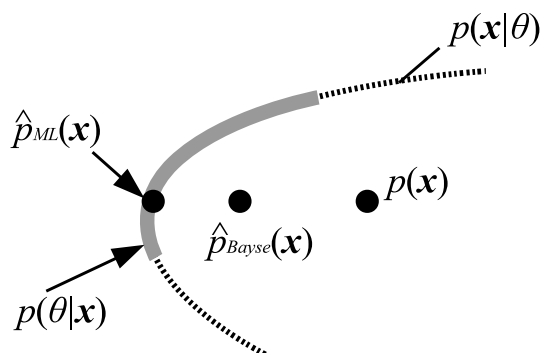


図 2.4: ベイズ推定法と最尤推定法の直感的違い

2.3.3 最大事後確率推定法

2.3.2 節で述べたように、ベイズ推定法ではモデル $p(\mathbf{x}|\theta)$ のパラメータ θ を確率変数として扱い、これを事後確率 $p(\theta|\mathcal{X})$ により周辺化することで事後予測分布を近似した。しかし、パラメータ θ の次元が高い場合には、式 (2.22) に含まれる θ および θ' に関する積分を高精度に計算することが困難となる [9]。そこで、モデル $p(\mathbf{x}|\theta)$ の事後確率 $p(\theta|\mathcal{X})$ に関する周辺化の操作を省略し、パラメータの事後確率 $p(\theta|\mathcal{X})$ を最大にするパラメータ値のみを用いて確率密度関数を近似する。これを最大事後確率 (Maximum a posteriori; MAP) 推定法と呼び、MAP 推定で求められるパラメータ $\hat{\theta}_{MAP}$ を最大事後確率推定量と呼ぶ。MAP 推定で得られる確率密度関数の推定は

$$\hat{p}_{MAP}(\mathbf{x}) = p(\mathbf{x}|\hat{\theta}_{MAP}), \quad \hat{\theta}_{MAP} = \operatorname{argmax}_{\theta} p(\theta|\mathcal{X}) \quad (2.23)$$

で与えられる。

最大事後確率推定量 $\hat{\theta}_{MAP}$ は、直接求めるよりも対数をとった事後確率を最大にするパラメータ値を求めるほうが計算が簡単になる場合が多い。また、式 (2.21) の分母 $p(\mathcal{X})$ がパラメータ θ に依存しないことを用いれば、最大事後確率推定量は

$$\hat{\theta}_{MAP} = \operatorname{argmax}_{\theta} \log p(\theta|\mathcal{X}) \quad (2.24)$$

$$= \operatorname{argmax}_{\theta} \left(\log p(\mathcal{X}|\theta) + \log p(\theta) \right) \quad (2.25)$$

で与えられる。ここで、右辺第 1 項 $\log p(\mathcal{X}|\theta)$ は対数尤度であり、この項のみを最大化する手法が最尤推定法であった。最尤推定法は、訓練標本数が少ない場合に尤度が非常に大きくなり、訓練標本に過剰に適合した確率密度関数が得られることがある。このような現象は過適合もしくは過学習と呼ばれる。そこで、MAP 推定法では対数尤度と共に対数事前確率 $\log p(\theta)$ を合わせて最大化することにより、最尤推定解 $\hat{p}_{ML}(\mathbf{x})$ よりも事前確率 $p(\theta)$ の大きいほうに補正する効果が得られる。

このように過適合を回避するための操作のことを罰則もしくは正則化と呼び、最大事後確率推定は罰則付き最尤推定法と呼ばれることもある。

2.4 ノンパラメトリック法

2.3節では、有限のパラメータで関数形が決まるパラメトリックモデルに関する確率密度関数の推定法を紹介した。これに対して、本節では、ノンパラメトリックなアプローチを紹介する。パラメトリック法では、選ばれたモデル $p(\mathbf{x}|\boldsymbol{\theta})$ が真の分布を表現するには貧弱なモデルである場合があり、これは予測性能が悪くなる要因となる。例えば、標本が多峰性の分布から観測されたのにも関わらず、単峰性のガウス分布でモデル化した場合に予測性能が悪くなる [10]。そこで、ノンパラメトリック法では、分布の形状についてわずかな仮定しかせず、主に簡単な頻度主義的なアプローチを用いる [6]。以降、例を用いてノンパラメトリック手法の説明を行う。

ノンパラメトリック手法で最も基本的なものがヒストグラム密度推定法である。例として1次元の連続変数 x の密度推定について考える。標準的なヒストグラムでは、 x の変域を幅 $\Delta_i > 0$ の別々の区間に区切り、 i 番目の区間に属する x の観測数 n_i を数える。観測総数を N とすると、各区間の確率密度は

$$p_i = \frac{n_i}{N\Delta_i} \quad (2.26)$$

と計算できる。一般的に、各区間幅 Δ_i は均等に $\Delta_i = \Delta$ ととることが多い。ヒストグラム法の特徴として、データ点が逐次的に与えられた場合でも、容易に適用できることが挙げられる。しかし、大きな問題点として推定された密度が各区間の縁で不連続になることが挙げられる。この点から、ヒストグラム法は1次元や2次元の簡便な可視化には役立つが、実際の問題にはほとんど応用できない。

2.4.1 カーネル密度推定法

カーネル密度推定法は、代表的なノンパラメトリック手法のひとつであり、多くの問題に応用されている。未知の確率密度 $p(\mathbf{x})$ から d 次元パターン \mathbf{x} の訓練標本 $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ が観測されたとき、この訓練標本から $p(\mathbf{x})$ を推定する問題を考える。ある注目点 \mathbf{x}' に関して、これを含むある小さな領域 R を考える。また領域 R の体積を $V := \int_R d\mathbf{x}$ とすると、あるパターン \mathbf{x} が領域 R に属する確率は

$$P := \int_R p(\mathbf{x}) d\mathbf{x} \quad (2.27)$$

である。確率 P の値は、注目点 \mathbf{x}' を用いて

$$P \approx Vp(\mathbf{x}') \quad (2.28)$$

と近似できる。一方、 N 個の訓練標本 \mathcal{X} をうち領域 R に属する個数を K で表せば、 P の値は

$$P \approx \frac{K}{N} \quad (2.29)$$

と近似することもできる。ゆえに、式 (2.28)、(2.29) より $p(\mathbf{x}')$ の値は次式で近似できる。

$$p(\mathbf{x}') \approx \frac{K}{NV} \quad (2.30)$$

ここで、領域 R に属する個数を K を数えるために、次の関数を定義する。

$$k(\mathbf{u}) = \begin{cases} 1, & |u_i| \leq \frac{1}{2} \quad (i = 1, \dots, d) \\ 0, & otherwise \end{cases} \quad (2.31)$$

これは原点を中心とする単位立方体を表している。式 (2.31) より、この立方体内部の総点数は

$$K = \sum_{i=1}^N k\left(\frac{\mathbf{x}' - \mathbf{x}_i}{h}\right) \quad (2.32)$$

となる。これを式 (2.30) に代入すると、 \mathbf{x}' に関する推定密度

$$p(\mathbf{x}') = \frac{1}{N} \sum_{i=1}^N \frac{1}{h^d} k\left(\frac{\mathbf{x}' - \mathbf{x}_i}{h}\right) \quad (2.33)$$

が得られる。ただし、一辺が h の d 次元超立方体の体積が $V = h^d$ であることを用いた。上式は関数 $k(\mathbf{u})$ の対称性を用いると、 \mathbf{x}' を中心とする立方体が 1 つあるのではなく、中心が点 \mathbf{x}_i にある N 個の立方体について総和をとったという、別の解釈ができる [10]。したがって、訓練標本 \mathcal{X} を用いて $p(\mathbf{x})$ を推定する問題において、式 (2.33) はカーネル密度推定法の一般形として導かれる。式 (2.33) において、 $k(\cdot)$ で表される関数をカーネル関数と呼び、特に式 (2.31) はパーセン窓関数と呼ばれる。但し、カーネル関数は

$$k(\mathbf{x}) \geq 0 \quad \forall \mathbf{x} \quad (2.34)$$

$$\int k(\mathbf{x}) d\mathbf{x} = 1 \quad (2.35)$$

を満たす必要がある。

式 (2.31) のようにパーセン窓関数を用いた場合、立方体の縁で不連続が生じてしまうため、一般的にはより滑らかなカーネル関数を選ばれる。その中でも代表的なカーネル関数として用いられるのが、ガウスクーネルであり、次式で与えられる。

$$k(\mathbf{x}) = \frac{1}{(2\pi)^{1/2}} \exp\left(-\frac{\langle \mathbf{x}, \mathbf{x} \rangle}{2}\right) \quad (2.36)$$

本研究においても、ガウスクーネルによるカーネル密度推定を導入した手法を提案する。

2.5 カーネル法

カーネル法とは、カーネルと呼ばれる関数を用いることで、入力パターンを元空間から高次元空間へ写像し、特徴空間上で線形分離を行う手法である。1960 年代に Aizermann らは、カーネル関数を特徴空間における内積として用いるという概念を機械学習に初めて導入した。これはポテンシャル関数法として知られている。その後、Vapnik らが Optimal Hyperplane Classifier (OHC) にカーネル法を適用した Support Vector Machine (SVM) を発表したことで脚光を浴びることとなり、カーネル法を用いた識別手法は広く研究されている [11, 12, 13]。

カーネル法の特徴として、高次元空間における相関を直接計算する代わりに、元空間のある関数値が高次元での内積に等しくなる関数を用いることで、非線形分離問題を扱い易くした点が挙げられる。一般的に、入力データを元空間から高次元空間へと写像すると、高次元での計算量は膨大になり、実時間での処理が困難となる。そこで、カーネル法は高次元空間での陽な計算を、カーネル

関数を用いることで元空間で暗に実現し、その計算量を実行可能なものとした。この特性はカーネルトリックと呼ばれている。他方で、カーネル法は、モジュール性の高さからも注目されている。多くのカーネル関数による識別手法では、そのアルゴリズムがカーネルの関数形に依存することがない。したがって、特定の問題に合わせてカーネル関数を選択することができる。また、データの次元や構造などにも依存せず、カーネル関数の計算のみを行えば、後の処理は汎用的に行えることが多い。このように、カーネル関数の計算処理と識別器構成の処理を分離できるのは、カーネル法の大きな特徴のひとつである [14, 15, 16].

2.5.1 Support Vector Machine

本節では、Support Vector Machine (SVM) を例に、カーネル法による線形識別器の非線形拡張について説明する。サポートベクトルマシン (support vector machine; SVM) は、マージン最大化基準による識別手法 OHC にカーネル法を適用した手法である [17]。特徴として、文字パターンの認識やグラフの認識など特定の問題ごとに適したカーネル関数を定義することにより、汎化性能を向上できる点が挙げられる [18, 19].

Optimal Hyperplane Classifier 入力パターンが d 次元ベクトル $\mathbf{x} = (x_1, \dots, x_d)^T$ で表されているとき、 N 個の入力信号 $\mathbf{x}_i, (i = 1, \dots, N)$ を 2 クラスに分類する問題を考える。二つのクラスが、入力空間の超平面で分離できるとき、この超平面を決定面と呼ぶ。決定面を与える関数を識別関数もしくは決定関数と呼び、次式で与えられる。

$$g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b \quad (2.37)$$

ここで、 \mathbf{w} は d 次元重みベクトル、 b はバイアス項である。このとき、あるクラスのデータがすべて式 (2.37) で与えられる超平面の正側 ($g(\mathbf{x}) > 0$) にあり、他のクラスのデータがすべて負側 ($g(\mathbf{x}) < 0$) にあるとき、2 クラス問題は線形分離可能であるという。SVM では、分離超平面とそれに最も近い標本との間の距離 (マージン) が最大になるように、分離超平面を決定する (図 2.5)。マージンを最大化することにより、新たに観測される未知データに対する汎化能力を高くすることができる。

入力データ \mathbf{x}_i がクラス 1 に属するとき $y_i = 1$ 、クラス 2 に属するとき $y_i = -1$ とラベルをつけると、線形分離可能であるとき、決定関数 (2.37) 式は次の条件を満たすことができる。

$$\begin{cases} \langle \mathbf{w}, \mathbf{x} \rangle + b > 0 & (y_i = 1) \\ \langle \mathbf{w}, \mathbf{x} \rangle + b < 0 & (y_i = -1) \end{cases} \quad (i = 1, \dots, N) \quad (2.38)$$

線形分離可能性より、 $\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$ を満たす教師データはない。したがって、クラス間の分離度を調整するために、式 (2.38) を次の条件で置き換える。

$$\begin{cases} \langle \mathbf{w}, \mathbf{x} \rangle + b \geq 1 & (y_i = 1) \\ \langle \mathbf{w}, \mathbf{x} \rangle + b \leq -1 & (y_i = -1) \end{cases} \quad (i = 1, \dots, N) \quad (2.39)$$

このとき、式 (2.39) は

$$y_i(\langle \mathbf{w}, \mathbf{x} \rangle + b) \geq 1 \quad (i = 1, \dots, N) \quad (2.40)$$

と等価になる。

ここで、図 2.5 に示す支持超平面とは、分離超平面から最短距離にある正例または負例のデータを通り、かつ、分離超平面と平行な超平面のことを指す。したがって、マージンを最大化すること

は、分離超平面と支持超平面の距離を最大化することである。支持超平面上の信号 \mathbf{x}_i と分離超平面との距離 a は

$$a = \frac{|g(\mathbf{x}_i)|}{\|\mathbf{w}\|} = \frac{\min_{j=1,\dots,N} |g(\mathbf{x}_j)|}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|} \quad (2.41)$$

で与えられる。式 (2.40), (2.41) より、最適化問題は次式で与えられる。

$$\min_{\mathbf{w}, b} \quad Q(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 \quad (2.42)$$

$$\text{s.t.} \quad y_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 \quad (i = 1, \dots, N) \quad (2.43)$$

制約条件付きの最適化問題は、Lagrange の未定乗数法の適用により制約無し of 最適化問題に書き換えることができる。 $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_N)$ を非負の Lagrange 乗数とすると、Lagrange 関数は

$$\min : \quad L(\mathbf{w}, b, \boldsymbol{\lambda}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_i^N \lambda_i \{y_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1\} \quad (2.44)$$

で与えられる。上式の最適解において、式 (2.44) は \mathbf{w}, b に関して最小で、 $\boldsymbol{\lambda}$ に関して最大となる。つまり、Lagrange の主問題 P と双対問題 D は

$$\text{P :} \quad \min_{\mathbf{w}, b} \sup_{\boldsymbol{\lambda}} L(\mathbf{w}, b, \boldsymbol{\lambda}) \quad (2.45)$$

$$\text{D :} \quad \max_{\boldsymbol{\lambda}} \inf_{\mathbf{w}, b} L(\mathbf{w}, b, \boldsymbol{\lambda}) \quad (2.46)$$

となる。双対問題 D の解 $\hat{\mathbf{w}}, \hat{b}$ において、 L は \mathbf{w}, b の偏微分が 0 となる。

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^N \lambda_i y_i \mathbf{x}_i = \mathbf{0} \quad (2.47)$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^N \lambda_i y_i = 0 \quad (2.48)$$

これらを式 (2.44) に代入すると、双対問題 D は

$$\max \quad Q(\boldsymbol{\lambda}) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \quad (2.49)$$

$$\text{s.t.} \quad \sum_{i=1}^N \lambda_i y_i = 0, \quad \lambda_i \geq 0 \quad (i = 1, \dots, N) \quad (2.50)$$

となる。最適化問題によって得られる \mathbf{w}, b を決定関数 (2.37) に代入することにより、データ \mathbf{x} は、 $g(\mathbf{x}) > 0$ ならクラス 1 に分類し、 $g(\mathbf{x}) < 0$ ならクラス 2 に分類する。 $g(\mathbf{x}) = 0$ のとき、 \mathbf{x} は境界上にあり分類できない。

ハードマージン Support Vector Machine OHC において、式 (2.42)-(2.43) や式 (2.49)-(2.50) に注目すると、いずれも標本ベクトル \mathbf{x} に関する内積のみで表現されている。そこで、

$$k(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle \quad (2.51)$$

で定義されるカーネル関数を用いれば、式 (2.49)-(2.50) は

$$\min \quad Q(\boldsymbol{\lambda}) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \quad (2.52)$$

$$\text{s.t.} \quad \sum_{i=1}^N \lambda_i y_i = 0, \quad \lambda_i \geq 0 \quad (i = 1, \dots, N) \quad (2.53)$$

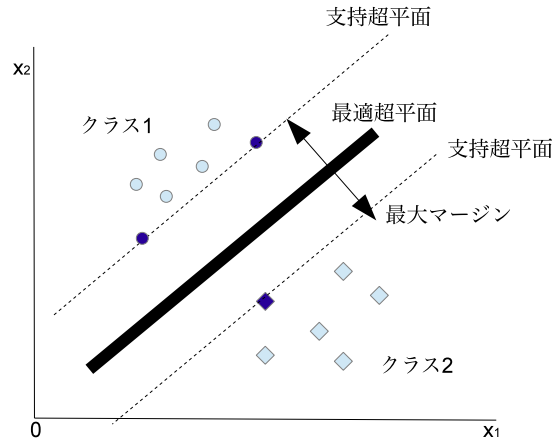


図 2.5: 2次元空間におけるハードマージン SVM

と書き換えられる。したがって、標本ベクトル \mathbf{x} の高次元空間への非線形写像 $\phi(\mathbf{x})$ についても同様に、カーネル関数

$$k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle \quad (2.54)$$

を用いれば、SVM の非線形拡張が容易に実現できる。

ソフトマージン Support Vector Machine ソフトマージンを伴う線形 SVM は、スラック変数 ξ_i を用いることで、2クラス問題が線形分離可能でない場合にも、SVM を適用できるように拡張されたものである (図 2.6)。次の最適問題を解くことで、識別に必要なバイアス項 b 、 d 次元重みベクトル \mathbf{w} とスラック変数 ξ が求められる。

$$\min \quad Q(\mathbf{w}, b, \xi) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \quad (2.55)$$

$$\text{s.t.} \quad y_i(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad (i = 1, \dots, N) \quad (2.56)$$

ここで、 \mathbf{x}_i は d 次元教師データ、 y_i はそれに対応するクラスラベル、 $\phi(\mathbf{x}_i)$ は \mathbf{x}_i の写像である。パラメータ C は式 (2.55) の右辺第 1 項のマージン最大化と第 2 項の誤認識の最小化のトレードオフを決定するマージンパラメータである。すなわち、 C の値を大きくすれば、教師データの識別率は上がるが、大きくしすぎると過学習が生じ、汎化能力が低下する。そのため、高い汎化能力を実現するためには、 C を適切な値に設定する必要がある。

2.5.2 マルチカーネル学習

マルチカーネル学習は、カーネル法の拡張手法である。基本的なカーネル法では、特定の問題ごとにある一つのカーネル関数を決定し、これを用いて識別問題を扱っている。また、カーネルはパラメータを含むため、カーネル関数の形と同時にパラメータの値も決定する必要がある。最適なパラメータを求めるためには、問題毎に交差検定などを行い、パラメータを調整しなければならない。

一方で、マルチカーネル学習法では、複数のカーネル関数群から最適な候補を選択し、それと同時にカーネルパラメータを自動的に決定するための処理を行う。マルチカーネル学習法を適用した

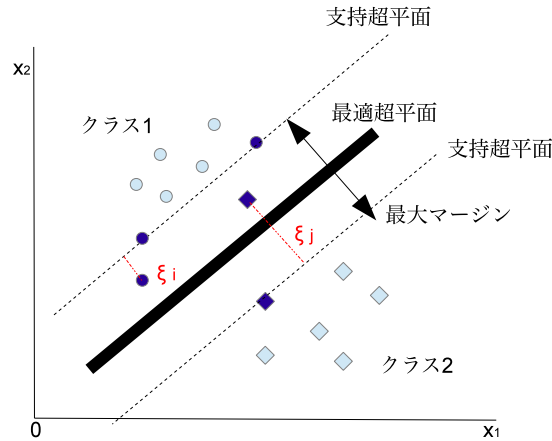


図 2.6: 2次元空間におけるソフトマージン SVM

手法の多くは、カーネル関数を複数のカーネル関数の線形和で表現し、各カーネル関数の重みパラメータを求める。つまり、カーネル関数 $k(\cdot, \cdot)$ は複数のカーネル関数 $k_m(\cdot, \cdot)$ ($m = 1, \dots, M$) を用いて

$$k(\mathbf{x}, \mathbf{z}) = \sum_{m=1}^M \eta_m k_m(\mathbf{x}, \mathbf{z}) \quad (2.57)$$

と表現される。複数のカーネルを用いることで、より柔軟なカーネル関数を表現することが可能となる。マルチカーネル学習では各カーネル関数の重みパラメータ η_m を更新する処理が必要となる。Gönen と Alpaydm は、マルチカーネル学習のアプローチを以下の 5 種類に分類した [20]。

1. 重みパラメータの学習を行わないもの。
2. ヒューリスティックなアプローチ。単カーネルを独立に用いた場合の結果を基に、重みパラメータの値を決定する。
3. 最適化問題によるアプローチ。重みパラメータに関する最適化問題を設定し、重みパラメータを決定する。
4. ベイズ理論的アプローチ。重みパラメータを確率変数として扱い、重みパラメータを推定する。
5. ブースティングアプローチ。性能評価に基づく停止則が止まるまで、新たなカーネルを追加する。

本研究では、最適化問題によるアプローチによる手法を提案する。

2.6 線形計画 MAP 識別法

線形計画 MAP 識別法 (Maximum A Posteriori based kernel classifier trained by Linear Programming; MAPLP) は、2010 年に Nopriadi と Yamashita によって提案された、カーネル法を用いた MAP 推定に基づく識別手法である [2, 21]。

2クラス分類問題について考える。訓練標本 $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ が観測されているとする。 $\mathbf{x}_i \in \mathbb{R}^d$ は d 次元のベクトルであり、 $y_i \in \{0, +1\}$ はクラスラベルである。このとき、以下に示す確率を定義する。

$$P(y), p(\mathbf{x}), p(\mathbf{x}|y), P(y|\mathbf{x}) \quad (2.58)$$

$P(y)$ はクラス y の生起確率、 $p(\mathbf{x})$ はクラスに依存しない標本 \mathbf{x} の確率密度関数、 $p(\mathbf{x}|y)$ はクラス y に属する \mathbf{x} の確率密度関数、 $P(y|\mathbf{x})$ は \mathbf{x} が生起したとき、そのクラスが y である確率を表している。上式の各項の間には

$$\sum_{y=0,1} P(y) = 1 \quad (2.59)$$

$$\sum_{y=0,1} P(y|\mathbf{x}) = 1 \quad (2.60)$$

$$p(\mathbf{x}) = \sum_{y=0,1} P(y)p(\mathbf{x}|y) \quad (2.61)$$

が成り立つ。さらに、ベイズの定理より

$$P(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)P(y)}{p(\mathbf{x})} \quad (y = 0, 1) \quad (2.62)$$

が成り立つ。前述したように、 $P(y|\mathbf{x})$ は未知のパターン \mathbf{x} が観測されたとき、それがに属するクラスが y である確からしさを表している。したがって、パターン \mathbf{x} を識別する際に、事後確率 $P(y|\mathbf{x})$ ($y = 0, 1$) が最大となるクラス y に識別するという方法をとる。つまり、識別法を

$$\hat{y} = \operatorname{argmax}_y P(y|\mathbf{x}) \quad (2.63)$$

とする (図 2.7)。このように、クラスの事後確率に基づく識別法をベイズ決定則と呼ぶ [5]。ベイズ決定則では、各クラスの識別関数として事後確率密度関数を用いている。識別結果のみに焦点をあ

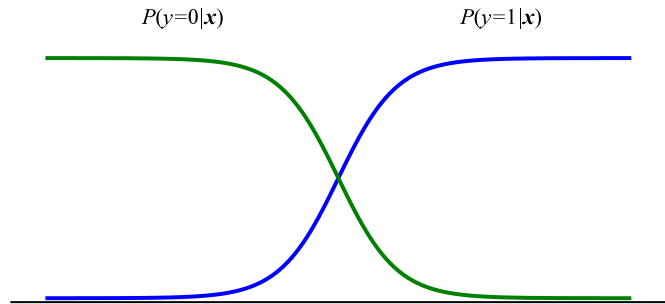


図 2.7: 事後確率による 2 クラス分類

てれば、各クラスの事後確率を厳密に求める必要はなく、識別関数の大小関係もしくは識別結果のみが一致していればよい。そこで MAPLP では、事後確率 $P(y|\mathbf{x})$ を直接計算する代わりに、ベイズ決定則 (2.63) を満たすような代理関数 $w(\mathbf{x}, y)$ を求める。

$$\operatorname{argmax}_y w(\mathbf{x}, y) = \operatorname{argmax}_y P(y|\mathbf{x}) \quad (2.64)$$

MAPLP では、図 2.8 で示すような代理関数が求まるように最適化問題を定義する。つまり、解が、

$$w(\mathbf{x}, y=0) = \begin{cases} 0, & \text{if } P(+1|\mathbf{x}) > P(0|\mathbf{x}) \\ 1, & \text{if } P(+1|\mathbf{x}) < P(0|\mathbf{x}) \end{cases} \quad (2.65)$$

$$w(\mathbf{x}, y=1) = \begin{cases} 1, & \text{if } P(+1|\mathbf{x}) > P(0|\mathbf{x}) \\ 0, & \text{if } P(+1|\mathbf{x}) < P(0|\mathbf{x}) \end{cases} \quad (2.66)$$

となる最適化問題を定義する。したがって、学習の最適化問題は次式で定義される。

$$\max_w \sum_{y=0,1} E_{\mathbf{x}} P(y|\mathbf{x}) \min(w(\mathbf{x}, y), 1) \quad (2.67)$$

$$\text{s.t.} \quad \sum_{y=0,1} E_{\mathbf{x}} [w(\mathbf{x}, y)] = 1 \quad (2.68)$$

$$w(\mathbf{x}, y) \geq 0, \quad \forall \mathbf{x} \quad (2.69)$$

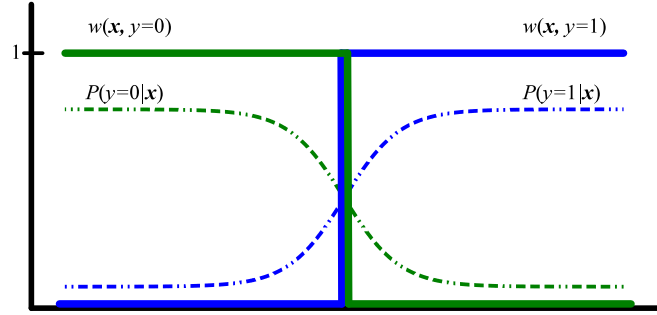


図 2.8: MAPLP で求める代理関数と真の事後確率との関係

式 (2.68)-(2.69) は確率密度関数の性質と一致している。ここで、 $w(\mathbf{x}, y)$ をカーネル関数でモデル化する。

$$w(\mathbf{x}, y) = \sum_{j=1}^N \alpha_{y,j} k(\mathbf{x}, \mathbf{x}_j), \quad y = 0, 1 \quad (2.70)$$

$\alpha_{y,j}$ は重みを表すパラメータである。目的関数 (2.67) は、期待値を標本平均に置き換え、学習標本に対して $P(y = y_i|\mathbf{x}_i) = 1, P(y \neq y_i|\mathbf{x}_i) = 0$ であることを用いれば、

$$\frac{1}{N} \sum_{i=1}^N \min \left(\sum_{j=1}^N \alpha_{y_i,j} k(\mathbf{x}_i, \mathbf{x}_j), 1 \right) \quad (2.71)$$

と書き換えられる。さらに、スラック変数 ξ_i を導入することにより、

$$\max_{\alpha_{y,j}, \xi_i} \sum_{j=1}^N \alpha_{y_i,j} k(\mathbf{x}_i, \mathbf{x}_j) - \xi_i \quad (2.72)$$

$$\text{s.t.} \quad \xi_i \geq 0 \quad (2.73)$$

$$\sum_{j=1}^N \alpha_{y_i,j} k(\mathbf{x}_i, \mathbf{x}_j) - \xi_i \leq 1, \quad i = 1, \dots, N \quad (2.74)$$

と書き換えられる。制約条件 (2.68),(2.69) は各々

$$\frac{1}{N} \sum_{y=0,1} \sum_{i=1}^N \sum_{j=1}^N \alpha_{y,j} k(\mathbf{x}_i, \mathbf{x}_j) = 1 \quad (2.75)$$

$$\alpha_{y,j} \geq 0, \quad y = 0, 1 \quad j = 1, \dots, N \quad (2.76)$$

と書き換えられる。制約条件 (2.69) は厳密に式変形を行うと

$$\sum_{j=1}^N \alpha_{y,j} k(\mathbf{x}, \mathbf{x}_j) \geq 0, \quad \forall \mathbf{x}, \quad y = 0, 1 \quad (2.77)$$

となることに注意が必要である。しかし、すべての \mathbf{x} について (2.77) を評価することは困難である。そこで、(2.76) と置き換えることで、最適化問題を容易に解くことが可能となる。ここで

$$\boldsymbol{\alpha}_0 = [\alpha_{0,1}, \dots, \alpha_{0,N}]^T \quad (2.78)$$

$$\boldsymbol{\alpha}_1 = [\alpha_{1,1}, \dots, \alpha_{1,N}]^T \quad (2.79)$$

$$\boldsymbol{\xi} = [\xi_1, \dots, \xi_N]^T \quad (2.80)$$

$$\mathbf{y} = [y_1, \dots, y_N]^T \quad (2.81)$$

$$\mathbf{z} = [\boldsymbol{\alpha}_0^T, \boldsymbol{\alpha}_1^T, \boldsymbol{\xi}^T]^T \quad (2.82)$$

$$\mathbf{c} = \begin{bmatrix} \mathbf{K}(\mathbf{1}_N - \mathbf{y}) \\ \mathbf{K}\mathbf{y} \\ -\mathbf{1}_N \end{bmatrix} \quad (2.83)$$

$$\mathbf{a} = \frac{1}{N} [\mathbf{1}_N^T \mathbf{K} | \mathbf{1}_N^T \mathbf{K} | \mathbf{0}_N^T]^T \quad (2.84)$$

$$\mathbf{A} = [\text{diag}(\mathbf{1}_N - \mathbf{y}) \mathbf{K} | \text{diag}(\mathbf{y}) \mathbf{K} | -\mathbf{I}_N] \quad (2.85)$$

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \dots & k(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & \dots & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \quad (2.86)$$

とおけば、最適化問題は

$$\max_{\mathbf{z}} \quad \langle \mathbf{c}, \mathbf{z} \rangle \quad (2.87)$$

$$\text{s.t.} \quad \langle \mathbf{a}, \mathbf{z} \rangle = 1 \quad (2.88)$$

$$\mathbf{A}\mathbf{z} \leq \mathbf{1}_N \quad (2.89)$$

$$\mathbf{z} \geq \mathbf{0}_{3N} \quad (2.90)$$

と線形計画問題に帰着する。式 (2.86) で定義される行列 \mathbf{K} はカーネルグラム行列と呼ばれる。線形計画問題を解くことにより、 $\boldsymbol{\alpha}$ が求まる。これを式 (2.70) に代入することで識別関数を得る。

第3章 提案手法

3.1 確率密度推定を用いた拡張手法

MAPLP では、式 (2.71) で示したように目的関数における期待値を訓練標本の平均値で近似している。これは、未知のパターンを考慮せずに訓練標本のみで学習を行っていることに相当する。したがって、訓練標本数が相対的に少ない場合に過学習を起し、未知のパターンに対する汎化能力が低下してしまう恐れがある。さらに、カーネル法を適用した手法であるにもかかわらず、カーネル法の拡張手法であるマルチカーネル学習が適用できないという欠点も挙げられる。この欠点は、先述したように期待値を標本平均で近似していることに由来する。

本研究では、各クラスにおける入力データの条件付き確率密度関数の推定による MAPLP 拡張手法の提案する。入力データの条件付き確率密度関数を予めモデル化することで、学習標本数が相対的に少ない場合でも過学習を緩和できると考えられる。さらに、マルチカーネル学習を適用することにより、カーネル関数の重みパラメータの自動計算が可能となり、交差検定等によるカーネルパラメータの最適値推定が簡略化される。加えて、汎化能力の高い学習が可能になると考えられる。

MAPLP の最適化問題 (2.67)-(2.69) は、スラック関数 $\xi(\mathbf{x})$ を導入すると、次式のように変形できる。

$$\max_{w, \xi} \sum_{y=0,1} E_{\mathbf{x}} \left[P(y|\mathbf{x})(w(\mathbf{x}, y) - \xi(\mathbf{x})) \right] \quad (3.1)$$

$$\text{s.t.} \quad \sum_{y=0,1} E_{\mathbf{x}} [w(\mathbf{x}, y)] = 1 \quad (3.2)$$

$$w(\mathbf{x}, y) \geq 0, \forall \mathbf{x} \quad y = 0, 1 \quad (3.3)$$

$$\xi(\mathbf{x}) \geq 0, \forall \mathbf{x} \quad (3.4)$$

$$w(\mathbf{x}, y) - \xi(\mathbf{x}) \leq 1, \forall \mathbf{x}, y \quad (3.5)$$

ここで、代理関数 $w(\mathbf{x}, y)$ とスラック関数 $\xi(\mathbf{x})$ はカーネル関数 $k(\cdot, \cdot)$ を用いて

$$w(\mathbf{x}, y) = \sum_{j=1}^N \alpha_{y,j} k(\mathbf{x}, \mathbf{x}_j), \quad y = 0, 1 \quad (3.6)$$

$$\xi(\mathbf{x}) = \sum_{j=1}^N \beta_j k(\mathbf{x}, \mathbf{x}_j) \quad (3.7)$$

とモデル化する。 $\alpha_{y,j}$, β_j は重みを表すパラメータである。ここで、(3.1) の目的関数を J とおけ

ば、 J は期待値の定義とベイズの定理より

$$J = \sum_{y=0,1} E_{\mathbf{x}} \left[P(y|\mathbf{x})(w(\mathbf{x}, y) - \xi(\mathbf{x})) \right] \quad (3.8)$$

$$= \sum_{y=0,1} \int p(\mathbf{x}) P(y|\mathbf{x})(w(\mathbf{x}, y) - \xi(\mathbf{x})) d\mathbf{x} \quad (3.9)$$

$$= \sum_{y=0,1} \int p(\mathbf{x}) \frac{p(\mathbf{x}|y)P(y)}{p(\mathbf{x})} (w(\mathbf{x}, y) - \xi(\mathbf{x})) d\mathbf{x} \quad (3.10)$$

$$= \sum_{y=0,1} P(y) \int p(\mathbf{x}|y)(w(\mathbf{x}, y) - \xi(\mathbf{x})) d\mathbf{x} \quad (3.11)$$

と展開できる。

提案手法では、各クラスにおける入力パターンの条件付き確率密度関数 $p(\mathbf{x}|y)$ のモデルとして (1) カーネル密度関数 (2) ガウス関数 (3) ガウス差分関数を用いる。入力データの条件付き確率密度関数を予めモデル化することで、訓練標本点に分布の幅をもたせ、学習標本数が相対的に少ない場合でも過学習を緩和できると考えられる。

3.1.1 カーネル密度関数を用いた拡張手法

2.4.1 節で紹介したように、カーネル密度推定は、分布を特定せずに観測値から分布を推定するノンパラメトリック手法である。主に、複雑な分布を推定する場合に有効である。本提案手法では、各クラス標本が未知の分布に従って観測されたものと仮定し、各クラスにおける入力パターンの条件付き確率密度関数 $p(\mathbf{x}|y)$ をカーネル密度推定によりモデル化する。

$$p(\mathbf{x}|y=0) = \frac{1}{N_0} \sum_{i=1}^{N_0} \phi(\mathbf{x} - \mathbf{x}_i^0) = \frac{1}{N_0} \sum_{i=1}^N (1 - y_i) \phi(\mathbf{x} - \mathbf{x}_i) \quad (3.12)$$

$$p(\mathbf{x}|y=1) = \frac{1}{N_1} \sum_{i=1}^{N_1} \phi(\mathbf{x} - \mathbf{x}_i^1) = \frac{1}{N_1} \sum_{i=1}^N y_i \phi(\mathbf{x} - \mathbf{x}_i) \quad (3.13)$$

$$\phi(\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2}\right) \quad (3.14)$$

N_0, N_1 は各クラスの訓練標本数、 σ^2 は分散を決めるパラメータである。これらを式 (3.11) に代入すると目的関数 J は

$$\begin{aligned} J &= \frac{P(0)}{N_0} \sum_{i=1}^N (1 - y_i) \sum_{j=1}^N (\alpha_{0,j} - \beta_j) \int \phi(\mathbf{x} - \mathbf{x}_i) k(\mathbf{x}, \mathbf{x}_j) d\mathbf{x} \\ &\quad + \frac{P(1)}{N_1} \sum_{i=1}^N y_i \sum_{j=1}^N (\alpha_{1,j} - \beta_j) \int \phi(\mathbf{x} - \mathbf{x}_i) k(\mathbf{x}, \mathbf{x}_j) d\mathbf{x} \end{aligned} \quad (3.15)$$

となる。ここで i, j 成分が $\mathbf{B}_{i,j} = \int \phi(\mathbf{x} - \mathbf{x}_i) k(\mathbf{x}, \mathbf{x}_j) d\mathbf{x}$ となる行列 \mathbf{B} を考える。カーネル関数として RBF カーネル

$$k(\mathbf{x}, \mathbf{z}) = \exp(-\gamma\|\mathbf{x} - \mathbf{z}\|^2) \quad (3.16)$$

を用いると、 $\mathbf{B}_{i,j}$ は次式のように展開される。

$$\mathbf{B}_{i,j} = \frac{1}{(2\pi\sigma^2)^{d/2}} \int \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{x}_i\|^2 - \gamma\|\mathbf{x} - \mathbf{x}_j\|^2\right) \quad (3.17)$$

$$= \frac{1}{(2\pi\sigma^2)^{d/2}} \int \exp\left\{-\frac{1}{2}\left(\frac{1}{\sigma^2} + 2\gamma\right)\|\mathbf{x}\|^2 + \langle \mathbf{x}, \left(\frac{1}{\sigma^2}\mathbf{x}_i + 2\gamma\mathbf{x}_j\right) \rangle - \frac{1}{2\sigma^2}\|\mathbf{x}_i\|^2 - \gamma\|\mathbf{x}_j\|^2\right\} d\mathbf{x} \quad (3.18)$$

ここで、ガウス関数の積分公式

$$\int \exp\left(-\frac{1}{2}\langle \mathbf{x}, \mathbf{A}\mathbf{x} \rangle + \langle \mathbf{b}, \mathbf{x} \rangle\right) = \sqrt{\frac{(2\pi)^d}{|\mathbf{A}|}} \exp\left(\frac{1}{2}\langle \mathbf{b}, \mathbf{A}^{-1}\mathbf{b} \rangle\right) \quad (3.19)$$

を用いて、 $\mathbf{A} = \left(\frac{1}{\sigma^2} + 2\gamma\right)\mathbf{I}_N$ 、 $\mathbf{b} = \frac{1}{\sigma^2}\mathbf{x}_i + 2\gamma\mathbf{x}_j$ とおけば

$$\mathbf{B}_{i,j} = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x}_i\|^2\right) \exp\left(-\gamma\|\mathbf{x}_j\|^2\right) \left(\frac{2\pi}{1/\sigma^2 + 2\gamma}\right)^{d/2} \exp\left\{\frac{1}{2}\left(\frac{1}{\sigma^2} + 2\gamma\right)^{-1}\left\|\frac{1}{\sigma^2}\mathbf{x}_i + 2\gamma\mathbf{x}_j\right\|^2\right\} \quad (3.20)$$

$$= \frac{1}{(1 + 2\gamma\sigma^2)^{d/2}} \int \exp\left(-\frac{\gamma}{1 + 2\gamma\sigma^2}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right) \quad (3.21)$$

となる。したがって、

$$\mathbf{z} = [\boldsymbol{\alpha}_0^T, \boldsymbol{\alpha}_1^T, \boldsymbol{\beta}^T]^T \quad (3.22)$$

とおけば、目的関数 J は

$$J = \frac{P(0)}{N_0} \sum_{i=1}^N (1 - y_i) \sum_{j=1}^N (\alpha_{0,j} - \beta_j) \mathbf{B}_{i,j} + \frac{P(1)}{N_1} \sum_{i=1}^N y_i \sum_{j=1}^N (\alpha_{1,j} - \beta_j) \mathbf{B}_{i,j} \quad (3.23)$$

$$= \frac{P(0)}{N_0} ((\mathbf{1}_N - \mathbf{y})^T \mathbf{B} \boldsymbol{\alpha}_0 - (\mathbf{1}_N - \mathbf{y})^T \mathbf{B} \boldsymbol{\beta}) + \frac{P(1)}{N_1} (\mathbf{y}^T \mathbf{B} \boldsymbol{\alpha}_1 - \mathbf{y}^T \mathbf{B} \boldsymbol{\beta}) \quad (3.24)$$

$$= \left[\frac{P(0)}{N_0} (\mathbf{1}_N - \mathbf{y})^T \mathbf{B} \mid \frac{P(1)}{N_1} \mathbf{y}^T \mathbf{B} \mid \left(\left(\frac{P(0)}{N_0} - \frac{P(1)}{N_1} \right) \mathbf{y} - \frac{P(0)}{N_0} \mathbf{1}_N \right)^T \mathbf{B} \right] \mathbf{z} \quad (3.25)$$

とおける。

次に、制約条件 (3.2) について考える。式 (3.2) の左辺は次式のように展開できる。

$$\sum_{y=0,1} E_{\mathbf{x}}[w(\mathbf{x}, y)] = \sum_{y=0,1} \int p(\mathbf{x}) w(\mathbf{x}, y) d\mathbf{x} \quad (3.26)$$

$$= \sum_{y=0,1} \int p(\mathbf{x}|y') p(y') w(\mathbf{x}, y) d\mathbf{x} \quad (3.27)$$

$$= P(0) \int p(\mathbf{x}|0) w(\mathbf{x}, 0) d\mathbf{x} + P(1) \int p(\mathbf{x}|1) w(\mathbf{x}, 0) d\mathbf{x} \\ + P(0) \int p(\mathbf{x}|0) w(\mathbf{x}, 1) d\mathbf{x} + P(1) \int p(\mathbf{x}|1) w(\mathbf{x}, 1) d\mathbf{x} \quad (3.28)$$

ここで、式 (3.21) を用いれば、

$$\int p(\mathbf{x}|0)w(\mathbf{x}|0)d\mathbf{x} = \frac{1}{N_0}(\mathbf{1}_N - \mathbf{y})^T \mathbf{B}\alpha_0 \quad (3.29)$$

$$\int p(\mathbf{x}|1)w(\mathbf{x}|0)d\mathbf{x} = \frac{1}{N_1}\mathbf{y}^T \mathbf{B}\alpha_0 \quad (3.30)$$

$$\int p(\mathbf{x}|0)w(\mathbf{x}|1)d\mathbf{x} = \frac{1}{N_0}(\mathbf{1}_N - \mathbf{y})^T \mathbf{B}\alpha_1 \quad (3.31)$$

$$\int p(\mathbf{x}|1)w(\mathbf{x}|1)d\mathbf{x} = \frac{1}{N_1}\mathbf{y}^T \mathbf{B}\alpha_1 \quad (3.32)$$

と計算できるので、

$$\mathbf{a}_{sub}^T = \left(\frac{P(0)}{N_0}\mathbf{1}_N + \left(\frac{P(1)}{N_1} - \frac{P(0)}{N_0} \right) \mathbf{y} \right)^T \mathbf{B} \quad (3.33)$$

とおくと、制約 (3.2) は

$$\left[\mathbf{a}_{sub}^T \mid \mathbf{a}_{sub}^T \mid \mathbf{0}_N^T \right] \mathbf{z} = 1 \quad (3.34)$$

と表現できる。

次に、制約条件 (3.3),(3.4) について考える。式 (3.6),(3.7) を用いれば、それぞれ

$$\sum_{j=1}^N \alpha_{0,j} k(\mathbf{x}, \mathbf{x}_j) \geq 0, \quad \forall \mathbf{x} \quad (3.35)$$

$$\sum_{j=1}^N \alpha_{1,j} k(\mathbf{x}, \mathbf{x}_j) \geq 0, \quad \forall \mathbf{x} \quad (3.36)$$

$$\sum_{j=1}^N \beta_j k(\mathbf{x}, \mathbf{x}_j) \geq 0, \quad \forall \mathbf{x} \quad (3.37)$$

となる。しかし、これをすべての \mathbf{x} について考えるのは困難である。したがって、制約条件 (3.3),(3.4) をそれぞれ

$$\alpha_{0,j} \geq 0, \quad (j = 0, \dots, N) \quad (3.38)$$

$$\alpha_{1,j} \geq 0, \quad (j = 0, \dots, N) \quad (3.39)$$

$$\beta_j \geq 0, \quad (j = 0, \dots, N) \quad (3.40)$$

と置き換え、評価を行う。

最後に、制約条件 (3.5) について考える。式 (3.5) の左辺は

$$w(\mathbf{x}, y) - \xi(\mathbf{x}) = \sum_{j=1}^N (\alpha_{y,j} - \beta_j) k(\mathbf{x}, \mathbf{x}_j) \quad (3.41)$$

となり、これがすべての \mathbf{x} について成立している必要がある。しかし、上式をすべての \mathbf{x} を網羅して考えることは難しいため、代表点のみで制約の評価を行う。そこで、代表点として標本 $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ について、次の制約条件を課す。

$$\sum_{j=1}^N (\alpha_{0,j} - \beta_j) k(\mathbf{x}, \mathbf{x}_j) \leq 1 \quad (3.42)$$

$$\sum_{j=1}^N (\alpha_{1,j} - \beta_j) k(\mathbf{x}, \mathbf{x}_j) \leq 1 \quad (3.43)$$

これらは、式 (2.86), (3.22) を用いれば、

$$\begin{bmatrix} \mathbf{K}|\mathbf{0}_{N,N}| - \mathbf{K} \end{bmatrix} \mathbf{z} \leq \mathbf{1}_N \quad (3.44)$$

$$\begin{bmatrix} \mathbf{K} - \mathbf{K}|\mathbf{0}_{N,N}| \end{bmatrix} \mathbf{z} \leq \mathbf{1}_N \quad (3.45)$$

以上をまとめると、カーネル密度関数を用いた MAPLP 拡張手法の最適化問題は、次の線形計画問題に帰着する。

$$\max_{\mathbf{z}} \quad \langle \mathbf{c}, \mathbf{z} \rangle \quad (3.46)$$

$$\text{s.t.} \quad \langle \mathbf{a}, \mathbf{z} \rangle = 1 \quad (3.47)$$

$$\mathbf{A}\mathbf{z} \leq \mathbf{1}_{2N} \quad (3.48)$$

$$\mathbf{z} \geq \mathbf{0}_{3N} \quad (3.49)$$

$$\mathbf{c} = \left[\frac{P(0)}{N_0}(\mathbf{1}_N - \mathbf{y})^T \mathbf{B} \mid \frac{P(1)}{N_1} \mathbf{y}^T \mathbf{B} \mid \left(\left(\frac{P(0)}{N_0} - \frac{P(1)}{N_1} \right) \mathbf{y} - \frac{P(0)}{N_0} \mathbf{1}_N \right)^T \mathbf{B} \right] \quad (3.50)$$

$$\mathbf{a} = \left[\mathbf{a}_{sub} \mid \mathbf{a}_{sub} \mid \mathbf{0}_N^T \right] \quad (3.51)$$

$$\mathbf{a}_{sub} = \left(\frac{P(0)}{N_0} \mathbf{1}_N + \left(\frac{P(1)}{N_1} - \frac{P(0)}{N_0} \right) \mathbf{y} \right)^T \mathbf{B} \quad (3.52)$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{K} & \mathbf{0}_{N,N} & -\mathbf{K} \\ \mathbf{0}_{N,N} & \mathbf{K} & -\mathbf{K} \end{bmatrix} \quad (3.53)$$

$$\mathbf{B} = (\mathbf{B}_{i,j}) \in \mathbb{R}^{N \times N} \quad (3.54)$$

$$\mathbf{B}_{i,j} = \frac{1}{(1 + 2\gamma\sigma^2)^{d/2}} \exp \left(-\frac{\gamma}{(1 + 2\gamma\sigma^2)^{d/2}} \|\mathbf{x}_i - \mathbf{x}_j\|^2 \right) \quad (3.55)$$

3.1.2 ガウス関数を用いた拡張手法

本提案手法では、各クラス標本が相異なるガウス分布に従って観測されたものと仮定し、各クラスにおける入力パターンの条件付き確率密度関数 $p(\mathbf{x}|y)$ をガウス分布によりモデル化する。

$$p(\mathbf{x}|y) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y) = \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}_y|^{1/2}} \exp \left\{ -\frac{1}{2} \langle (\mathbf{x} - \boldsymbol{\mu}_y), \boldsymbol{\Sigma}_y^{-1} (\mathbf{x} - \boldsymbol{\mu}_y) \rangle \right\} \quad (3.56)$$

$\boldsymbol{\mu}_y$ は各クラスの標本平均ベクトル、 $\boldsymbol{\Sigma}_y$ は各クラスの標本分散共分散行列を決めるパラメータである。これを式 (3.11) に代入すると目的関数 J は

$$J = P(0) \sum_{j=1}^N (\alpha_{0,j} - \beta_j) \int p(\mathbf{x}|0) k(\mathbf{x}, \mathbf{x}_j) d\mathbf{x} + P(1) \sum_{j=1}^N (\alpha_{1,j} - \beta_j) \int p(\mathbf{x}|1) k(\mathbf{x}, \mathbf{x}_j) d\mathbf{x} \quad (3.57)$$

となる。ここで、 j 成分が $g_{y,j} = \int p(\mathbf{x}|y)k(\mathbf{x}, \mathbf{x}_j)d\mathbf{x}$ となるベクトルを \mathbf{g}_y とおき、これについて計算を行う。

$$g_{y,j} = \int p(\mathbf{x}|y)k(\mathbf{x}, \mathbf{x}_j)d\mathbf{x} \quad (3.58)$$

$$= \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_y|^{1/2}} \int \exp\left(-\frac{1}{2}\langle(\mathbf{x} - \boldsymbol{\mu}_y), \boldsymbol{\Sigma}_y^{-1}(\mathbf{x} - \boldsymbol{\mu}_y)\rangle - \gamma\|\mathbf{x} - \mathbf{x}_j\|^2\right)d\mathbf{x} \quad (3.59)$$

$$= \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_y|^{1/2}} \exp\left(-\frac{1}{2}\langle\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y^{-1}\boldsymbol{\mu}_y\rangle\right) \exp(-\gamma\|\mathbf{x}_j\|^2) \\ \times \int \exp\left(-\frac{1}{2}\langle\mathbf{x}, (\boldsymbol{\Sigma}_y^{-1} + 2\gamma\mathbf{I})\mathbf{x}\rangle + \langle\mathbf{x}, \boldsymbol{\Sigma}_y^{-1}\boldsymbol{\mu}_y + 2\gamma\mathbf{x}_j\rangle\right)d\mathbf{x} \quad (3.60)$$

また、ガウス関数の積分公式 (3.19) を用いれば、

$$g_{y,j} = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_y|^{1/2}} \exp\left(-\frac{1}{2}\langle\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y^{-1}\boldsymbol{\mu}_y\rangle\right) \exp(-\gamma\|\mathbf{x}_j\|^2) \\ \times \frac{(2\pi)^{d/2}}{|\boldsymbol{\Sigma}_y^{-1} + 2\gamma\mathbf{I}|^{1/2}} \exp\left(\frac{1}{2}\langle(\boldsymbol{\Sigma}_y^{-1}\boldsymbol{\mu}_y + 2\gamma\mathbf{x}_j), (\boldsymbol{\Sigma}_y^{-1} + 2\gamma\mathbf{I})^{-1}(\boldsymbol{\Sigma}_y^{-1}\boldsymbol{\mu}_y + 2\gamma\mathbf{x}_j)\rangle\right) \\ = \frac{1}{|\mathbf{I} + 2\gamma\boldsymbol{\Sigma}_y|^{1/2}} \exp\left(-\frac{1}{2}\langle\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y^{-1}\boldsymbol{\mu}_y\rangle - \gamma\|\mathbf{x}_j\|^2\right) \\ + \frac{1}{2}\langle(\boldsymbol{\Sigma}_y^{-1}\boldsymbol{\mu}_y + 2\gamma\mathbf{x}_j), \boldsymbol{\Sigma}_y(\mathbf{I} + 2\gamma\boldsymbol{\Sigma}_y)^{-1}(\boldsymbol{\Sigma}_y^{-1}\boldsymbol{\mu}_y + 2\gamma\mathbf{x}_j)\rangle \quad (3.61)$$

$$= \frac{1}{|\mathbf{I} + 2\gamma\boldsymbol{\Sigma}_y|^{1/2}} \exp\left(-\gamma\langle\boldsymbol{\mu}_y, (\mathbf{I} + 2\gamma\boldsymbol{\Sigma}_y)^{-1}\boldsymbol{\mu}_y\rangle - \gamma\langle\mathbf{x}_j, (\mathbf{I} + 2\gamma\boldsymbol{\Sigma}_y)^{-1}\mathbf{x}_j\rangle\right) \\ + 2\gamma\langle\boldsymbol{\mu}_y, (\mathbf{I} + 2\gamma\boldsymbol{\Sigma}_y)^{-1}\mathbf{x}_j\rangle \quad (3.62)$$

$$= \frac{1}{|\mathbf{I} + 2\gamma\boldsymbol{\Sigma}_y|^{1/2}} \exp\left(\gamma\langle(\boldsymbol{\mu}_y - \mathbf{x}_j), (\mathbf{I} + 2\gamma\boldsymbol{\Sigma}_y)^{-1}(\boldsymbol{\mu}_y - \mathbf{x}_j)\rangle\right) \quad (3.63)$$

と計算できる。これを用いれば、目的関数 J は

$$J = P(0) \sum_{j=1}^N (\alpha_{0,j} - \beta_j) g_{0,j} + P(1) \sum_{j=1}^N (\alpha_{1,j} - \beta_j) g_{1,j} \quad (3.64)$$

$$= \left[P(0)\mathbf{g}_0^T \mid P(1)\mathbf{g}_1^T \mid - (P(0)\mathbf{g}_0 + P(1)\mathbf{g}_1)^T \right] \mathbf{z} \quad (3.65)$$

と求まる。次に制約条件 (3.2) について考える。これは式 (3.28) のように展開できる。ここで、式 (3.63) を用いれば

$$\int p(\mathbf{x}|0)w(\mathbf{x}|0)d\mathbf{x} = \mathbf{g}_0^T \boldsymbol{\alpha}_0 \quad (3.66)$$

$$\int p(\mathbf{x}|1)w(\mathbf{x}|0)d\mathbf{x} = \mathbf{g}_1^T \boldsymbol{\alpha}_0 \quad (3.67)$$

$$\int p(\mathbf{x}|0)w(\mathbf{x}|1)d\mathbf{x} = \mathbf{g}_0^T \boldsymbol{\alpha}_1 \quad (3.68)$$

$$\int p(\mathbf{x}|1)w(\mathbf{x}|1)d\mathbf{x} = \mathbf{g}_1^T \boldsymbol{\alpha}_1 \quad (3.69)$$

$$(3.70)$$

が求められる。したがって、制約条件 (3.2) は

$$\left[(P(0)\mathbf{g}_0 + P(1)\mathbf{g}_1)^T \mid (P(0)\mathbf{g}_0 + P(1)\mathbf{g}_1)^T \mid \mathbf{0}_N^T \right] = 1 \quad (3.71)$$

と表現できる.

制約条件 (3.3)-(3.5) に関しては, 3.1.1 節と同様に評価を行う.

以上をまとめると, ガウス関数を用いた MAPLP 拡張手法の最適化問題は, 次の線形計画問題に帰着する.

$$\max_{\mathbf{z}} \quad \langle \mathbf{c}, \mathbf{z} \rangle \quad (3.72)$$

$$\text{s.t.} \quad \langle \mathbf{a}, \mathbf{z} \rangle = 1 \quad (3.73)$$

$$\mathbf{A}\mathbf{z} \leq \mathbf{1}_{2N} \quad (3.74)$$

$$\mathbf{z} \geq \mathbf{0}_{3N} \quad (3.75)$$

$$\mathbf{c} = [P(0)\mathbf{g}_0^T \mid P(1)\mathbf{g}_1^T \mid -(P(0)\mathbf{g}_0 + P(1)\mathbf{g}_1)^T]^T \quad (3.76)$$

$$\mathbf{a} = [(P(0)\mathbf{g}_0 + P(1)\mathbf{g}_1)^T \mid (P(0)\mathbf{g}_0 + P(1)\mathbf{g}_1)^T \mid \mathbf{0}_N^T]^T \quad (3.77)$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{K} & \mathbf{0}_{N,N} & -\mathbf{K} \\ \mathbf{0}_{N,N} & \mathbf{K} & -\mathbf{K} \end{bmatrix} \quad (3.78)$$

$$\mathbf{g}_y = (\mathbf{g}_{y,j}) \in \mathbb{R}^{N \times 1} \quad (3.79)$$

$$g_{y,j} = \frac{1}{|\mathbf{I} + 2\gamma\boldsymbol{\Sigma}_y|^{1/2}} \exp\left(\gamma\langle(\boldsymbol{\mu}_y - \mathbf{x}_j), (\mathbf{I} + 2\gamma\boldsymbol{\Sigma}_y)^{-1}(\boldsymbol{\mu}_y - \mathbf{x}_j)\rangle\right) \quad (3.80)$$

3.1.3 ガウス差分関数を用いた拡張手法

ガウス差分関数とは, 主に信号処理においてエッジ検出などに用いられる窓関数のことである. 図 3.1 にガウス差分関数とガウス関数の波形の違いを示した. 図 3.1 から分かるように, ガウス差分関数はガウス関数の極をつぶし, ガウス関数の平均値よりも外側に極をもつ形になっていることがわかる.

識別問題では, 経験的にエラー標本は分布境界面に出現することが多い. そこで, 本提案手法では, 分布境界面付近の損失を大きく評価するために, 各クラスの入力パターンの条件付き確率密度関数 $p(\mathbf{x}|y)$ をガウス差分関数によりモデル化する.

$$p(\mathbf{x}|y) = \left(1 + \frac{\kappa}{\nu^d - 1}\right) \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_y, \nu^2\boldsymbol{\Sigma}_y) - \frac{\kappa}{\nu^d - 1} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y) \quad (3.81)$$

κ は分布の極の尖度を定めるパラメータ ($0 < \kappa < 1$), ν は分布の極の広がりを決めるパラメータ ($\nu > 1$) である. これを式 (3.11) に代入すると目的関数は

$$J = P(0) \sum_{j=1}^N (\alpha_{0,j} - \beta_j) \int p(\mathbf{x}|0)k(\mathbf{x}, \mathbf{x}_j)d\mathbf{x} + P(1) \sum_{j=1}^N (\alpha_{1,j} - \beta_j) \int p(\mathbf{x}|1)k(\mathbf{x}, \mathbf{x}_j)d\mathbf{x} \quad (3.82)$$

となる. ここで, j 成分が $d_{y,j} = \int p(\mathbf{x}|y)k(\mathbf{x}, \mathbf{x}_j)d\mathbf{x}$ となるベクトルを \mathbf{d}_y とおき, これについて計算を行う.

$$d_{y,j} = \int p(\mathbf{x}|y)k(\mathbf{x}, \mathbf{x}_j)d\mathbf{x} \quad (3.83)$$

$$= \left(1 + \frac{\kappa}{\nu^d - 1}\right) \int \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_y, \nu^2\boldsymbol{\Sigma}_y)k(\mathbf{x}, \mathbf{x}_j)d\mathbf{x} - \frac{\kappa}{\nu^d - 1} \int \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)k(\mathbf{x}, \mathbf{x}_j)d\mathbf{x} \quad (3.84)$$

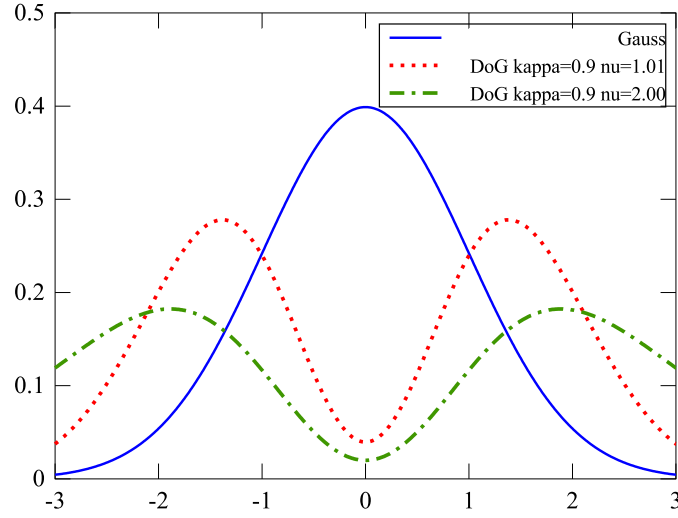


図 3.1: ガウス差分関数とガウス関数の波形 (平均 0, 分散 1)

と展開できる. ここで, 式 (3.63) より $g_{y,j}$ を Σ_y の関数 $g_{y,j}(\Sigma_y)$ とすると, $d_{y,j}$ は

$$d_{y,j} = \left(1 + \frac{\kappa}{\nu^d - 1}\right) g_{y,j}(\nu^2 \Sigma_y) - \frac{\kappa}{\nu^d - 1} g_{y,j}(\Sigma_y) \quad (3.85)$$

と求まる. これを用いれば, 目的関数 J は

$$J = P(0) \sum_{j=1}^N (\alpha_{0,j} - \beta_j) d_{0,j} + P(1) \sum_{j=1}^N (\alpha_{1,j} - \beta_j) d_{1,j} \quad (3.86)$$

$$= \left[P(0) \mathbf{d}_0^T \mid P(1) \mathbf{d}_1^T \mid - (P(0) \mathbf{d}_0 + P(1) \mathbf{d}_1)^T \right] \mathbf{z} \quad (3.87)$$

と求まる. 次に制約条件 (3.2) について考える. これは式 (3.28) のように展開できる. ここで, 式 (3.63) を用いれば

$$\int p(\mathbf{x}|0) w(\mathbf{x}|0) d\mathbf{x} = \mathbf{d}_0^T \boldsymbol{\alpha}_0 \quad (3.88)$$

$$\int p(\mathbf{x}|1) w(\mathbf{x}|0) d\mathbf{x} = \mathbf{d}_1^T \boldsymbol{\alpha}_0 \quad (3.89)$$

$$\int p(\mathbf{x}|0) w(\mathbf{x}|1) d\mathbf{x} = \mathbf{d}_0^T \boldsymbol{\alpha}_1 \quad (3.90)$$

$$\int p(\mathbf{x}|1) w(\mathbf{x}|1) d\mathbf{x} = \mathbf{d}_1^T \boldsymbol{\alpha}_1 \quad (3.91)$$

$$(3.92)$$

が求められる. したがって, 制約条件 (3.2) は

$$\left[(P(0) \mathbf{d}_0 + P(1) \mathbf{d}_1)^T \mid (P(0) \mathbf{d}_0 + P(1) \mathbf{d}_1)^T \mid \mathbf{0}_N^T \right] = 1 \quad (3.93)$$

と表現できる.

制約条件 (3.3)-(3.5) に関しては, 3.1.1 節と同様に評価を行う.

以上をまとめると、ガウス差分関数を用いた MAPLP 拡張手法の最適化問題は、次の線形計画問題に帰着する。

$$\max_{\mathbf{z}} \quad \langle \mathbf{c}, \mathbf{z} \rangle \quad (3.94)$$

$$\text{s.t.} \quad \langle \mathbf{a}, \mathbf{z} \rangle = 1 \quad (3.95)$$

$$\mathbf{A}\mathbf{z} \leq \mathbf{1}_{2N} \quad (3.96)$$

$$\mathbf{z} \geq \mathbf{0}_{3N} \quad (3.97)$$

$$\mathbf{c} = [P(0)\mathbf{d}_0^T \mid P(1)\mathbf{d}_1^T \mid -(P(0)\mathbf{d}_0 + P(1)\mathbf{d}_1)^T]^T \quad (3.98)$$

$$\mathbf{a} = [(P(0)\mathbf{d}_0 + P(1)\mathbf{d}_1)^T \mid (P(0)\mathbf{d}_0 + P(1)\mathbf{d}_1)^T \mid \mathbf{0}_N^T]^T \quad (3.99)$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{K} & \mathbf{0}_{N,N} & -\mathbf{K} \\ \mathbf{0}_{N,N} & \mathbf{K} & -\mathbf{K} \end{bmatrix} \quad (3.100)$$

$$\mathbf{d}_y = (d_{y,j}) \in \mathbb{R}^{N \times 1} \quad (3.101)$$

$$d_{y,j} = \left(1 + \frac{\kappa}{\nu^d - 1}\right) \frac{1}{|\mathbf{I} + 2\gamma\nu^2\boldsymbol{\Sigma}_y|^{1/2}} \exp\left(\gamma\langle(\boldsymbol{\mu}_y - \mathbf{x}_j), (\mathbf{I} + 2\gamma\nu^2\boldsymbol{\Sigma}_y)^{-1}(\boldsymbol{\mu}_y - \mathbf{x}_j)\rangle\right) - \frac{\kappa}{\nu^d - 1} \frac{1}{|\mathbf{I} + 2\gamma\boldsymbol{\Sigma}_y|^{1/2}} \exp\left(\gamma\langle(\boldsymbol{\mu}_y - \mathbf{x}_j), (\mathbf{I} + 2\gamma\boldsymbol{\Sigma}_y)^{-1}(\boldsymbol{\mu}_y - \mathbf{x}_j)\rangle\right) \quad (3.102)$$

3.1.4 制約追加法

3.1.1-3.1.3 節では、制約条件 (3.5) を標本 $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ のみで評価した。本節では、制約条件 (3.5) を厳密に評価するために、式 (3.5) の左辺で定義される関数 $f_y(\mathbf{x})$ の極大値をとる点を求め、これを新たな評価点として追加する。

$$f_y(\mathbf{x}) = w(\mathbf{x}, y) - \xi(\mathbf{x}), \quad (y = 0, 1) \quad (3.103)$$

$$= \sum_{j=1}^N (\alpha_{y,j} - \beta_j) k(\mathbf{x}, \mathbf{x}_j), \quad (y = 0, 1) \quad (3.104)$$

$$= \sum_{j=1}^N (\alpha_{y,j} - \beta_j) \exp(\gamma\|\mathbf{x} - \mathbf{x}_j\|^2), \quad (y = 0, 1) \quad (3.105)$$

関数 $f_y(\mathbf{x})$ の勾配は

$$\nabla_{\mathbf{x}} f_y(\mathbf{x}) = \sum_{j=1}^N -2\gamma(\alpha_{y,j} - \beta_j) \exp(-\gamma\|\mathbf{x} - \mathbf{x}_j\|^2)(\mathbf{x} - \mathbf{x}_j) \quad (3.106)$$

となる。各標本点 $\mathbf{x}_1, \dots, \mathbf{x}_N$ から勾配法で極大値を求め、極大値をとる点 \mathbf{x}' において

$$f_y(\mathbf{x}') > 1 \quad (3.107)$$

であるとき、制約

$$\sum_{j=1}^N (\alpha_{y,j} - \beta_j) k(\mathbf{x}', \mathbf{x}_j) \leq 1, \quad (y = 0, 1) \quad (3.108)$$

を追加し、再び最適化問題を解く。制約追加法を適用した確率密度推定による拡張手法の最適化アルゴリズムを Algorithm 1 に示す。

Algorithm 1 制約追加法を適用した確率密度推定による拡張手法

Require: 訓練標本 $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, カーネルパラメータ: γ , モデルパラメータ: σ (KDE 拡張法), ν, κ (DoG 拡張法)

Ensure: 解 $\mathbf{z} = [\boldsymbol{\alpha}_0^T, \boldsymbol{\alpha}_1^T, \boldsymbol{\beta}^T]$

- 1: 標本から各クラスの事前確率 $P(y=0), P(y=1)$ を推定する
- 2: 目的関数の係数ベクトル \mathbf{c} を求める.
- 3: 制約条件 (3.2) の係数ベクトル \mathbf{a} を求める.
- 4: カーネルグラム行列 \mathbf{K} を求める
- 5: 制約条件 (3.5) の係数行列 \mathbf{A} を求める
- 6: **for** $i = 1$ to i_{max} **do**
- 7: 線形計画問題

$$\begin{aligned} \max_{\mathbf{z}} \quad & \langle \mathbf{c}, \mathbf{z} \rangle \\ \text{s.t.} \quad & \langle \mathbf{a}, \mathbf{z} \rangle = 1 \\ & \mathbf{A}\mathbf{z} \leq \mathbf{1}_{2N} \\ & \mathbf{z} \geq \mathbf{0}_{3N} \end{aligned}$$

を解く.

- 8: 各標本点 $\mathbf{x}_1, \dots, \mathbf{x}_N$ を初期値として, 勾配法により関数

$$f_y(\mathbf{x}) = w(\mathbf{x}, y) - \xi(\mathbf{x}), \quad (y = 0, 1)$$

の極大点 $\mathbf{z}_t (t = 1, \dots, T)$ を求める.

- 9: **for** $t = 1$ to T **do**
- 10: endflag = true
- 11: **if** $w(\mathbf{z}_t, y) - \xi(\mathbf{x}) > 1$ **then**
- 12: endflag = false
- 13: ベクトル $\mathbf{k}(\mathbf{z}_t) = [k(\mathbf{z}_t, \mathbf{x}_1), \dots, k(\mathbf{z}_t, \mathbf{x}_N)]^T$ を求め, 制約を追加する.

$$\mathbf{K} \leftarrow \begin{bmatrix} \mathbf{K} \\ \mathbf{k}(\mathbf{z}_t)^T \end{bmatrix}, \quad \mathbf{A} \leftarrow \begin{bmatrix} \mathbf{K} & \mathbf{0}_{N,N} & -\mathbf{K} \\ \mathbf{0}_{N,N} & \mathbf{K} & -\mathbf{K} \end{bmatrix}$$

- 14: **end if**
 - 15: **end for**
 - 16: **if** endflag==ture **then**
 - 17: アルゴリズムを終了し, 解を得る.
 - 18: **end if**
 - 19: **end for**
-

3.2 マルチカーネル学習

3.1節では、各クラスにおける入力データの条件付き確率密度関数 $p(\mathbf{x}|y)$ の推定による拡張手法について説明した。本節では、これらの拡張手法に対してマルチカーネル学習を適用した手法を提案する。2.5.2節で述べたように、カーネル関数を複数のカーネル関数の線形和、つまり

$$k(\mathbf{x}, \mathbf{z}) = \sum_{m=1}^M \eta_m k_m(\mathbf{x}, \mathbf{z}) \quad (3.109)$$

で表現する。 η_m は各カーネル関数の重みパラメータ、 M は単カーネルの総数を表す。提案手法におけるカーネル関数を、RBFカーネル (3.16) を用いて

$$k(\mathbf{x}, \mathbf{z}) = \sum_{m=1}^M \eta_m \exp(-\gamma_m \|\mathbf{x} - \mathbf{z}\|^2) \quad (3.110)$$

と定義する。

3.1節では、式 (3.6), (3.7) の重みパラメータ $\alpha_0, \alpha_1, \beta$ に関する最適化問題を定義した。これらの最適化問題に対して式 (3.109) で定義されるカーネル関数を導入すると、 η に関する最適化問題に変形することができる。そこで、マルチカーネル学習適用法では、始めに η を固定し $\alpha_0, \alpha_1, \beta$ に関する最適化問題を解き、次に $\alpha_0, \alpha_1, \beta$ を固定し η に関する最適化問題を解く、という交互最適化により解を得る。

3.2.1 カーネル密度関数を用いた拡張手法への適用法

3.1.1節に関して、マルチカーネル関数 (3.110) を導入することを考える。行列 B_m, B およびカーネルグラム行列 K を

$$[B_m]_{i,j} = \int \phi(\mathbf{x} - \mathbf{x}_i) k_m(\mathbf{x}, \mathbf{x}_j) d\mathbf{x} \quad (3.111)$$

$$= \frac{1}{(1 + 2\gamma_m \sigma^2)^{d/2}} \int \exp\left(-\frac{\gamma_m}{1 + 2\gamma_m \sigma^2} \|\mathbf{x}_i - \mathbf{x}_j\|^2\right) \quad (3.112)$$

$$B = \sum_{m=1}^M \eta_m B_m \quad (3.113)$$

$$K_m = \begin{bmatrix} k_m(\mathbf{x}_1, \mathbf{x}_1) & \dots & k_m(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ k_m(\mathbf{x}_N, \mathbf{x}_1) & \dots & k_m(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \quad (3.114)$$

$$K = \sum_{m=1}^M \eta_m K_m \quad (3.115)$$

と定義すると、 $\mathbf{z} = [\alpha_0^T, \alpha_1^T, \beta^T]^T$ に関する最適化問題は式 (3.46)-(3.53) と同様に与えられる。

一方で、 η に関する目的関数は

$$J = \sum_{m=1}^M \eta_m \left[\frac{P(0)}{N_0} (\mathbf{1}_N - \mathbf{y})^T B_m (\alpha_0 - \beta) + \frac{P(1)}{N_1} \mathbf{y}^T B_m (\alpha_1 - \beta) \right] \quad (3.116)$$

である。また制約 (3.2) を $\boldsymbol{\eta}$ に関して変形すると、式 (3.33), (3.34) より

$$\sum_{m=1}^M \eta_m \left[\left(\frac{P(0)}{N_0} \mathbf{1}_N + \left(\frac{P(1)}{N_1} - \frac{P(0)}{N_0} \right) \mathbf{y} \right)^T \mathbf{B}_m(\boldsymbol{\alpha}_0 + \boldsymbol{\alpha}_1) \right] = 1 \quad (3.117)$$

と求まる。次に、制約条件 (3.5) について考える。式 (3.42), (3.43) にカーネル関数 (3.109) を代入すると

$$\sum_m^M \eta_m \sum_{j=1}^N (\alpha_{0,j} - \beta_j) k_m(\mathbf{x}, \mathbf{x}_j) \leq 1 \quad (3.118)$$

$$\sum_m^M \eta_m \sum_{j=1}^N (\alpha_{1,j} - \beta_j) k_m(\mathbf{x}, \mathbf{x}_j) \leq 1 \quad (3.119)$$

となる。式 (3.115) を用いれば

$$\sum_m^M \eta_m \left[\mathbf{K}_m(\boldsymbol{\alpha}_0 - \boldsymbol{\beta}) \right] \leq \mathbf{1}_N \quad (3.120)$$

$$\sum_m^M \eta_m \left[\mathbf{K}_m(\boldsymbol{\alpha}_1 - \boldsymbol{\beta}) \right] \leq \mathbf{1}_N \quad (3.121)$$

と表される。

最後に、 $\boldsymbol{\eta}$ 関する制約として、

$$\eta_m \geq 0, \quad (m = 1, \dots, M) \quad (3.122)$$

を課す。これに加えて、 $\boldsymbol{\eta}$ が一意に定まるように

$$\sum_{m=1}^M \eta_m = 1 \quad (3.123)$$

を課す。

以上をまとめると、 $\boldsymbol{\eta}$ に関する最適化問題は、次の線形計画問題に帰着する。

$$\max_{\boldsymbol{\eta}} \quad \langle \mathbf{c}_{\boldsymbol{\eta}}, \boldsymbol{\eta} \rangle \quad (3.124)$$

$$\text{s.t.} \quad \langle \mathbf{a}_{\boldsymbol{\eta}}, \boldsymbol{\eta} \rangle = 1 \quad (3.125)$$

$$\mathbf{A}_{\boldsymbol{\eta}} \boldsymbol{\eta} \leq \mathbf{1}_{2M} \quad (3.126)$$

$$\mathbf{1}_M^T \boldsymbol{\eta} = 1 \quad (3.127)$$

$$\boldsymbol{\eta} \geq \mathbf{0}_M \quad (3.128)$$

$$\begin{aligned} \mathbf{c}_\eta &= \left[\frac{P(0)}{N_0} (\mathbf{1}_N - \mathbf{y})^T \mathbf{B}_1 (\boldsymbol{\alpha}_0 - \boldsymbol{\beta}) + \frac{P(1)}{N_1} \mathbf{y}^T \mathbf{B}_1 (\boldsymbol{\alpha}_1 - \boldsymbol{\beta}) \right] \cdots \\ &\quad \cdots \left[\frac{P(0)}{N_0} (\mathbf{1}_N - \mathbf{y})^T \mathbf{B}_M (\boldsymbol{\alpha}_0 - \boldsymbol{\beta}) + \frac{P(1)}{N_1} \mathbf{y}^T \mathbf{B}_M (\boldsymbol{\alpha}_1 - \boldsymbol{\beta}) \right]^T \end{aligned} \quad (3.129)$$

$$\begin{aligned} \mathbf{a}_\eta &= \left[\left(\frac{P(0)}{N_0} (\mathbf{1}_N - \mathbf{y}) + \frac{P(1)}{N_1} \mathbf{y} \right)^T \mathbf{B}_1 (\boldsymbol{\alpha}_0 + \boldsymbol{\alpha}_1) \right] \cdots \\ &\quad \cdots \left[\left(\frac{P(0)}{N_0} (\mathbf{1}_N - \mathbf{y}) + \frac{P(1)}{N_1} \mathbf{y} \right)^T \mathbf{B}_M (\boldsymbol{\alpha}_0 + \boldsymbol{\alpha}_1) \right]^T \end{aligned} \quad (3.130)$$

$$\mathbf{A}_\eta = \begin{bmatrix} A_{\eta,0} \\ A_{\eta,1} \end{bmatrix} \quad (3.131)$$

$$\mathbf{A}_{\eta,0} = \left[K_1(\boldsymbol{\alpha}_0 - \boldsymbol{\beta}) \mid \cdots \mid K_M(\boldsymbol{\alpha}_0 - \boldsymbol{\beta}) \right] \quad (3.132)$$

$$\mathbf{A}_{\eta,1} = \left[K_1(\boldsymbol{\alpha}_1 - \boldsymbol{\beta}) \mid \cdots \mid K_M(\boldsymbol{\alpha}_1 - \boldsymbol{\beta}) \right] \quad (3.133)$$

3.2.2 ガウス関数を用いた拡張手法への適用法

3.1.2節に関して、マルチカーネル関数 (3.110) を導入することを考える。ベクトル $\mathbf{g}_{y,m}$, \mathbf{g}_y を

$$[\mathbf{g}_{y,m}]_j = \int p(\mathbf{x}|y) k(\mathbf{x}, \mathbf{x}_j) d\mathbf{x} \quad (3.134)$$

$$= \frac{1}{|\mathbf{I} + 2\gamma_m \boldsymbol{\Sigma}_y|^{1/2}} \exp \left(\gamma_m \langle (\boldsymbol{\mu}_y - \mathbf{x}_j), (\mathbf{I} + 2\gamma_m \boldsymbol{\Sigma}_y)^{-1} (\boldsymbol{\mu}_y - \mathbf{x}_j) \rangle \right) \quad (3.135)$$

$$\mathbf{g}_y = \sum_{m=1}^M \eta_m \mathbf{g}_{y,m} \quad (3.136)$$

と定義し、式 (3.114)-(3.115) で定義されるカーネルグラム行列を用いると、 $\mathbf{z} = [\boldsymbol{\alpha}_0^T, \boldsymbol{\alpha}_1^T, \boldsymbol{\beta}^T]^T$ に関する最適化問題は式 (3.72)-(3.78) と同様に与えられる。

一方で、 $\boldsymbol{\eta}$ に関する目的関数は

$$J = \sum_{m=1}^M \eta_m \left[P(0) \mathbf{g}_{0,m}^T (\boldsymbol{\alpha}_0 - \boldsymbol{\beta}) + P(1) \mathbf{g}_{1,m}^T (\boldsymbol{\alpha}_1 - \boldsymbol{\beta}) \right] \quad (3.137)$$

である。また、制約 (3.2) を $\boldsymbol{\eta}$ に関して変形すると、式 (3.71) より

$$\sum_{m=1}^M \eta_m \left[(P(0) \mathbf{g}_{0,m} + P(1) \mathbf{g}_{1,m})^T (\boldsymbol{\alpha}_0 + \boldsymbol{\alpha}_1) \right] \quad (3.138)$$

と求まる。制約 (3.5) および $\boldsymbol{\eta}$ に関する制約は、3.2.1節と同様に式 (3.120)-(3.121) および (3.122)-(3.123) を課す。

以上をまとめると、 $\boldsymbol{\eta}$ に関する最適化問題は式 (3.124)-(3.133) に帰着する。ただし、 \mathbf{c}_η , \mathbf{a}_η は

次式で定義される.

$$\begin{aligned} \mathbf{c}_\eta = & \left[P(0)\mathbf{g}_{0,1}^T(\boldsymbol{\alpha}_0 - \boldsymbol{\beta}) + P(1)\mathbf{g}_{1,1}^T(\boldsymbol{\alpha}_1 - \boldsymbol{\beta}) \middle| \cdots \right. \\ & \left. \cdots \middle| P(0)\mathbf{g}_{0,M}^T(\boldsymbol{\alpha}_0 - \boldsymbol{\beta}) + P(1)\mathbf{g}_{1,M}^T(\boldsymbol{\alpha}_1 - \boldsymbol{\beta}) \right]^T \end{aligned} \quad (3.139)$$

$$\begin{aligned} \mathbf{a}_\eta = & \left[(P(0)\mathbf{g}_{0,1} + P(1)\mathbf{g}_{1,1})^T(\boldsymbol{\alpha}_0 + \boldsymbol{\alpha}_1) \middle| \cdots \right. \\ & \left. \cdots \middle| (P(0)\mathbf{g}_{0,M} + P(1)\mathbf{g}_{1,M})^T(\boldsymbol{\alpha}_0 + \boldsymbol{\alpha}_1) \right]^T \end{aligned} \quad (3.140)$$

3.2.3 ガウス差分関数を用いた拡張手法への適用法

3.1.3 節に関して, マルチカーネル関数 (3.110) を導入することを考える. 式 (3.134)-(3.135) を Σ_y の関数として

$$[\mathbf{g}_{y,m}(\Sigma_y)]_j = \frac{1}{|\mathbf{I} + 2\gamma_m \Sigma_y|^{1/2}} \exp\left(\gamma_m \langle (\boldsymbol{\mu}_y - \mathbf{x}_j), (\mathbf{I} + 2\gamma_m \Sigma_y)^{-1}(\boldsymbol{\mu}_y - \mathbf{x}_j) \rangle\right) \quad (3.141)$$

とする. これを用いて, ベクトル $\mathbf{d}_{y,m}$, \mathbf{d}_y を

$$[\mathbf{d}_{y,m}]_j = \int p(\mathbf{x}|y)k(\mathbf{x}, \mathbf{x}_j)d\mathbf{x} \quad (3.142)$$

$$= \left(1 + \frac{\kappa}{\nu^d - 1}\right)[\mathbf{g}_{y,m}(\nu^2 \Sigma_y)]_j - \frac{\kappa}{\nu^d - 1}[\mathbf{g}_{y,m}(\Sigma_y)]_j \quad (3.143)$$

$$\mathbf{d}_y = \sum_{m=1}^M \eta_m \mathbf{d}_{y,m} \quad (3.144)$$

と定義し, 式 (3.114)-(3.115) で定義されるカーネルグラム行列を用いると, $\mathbf{z} = [\boldsymbol{\alpha}_0^T, \boldsymbol{\alpha}_1^T, \boldsymbol{\beta}^T]^T$ に関する最適化問題は式 (3.94)-(3.100) と同様に与えられる.

一方で, $\boldsymbol{\eta}$ に関する目的関数は

$$J = \sum_{m=1}^M \eta_m \left[P(0)\mathbf{d}_{0,m}^T(\boldsymbol{\alpha}_0 - \boldsymbol{\beta}) + P(1)\mathbf{d}_{1,m}^T(\boldsymbol{\alpha}_1 - \boldsymbol{\beta}) \right] \quad (3.145)$$

である. また, 制約 (3.2) を $\boldsymbol{\eta}$ に関して変形すると, 式 (3.71) より

$$\sum_{m=1}^M \eta_m \left[(P(0)\mathbf{d}_{0,m} + P(1)\mathbf{d}_{1,m})^T(\boldsymbol{\alpha}_0 + \boldsymbol{\alpha}_1) \right] \quad (3.146)$$

と求まる. 制約 (3.5) および $\boldsymbol{\eta}$ に関する制約は, 3.2.1 節と同様に式 (3.120)-(3.121) および (3.122)-(3.123) を課す.

以上をまとめると, $\boldsymbol{\eta}$ に関する最適化問題は式 (3.124)-(3.133) に帰着する. ただし, \mathbf{c}_η , \mathbf{a}_η は

次式で定義される.

$$\begin{aligned} \mathbf{c}_\eta &= \left[P(0)\mathbf{d}_{0,1}^T(\boldsymbol{\alpha}_0 - \boldsymbol{\beta}) + P(1)\mathbf{d}_{1,1}^T(\boldsymbol{\alpha}_1 - \boldsymbol{\beta}) \mid \cdots \right. \\ &\quad \left. \cdots \mid P(0)\mathbf{d}_{0,M}^T(\boldsymbol{\alpha}_0 - \boldsymbol{\beta}) + P(1)\mathbf{d}_{1,M}^T(\boldsymbol{\alpha}_1 - \boldsymbol{\beta}) \right]^T \end{aligned} \quad (3.147)$$

$$\begin{aligned} \mathbf{a}_\eta &= \left[(P(0)\mathbf{d}_{0,1} + P(1)\mathbf{d}_{1,1})^T(\boldsymbol{\alpha}_0 + \boldsymbol{\alpha}_1) \mid \cdots \right. \\ &\quad \left. \cdots \mid (P(0)\mathbf{d}_{0,M} + P(1)\mathbf{d}_{1,M})^T(\boldsymbol{\alpha}_0 + \boldsymbol{\alpha}_1) \right]^T \end{aligned} \quad (3.148)$$

3.2.4 制約追加法

3.1.4 節と同様に, 制約条件 (3.5) を厳密に評価するために, 式 (3.5) の左辺で定義される関数 $f_y(\mathbf{x})$ の極大値をとる点を求め, これを新たな評価点として追加する.

$$f_y(\mathbf{x}) = w(\mathbf{x}, y) - \xi(\mathbf{x}), \quad (y = 0, 1) \quad (3.149)$$

$$= \sum_{m=1}^M \eta_m \sum_{j=1}^N (\alpha_{y,j} - \beta_j) k_m(\mathbf{x}, \mathbf{x}_j), \quad (y = 0, 1) \quad (3.150)$$

$$= \sum_{m=1}^M \eta_m \sum_{j=1}^N (\alpha_{y,j} - \beta_j) \exp(\gamma_m \|\mathbf{x} - \mathbf{x}_j\|^2), \quad (y = 0, 1) \quad (3.151)$$

関数 $f_y(\mathbf{x})$ の勾配は

$$\nabla_{\mathbf{x}} f_y(\mathbf{x}) = \sum_{j=1}^N -2(\alpha_{y,j} - \beta_j) \sum_{m=1}^M \eta_m \gamma_m \exp(-\gamma_m \|\mathbf{x} - \mathbf{x}_j\|^2) (\mathbf{x} - \mathbf{x}_j) \quad (3.152)$$

となる. 各標本点 $\mathbf{x}_1, \dots, \mathbf{x}_N$ から勾配法で極大値を求め, 極大値をとる点 \mathbf{x}' において

$$f_y(\mathbf{x}') > 1 \quad (3.153)$$

であるとき, 制約

$$\sum_{m=1}^M \eta_m \sum_{j=1}^N (\alpha_{y,j} - \beta_j) k_m(\mathbf{x}', \mathbf{x}_j) \leq 1, \quad (y = 0, 1) \quad (3.154)$$

を追加し, 再び最適化問題を解く. 約追加法を適用したマルチカーネル学習適用手法の最適化アルゴリズムを Alorithm 2 に示す.

Algorithm 2 制約追加法を適用したマルチカーネル学習適用手法

Require: 訓練標本 $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, カーネルパラメータの候補群: $\gamma_1, \dots, \gamma_M$, モデルパラメータ: σ (KDE 拡張法), ν, κ (DoG 拡張法)

Ensure: 解 $\mathbf{z} = [\boldsymbol{\alpha}_0^T, \boldsymbol{\alpha}_1^T, \boldsymbol{\beta}^T]$, $\boldsymbol{\eta}$

- 1: 標本から各クラスの事前確率 $P(y=0), P(y=1)$ を推定する
- 2: $\boldsymbol{\eta}$ の初期値を $\eta_m = 1/M, m = 1, \dots, M$ と設定する.
- 3: **for** $i_1 = 1$ to i_1^{max} **do**
- 4: # $\boldsymbol{\eta}$ を固定して, $\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \boldsymbol{\beta}$ を更新する.
- 5: 目的関数の係数ベクトル \mathbf{c} を求める.
- 6: 制約条件 (3.2) の係数ベクトル \mathbf{a} を求める.
- 7: カーネルグラム行列 \mathbf{K} を求める
- 8: 制約条件 (3.5) の係数行列 \mathbf{A} を求める
- 9: **for** $i_2 = 1$ to i_2^{max} **do**
- 10: 線形計画問題

$$\begin{aligned} \max_{\mathbf{z}} \quad & \langle \mathbf{c}, \mathbf{z} \rangle \\ \text{s.t.} \quad & \langle \mathbf{a}, \mathbf{z} \rangle = 1 \\ & \mathbf{A}\mathbf{z} \leq \mathbf{1}_{2N} \\ & \mathbf{z} \geq \mathbf{0}_{3N} \end{aligned}$$

を解く.

- 11: 各標本点 $\mathbf{x}_i (i = 1, \dots, N)$ を初期値として, 勾配法により関数

$$f_y(\mathbf{x}) = w(\mathbf{x}, y) - \xi(\mathbf{x}), \quad (y = 0, 1)$$

の極大点 $\mathbf{z}_t (t = 1, \dots, T)$ を求める.

- 12: **for** $t = 1$ to T **do**
- 13: endflag = true
- 14: **if** $w(\mathbf{z}_t, y) - \xi(\mathbf{x}) > 1$ **then**
- 15: endflag = false
- 16: ベクトル $\mathbf{k}(\mathbf{z}_t) = [k(\mathbf{z}_t, \mathbf{x}_1), \dots, k(\mathbf{z}_t, \mathbf{x}_N)]^T$ を求め, 制約を追加する.

$$\mathbf{K} \leftarrow \begin{bmatrix} \mathbf{K} \\ \mathbf{k}(\mathbf{z}_t)^T \end{bmatrix}, \quad \mathbf{A} \leftarrow \begin{bmatrix} \mathbf{K} & \mathbf{0}_{N,N} & -\mathbf{K} \\ \mathbf{0}_{N,N} & \mathbf{K} & -\mathbf{K} \end{bmatrix}$$

- 17: **end if**
 - 18: **end for**
 - 19: **if** endflag==ture **then**
 - 20: i_2 についての For を抜ける.
 - 21: **end if**
 - 22: **end for** i_2
-

```

23: #  $\alpha_0, \alpha_1, \beta$  を固定して,  $\eta$  を更新する.
24: 目的関数の係数ベクトル  $\mathbf{c}_\eta$  を求める.
25: 制約条件 (3.2) の係数ベクトル  $\mathbf{a}_\eta$  を求める.
26: カーネルグラム行列  $\mathbf{K}_m (m = 1, \dots, M)$  を求める.
27: 制約条件 (3.5) の係数行列  $\mathbf{A}_\eta$  を求める
28: for  $i_3 = 1$  to  $i_3^{max}$  do
29:     線形計画問題

```

$$\begin{aligned}
& \max_{\eta} && \langle \mathbf{c}_\eta, \eta \rangle \\
& \text{s.t.} && \langle \mathbf{a}_\eta, \eta \rangle = 1 \\
& && \mathbf{A}_\eta \eta \leq \mathbf{1}_{2M} \\
& && \mathbf{1}_M^T \eta = 1 \\
& && \eta \geq \mathbf{0}_M
\end{aligned}$$

を解く.

```

30: 各標本点  $\mathbf{x}_i (i = 1, \dots, N)$  を初期値として, 勾配法により関数

```

$$f_y(\mathbf{x}) = w(\mathbf{x}, y) - \xi(\mathbf{x}), \quad (y = 0, 1)$$

の極大点 $\mathbf{z}_t (t = 1, \dots, T)$ を求める.

```

31: for  $t = 1$  to  $T$  do
32:     endflag = true
33:     if  $w(\mathbf{z}_t, y) - \xi(\mathbf{x}) > 1$  then
34:         endflag = false
35:         ベクトル  $\mathbf{k}_m(\mathbf{z}_t) = [k(\mathbf{z}_t, \mathbf{x}_1), \dots, k(\mathbf{z}_t, \mathbf{x}_N)]^T, (m = 1, \dots, M)$  を求め,
            カーネルグラム行列  $\mathbf{K}_m, (m = 1, \dots, M)$  を更新し, 制約を追加する.

```

$$\mathbf{K}_m \leftarrow \begin{bmatrix} \mathbf{K}_m \\ \mathbf{k}_m(\mathbf{z}_t)^T \end{bmatrix}, \quad (m = 1, \dots, M)$$

$$\mathbf{A}_{\eta,0} \leftarrow \left[K_1(\alpha_0 - \beta) \mid \dots \mid K_M(\alpha_0 - \beta) \right]$$

$$\mathbf{A}_{\eta,1} \leftarrow \left[K_1(\alpha_1 - \beta) \mid \dots \mid K_M(\alpha_1 - \beta) \right]$$

$$\mathbf{A}_\eta \leftarrow \begin{bmatrix} \mathbf{A}_{\eta,0} \\ \mathbf{A}_{\eta,1} \end{bmatrix}$$

```

36:         end if
37:     end for
38:     if endflag == true then
39:          $i_3$  についての For を抜ける.
40:     end if
41: end for  $i_3$ 
42: if  $\alpha_0, \alpha_1, \beta, \eta$  が収束する then
43:     アルゴリズムを終了し, 解を得る
44: end if
45: end for

```

第4章 計算機実験

4.1 オープンデータベースを用いた2クラス識別問題

4.1.1 実験環境

データセット 識別実験を行うために、文献 [22, 23] で利用されている2クラス識別問題を用いた。これらは、UCI Machine Learning Repository, DELVE Datasets および STATLOG のオープンデータベースから作成された13種類のデータセット “banana”, “breast-cancer”, “diabetes”, “flare-solar”, “german”, “heart”, “image”, “ringworm”, “splice”, “thyroid”, “titanic”, “twonorm”, “waveform” から、100あるいは20の学習用、試験用の組を作成したものである。いくつかのデータセットは2クラス問題ではないためランダムに2クラスに分離されている。識別問題の属性を表4.1に示す。

ソルバ 線形計画問題のソルバとしてIBM社のCPLEXを用いた [24]。

パラメータ カーネル関数としてRBFカーネル (3.16) を用いた。カーネルパラメータ γ は各識別問題に対して

$$\log_{10} \gamma \in \{-2.0, -1.9, \dots, 2.0\} \quad (4.1)$$

から最適値を選択した。マルチカーネル学習におけるカーネルパラメータ $\gamma_m, (m = 1, \dots, M)$ は

$$\log_{10} \gamma_m \in \{-2.0, -1.8, -1.6, \dots, 2.0\} \quad (4.2)$$

を用いた。3.1.1-3.2.1節でカーネル密度推定に用いる分散パラメータ σ も同様に

$$\log_{10} \sigma \in \{-2.0, -1.9, \dots, 2.0\} \quad (4.3)$$

から選択した。3.1.3-3.2.3節でガウス差分関数の尖度を定めるパラメータ κ および、極の広がりを定めるパラメータ ν は [25] を参考した。 κ は $\kappa = 0.9$ で固定し、 ν は

$$\log_{10} \nu \in \{0.1, 0.2, \dots, 1\} \quad (4.4)$$

から選択した。これらのパラメータの最適値は各識別問題に対して、データセット No.1 を用いた5分割交差検定で決定した。

4.1.2 実験結果

実験1：確率密度推定を用いた拡張手法の比較実験 「確率密度推定を適用した拡張手法」と「確率密度推定および制約追加法を適用した拡張手法」の比較実験の結果を示す。各識別問題に対して、データセット No.1 を用いた5分割交差検定で決定した最適パラメータを表4.2に示す。全デー

表 4.1: 識別問題の属性

識別問題	学習標本数	テスト標本数	次元数	データセット数
Banana	400	4900	2	100
Breast-cancer	200	77	9	100
Diabetis	268	300	8	100
Flare-Solar	666	400	9	100
German	700	300	20	100
Heart	170	100	13	100
Image	1300	1010	18	20
Ringnorm	400	7000	20	100
Splice	1000	2175	60	20
Thyroid	140	75	5	100
Titanic	150	2051	3	100
Twonorm	400	7000	20	100
Waveform	400	4600	21	100

タセットを用いた識別実験の平均識別率と標準偏差を表 4.3 に示す。表中で、最も識別率の高い結果を Bold font で表し、次に高い結果に under line を付して表した。各手法の平均識別率に対し Nemenyi 検定を行った結果を表 4.4 に示す。表中で、“1” は行方向の手法が列方向の手法よりも有意差に識別性能が高いことを示し、“0” は識別性能に有意差が無かったことを示す。また、“-1” は行方向の手法が列方向の手法よりも有意差に識別性能が低いことを示す。

実験 2：マルチカーネル学習を適用した拡張手法の性能評価実験 「マルチカーネル学習を適用した拡張手法」と「マルチカーネル学習および制約追加法を適用した拡張手法」の比較実験の結果を示す。各識別問題に対して、データセット No.1 を用いた 5 分割交差検定で決定した最適パラメータを表 4.5 に示す。全データセットを用いた識別実験の平均識別率と標準偏差を表 4.6 に示した。表中で、最も識別率の高い結果を Bold font で表し、次に高い結果に under line を付して表した。各手法の平均識別率に対し Nemenyi 検定を行った結果を表 4.7 に示す。表中で、“1” は行方向の手法が列方向の手法よりも有意差に識別性能が高いことを示し、“0” は識別性能に有意差が無かったことを示す。また、“-1” は行方向の手法が列方向の手法よりも有意差に識別性能が低いことを示す。

4.1.3 考察

実験 1：確率密度推定を用いた拡張手法の性能評価実験 KDE 拡張法が最も bold font が多い結果となった。その次に、制約追加法を適用した KDE 拡張法が MAPLP よりも高い識別性能をもつ結果となった。Gauss 拡張法および DoG 拡張法は、ほとんどの識別問題で MAPLP を下回る結果となった。すなわち、本実験において、各クラスにおける入力データの確率密度関数 $p(x|y)$ をノンパラメトリック手法で推定する拡張手法が、パラメトリック手法で推定する手法よりも全体的に高い識別性能を示した。

識別問題 “splice” の結果に注目すると、Gauss 拡張法および DoG 拡張法は MAPLP よりも 5% 以上高い識別性能を示している。これは他の識別問題と比較して大きな性能向上を示した結果である。表 4.1 より “splice” は他の識別問題と比較して、高次元の特徴をもつ問題であることがわかる。この結果より、Gauss 拡張法および DoG 拡張法は相対的に高次元の特徴をもつ識別問題に対して、MAPLP よりも高い識別性能をもつ可能性があると考えられる。もちろん、本実験では、高次元の特徴をもつ識別問題が “splice” のみであるため、一概に上記の仮定が成り立つと言えるわけではない。今後、追加実験を行い検討する必要があると考えられる。

Nemenyi 検定の結果から、KDE 拡張法は Gauss 拡張法および DoG 拡張法よりも有意に識別性能が高いことが分かった。また、MAPLP は DoG 拡張法よりも有意に識別性能が高いことが分かった。KDE 拡張法と MAPLP には識別性能の有意差がないことを確認した。

訓練標本数の違いにおける識別性能を評価するため、表 4.1 における訓練標本数を 20%, 40%, 60%, 80% に削減した場合の識別実験を追加で行った。その結果を図 4.3 に示す。KDE 拡張法では “heart”, “ringnorm” において MAPLP よりも訓練標本数に対する識別率の減少度が大きいことがわかった。Gauss 拡張法ではすべての識別問題において訓練標本数に対する識別率の減少度は MAPLP と同程度であった。DoG 拡張法では、表 4.3 で高い識別性能を示した “splice” において、60% を境に急激に識別性能が低下した。しかし、“image”, “ringnorm”, “titanic”, “waveform” のように、訓練標本数が低下すると識別性能が向上するものもあった。この振る舞いは、従来手法や他の拡張法にはないものであった。図 4.3 の結果より、3.1 節で述べたように、確率密度関数 $p(\mathbf{x}|y)$ をモデル化により相対的に少ない標本数に対して、提案手法が従来手法と比べて性能向上するとは言えないことが分かった。

実験 2: マルチカーネル学習を適用した拡張手法の性能評価実験 MAPLP が最も bold font が多く、KDE 拡張法が最も under line が多い結果となった。Gauss 拡張法はすべての識別問題で MAPLP を下回る結果となった。DoG 拡張法は、識別問題 “splice” の結果に注目すると MAPLP よりも 4% 以上高い識別性能を示している。これはマルチカーネル学習を適用して唯一従来手法を上回る結果である。3.1.3 節に述べたように、DoG 拡張法は各クラスにおける入力データの確率密度関数 $p(\mathbf{x}|y)$ をガウス差分関数でモデル化することで、クラス境界付近のエラーデータの評価をガウスモデルよりも厳しき行うように設計した手法である。マルチカーネル学習適用手法において Gauss 拡張法と比較すると、10 種の識別問題で DoG モデルの効果が得られていると考えられる。

Nemenyi 検定の結果から、MAPLP は Gauss 拡張法および DoG 拡張法よりも有意に識別性能が高いことが分かった。これは、表 4.6 の結果と明らかに一致する結果となった。

カーネルの重みパラメータの初期値の違いを評価するために、単カーネルで用いた最適パラメータ (表 4.2) を基に初期値を調整した場合の識別実験を行い、この結果を表 4.9 に示した。表 4.9 から、初期値を単カーネル手法の最適値で調整した場合の KDE 拡張法が最も bold font が多い結果となった。一方で、表 4.6 にも示したように、初期値を一定値とした場合では、多くの識別問題で識別性能が低下した。この結果から、マルチカーネル学習適用法の識別性能はカーネル関数の重みパラメータの初期値に大きく依存している、と考えられる。

表 4.5 から分かるように、マルチカーネル学習の適用により交差検定によるカーネルパラメータの最適値推定が不要になった。これにより、計算時間が削減できると考えられる。本実験では、計算時間の厳密な測定を行わなかった。今後、計算時間を厳密に測定し、評価する必要があると考えられる。また、単カーネルの総数による計算量の違いについても検討が必要である。

実験1と実験2の比較 表4.3と表4.6の提案手法について、識別率の比較を表4.8に示した。KDE拡張法では、マルチカーネル学習の適用により“twonorm”について識別率が向上した。また、識別率の低下が5%未満の識別問題は7つであった。一方で、制約追加法を適用した手法では、マルチカーネル学習を適用したことで多くの識別問題で識別性能が低下した。Gauss拡張法ではマルチカーネル学習の適用によりすべての識別問題で識別性能が低下した。識別率の低下が5%未満の識別問題は4つであった。一方で、制約追加法を適用した手法では、マルチカーネル学習の適用により、“breast-cancer”、“german”において識別率が向上した。DoG拡張法では、マルチカーネル学習の適用により“banana”、“titanic”について識別率が向上した。特に、“titanic”については確率密度推定による手法よりも15%の識別率向上を示した。制約追加法を適用した手法ではマルチカーネル学習の適用により、“breast-cancer”、“diabetes”、“german”、“titanic”、“twonorm”、“waveform”において識別率が向上した。また識別率の低下が5%未満であったものは無かった。

表 4.2: 実験1：確率密度推定を適用した拡張手法の比較実験における最適パラメータ

識別問題	従来手法	提案手法					制約追加法を用いた提案手法				
	MAPLP	KDE		G	DoG		KDE		G	DoG	
	$\log_{10} \gamma$	$\log_{10} \gamma$	$\log_{10} \sigma$	$\log_{10} \gamma$	$\log_{10} \gamma$	$\log_{10} \nu$	$\log_{10} \gamma$	$\log_{10} \sigma$	$\log_{10} \gamma$	$\log_{10} \gamma$	$\log_{10} \nu$
Banana	0.5	1.0	-0.7	-0.7	-1.4	0.6	1.0	-0.8	-0.7	-1.4	0.6
Breast-cancer	-0.6	-0.5	-0.7	-0.8	-1.0	0.1	-0.5	-0.7	-0.8	-1.0	0.1
Diabetes	-0.3	-0.4	-0.5	-0.5	-0.8	0.2	-0.4	-0.5	-0.5	-0.8	0.2
Flare-Solar	0.0	0.0	-0.6	0.2	-1.3	1.0	0.0	-0.6	0.2	-1.3	1.0
German	-0.5	-1.0	-0.5	-1.0	-1.5	0.1	-1.0	-0.5	-1.0	-1.5	0.1
Heart	-0.8	-1.0	-0.4	0.3	0.2	0.1	-1.0	0.6	0.3	0.2	0.1
Image	1.0	0.5	-1.5	0.1	-1.4	0.2	0.5	-1.5	-0.8	-0.4	0.1
Ringnorm	-1.0	-0.7	-1.1	-0.8	-1.6	0.3	-0.7	-2.0	-0.8	-0.7	0.2
Splice	1.0	-1.3	-0.5	-1.4	-1.5	0.1	-1.4	-2.0	-1.4	-1.5	0.1
Thyroid	0.9	1.0	-1.5	0.4	0.3	0.1	0.4	-1.0	0.4	0.6	0.2
Titanic	-0.5	-0.2	-0.3	2.0	-0.9	0.4	0.0	-0.4	2.0	-0.5	0.1
Twonorm	-1.2	-0.5	-0.7	-0.8	-2.0	0.3	-1.4	-1.9	-0.8	-2.0	0.3
Waveform	-0.7	-0.7	-0.5	-1.2	-1.5	0.1	-0.6	-2.0	-1.2	-1.5	0.1

表 4.3: 実験 1 : 確率密度推定を適用した拡張手法の平均識別率 [%] と標準偏差

識別問題	従来手法	提案手法			制約追加法を用いた提案手法		
	MAPLP	KDE	G	DoG	KDE	G	DoG
Banana	89.2 ± 0.6	89.1 ± 0.5	57.4 ± 2.3	53.7 ± 5.0	89.2 ± 0.5	55.8 ± 2.8	51.3 ± 5.7
Breast-cancer	74.4 ± 4.2	73.5 ± 4.2	<u>74.2 ± 4.2</u>	74.1 ± 4.5	73.5 ± 4.2	64.5 ± 4.9	63.7 ± 4.5
Diabetis	75.3 ± 1.7	75.8 ± 1.7	75.3 ± 1.7	73.3 ± 2.2	75.8 ± 1.7	72.5 ± 2.3	64.6 ± 3.8
Flare-Solar	64.7 ± 1.9	65.9 ± 1.8	63.9 ± 2.2	61.2 ± 4.0	65.9 ± 1.8	62.8 ± 2.1	60.5 ± 4.4
German	74.7 ± 2.4	74.8 ± 2.2	74.1 ± 2.3	72.2 ± 2.2	74.8 ± 2.2	66.9 ± 3.2	68.6 ± 3.8
Heart	82.5 ± 3.9	<u>82.0 ± 3.8</u>	80.9 ± 3.8	79.5 ± 4.0	78.8 ± 4.4	81.4 ± 3.8	80.7 ± 4.2
Image	96.2 ± 0.5	96.5 ± 0.6	78.4 ± 2.4	74.9 ± 6.3	96.5 ± 0.6	76.7 ± 2.3	76.0 ± 3.2
Ringnorm	81.2 ± 3.3	89.3 ± 1.4	79.9 ± 3.9	72.6 ± 1.7	89.4 ± 1.4	80.9 ± 2.2	73.9 ± 6.9
Splice	75.0 ± 0.8	85.3 ± 1.0	80.8 ± 1.5	83.3 ± 1.4	80.5 ± 1.6	81.1 ± 1.7	<u>83.2 ± 1.4</u>
Thyroid	95.3 ± 2.3	95.7 ± 2.2	92.4 ± 3.1	88.5 ± 3.0	<u>95.6 ± 2.2</u>	87.8 ± 3.7	75.4 ± 5.0
Titanic	76.4 ± 1.4	77.0 ± 1.8	74.1 ± 3.7	52.6 ± 12.2	<u>76.8 ± 1.7</u>	75.2 ± 3.0	55.8 ± 15.2
Twonorm	97.4 ± 0.2	97.1 ± 0.2	97.3 ± 0.3	97.0 ± 0.4	97.1 ± 0.3	<u>97.3 ± 0.2</u>	96.5 ± 1.2
Waveform	88.6 ± 1.2	89.7 ± 0.4	85.6 ± 1.7	83.6 ± 4.0	<u>88.9 ± 0.8</u>	83.1 ± 1.6	81.4 ± 2.0
# of bold	3	9	0	0	6	0	0
# of under line	1	1	1	0	3	1	1

表 4.4: 実験 1 : 平均識別率に対する Nemenyi 検定 (有意水準 $\alpha = 0.05$) の結果

		従来手法	提案手法			制約追加法			総和	
		MAPLP	KDE	G	DoG	KDE	G	DoG	# "1"	# "-1"
従来手法	MAPLP	—	0	0	1	0	0	1	2	0
提案手法	KDE	0	—	0	1	0	1	1	3	0
	G	0	0	—	0	0	0	0	0	0
	DoG	-1	-1	0	—	-1	0	0	0	3
制約追加法	KDE	0	0	0	1	—	0	1	2	0
	G	0	-1	0	0	0	—	0	0	1
	DoG	-1	-1	0	0	-1	0	—	0	3
平均順位		<u>2.8</u>	2.0	3.9	5.4	<u>2.8</u>	4.8	6.2	—	

表 4.5: 実験 2 : マルチカーネル学習を適用した拡張手法における最適パラメータ

識別問題	提案手法		制約追加法を用いた提案手法	
	KDE	DoG	KDE	DoG
	$\log_{10} \sigma$	$\log_{10} \nu$	$\log_{10} \sigma$	$\log_{10} \nu$
Banana	0.6	0.6	-2.0	1.0
Breast-cancer	0.2	0.5	-1.4	0.3
Diabetis	0.5	0.4	0.5	0.4
Flare-Solar	0.3	0.3	1.5	0.3
German	0.1	0.3	-0.8	0.2
Heart	0.2	0.3	-1.0	0.3
Image	0.1	0.3	-1.3	0.3
Ringnorm	0.1	0.3	-0.2	0.3
Splice	-0.1	0.1	-0.4	0.1
Thyroid	1.3	0.1	-0.5	0.1
Titanic	-1.5	0.5	-1.1	0.5
Twonorm	0.0	0.2	-1.4	0.2
Waveform	0.3	0.3	-0.3	0.3

表 4.6: 実験 2 : マルチカーネル学習を適用した拡張手法の平均識別率 [%] と標準偏差

識別問題	従来手法	提案手法			制約追加法を用いた提案手法		
	MAPLP	KDE	G	DoG	KDE	G	DoG
Banana	89.2 ± 0.6	52.6 ± 4.4	55.2 ± 2.0	54.1 ± 3.9	<u>65.2 ± 10.6</u>	55.2 ± 2.0	47.6 ± 4.1
Breast-cancer	74.4 ± 4.2	72.0 ± 4.3	71.2 ± 4.5	62.2 ± 7.7	<u>69.5 ± 6.5</u>	71.2 ± 4.5	<u>72.2 ± 4.5</u>
Diabetis	75.3 ± 1.7	<u>73.9 ± 2.2</u>	65.2 ± 2.2	72.7 ± 2.2	58.1 ± 14.9	65.2 ± 2.2	72.7 ± 2.2
Flare-Solar	64.7 ± 1.9	<u>62.4 ± 5.1</u>	55.3 ± 1.8	58.7 ± 6.3	54.2 ± 3.5	55.3 ± 1.8	58.7 ± 6.3
German	74.7 ± 2.4	71.3 ± 2.6	70.2 ± 2.0	70.3 ± 2.7	65.1 ± 11.9	70.2 ± 2.0	<u>72.3 ± 1.9</u>
Heart	82.5 ± 3.9	<u>78.2 ± 4.4</u>	65.0 ± 9.5	77.6 ± 4.2	55.3 ± 11.4	65.0 ± 9.5	77.6 ± 4.2
Image	96.2 ± 0.5	72.8 ± 1.4	69.4 ± 2.4	<u>75.3 ± 1.5</u>	70.3 ± 2.9	69.4 ± 2.4	75.2 ± 1.5
Ringnorm	81.2 ± 3.3	<u>74.1 ± 1.5</u>	52.1 ± 1.7	72.5 ± 2.3	65.4 ± 12.7	52.1 ± 1.7	72.5 ± 2.3
Splice	75.0 ± 0.8	79.3 ± 1.9	63.2 ± 7.4	79.3 ± 1.7	58.0 ± 10.4	63.2 ± 7.4	79.3 ± 1.7
Thyroid	95.3 ± 2.3	66.5 ± 18.1	75.0 ± 4.1	74.9 ± 4.4	<u>75.4 ± 5.2</u>	75.0 ± 4.1	74.9 ± 4.4
Titanic	76.4 ± 1.4	<u>70.8 ± 3.4</u>	69.7 ± 3.0	68.0 ± 1.8	69.1 ± 11.5	69.7 ± 3.0	64.4 ± 9.2
Twonorm	97.4 ± 0.2	<u>97.3 ± 0.3</u>	87.5 ± 8.7	97.0 ± 0.5	78.8 ± 16.5	87.5 ± 8.7	97.0 ± 0.5
Waveform	88.6 ± 1.2	<u>86.5 ± 0.8</u>	70.1 ± 3.2	82.5 ± 1.7	67.9 ± 5.0	70.1 ± 3.2	82.5 ± 1.7
# of bold	12	0	0	1	0	0	1
# of under line	0	7	0	1	2	0	2

表 4.7: 実験 2: 平均識別率に対する Nemenyi 検定 (有意水準 $\alpha = 0.05$) の結果

		従来手法	提案手法			制約追加法			総和	
		MAPLP	KDE	G	DoG	KDE	G	DoG	# “1”	# “-1”
従来手法	MAPLP	—	0	1	1	1	1	1	5	0
提案手法	KDE	0	—	0	0	1	0	0	1	0
	G	-1	0	—	0	0	0	0	0	1
	DoG	-1	0	0	—	0	0	0	0	1
制約追加法	KDE	-1	-1	0	0	—	0	0	0	2
	G	-1	0	0	0	0	—	0	0	1
	DoG	-1	0	0	0	0	0	—	0	1
平均順位		1.2	<u>3.1</u>	5.0	4.0	5.7	5.2	3.7	—	

表 4.8: 確率密度推定適用手法 (実験 1) とマルチカーネル学習適用手法 (実験 2) の識別率の差

識別問題	提案手法			制約追加法を用いた提案手法		
	KDE	G	DoG	KDE	G	DoG
Banana	-36.4	-2.1	0.4	-24.0	-0.5	-3.7
Breast-cancer	-1.5	-3.0	-11.9	-4.0	6.7	8.4
Diabetis	-1.9	-10.1	-0.6	-17.7	-7.3	8.2
Flare-Solar	-3.5	-8.6	-2.5	-11.7	-7.6	-1.7
German	-3.5	-3.9	-1.9	-9.7	3.3	3.7
Heart	-3.8	-15.9	-1.8	-23.6	-16.4	-3.1
Image	-23.8	-9.0	0.4	-26.2	-7.4	-0.8
Ringnorm	-15.2	-27.8	-0.1	-24.0	-28.8	-1.5
Splice	-6.0	-17.7	-4.0	-22.5	-17.9	-3.9
Thyroid	-29.2	-17.4	-13.6	-20.2	-12.8	-0.5
Titanic	-6.2	-4.4	15.4	-7.6	-5.5	8.6
Twonorm	0.2	-9.9	0.0	-18.3	-9.9	0.5
Waveform	-3.2	-15.5	-1.1	-21.0	-13.0	1.1
# of diff. ≥ 0.0	1	0	3	0	2	6
# of diff. ≥ -5.0	7	4	11	1	3	13

表 4.9: 初期値の違いによるマルチカーネル学習適用手法 (制約追加法適用無し) の平均識別率 [%] と標準偏差の比較

識別問題	従来手法	各カーネルの重みパラメータの初期値					
		一定			単カーネル手法の最適値を用いて調整		
	MAPLP	KDE	G	DoG	KDE	G	DoG
Banana	89.2 ± 0.6	52.6 ± 4.4	55.2 ± 2.0	54.1 ± 3.9	<u>89.1 ± 0.5</u>	56.4 ± 2.5	53.7 ± 4.9
Breast-cancer	74.4 ± 4.2	72.0 ± 4.3	71.2 ± 4.5	62.2 ± 7.7	73.7 ± 4.2	<u>74.2 ± 4.2</u>	74.1 ± 4.5
Diabetis	75.3 ± 1.7	73.9 ± 2.2	65.2 ± 2.2	72.7 ± 2.2	75.8 ± 1.7	<u>75.6 ± 1.8</u>	73.3 ± 2.2
Flare-Solar	<u>64.7 ± 1.9</u>	62.4 ± 5.1	55.3 ± 1.8	58.7 ± 6.3	65.9 ± 1.8	64.0 ± 2.4	59.5 ± 4.4
German	<u>74.7 ± 2.4</u>	71.3 ± 2.6	70.2 ± 2.0	70.3 ± 2.7	74.8 ± 2.2	74.1 ± 2.3	71.4 ± 2.1
Heart	82.5 ± 3.9	78.2 ± 4.4	65.0 ± 9.5	77.6 ± 4.2	<u>82.0 ± 3.8</u>	80.4 ± 3.4	79.3 ± 4.4
Image	<u>96.2 ± 0.5</u>	72.8 ± 1.4	69.4 ± 2.4	75.3 ± 1.5	96.4 ± 0.7	77.4 ± 2.4	75.0 ± 5.9
Ringnorm	<u>81.2 ± 3.3</u>	74.1 ± 1.5	52.1 ± 1.7	72.5 ± 2.3	88.8 ± 1.6	79.9 ± 3.9	67.5 ± 8.1
Splice	75.0 ± 0.8	79.3 ± 1.9	63.2 ± 7.4	79.3 ± 1.7	82.5 ± 1.4	80.8 ± 1.5	<u>80.9 ± 1.5</u>
Thyroid	<u>95.3 ± 2.3</u>	66.5 ± 18.1	75.0 ± 4.1	74.9 ± 4.4	95.7 ± 2.2	92.4 ± 3.1	88.4 ± 3.1
Titanic	<u>76.4 ± 1.4</u>	70.8 ± 3.4	69.7 ± 3.0	68.0 ± 1.8	76.9 ± 2.0	74.1 ± 3.7	70.1 ± 9.0
Twonorm	97.4 ± 0.2	97.3 ± 0.3	87.5 ± 8.7	97.0 ± 0.5	<u>97.3 ± 0.2</u>	97.3 ± 0.3	97.0 ± 0.4
Waveform	88.6 ± 1.2	86.5 ± 0.8	70.1 ± 3.2	82.5 ± 1.7	89.0 ± 0.6	85.6 ± 1.7	79.2 ± 5.2
# of bold	4	0	0	0	9	0	0
# of under line	6	0	0	0	3	2	1

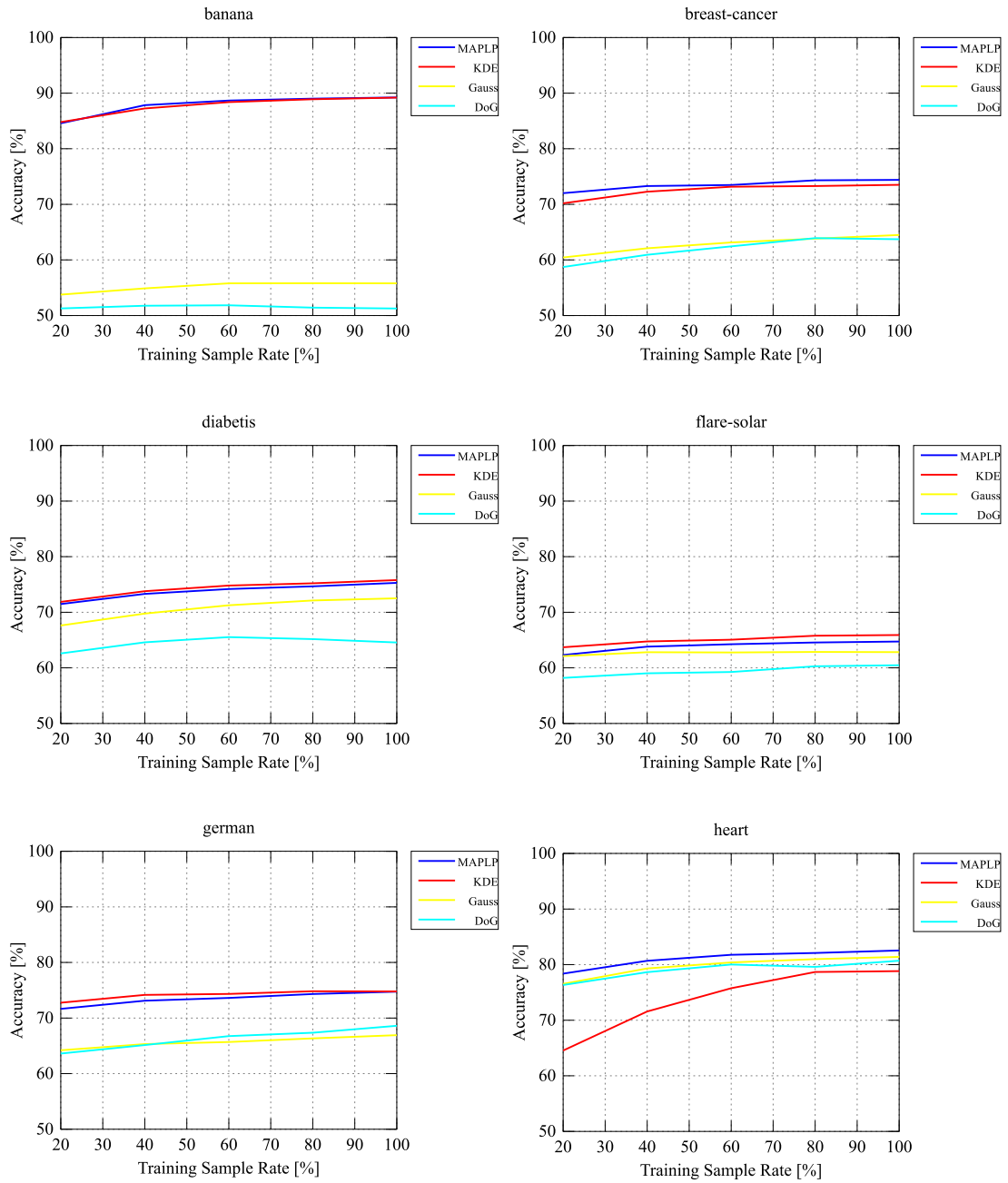


図 4.1: 訓練標本数の違いによる識別率の変化 (制約追加法を適用した確率密度推定による拡張手法)

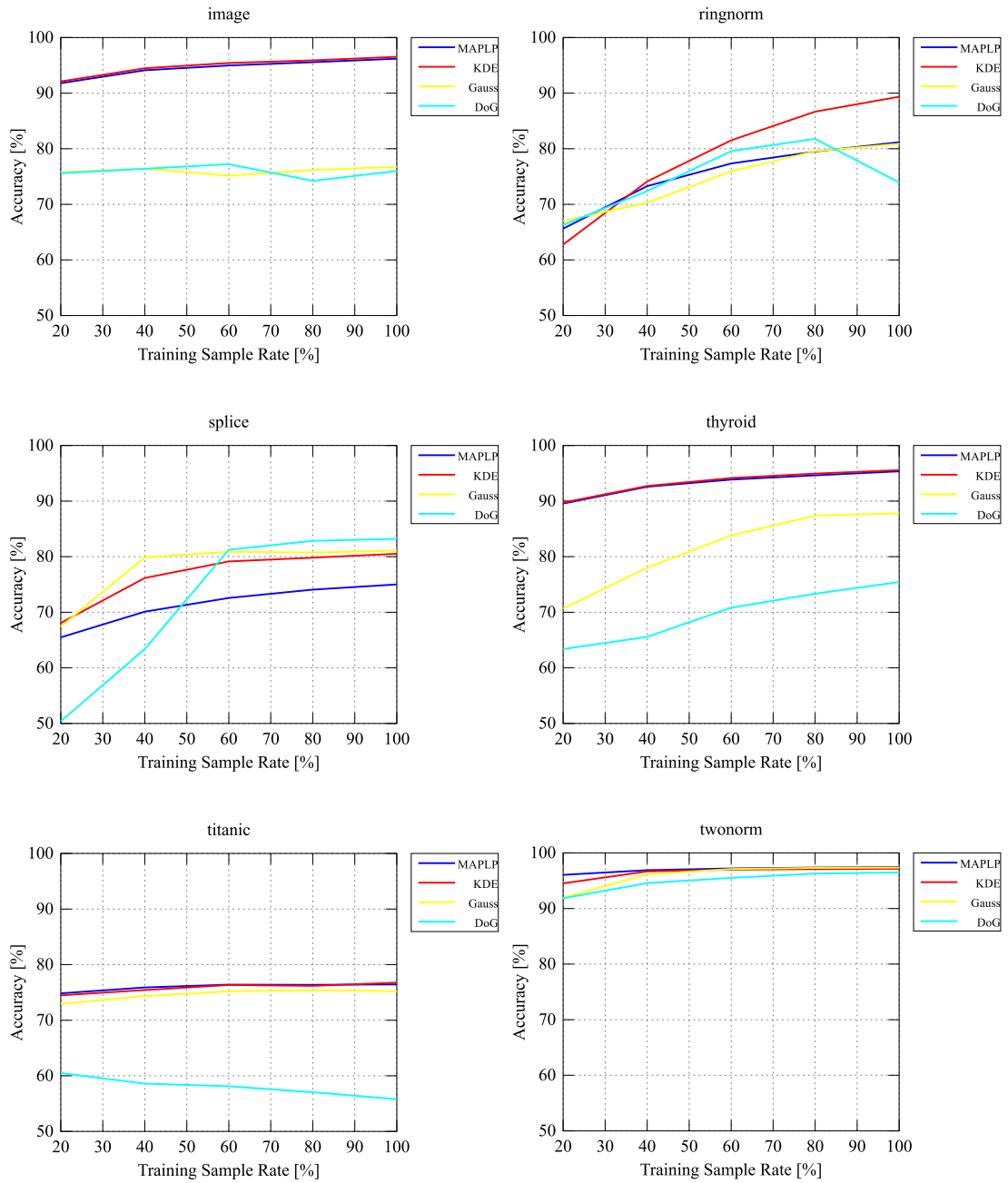


図 4.2: 訓練標本数の違いによる識別率の変化 (制約追加法を適用した確率密度推定による拡張手法)

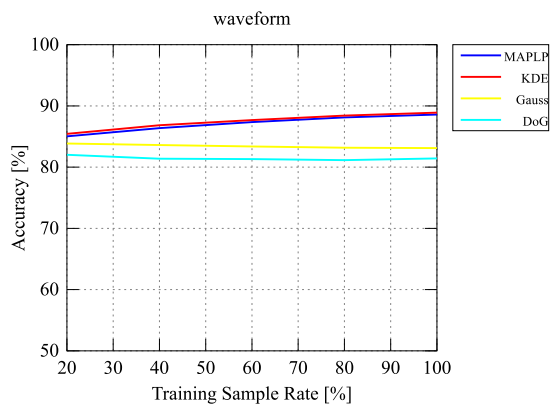


図 4.3: 訓練標本数の違いによる識別率の変化 (制約追加法を適用した確率密度推定による拡張手法)

第5章 まとめ

5.1 総括

本論文では、はじめにパターン認識系の概要、既存の識別器の構成法、パラメータ推定法、カーネル法、線形計画 MAP 識別法について述べた。線形計画 MAP 識別法では、最適化問題に含まれる期待値の計算を標本平均で近似するようにして識別器を設計している。これは、訓練標本のみを学習し、未知の標本を考慮していない設計であるといえる。そこで提案手法では、各クラスにおける入力データの条件付き確率密度関数をモデル化し、最適化問題における期待値を近似無しで定式化した。さらに、確率密度関数のモデル化により、従来手法では導入できなかったマルチカーネル学習を適用した拡張手法を提案した。マルチカーネル学習の導入により、交差検定によるカーネル選択やカーネルパラメータの最適値計算が不要になるという利点がある。

性能評価として、オープンデータセットを用いた計算機実験を行い、多くの識別問題に対してカーネル密度推定による拡張手法の有効性を示した。特に、高次元の特徴をもつ識別問題において、カーネル密度推定による拡張手法、ガウスモデルによる拡張手法、ガウス差分関数による拡張手法は従来手法よりも 5% 以上の識別率向上を示した。マルチカーネル学習を適用した提案手法では、各カーネルの重みパラメータの初期値を一定にした場合、多くの識別問題で識別性能が低下した。これと同時に、数種の識別問題では、平均識別率は低いものの標準偏差の値が大きいため、高い識別率を示したデータセットも存在することを確認した。他方で、初期値を単カーネルにおける最適値を用いて調整した場合では、従来手法と比較して同程度以上の識別性能をもつことを確認した。この結果から、マルチカーネル学習適用法の識別性能はカーネル関数の重みパラメータの初期値に大きく依存している、ということが分かった。

5.2 今後の課題

今後の課題を以下に挙げる。

1. 提案手法の性能評価のために、より多くの識別問題に対して実験を行う必要がある。また、他の識別手法との比較実験を行う必要がある。
2. 最適化問題における制約条件 (3.3)-(3.4) をより広い可能領域で評価する方法が考えられる。本研究では、制約条件 (3.3)-(3.4) の評価を

$$\alpha_{0,j} \geq 0, \quad (j = 0, \dots, N) \quad (5.1)$$

$$\alpha_{1,j} \geq 0, \quad (j = 0, \dots, N) \quad (5.2)$$

$$\beta_j \geq 0, \quad (j = 0, \dots, N) \quad (5.3)$$

と置き替えることで、評価を起こった。これは、制約条件 (3.3)-(3.4) をすべての x について考えるのは困難であるためであった。しかしながら、置き換えられた制約条件は本来の制約

条件 (3.3)-(3.4) に対して，解の可能領域を大幅に狭めた制約となってしまう。そこで，制約条件 (3.3)-(3.4) を標本点について満たすような制約条件を与えることを考える。つまり，

$$\sum_{j=1}^N \alpha_{0,j} k(\mathbf{x}_i, \mathbf{x}_j) \geq 0, \quad (i = 1, \dots, N) \quad (5.4)$$

$$\sum_{j=1}^N \alpha_{1,j} k(\mathbf{x}_i, \mathbf{x}_j) \geq 0, \quad (i = 1, \dots, N) \quad (5.5)$$

$$\sum_{j=1}^N \beta_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0, \quad (i = 1, \dots, N) \quad (5.6)$$

を制約条件として用いる方法が考えられる。代理関数 $w(\mathbf{x}, y)$ とスラック関数 $\xi(\mathbf{x})$ の重みパラメータである $\alpha_{y,j}, \beta_j$ ($j = 1, \dots, N$) に対する非負条件を排除することができるため，より広い可能領域から解を求めることができると考えられる。この際，可能領域が有界でなく目的関数が発散してしまう場合があることに注意する必要がある。

3. マルチカーネル学習法における各カーネルの重みパラメータの決定手法について，既存手法を導入し，比較・検討を行う必要があると考えられる。2.5.2 節でも述べたように，カーネルの重みパラメータの決定方法は数多く提案されている。文献 [20] で紹介されている更新手法を適用し，実験および検討を行う必要がある。例えば，文献 [26] では半正定値計画問題による更新手法が提案されている。
4. マルチカーネル学習において，カーネルパラメータの選択方法について検討する必要がある。本実験では，式 (4.2) の値を各単カーネルに適用したが，これらのパラメータ選択の方法については検討できなかった。カーネルパラメータの変域や各カーネルのカーネルパラメータの間隔などを調整し，精度を比較する必要がある。

謝辞

本研究を進め論文をまとめるにあたり，懇切丁寧に様々なご指導いただきました鷺沢嘉一助教に心より感謝申し上げます。また，同期である修士2年の皆さまをはじめ，研究に協力してくださった鷺沢研究室の皆さまに深く感謝申し上げます。

参考文献

- [1] 飯島泰蔵, パターン認識理論, 森北出版, 1989.
- [2] Nopriadi and Y. Yamashita, “Maximum a posteriori based kernel classifier trained by linear programming,” Structural, Syntactic, and Statistical Pattern Recognition, pp.493–502, Springer, 2010.
- [3] H. Sato and Y. Washizawa, “A novel EEG-based spelling system using N100 and P300,” EHealth-For Continuity of Care: Proceedings of MIE2014, vol.205, p.428, 2014.
- [4] H. Sato and Y. Washizawa, “N100-P300 Speller BCI with detection of user’s input intention,” Proceedings of the 6th International Brain-Computer Interface Conference 2014, pp.172–175, 2014. DOI:10.3217/978-3-85125-378-8-43.
- [5] 石井健一郎, 上田修功, 前田英作, 村瀬洋, わかりやすいパターン認識, オーム社, 1998.
- [6] 石井健一郎, 上田修功, 続・わかりやすいパターン認識-教師なし学習入門-, オーム社, 2014.
- [7] R.O. Duda, P.E. Hart, Dacid G. Stork(原著), 尾上守夫(監訳), パターン識別, 株式会社新技術コミュニケーションズ, 2001.
- [8] 杉山将, 統計的機械学習-生成モデルに基づくパターン認識-, オーム社, 2009.
- [9] 杉山将, 機械学習のための確率と統計, 講談社, 2015.
- [10] C. M. Bishop(原著), 元田浩, 栗田多喜夫, 樋口知之, 松本裕治, 村田昇(監訳), パターン認識と機械学習 上 バイズ理論による統計的予測, シュブリンガー・ジャパン, 2007.
- [11] S. Mika, G. Rätsch, J. Weston, B. Scholköpfung, and K.R. Müllert, “Fisher discriminant analysis with kernels,” Neural Networks for Signal Processing IX, vol.1, p.1, 1999.
- [12] 鷺沢嘉一, “カーネル部分空間法によるパターン認識,” Master’s thesis, 東京工業大学, 2004.
- [13] Y. Washizawa, “Subset kernel principal component analysis,” Machine Learning for Signal Processing, 2009. MLSP 2009. IEEE International Workshop on IEEE, pp.1–6 2009.
- [14] J. Shawe-Taylor, Nello Cristianini(原著), 大北剛(監訳), カーネル法によるパターン解析, 共立出版, 2010.
- [15] C. M. Bishop(原著), 元田浩, 栗田多喜夫, 樋口知之, 松本裕治, 村田昇(監訳), パターン認識と機械学習 下 バイズ理論による統計的予測, シュブリンガー・ジャパン, 2007.
- [16] 赤穂昭太郎, カーネル多変量解析-非線形データ解析の新しい展開-, 岩波書店, 2008.

- [17] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol.20, no.3, pp.273–297, 1995.
- [18] 阿部重夫, パターン認識のためのサポートベクトルマシン入門, 森北出版, 2011.
- [19] 竹内一郎, 鳥山昌幸, 機械学習プロフェッショナルシリーズ サポートベクトルマシン, 講談社, 2015.
- [20] M. Gönen and E. Alpaydm, “Multiple kernel learning algorithms,” *The Journal of Machine Learning Research*, vol.12, pp.2211–2268, 2011.
- [21] Nopriadi and Y. Yamashita, “A new approach to a maximum a posteriori-based kernel classification method,” *Neural Networks*, vol.33, pp.247–256, 2012.
- [22] G. Rätsch, T. Onoda, and K.R. Müller, “Soft margins for AdaBoost,” *Machine Learning*, vol.42, no.3, pp.287–320, 2001.
- [23] G. Rätsch, “Ensemble learning methods for classification,” Diploma thesis (in german), april 1998.
- [24] IBM Inc., “IBM ILOG CPLEX optimization studio v12.6 documentation,” 2013.
- [25] T. Yokota and Y. Yamashita, “A quadratically constrained MAP classifier using the mixture of Gaussians models as a weight function,” *Neural Networks and Learning Systems, IEEE Transactions on*, vol.24, no.7, pp.1127–1140, 2013.
- [26] G.R. Lanckriet, N. Cristianini, P. Bartlett, L.E. Ghaoui, and M.I. Jordan, “Learning the kernel matrix with semidefinite programming,” *The Journal of Machine Learning Research*, vol.5, pp.27–72, 2004.