

サポートカーネルマシン近似解に基づく 高速・高精度な能動学習

篠原 靖志

博士(情報学)

総合研究大学院大学
複合科学研究科
情報学専攻

平成 20 年度
(2008)

2008 年 9 月

本論文は総合研究大学院大学複合科学研究科情報学専攻に
博士(情報学)授与の要件として提出した博士論文である。

審査委員会

主査	高須淳宏 教授	総合研究大学院大学国立情報学研究所
	北本朝展 准教授	総合研究大学院大学国立情報学研究所
	佐藤 健 教授	総合研究大学院大学国立情報学研究所
	佐藤真一 教授	総合研究大学院大学国立情報学研究所
	山田誠二 教授	総合研究大学院大学国立情報学研究所

(主査以外はアルファベット順)

Fast and Precise Active Learning
based on Approximate Solution
of Support Kernel Machines

Yasusi Sinohara

DOCTOR OF
PHILOSOPHY

Department of Informatics
School of Multidisciplinary Sciences
The Graduate University for Advanced Studies (SOKENDAI)

September, 2008

A dissertation submitted to
the Department of Informatics,
School of Multidisciplinary Sciences,
The Graduate University for Advanced Studies (SOKENDAI)
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Advisory Committee

Prof. Atsuhiko Takasu (Chair)	National Institute of Informatics/ The Graduate University for Advanced Studies
Assoc. Prof. Asanobu Kitamoto	National Institute of Informatics/ The Graduate University for Advanced Studies
Prof. Ken Sato	National Institute of Informatics/ The Graduate University for Advanced Studies
Prof. Shin'ichi Sato	National Institute of Informatics/ The Graduate University for Advanced Studies
Prof. Seiji Yamada	National Institute of Informatics/ The Graduate University for Advanced Studies

(Alphabet order of last name except chair)

内容梗概

本論文は、事例データの学習に基づいて自動判定システムを効率的に構築するために、専門家とシステムのインタラクションによって、学習用事例データベースの効率的構築、判別に有効な特徴（正確には、その内積であるカーネル）の組み合わせの選択、高精度の判別関数の学習を平行して行う能動学習手法について提案するものである。

「サポートベクターマシン」などの汎化誤差の解析に基づく近年の機械学習技術の進歩により、「機器状態」とその「診断結果」など「入力」と「ラベル」とが対となったラベル付き学習用事例データベースが存在すれば、高精度の判別関数を学習することは比較的容易になっている。デジタルカメラをはじめ各種センサと記録装置の高性能化・低廉化により、機器状態などの「入力」のデータベース化は容易に行なえるようになったが、各「機器状態」に対する「診断結果」など各「入力」の「ラベル」は、専門家などに「ラベル付け」をしてもらう必要があるため、ラベル付き学習用データベースを用意することが困難な場合が少なくない。そこで、極力少数の入力に対するラベル付けで高精度の判別関数を学習できる手法として「能動サポートベクターマシン」などの能動学習が提案されている。これは、現在までに得られたラベルによる学習結果（判別関数）に基づいて、コンピュータがデータベース中からラベル付けが有用な入力を能動的に選び出し、専門家にラベル付けをしてもらうことを繰り返すことで、ラベル付き学習用事例データベースを効率的に構築しながら高精度の判別関数の学習を行なう手法である。

ただし、従来の能動学習では、通常、判別に有効な特徴は事前にわかっているとの前提があった。しかし、十分な事前知識がなければ、能動学習を始める前に判別に有効な特徴を選定することは本来難しい。代表的な能動サポートベクターマシンでは、適切な特徴（厳密には、カーネル）が選ばれていれば非常に効率的に能動学習が進むのに、不適切な特徴（カーネル）が選ばれた場合には学習が途中で止まり、効率的な能動学習ができなくなる。

そこで本研究では、各入力のラベルだけでなく、判別に有効なカーネル

も不明な状況から出発し、単に高精度の判別結果を得るためだけでなく、判別に有効なカーネルを見極めるために専門家によるラベル付けを用いる、高速で精度の高い能動学習手法を提案する。

本研究での基本的なアプローチは、特徴選択を一般化し高精度の判別に有効なカーネルの最適な組み合わせを見つけ出す多カーネル学習の問題と捉え、多カーネル学習アルゴリズムを能動化する。我々はまず、代表的な多カーネル学習であるサポートカーネルマシンを基本アルゴリズムとして、これを能動学習化した「単純能動サポートカーネルマシン」を提案する。これにより、特徴選択（正確には、カーネルの組み合わせの選択）と高精度な診断基準の構築の両者に有効な能動学習を実現する。さらに、サポートカーネルマシンの近似解を活用することで、最終的な判別関数の高い判別精度を低下させることなく、対話環境での単純能動サポートカーネルマシンの応答の遅さを解決した2段階能動サポートカーネルマシンを提案する。

1章では本研究の背景と目的を述べる。

2章では、本研究の基礎となるサポートベクターマシン、サポートカーネルマシン、能動学習とそのためのサンプリング戦略についての既存研究、サポートベクターマシンによる能動学習の課題である「カーネル事前選択問題」について述べる。

3章では、まず「カーネル事前選択問題」解決の基本手法として、代表的な多カーネル学習手法である「サポートカーネルマシン」を能動学習化した「単純能動サポートカーネルマシン」を提案する。しかし、本能動学習手法は、能動学習時に専門家とのインタラクションを行なうには、応答が遅い。これは、単純能動サポートカーネルマシンでは、能動学習時に、新たなラベルを取得するたびに、最適な判別関数を再学習してラベル付け対象を選択することに起因する。サポートカーネルマシンは、カーネルの組み合わせの最適化も同時に行なうため判別関数の学習に計算時間を要する。この結果、単純能動サポートカーネルマシンでは、専門家がラベル付けをしてシステムが次のラベル付け対象を専門家に提示するまでの時間が掛かり過ぎる場合が生じる。

そこで、能動学習段階では、(1) サポートカーネルマシンの近似解による判別関数を用いてラベル付け対象を選択して、能動学習時の応答性を確保する、(2) 能動学習が終了した段階でサポートカーネルマシンを完全に最適化して判別関数の判別精度の高さを確保する、というアプローチを取る2段階能動サポートカーネルマシンを提案する。

さらに,このような近似を行なっても,最適な単一のカーネルを用いたサポートベクターマシンとほぼ同等の判別精度を持つ判別関数が得られることを実験により示す.

4章では,能動サポートカーネルマシンに適したサンプリング戦略の検討を行う.特に,RBFカーネルはサポートベクターマシンで有効性が知られているため,異なる半径のRBFカーネルを用いる能動サポートカーネルマシンについて検討を行う.そして,ラベル付け対象の選択戦略として能動学習で通常用いられるマージン戦略よりも,適切なカーネルの組み合わせに,より少ないラベル付けで収束し易い戦略として,SKM-SHIFT戦略を提案する.その基本的考え方は,最適なカーネルの組み合わせが絞り込めていない初期段階では極力多様な入力に対するラベルを得てカーネルの組み合わせの絞込みを重視するのに対して,カーネルの組み合わせが最適値に近づいた後半では判定精度向上に重点を置くというものである.そして,2段階サポートカーネルマシンとSKM-SHIFTを組み合わせた能動学習手法の有効性を標準的ベンチマーク問題により示す.

5章では,実問題への適用として,配電柱にある腕金と呼ばれる金属部材の表面写真に基づく再利用判定への適用例を示す.カーネルと判別面を同時に学習する能動サポートカーネルマシンの利点として,SKM-SHIFTにより学習されたカーネルの分析から能動学習の早い段階で学習用入力データの偏りを発見できたことを示す.

最後に第6章で本論文の成果をまとめる.

本研究の最大の成果は,従来は既知としていた判別に有効なカーネルの選択を含めて能動学習を行ない,かつ,専門家との円滑なインタラクションを妨げない能動学習時の高い応答性と最終的な判別関数の判別精度の高さとを両立させた実用性の高い能動学習手法を開発した点にある.

Abstract

In this thesis, we propose an active learning method for highly precise classification which simultaneously performs efficient interactive labeling of unlabeled data and the selection of a proper combination of features or kernels and the learning of a precise discriminant function using the combined kernels and labeled data.

By recent progress of machine learning methods such as “support vector machines”, we can get a highly precise discriminant function relatively easily if we have a labeled database composed of pairs of an “input” (such as a state of an equipment) and a “label” (such as a diagnostic result).

Though it becomes easier to prepare databases of “inputs”, such as states of equipments, because of development of sensory and storage devices, it is still difficult to prepare labeled databases because labels of inputs, such as diagnostic results of states of equipments, are usually given by human experts. To get a highly precise discriminant function by labeling a smaller number of samples, active learning methods such as “active support vector machines” have been proposed. In active learning, the system first selects an unlabeled sample from the database effective to learn a precise discriminant function if labeled, then a human expert labels the selected sample and this process is repeated until a sufficient number of samples are labeled. By this process, the active learning algorithm performs both the construction of a labeled database and the learning of a precise discriminant function.

However, conventional active learning usually assumes that the features effective for the target classification task are known in advance. But it is difficult to identify such features before active learning. In case of active support vector machines, if proper features or kernels are selected, the learning proceeds very efficiently but if inadequate features or kernels are selected, the learning process stops on the way and the effective learning

becomes impossible.

In this study to overcome this issue of “kernel pre-selection”, we propose a fast and precise active learning method to use the labeling for learning a precise discriminant function as well as for selecting effective features or kernels.

Our basic approach is to generalize feature selection as selection of a combination of kernels and to treat the problem as a multi-kernel learning problem. We use the support kernel machines as our base multiple kernel learning algorithm and propose the “basic active support kernel machines” at first. To overcome the slow response of the basic support kernel machines, we further propose the “two-phased support kernel machines” based on approximate solutions of support kernel machines.

In chapter 1, we describe our research background and objectives.

In chapter 2, as bases of our study, we introduce the support vector machines, the support kernel machines and active learning methods and their sampling strategies. We also point out the issue of “kernel pre-selection” of the active support vector machines.

In chapter 3, we first propose the “basic active support kernel machines” which is an active learning version of a typical multiple kernel learning method “support kernel machines”.

In the basic active support kernel machines, we re-learn the optimal discriminant function every time after labeling and then select a sample for labeling. Because the support kernel machine optimizes the discriminant function as well as the combination of kernels at the same time, it takes too much time to show a next labeling sample for human experts after his labeling of one sample.

We therefore propose the “two-phased active support kernel machines” which achieves quick response in the active learning phase by selecting a sample to label using the discriminant function based on an approximate solution of support kernel machines, and also learns a highly precise discriminant function by the complete optimization of the support kernel machine after active learning phase.

We further show experimentally that we can get discriminant functions by the two-phased active support kernel machines as precise as the ones of the support kernel machines using the best single kernels.

In chapter 4, we investigate the sampling strategy for active support kernel machines using RBF kernels which are known to be effective in case of support vector machines.

For the active support kernel machines using RBF kernels with different diameters, we propose the “SKM-SHIFT strategy” as a sampling strategy which can learn a proper combination of kernels in fewer number of labeling than the “margin strategy” typically used in active learning methods. It emphasizes to narrow down the combination of kernels by getting labels of mutually dissimilar samples in the initial phase, and emphasizes to improve the precision of discriminant function after the combination of kernels approaches to the optimal. We show the effectiveness of the combination of the two-phased active support kernel machines and the SKM-SHIFT strategy using a standard benchmark problem.

In chapter 5, as an application to a real problem, we apply our active learning algorithm to the problem to judge whether a metal equipment attached to electric poles to reuse or not based on its surface image. We show that a bias of input images can be found by analyzing the learned combination of kernels. It is the effect of the active learning method which learns a discrimination function and a combination of kernels simultaneously.

In chapter 6, we summarize our results in this thesis.

The most contribution of this study is that we relax the assumption that the kernels effective to the target classification are known in advance and develop an active learning method which can select an effective combination of kernels and which achieves quick response for smooth interaction with human experts in active learning phase and high precision of learned discriminant functions simultaneously.

目次

内容梗概	i
Abstract	iii
第1章 序論	1
1.1 はじめに	1
1.2 研究の位置づけとアプローチ	2
1.3 論文の構成と概要	3
第2章 機械学習および能動学習関連研究	7
2.1 サポートベクターマシン (SVM)	7
2.1.1 Radius-Margin 上界	8
2.1.2 ハードマージン・サポートベクターマシン – 分離可能ケース	11
2.1.3 ソフトマージン・サポートベクターマシン – 分離不能ケース	12
2.1.4 双対問題	13
2.1.5 最小超球半径	16
2.1.6 カーネル	16
2.1.7 代表的解法 – SMO アルゴリズム	19
2.1.8 スパン上界	24
2.2 サポートカーネルマシーン (SKM)	28
2.2.1 概要	28
2.2.2 Radius-Margin 上界に基づく定式化	28
2.2.3 等価問題	31
2.2.4 代表的バッチ解法	36
2.3 能動学習とサンプリング戦略	39
2.3.1 能動学習	40
2.3.2 能動学習におけるサンプリング戦略	41

2.3.3	サポートベクターマシンにおけるサンプリング戦略	42
2.3.4	最適化の視点からの MARGIN 戦略の意味	46
2.4	サポートベクターマシンにおけるカーネル選択問題	48
2.4.1	バッチ版サポートベクターマシンでのカーネル選択	48
2.4.2	能動サポートベクターマシンでのカーネル事前選択 問題	50
2.5	本章のまとめ	52
第 3 章	サポートカーネルマシンに基づく能動学習の提案	53
3.1	カーネル事前選択問題解決のアプローチ	53
3.2	単純能動サポートカーネルマシンの概要	54
3.2.1	能動サポートカーネルマシンの基本アルゴリズム	54
3.2.2	簡単な適用例	57
3.3	2 段階能動サポートカーネルマシン	58
3.3.1	サポートカーネルマシン能動化のアプローチ	58
3.3.2	アルゴリズムの詳細	59
3.3.3	2 段階サポートカーネルマシンの収束性	65
3.3.4	実装	66
3.3.5	計算量	67
3.4	実験	69
3.4.1	実験システム	69
3.4.2	実験データと評価項目	69
3.4.3	能動サポートベクターマシン v.s. 能動サポートカー ネルマシン	71
3.4.4	能動サポートカーネルマシンにおけるモードの影響	81
3.4.5	考察	84
3.5	本章のまとめ	93
第 4 章	能動サポートカーネルマシン向きサンプリング戦略	97
4.1	サンプリング戦略 SKM-SHIFT の提案	97
4.1.1	アプローチ	97
4.1.2	SKM-SHIFT 戦略	98
4.1.3	サンプリング対象の絞込みによる高速化	100
4.2	実験	100
4.2.1	単カーネルサポートベクターマシンとサポートカー ネルマシンの比較	101

4.2.2	能動サポートカーネルマシンでのサンプリング戦略 の比較	103
4.2.3	カーネル配分の推移	105
4.2.4	考察	107
4.3	本章のまとめ	108
第 5 章	配電柱腕金の再利用判定への適用	109
5.1	背景	109
5.2	配電柱腕金再利用判定システムの概要	110
5.2.1	システムの概要	110
5.3	画像認識用特徴抽出	111
5.3.1	画像データの粗視化	111
5.3.2	色ヒストグラム	112
5.4	実験	114
5.5	考察	115
5.6	本章のまとめ	117
第 6 章	結論	119
	謝辞	120
	参考文献	122
	研究業績	126

定義・問題・補題・定理・系 一覧

定義 2.1.1: マージン	8
定理 2.1.1: Radius-Margin 上界	9
問題 2.1.1: SVM 主問題	12
問題 2.1.2: SVM 双対問題	14
定義 2.1.2: 線形カーネル	16
定義 2.1.3: p 次多項式カーネル	16
定義 2.1.4: RBF カーネル	17
補題 2.1.1: RBF SVM-線形 SVM の関係	17
定義 2.1.5: 部分照合カーネル	18
定義 2.1.6: 和カーネル	19
定義 2.1.7: 積カーネル	19
定義 2.1.8: 畳込みカーネル	19
補題 2.1.2: SVM 収束条件	20
定義 2.1.9: τ -違反ペア; τ -violating pair	21
定義 2.1.10: 最大違反ペア; the most violating pair	21
補題 2.1.3: SMO の収束判定	21
補題 2.1.4: SVM 双対ギャップの上限	22
定義 2.1.11: スパン	25
補題 2.1.5: スパンの行列表現	25
定理 2.1.2: 追加学習による汎化誤差増加幅	26
系 2.1.1: 能動学習時の汎化誤差指標の増加幅	27

問題 2.2.1: SKM 問題	30
問題 2.2.2: SKM_ORIGINAL	32
問題 2.2.3: SKM_SCOP	32
問題 2.2.4: SKM_BACH	32
問題 2.2.5: SKM_DIRECT	33
問題 2.2.6: SKM_BetaFree	33
定理 2.2.1: SKM 等価問題	33
問題 2.2.7: SKM-ILP 問題	38
定理 2.3.1: SMO 1 回による追加汎化誤差増加幅	47
補題 3.3.1: 2 段階 SKM の収束性 (1)	65
補題 3.3.2: 2 段階 SKM の収束性 (2)	65

アルゴリズム 目次

アルゴリズム 2.1.1	SMO アルゴリズム	23
アルゴリズム 2.2.1	SKM-ILP の基本解法	38
アルゴリズム 2.3.1	能動学習基本アルゴリズム	40
アルゴリズム 2.3.2	COMB	45
アルゴリズム 2.3.3	SHIFT (Exploration-Based Active Learner)	45
アルゴリズム 3.2.1	能動サポートカーネルマシン	55
アルゴリズム 3.3.1	2 段階能動サポートカーネルマシン (toplevel)	61
アルゴリズム 3.3.2	2 段階能動サポートカーネルマシン (update)	62
アルゴリズム 3.3.3	2 段階能動サポートカーネルマシン (intercept)	62
アルゴリズム 4.1.1	SKM-SHIFT 戦略	99

表 目 次

表 3.1	2 段階 SKM のモードと既存解法の関係	64
表 3.2	実験データベースと能動 SKM の最終誤答率	70
表 3.3	能動 SKM と最良能動 SVM の誤答率の比較 (中間)	79
表 3.4	能動 SKM と最良能動 SVM の誤答率の比較 (最終)	80
表 3.5	2 段階能動 SKM のモード	81
表 3.6	モード basic と wss2 の誤答率の比較 (中間)	83
表 3.7	モード basic と wss2 の誤答率の比較 (最終)	84
表 4.1	SKM における最適配分 β^*	102
表 5.1	腕金再利用判定におけるカーネル配分	115

目次

図 2.1	サポートベクターマシンの概念図 (1) 特徴ベクトル空間	8
図 2.2	サポートベクターマシンの概念図 (2) マージン最大化	9
図 2.3	サポートベクターマシンの概念図 (3) Radius-Margin 上限	10
図 2.4	楕円境界に対する斉次多項式能動 SVM の判別面	51
図 2.5	楕円境界に対する斉次多項式能動 SVM の性能差	51
図 3.1	楕円境界に対する能動 SVM と能動 SKM の判別面の比較	55
図 3.2	楕円境界に対する能動 SVM と能動 SKM の誤答率の比較	56
図 3.3	楕円境界に対する能動 SKM の β の変化	56
図 3.4	ベンチマーク結果 [5 データベース] (1)	72
図 3.5	ベンチマーク結果 [5 データベース] (2)	73
図 3.6	ベンチマーク結果 [5 データベース] (3)	74
図 3.7	ベンチマーク結果 [20 データベース] (1-1)	76
図 3.8	ベンチマーク結果 [20 データベース] (1-2)	77
図 3.9	ベンチマーク結果 [20 データベース] (1-3)	78
図 3.10	ベンチマーク結果 [20 データベース] (2-1)	85
図 3.11	ベンチマーク結果 [20 データベース] (2-2)	86
図 3.12	ベンチマーク結果 [20 データベース] (2-3)	87
図 3.13	ベンチマーク結果 [20 データベース] (3-1)	88
図 3.14	ベンチマーク結果 [20 データベース] (3-2)	89
図 3.15	ベンチマーク結果 [20 データベース] (3-3)	90
図 3.16	近似による逐次更新と最近傍選択 (1)	94
図 3.17	近似による逐次更新と最近傍選択 (2)	95
図 4.1	能動 SKM の応答時間とサポートベクター数の推移	104
図 4.2	能動 SKM での正解率の推移	104
図 4.3	能動 SKM でのカーネル配分 β の平均的推移	105
図 4.4	能動 SKM でのカーネル配分誤差 $\ \beta - \beta^*\ $ の推移	106

図 5.1	腕金鍍画像に基づく判定システム [28] の概念図	110
図 5.2	画像データの粗視化	112
図 5.3	HSV 六角錐カラーモデル	113
図 5.4	HSV ヒストグラム	113
図 5.5	腕金再利用判定におけるカーネル配分, 正解率の推移	116

第1章 序論

1.1 はじめに

企業間競争の激化により各企業では、サービス品質や安全性などを低下させず、かつ、コストを一層削減することが求められている。

設備の保守運用分野においても、従来は、一定時間が経過した部品は交換するなど、時間ベースの管理 (TBM; Time-Based Management) が多く用いられてきた。しかし、安全性を確保しながら一層の設備の有効活用を図るため、近年、設備状態に応じて交換時期を早めたり、交換時期を延ばしたりする状態ベースの管理 (CBM; Condition-Based Maintenance) への移行が進みつつある。このため、設備状態を的確に判断して、使用継続や補修・交換などの判断を的確に行う必要性が高まっている。一方で、熟練技術者の大量退職などにより、これらの判断を熟練した専門家のみ reliant することも困難となりつつある。

このため、専門家の診断ノウハウをコンピュータに効率的に取り込み、的確な診断が行えるようにする技術が求められている。このための有望な技術が機械学習である。機械学習により、「(機器) 状態」と「診断(判定) 結果」とがペアとなった事例データベースがあれば、診断(判定) 基準の構築が可能となる。特に、近年、マージン最大化原理に基づくサポートベクターマシン (Support Vector Machines; SVM) など、判定誤差に関する新しい理論に基づく学習方式の登場により、高い精度の判別規則の学習が可能となっている。文書分類、画像判定などさまざまな分野で、サポートベクターマシンを用いた自動分類システム、自動判定システムが作成されている。

ただし、事例データベースを準備することは必ずしも容易ではない。

近年の映像機器や各種センサーの進展と記録装置の大容量化により、「機器状態」など「入力」のみのデータベースは比較的容易に構築できる。さらに、各入力に対するラベルが得られれば、学習のための事例データベースとなるが、このためには、専門家などが、例えば、機器状態に基づいた診

断を行い, 各入力 (機器状態) にラベル (診断結果) を与える「ラベル付け」が必要となる。この作業は, 専門家を拘束して行う必要があり, コスト・時間を要する作業である。従って, このラベル付け作業の効率化が, 機械学習を活用した自動判定システム構築においては重要となる。

このために有望な技術として能動学習がある。能動学習は, 大量のラベルなしデータ (入力) があり, かつ, そのラベルを得るためにコストを要する時, ラベル付けにより高い判別精度向上が見込まれるデータを, 計算機が能動的に選択して, 専門家にラベル付けしてもらうことで, 効率的に判別精度を向上させる学習法である。このため, 比較的少数の事例に対するラベルづけで精度の高い診断基準を得ることが可能となり, 専門家の拘束時間などのコストを抑えることが可能となる。

ただし, 従来の能動学習では, 判定に有効な特徴が何か (例えば, 色か形か) が事前知識によってわかっているとの前提で学習が進められる。従って, 従来の能動学習を適用するには, 学習を始める前に学習で用いる特徴 (サポートベクターマシンの場合には, 特徴ベクトルの内積であるカーネル) を事前に決定する必要がある。しかし, どのような特徴 (カーネル) が高精度の判定に有効かは, 各入力に対するラベルの分布状況を知る前には見極めることは本来難しく, 対象データに適した特徴 (カーネル) が選ばれるとは限らない。

1.2 研究の位置づけとアプローチ

本研究では, 各入力のラベルだけでなく, どの特徴 (正確には, カーネル) が判定に有効であるかも不明な状況から出発し, 単に高精度の判定結果を得るためだけでなく, 判定に有効な特徴 (カーネル) を見極めるためにラベル付けを用いる能動学習手法を提案する。

本研究では, 特徴選択を一般化し高精度診断のために有効なカーネルの最適な組み合わせを見つけ出す多カーネル学習の問題と捉える。

これは, あらかじめ与えられた複数の候補カーネルの組み合わせでカーネルを表すことで候補カーネルの組み合わせ方も判別関数と同時に学習するものである。サポートベクター型の多カーネル学習としては, サポートカーネルマシン (SKM; Support Kernel Machine) が知られている。

そこで, 本研究では, 専門家と計算機がインタラクティブに対話を行う環境で, サポートカーネルマシンを用いた能動学習により, カーネル選択, 判別関数の学習, ラベル付きデータベース構築を同時に行う効率的な能動

学習手法を開発する.

本研究では以下の制約を重視する.

インタラクティブ応答性 専門家を拘束してインタラクティブにラベル付けを行うことを想定する. このため, 一定時間以内にラベル付け対象事例を選択して専門家にフィードバックできる応答性の高さが重要となる.

大量未処理データに対する対応 ラベルなしデータは大量に存在するため, 画像処理における各種画像特徴など, 使用するすべての特徴の値, または, カーネルを全データに対して事前に計算して, 保持することは, 容量制約や時間制約から, 困難な場合も少なくない. このような未処理データに対するデータ特徴値の取得に要する時間も考慮して, ラベル付け対象事例の選択などを行う必要がある.

1.3 論文の構成と概要

第2章「機械学習および能動学習の関連研究」では, 代表的な多カーネル学習であるサポートカーネルマシンとその基礎となるサポートベクターマシンに関わる関連研究, および, 能動学習とそのためのサンプリング戦略についての既存研究について述べる.

そして, 能動サポートベクターマシンの適用の際に問題となるカーネル事前選択問題について述べる.

3章「サポートカーネルマシンに基づく能動学習の提案」では, まず, 能動サポートベクターマシンにおけるカーネル事前選択の困難を解決するための方法として, 単純能動サポートカーネルマシンを導入する. 更に, 単純能動サポートカーネルマシンの問題点である応答の遅さを解決する方策として, 高い応答性を持つ形でサポートカーネルマシンを能動化した2段階能動サポートカーネルマシンを提案する. その基本的な考え方は, 専門家に対してラベルを知りたい事例を迅速に提示して「ラベル」を蓄積する「能動学習段階」と, 一定量のラベルの蓄積後, 最適なカーネル組み合わせの選択と判定基準の構築を行う「完全最適化段階」との2段階を考え, 能動学習段階においては応答性を重視したアルゴリズム設計を, 完全最適化段階においては解の最適性を重視したアルゴリズム設計を行うというものである.

第 1 章 序論

さらに、複数の単カーネルサポートベクターマシン、単純能動サポートカーネルマシン、2 段階能動サポートカーネルマシンを、特性の異なる 13 個のデータベースに適用した実験により、以下を明らかにする。

- 単カーネルサポートベクターマシンを用いた場合、カーネルの選択により能動学習の効率、および、判定精度に大きな差が生じる。
- 同一の要素カーネル集合を用いた能動サポートカーネルマシンは、多様なデータベースに対して、個々のデータベースに対して最適化した単カーネルサポートベクターマシンと同等の学習精度を持つ。これは、能動サポートカーネルマシンが、能動学習の過程で、各データベースに適した合成カーネルを学習するためである。
- 単純能動サポートカーネルマシンが 1 件数 10 秒の応答時間を要する場合でも、2 段階能動サポートカーネルマシンは 1 秒以内の高速な応答が可能である

第 4 章「能動サポートカーネルマシン向きサンプリング戦略」では、最適なカーネルの組み合わせを早期に見出す能動サポートカーネルマシンのためのサンプリング手法 SKM-SHIFT を提案する。その基本的考え方は、最適なカーネルの組み合わせが絞り込めていない初期段階では極力多様な事例に対するラベルを得てカーネルの組み合わせの絞込みを重視するのに対して、カーネルの組み合わせが最適値に近づいた後半では判定精度向上に重点を置くというものである。

そして、異なる半径の RBF カーネルを要素カーネルに用いる場合について、2 段階能動サポートカーネルマシンと SKM-SHIFT を組み合わせた能動学習手法の有効性を、標準的ベンチマーク問題である米国郵政公社 (USPS) の手書き数字認識問題への適用を通じて示す。

第 5 章「配電柱腕金の再利用判定への適用」では、具体的な設備診断問題への適用例として、配電柱につけられた腕金と呼ばれる金属部材の表面写真に基づく再利用判定システムへの適用例を示す。

本適用では、要素カーネルとして、画像中の各位置での色の類似度に関するカーネルと、画像全体での色の出現頻度に関するカーネルの 2 種類を用いた。この結果、ラベル付けデータ数が 100 個付近で正解率 99% という高い認識精度が得られた。ただし、本来、再利用判定は色の種類・頻度のみにより位置に影響されないとされているのに、画像中の各位置での色の類似度に関するカーネルのみがサポートカーネルとなった。特に、カーネ

ル重みの収束性が高いSKM-SHIFT 戦略を用いた場合, 100 サンプル程度でこの傾向が顕著となった. このため, 画像データに位置に関する偏りがあることが早期に発見可能であった.

この偏りは転置画像も学習データとすることで解消できるが, 判別関数のみでなく, カーネルも同時に学習する本手法の特徴により, データの偏りを早期に発見することができ, 早期に対策をとることが可能となった.

最後に第6章「まとめ」で本論文の成果をまとめる.

本報告の主な成果は, 多カーネル学習の枠組みにより, 能動学習の中で従来見落とされていた特徴選択を判別関数の学習と統一的に扱い, 能動学習時の応答が速く, かつ, 最適な特徴(カーネルの組み合わせ)を早く見つけ易く, また, 最終的な判定精度が高い能動学習手法を提案した点にある.

第2章 機械学習および能動学習 関連研究

本章では,2.1節, 2.2節 で, サポートベクターマシンとその多カーネル学習版であるサポートカーネルマシンの考え方, 定式化, 解法について整理する. さらに,2.3節で, 能動学習に関して能動サポートベクターマシンとサンプリング戦略について述べる. 最後に,2.4節で, 能動サポートベクターマシン適用上の障害となる「カーネルの事前選択問題」について述べる.

2.1 サポートベクターマシン (SVM)

今, 入力集合を X , 2値判定のラベル集合を $Y = \{-1, +1\}$ とする.

この時, サポートベクターマシンは, 訓練用に与えられた N 個の入力 $x_i \in X$ とラベル $y_i \in Y$ の組 $\{(x_1, y_1), \dots, (x_N, y_N)\}$ から, 未知の入力 $x \in X$ に対する2値判定結果 $y \in Y$ を精度良く推定する判別関数 $f(x)$ を学習する機械学習方式である.

学習に用いる入力データとラベルの組の集合 $\{(x_1, y_1), \dots, (x_N, y_N)\}$ を, 学習用データ, または, 訓練データと呼ぶ.

サポートベクターマシンでは, 入力 x に対して m 次元の特徴ベクトル $\phi(x) \in \mathcal{R}^m$ を与える特徴ベクトル関数 $\phi: X \mapsto \mathcal{R}^m$ を考え, 特徴ベクトル空間 $\phi(X)$ での線形判別を考える (図 2.1).

線形判別面の垂線方向 w , 切片 b とすると線形判別関数 $f(x)$, 線形判別結果 y は以下となる.

$$f(x) = w^\top \phi(x) + b$$
$$y = \text{sgn}(f(x)) = \begin{cases} +1 & \text{if } f(x) = w^\top \phi(x) + b \geq 0 \\ -1 & \text{if } f(x) = w^\top \phi(x) + b < 0 \end{cases}$$

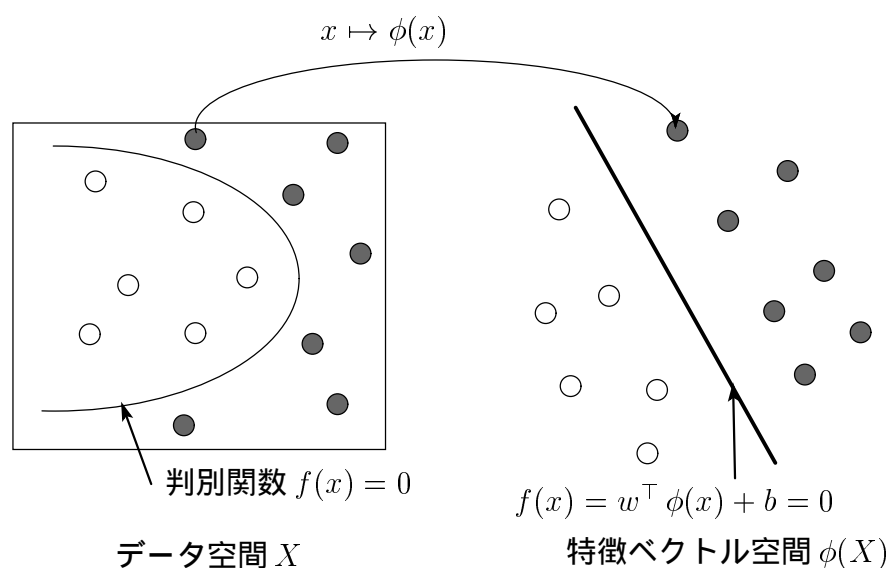


図 2.1: サポートベクターマシンの概念図 (1) 特徴ベクトル空間

サポートベクターマシンでは, Vapnik-Chervonenkis (VC) 次元の理論により求めた線形判別の汎化誤差 (未知データに対する誤等率) の上限値に基づいて, 線形判別関数 $f(x) = w^\top \phi(x) + b$ の中で, 汎化誤差の上限値が最小のものを求めることで, 精度の高い 2 値判別を行う.

2.1.1 Radius-Margin 上界

与えられた N 個の事例 $\{(x_i, y_i) | i = 1, \dots, N\}$ すべてを正しく判別できる線形判別関数 $f(x)$ が存在する場合を線形分離可能 (separable), そうでない場合を線形分離不能 (non-separable) と呼ぶ.

例えば, 特徴空間 $\phi(X)$ の次元数 m が訓練データの数 N よりも大きく, 同一の x に対して同一の判定結果 y となる場合は, 必ず, 訓練データは超平面 $f(x) = 0$ によって線形分離可能となる.

ここで, マージン $M(L; f)$ を以下のように定義する.

定義 2.1.1 (マージン) 判別関数 $f(x) = w^\top \phi(x) + b$ と, 訓練データ $L = \{(x_i, y_i) | i = 1, \dots, N\}$ が与えられた時, マージン $M(L; f)$ は, 特徴空間 $\phi(X)$ において判別面 $f(x) = 0$ に最も近い正例, 負例への距離である.

$$M(L; f) \equiv \min_{(x_i, y_i) \in L} \frac{|w^\top \phi(x_i) + b|}{\|w\|} = \min_{(x_i, y_i) \in L} \frac{y_i (w^\top \phi(x_i) + b)}{\|w\|}$$

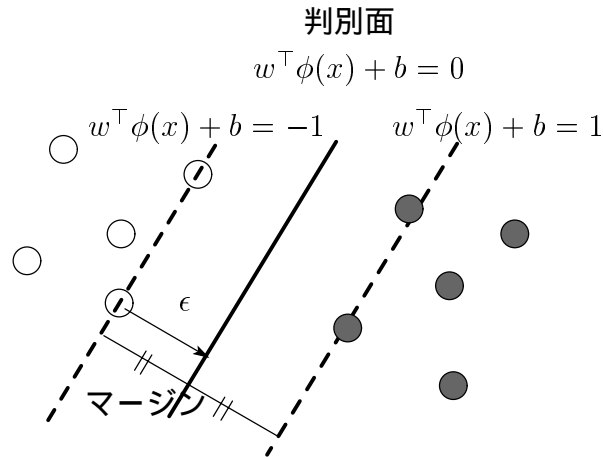


図 2.2: サポートベクターマシンの概念図 (2) マージン最大化

この時, Vapnik らは, $\phi(x) = x$ で, 訓練データが線形分離可能な場合に対して, 以下を示した [26].

定理 2.1.1 (Radius-Margin 上界) 入力空間 X が半径 R の超球に内包される時 ($\|X - c\| \leq R^2$), (X, Y) 上の任意の確率分布 P に対して, 線形判別関数 $f(x) = w^\top x + b$ の誤判定確率は以下で与えられる.

$$P_{err}(w, b) \equiv P_{(x_i, y_i) \in X \times Y} (\text{sgn}(f(x_i)) \neq y_i)$$

この時, 平均誤判定確率の上限は以下により与えられる.

$$E_L \{P_{err}(A(L))\} \leq \frac{1}{l} E_L \left\{ \frac{R^2(L)}{M(L; A(L))^2} \right\}$$

ここで, $R(L)$ は訓練データ L を囲む特徴ベクトル空間での最小超球の半径, $A(L)$ は訓練データ L に対して最大マージンを与える判別関数 $f(x)$ である.

$$R(L) \equiv \min_c \max_{(x_i, y_i) \in L} \|x_i - c\| \quad (2.1)$$

$$A(L) \equiv \underset{f}{\text{argmax}} M(L; f) \quad (2.2)$$

平均 $E_L \{g(L)\}$ は, サイズ $l-1$ の訓練データ $L \subset X \times Y$ をランダムに発生させて $g(L)$ を求めることを無限回繰り返して得た $g(L)$ の平均値である. □

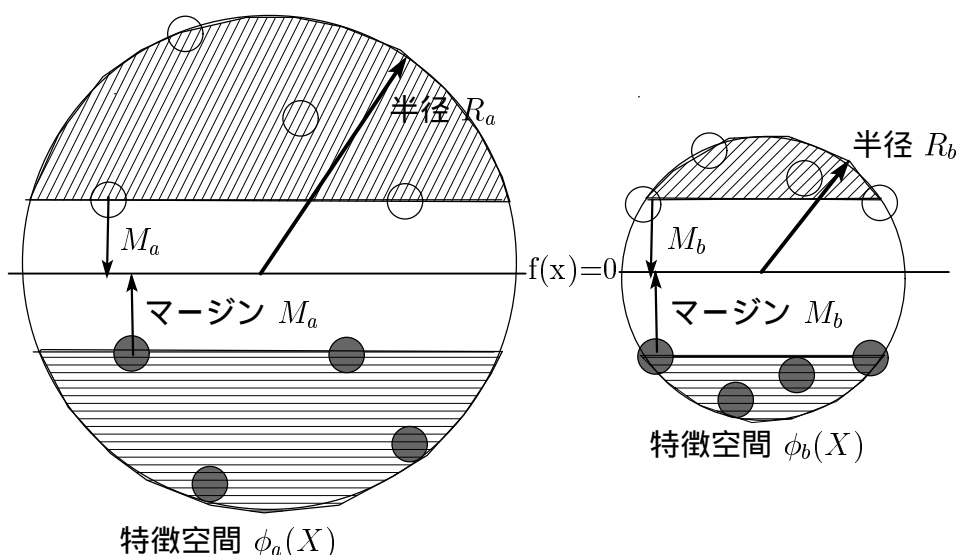


図 2.3: サポートベクターマシンの概念図 (3) Radius-Margin 上限

従って, 一般の特徴ベクトル関数 $\phi(x)$ に対しても, 訓練データの特徴ベクトル空間 $\phi(L)$ を囲む最小超球半径 R とマージン M との比 R/M が汎化誤差の上限の良い推定値となる. 従って, この Radius-Margin 上限を最小化する判別面 $f(x) = 0$ が汎化誤差の小さい良い判別面を与えられる可能性が高い.

半径 R とマージン M の比が小さいほど, 良い判別面となることは, 以下のように解釈すると直感的に理解できる.

マージンは, 判別関数 $f(x)$ により正しく判定されたある事例 x が, 誤判定となるための特徴空間 $\phi(X)$ 上での最小移動距離である (図 2.2).

$$M = \min_{\epsilon \in \mathcal{R}^m} \|\epsilon\|$$

subject to $\text{sgn}(w^\top \phi(x_i) + b) = y_i, \quad i = 1, \dots, N$

$$\text{sgn}(w^\top (\phi(x_j) + \epsilon) + b) \neq y_j \quad \text{for some } j, \epsilon$$

従って, マージン M が小さければ, 得られた訓練データ x_i に僅かな誤差が混入する (厳密には, 特徴ベクトル $\phi(x)$ に誤差 ϵ が混入し $\phi(x) + \epsilon$ となる) ことで判定結果が変わってしまうのに対して, マージンが大きい場合は, 多少の誤差が混入しても判定結果は変わらず安定している. このため, マージンが大きいほど, 入力誤差に対して頑健な判定を行うことができる (マージン最大化原理).

ただし, 特徴ベクトル空間が異なる場合, マージン M の大きさは判別関数の頑健性の指標としては意味を持たない. 例えば, 特徴ベクトルを

$\phi(x) \mapsto a\phi(x)$ と a 倍すると, 得られる判別関数の判別性能は変わらないのに, 特徴ベクトル空間 $\phi(X)$ 自身が a 倍に拡大されるためマージンも a 倍になる.

判別関数の頑健性を示す指標として意味を持つのは, 特徴空間を含む最小超球の半径 $R = R(L)$ など特徴空間の大きさを示す 1 次元量で正規化したマージン M/R である. (図 2.3).

M/R が大きいほど, すなわち, radius-margin 上限 R/M が小さいほど, 汎化誤差の小さい, 頑健な判別が可能となる.

2.1.2 ハードマージン・サポートベクターマシン – 分離可能ケース

ハードマージン・サポートベクターマシン (SVM; Support Vector Machines) は, 訓練データ $L = \{(x_i, y_i) | i = 1, \dots, N\}$ が特徴空間 $\phi(X)$ で線形分離可能な場合に, 定理 2.1.1 に基づき, Radius-Margin 上界の最小化を直接定式化したものである.

線形分離可能性より, 適当な $f(x) = w^\top \phi(x) + b$ に対して下記が成立する.

$$y_i f(x_i) > 0$$

マージンは $M(L) = \min y_i (w^\top \phi(x) + b) / \|w\|$ であり w に関してスケール不変である. このため, $\min y_i (w^\top \phi(x) + b) = 1$ と規格化することにより $M(L) = 1 / \|w\|$ となる. $R(L) = R$ とすると, Radius-Margin 上界 $R(L)/M(L) = R \|w\|$ となり, その最小化は以下により定式化できる.

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} R^2 \|w\|^2 \\ \text{subject to} \quad & y_i (w^\top \phi(x_i) + b) \geq 1 \quad (i = 1, \dots, n) \end{aligned} \quad (2.3)$$

ハードマージン・サポートベクターマシンでは, 上記の最小化問題を解くことで判別関数 $f(x) = w^\top \phi(x) + b$ を決定する.

なお, 与えられた特徴ベクトル関数 $\phi(x)$ と訓練データ L により, R は一意に定まるため, 式 (2.3) の最大化においては R は省略できる.

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} \|w\|^2 \\ \text{subject to} \quad & y_i (w^\top \phi(x_i) + b) \geq 1 \quad (i = 1, \dots, n) \end{aligned} \quad (2.4)$$

サポートベクターマシンの定式化としては、通常は式 (2.4) が用いられるが、本論文では、複数の特徴ベクトルを扱うサポートカーネルマシンに拡張するため、特徴ベクトル関数 $\phi(x)$ による半径 $R = R(L)$ の違いを明示的に含む定式化 (2.3) を標準的定式化とする。

2.1.3 ソフトマージン・サポートベクターマシン —分離不能ケース

ソフトマージン・サポートベクターマシンは、訓練データが特徴空間で線形分離可能でない場合のサポートベクターマシンである。

ソフトマージン・サポートベクターマシンでは、多少の判別誤りは許すように制約を緩める。すなわち、 $y_i f(x_i) \geq 1$ という制約条件が必ずしも成立せず、いくつかの事例で、 $y_i f(x_i) = 1 - \xi_i$ ($\xi_i > 0$) となることを許容する。そして制約違反 ξ_i が極力小さくなるようにペナルティ項 $\sum_i l(\xi_i)$ を導入する。

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} R^2 \|w\|^2 + \sum_i l(\xi_i) & (2.5) \\ \text{subject to} \quad & y_i(w^\top \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, N \end{aligned}$$

L1-サポートベクターマシン

ペナルティ $l(\xi_i)$ としては、L1 ノルム $l(\xi_i) = C \xi_i$ を用いることが一般的である。この L1-サポートベクターマシンの定式化は下記となる。

問題 2.1.1 (SVM 主問題)

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} R^2 \|w\|^2 + C \sum_i \xi_i & (2.6) \\ \text{subject to} \quad & y_i(w^\top \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad (i = 1, \dots, N) \end{aligned}$$

第 1 項はマージンすなわち、判別面 $f(x) = 0$ と最も近い整合的正例、負例 ($f(x_i) = \pm 1, \xi_i = 0$) との距離 $\frac{1}{\|w\|}$ の最大化に、第 2 項は訓練データで

の判別モデルとの不整合 (例外) ξ_i の最小化に対応する。 C は、両者のトレードオフを調整する。

以下では、問題 2.1.1(SVM 主問題) の最適値をサポートベクターマシンの「汎化誤差指標」と呼ぶこととする。

L2-サポートベクターマシン

ペナルティ $l(\xi_i)$ として L2 ノルム $l(\xi_i) = \frac{1}{2}C \|\xi_i\|^2$ を用いる場合を L2-サポートベクターマシンと呼ぶ。

$$\frac{1}{2} R^2 \|w\|^2 + \frac{C}{2} \sum_i \|\xi_i\|^2 = \frac{1}{2} R^2 \left\| \left(w, \frac{\sqrt{C}}{R} \xi \right) \right\|^2$$

$$y_i \left(w^\top \phi(x_i) + \left(\frac{\sqrt{C}}{R} \xi_i \right) \left(\frac{R}{\sqrt{C}} y_i \right) + b \right) \geq 1$$

であるから、L2-サポートベクターマシンは、特徴空間を以下のように $(\dim(\phi(X)) + \text{サンプル数 } n)$ の次元に拡張したハードマージン・サポートベクターマシンと等価である。

$$\phi(x_i) \mapsto \begin{pmatrix} \phi(x_i) \\ \frac{R}{\sqrt{C}} y_i \mathbf{e}_i \end{pmatrix}$$

ただし、 \mathbf{e}_i は、第 i 要素が 1 の単位ベクトルである。

この時、ハードマージン・サポートベクターマシンの解 w' は下記となる。

$$w' = \begin{pmatrix} w \\ \frac{\sqrt{C}}{R} \xi \end{pmatrix}$$

2.1.4 双対問題

L1-サポートベクターマシンの双対問題

問題 2.1.1(SVM 主問題) の (x_i, y_i) に対する制約のラグランジュ未定乗数を $y_i \alpha_i \geq 0$ とする。この時、

$$S(w, b, \xi_i, \alpha) = \frac{1}{2} R^2 \|w\|^2 + C \sum_i \xi_i + \sum_i \alpha_i y_i (1 - \xi_i - y_i (w^\top \phi(x_i) + b))$$

と定義すると, 問題 2.1.1 は

$$\min_{w,b,\xi_i} \max_{\alpha} S(w, b, \xi_i, \alpha) \text{ subject to } \xi_i \geq 0, y_i \alpha_i \geq 0$$

と等価である. KKT 条件は,

$$w = \frac{1}{R^2} \sum_i \alpha_i \phi(x_i), \quad 0 \leq y_i \alpha_i \leq C, \quad \sum_i \alpha_i = 0 \quad (2.7)$$

従って, 双対問題 $\max_{\alpha} \min_{w,b,\xi_i} S(w, b, \xi, \alpha)$ は下記となる.

問題 2.1.2 (SVM 双対問題)

$$\max_{\alpha} \underbrace{\sum_i \alpha_i y_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j)}_{S(\alpha)} / R^2 \quad (2.8)$$

$$\text{subject to } \sum \alpha_i = 0, \quad \alpha_i \in [A_i, B_i] \quad (2.9)$$

ただし,

$$[A_i, B_i] = \begin{cases} [0, C] & \text{for } y_i = 1 \\ [-C, 0] & \text{for } y_i = -1 \end{cases}$$

$$K(x_i, x_j) = \phi(x_i)^\top \phi(x_j)$$

なお, 特徴ベクトルの内積 $K(x, z) = \phi(x)^\top \phi(z)$ をカーネルと呼ぶ.

双対性から, $S(\alpha)$ はサポートベクターマシンの汎化誤差 (式 (2.6)) の下限を与え, $\max S(\alpha)$ はサポートベクターマシンの汎化誤差に等しい.

問題 2.1.2 は, 高次元特徴ベクトル $\phi(x)$ を明示的に扱う必要がなく, 各訓練データのペア i, j に対するカーネル $K(x_i, x_j)$ (スカラー量) とラベル y_i のみにより定まるという特徴を持つ.

問題 2.1.2 は, α に関する 2 次計画問題であるから, 疎な最適解 α^* を持つ.

この時, $\alpha_i^* \neq 0$ なる事例 (x_i, y_i) をサポートベクターと呼ぶ. $|\alpha_i| = C$ なるサポートベクターを束縛サポートベクター, $0 < |\alpha_i| < C$ なるサポートベクターを自由サポートベクターと呼ぶ.

また, 判別関数 $f(x)$ は, サポートベクターの重み α^* とカーネル $K(x_i, x)$ により下記に定まる.

$$f(x) = w^\top \phi(x) + b = \sum_i \alpha_i^* \frac{K(x_i, x)}{R^2} + b^*$$

ただし,

$$\max_{\alpha_j < B_j} y_j - \sum_s \alpha_s^* \frac{K(x_s, x_j)}{R^2} \leq b^* \leq \min_{\alpha_i > A_i} y_i - \sum_s \alpha_s^* \frac{K(x_s, x_i)}{R^2}$$

自由サポートベクターが存在する時は,

$$b^* = \max_{\alpha_j < B_j} y_j - \sum_s \alpha_s^* \frac{K(x_s, x_j)}{R^2} = \min_{\alpha_i > A_i} y_i - \sum_s \alpha_s^* \frac{K(x_s, x_i)}{R^2}$$

なお, 以下が成立する.

$$\begin{cases} y_i f(x_i) > 1 & (\alpha_i^* = 0) \\ y_i f(x_i) = 1 & (0 < y_i \alpha_i^* < C) \\ y_i f(x_i) < 1 & (\alpha_i^* = y_i) \end{cases} \Leftrightarrow \begin{cases} f(x_i) > y_i & (\alpha_i^* = A_i) \\ f(x_i) = y_i & (A_i < \alpha_i^* < B_i) \\ f(x_i) < y_i & (\alpha_i^* = B_i) \end{cases} \quad (2.10)$$

すなわち, $y_i f(x_i) - 1$ の符号が正, ゼロ, 負の時, 各々非サポートベクター, 自由サポートベクター, 束縛サポートベクターとなる.

ハードマージンサポートベクターマシンの双対問題

ハードマージン・サポートベクターマシン (2.3) の双対問題は, 問題 2.1.2 (SVM 双対問題) において $C = \infty$ とした場合である.

L2-サポートベクターマシンの双対問題

L2-サポートベクターマシンの双対問題は, 特徴ベクトルの対応関係から, カーネルとして $K + R^2/C \cdot \mathbf{I}$ を用いたハードマージン・サポートベクターマシンとなる. ただし, \mathbf{I} は $n \times n$ の単位行列である.

2.1.5 最小超球半径

Radius-Margin 上界にあらわれるデータ半径 R は, データを囲む最小超球の中心を c とすると, 定義 (式 (2.1)) から, 以下で与えられる.

$$\begin{aligned} R^2 &= \max_{R^2, c} \min_i \|\phi(x_i) - c\|^2 \\ &= \min_{\sum_i \beta_i = 1, \beta_i \geq 0} \text{diag}(K)^\top \beta - \beta^\top K \beta \end{aligned}$$

なお, $\text{diag}(A)$ は, 行列 A の対角成分からなる列ベクトルである. 中心 c は, サンプル重み β_k による重心で, $c = \sum_i \beta_i \phi(x_i)$ で与えられる.

本問題は β に関する 2 次計画問題で, SVDD (Support Vector Data Description) [24] と呼ばれる問題の 1 種である. 後述する SMO アルゴリズムと同様の手法で高速に解くことができる.

2.1.6 カーネル

カーネルは, (非線形) 特徴ベクトル $\phi(x)$ の内積である. 明示的に $\phi(x)$ が特定されていない場合でも, 入力空間の任意の n 個の点 x_1, \dots, x_n のカーネル行列 $[K(x_i, x_j)]_{i, j \in \{1, \dots, n\}}$ の固有値が正の場合,

$$K(x, z) = \sum_{k=1}^{\infty} \lambda_k \phi_k(x) \phi_k(z), \quad \lambda_k \geq 0$$

なる λ_k と $\phi_k(x)$ が存在し, 対応する特徴ベクトルは, 下記となる.

$$\phi(x) = (\sqrt{\lambda_1} \phi_1(x), \sqrt{\lambda_2} \phi_2(x), \dots)^\top$$

代表的なカーネルとして以下が挙げられる.

定義 2.1.2 (線形カーネル) $K(x, z) = x^\top z$

対応する特徴ベクトルは $\phi(x) = x$.

定義 2.1.3 (p 次多項式カーネル) $K(x, z) = (x^\top z)^p$

多項係数 $\binom{p}{k_1, \dots, k_m} = \frac{p!}{k_1! \dots k_m!}$ とすると, 対応する特徴ベクトルは

$$\begin{aligned} \phi(x) &= (\phi_{k_1, \dots, k_m}(x))_{(k_1, \dots, k_m) | \sum k_j = p, k_j \geq 0}^\top \\ \phi_{k_1, \dots, k_m}(x) &= \sqrt{\binom{p}{k_1, \dots, k_m}} x_1^{k_1} \dots x_m^{k_m} \end{aligned}$$

この時,

$$\begin{aligned} K(x, z) &= \phi(x)^\top \phi(z) = \sum_{(k_1, \dots, k_m) | \sum k_j = p, k_j \geq 0} \binom{p}{k_1, \dots, k_m} \prod_{j=1}^m (x_j z_j)^{k_j} \\ &= (x^\top z)^p \end{aligned}$$

上記は正確には, 斉次多項式カーネルで原点対象であるため, この線形和による判別関数も原点对称なものに限定される. このため, 通常は, $(x^\top, \sqrt{a})^\top$ に対する p 次多項式カーネルである非斉次多項式カーネル

$$K(x, z) = (a + x^\top z)^p$$

を用いることが一般的である.

定義 2.1.4 (RBF カーネル) $K(x, z) = \exp(-\gamma \|x - z\|^2)$

$x \in \mathcal{R}^m$ の時, 対応する特徴ベクトルは各要素が m 次元空間上の各点に対応する無限次元ベクトルとなる.

$$\begin{aligned} \phi(x) &= (\dots, \phi_t(x), \dots)_{t \in \mathcal{R}^m}^\top \\ \phi_t(x) &= \left(\frac{4\gamma}{\pi}\right)^{m/4} \exp(-2\gamma \|x - t\|^2) \end{aligned}$$

この時,

$$\begin{aligned} K(x, z) &= \phi(x)^\top \phi(z) = \int_{t \in \mathcal{R}^m} \phi_t(x) \phi_t(z) dt \\ &= \left(\frac{4\gamma}{\pi}\right)^{m/2} \int_{t \in \mathcal{R}^m} \exp(-2\gamma (\|x - t\|^2 + \|z - t\|^2)) dt \\ &= \exp(-\gamma \|x - z\|^2) \end{aligned}$$

なお, 線形カーネルは定義より 1 次多項式カーネルである. また, 線形カーネルを用いた線形サポートベクターマシンは, RBF カーネルを用いた RBF-サポートベクターマシンで $\gamma \rightarrow 0$ の場合に等しい.

補題 2.1.1 (RBF SVM-線形 SVM の関係) RBF-サポートベクターマシンは, $\gamma \approx 0$ の時 $\phi(x) = x$ による線形カーネルサポートベクターマシンに一致する. サンプル重み α_i は両方で共通である.

証明 RBF カーネルに対する最小超球半径 R , 線形カーネルに対する最小超球半径 R_{lin} は下記にて与えられる.

$$\begin{aligned} R^2 &= \min_{\sum_k \beta_k = 1, \beta_k \geq 0} \text{diag}(K)^\top \beta - \beta^\top K \beta \\ &= \min_{\sum_k \beta_k = 1, \beta_k \geq 0} \sum_i \beta_i - \sum_{i,j} \beta_i \beta_j \exp(-\gamma \|x_i - x_j\|^2) \\ R_{lin}^2 &= \min_{\sum_k \beta_k = 1, \beta_k \geq 0} \sum_i \text{diag}(x^\top x)^\top \beta - \beta^\top (x^\top x) \beta \end{aligned}$$

$\gamma \approx 0$ の時 $\exp(-\gamma(x-z)^2) \approx 1 - \gamma(x-z)^2$, $\sum_k \beta_k = 1$ に留意すると,

$$\begin{aligned} R^2 &\approx \sum_{i,j} \beta_i \beta_j \gamma \|x_i - x_j\|^2 \\ &= 2\gamma \left(\sum_i \beta_i x_i^2 - \sum_{i,j} \beta_i \beta_j x_i x_j \right) \\ &= 2\gamma \left(\sum_i \text{diag}(x^\top x)^\top \beta - \beta^\top (x^\top x) \beta \right) = 2\gamma R_{lin}^2 \end{aligned}$$

従って, $\gamma \approx 0$ の時 RBF カーネルの最小半径 R と線形カーネルの最小半径 R_{lin} を与える β は同一で, $R \approx \sqrt{2\gamma} R_{lin}$ である.

RBF カーネルによる判別関数 $f(x) = \sum_i \alpha_i \frac{K(x_i, x)}{R^2} + b$ は, $\gamma \approx 0$ の時 $K(x, z) = \exp(-\gamma(x-z)^2) \approx 1 - \gamma(x-z)^2$, $\sum_i \alpha_i = 0$ に留意すると, 下記の近似が成立する.

$$\begin{aligned} f(x) &\approx \sum_i \alpha_i \frac{1 - \gamma \|x - x_i\|^2}{2\gamma R_{lin}^2} + b \\ &= \left(\sum_i \alpha_i \right) \frac{1 - \gamma \|x\|^2}{2\gamma R_{lin}^2} + \sum_i \alpha_i \frac{2\gamma x_i^\top x}{2\gamma R_{lin}^2} + \left(b - \sum_i \alpha_i \frac{\gamma \|x_i\|^2}{2\gamma R_{lin}^2} \right) \\ &= \sum_i \alpha_i \frac{x_i^\top x}{R_{lin}^2} + \left(b - \frac{1}{2} \sum_i \alpha_i \frac{\|x_i\|^2}{R_{lin}^2} \right) \end{aligned}$$

最後の式は, 線形サポートベクトルマシンによる判別関数である. \square

定義 2.1.5 (部分照合カーネル)

$$K_{l,r}(x_i, x_j) = I(x_i(l:r) = x_j(l:r))$$

文字列 x_i, x_j に対して, 各々の l 番目から r 番目までの文字列からなる部分文字列が一致するならば 1 を, 不一致ならば 0 を返す.

さらに, カーネルを合成することで新たなカーネルを得ることができる.

定義 2.1.6 (和カーネル)

$$K(x_i, x_j) = \sum_{k=1}^M \beta_k K_k(x_i, x_j)$$

$$\phi(x) = (\sqrt{\beta_1} \phi_1(x), \dots, \sqrt{\beta_M} \phi_M(x))^\top \quad \text{where } \beta_k \geq 0$$

定義 2.1.7 (積カーネル)

$$K(x_i, x_j) = \prod_{k=1}^M K_k(x_i, x_j)$$

$$\phi_k(x) = (\phi_{k,1}(x), \dots, \phi_{k,m_k}(x))^\top$$

$$\phi(x) = \text{vec} \left(\bigotimes_{k=1}^M \phi_k(x) \right)$$

ただし, \bigotimes は直積, vec は $m_1 \times \dots \times m_M$ 行列を, $(m_1 \cdot \dots \cdot m_M) \times 1$ の列ベクトルに変換する演算.

定義 2.1.8 (畳込みカーネル)

$$K(x_i, x_j) = \sum_{\substack{s_i \in S(x_i) \\ s_j \in S(x_j)}} \beta_{s_i|x_i} \beta_{s_j|x_j} K_S(s_i, s_j)$$

$$\phi(x) = \sum_{s \in S(x)} \beta_{s|x} \phi_S(s)$$

ただし, $S(x)$ は x の部分構造集合

2.1.7 代表的解法 – SMO アルゴリズム

サポートベクターマシンの代表的解法として, Plattらにより提案された Sequential Minimal Optimization (SMO) アルゴリズムがある [20].

SMO では, 高速に解を得られる部分問題の最適化 (Minimal Optimization) を繰り返して解を改善することで問題 2.1.2 の最適解 α^* を得る.

なお, 以下では $S(\alpha)$ の α, α_i に関する勾配を g, g_i と記す.

$$g = \frac{dS(\alpha)}{d\alpha} = (g_1, \dots, g_N)$$

$$g_i = \frac{\partial S(\alpha)}{\partial \alpha_i} = y_i - \sum_j \alpha_j \frac{K(x_j, x_i)}{R^2} = y_i - f(x_i) + b$$

部分問題

制約 (2.9) を満たす可能解 $\alpha_k^0, k = 1, \dots, N$ が与えられた時, SMO では, 問題 2.1.2 で適当な 2 変数 α_i, α_j のみを可変とし, 他の変数 $\alpha_k = \alpha_k^0 (k \notin \{i, j\})$ と固定した最適化問題を解くことで解 α^0 を改善する.

$\sum_k \alpha_k = \sum_k \alpha_k^0 = 0$ の制約より, $\alpha_i + \alpha_j = \alpha_i^0 + \alpha_j^0$ であるから, $\alpha_i = \alpha_i^0 - \lambda, \alpha_j = \alpha_j^0 + \lambda$ となる. すなわち, $\alpha = \alpha^0 + \lambda(\mathbf{e}_j - \mathbf{e}_i)$ と表せる. 従って, 部分問題は下記の変数 λ に対する 2 次計画問題となる.

$$\begin{aligned} \max_{\lambda} \quad & S(\alpha^0 + \lambda(\mathbf{e}_j - \mathbf{e}_i)) \\ \text{subject to} \quad & \alpha_i^0 - \lambda \in [A_i, B_i] \\ & \alpha_j^0 + \lambda \in [A_j, B_j] \end{aligned} \quad (2.11)$$

$D_{i,j}$ を, 特徴ベクトル $\phi(x_i)$ と $\phi(x_j)$ との距離とする.

$$D_{i,j}^2 \equiv \|\phi(x_j) - \phi(x_i)\|^2 = K(x_i, x_i) + K(x_j, x_j) - 2K(x_i, x_j)$$

この時, 以下が成立する.

$$\begin{aligned} S(\alpha + \lambda(\mathbf{e}_j - \mathbf{e}_i)) &= S(\alpha) + g^\top(\mathbf{e}_j - \mathbf{e}_i)\lambda - \frac{1}{2}(\mathbf{e}_j - \mathbf{e}_i)^\top K(\mathbf{e}_j - \mathbf{e}_i)\lambda^2 \\ &= S(\alpha) + (g_j - g_i)\lambda - \frac{D_{i,j}^2}{2R^2}\lambda^2 \\ &= S(\alpha) + \frac{R^2}{2D_{i,j}^2}(g_j - g_i)^2 - \frac{D_{i,j}^2}{2R^2}\left(\lambda - \frac{R^2}{D_{i,j}^2}(g_j - g_i)\right)^2 \end{aligned}$$

従って, 解 λ は以下の 3 点の中で $S(\alpha^0 + \lambda(\mathbf{e}_j - \mathbf{e}_i))$ を最小とするもので, 高速に求めるられる.

$$\begin{cases} \lambda = \max(A_j - \alpha_j^0, \alpha_i^0 - B_i) & (\lambda \text{ の下限}) \\ \lambda = \min(B_j - \alpha_j^0, \alpha_i^0 - A_i) & (\lambda \text{ の上限}) \\ \lambda = \frac{R^2}{D_{i,j}^2}(g_j - g_i) & (\lambda \text{ に関する 2 次関数 } S(\cdot) \text{ の最小点}) \end{cases}$$

$g_j - g_i > 0$ の時 $S(\alpha + \lambda(\mathbf{e}_j - \mathbf{e}_i)) > S(\alpha)$ が成立する.

収束条件

補題 2.1.2 (SVM 収束条件) α が問題 2.1.2 の最適解となっている時かつその時に限り, $\alpha_i > A_i, \alpha_j < B_j$ となる任意の (i, j) に対して

$$g_j - g_i \leq 0$$

が成立する.

この時, 判別関数 $f(x)$ の切片 b の存在範囲は

$$g_j \leq b \leq g_i$$

証明 $g_j - g_i = (y_j - f(x_j)) - (y_i - f(x_i))$ が成立し, $\alpha_i > A_i$ の時 $y_i \geq f(x_i)$, $\alpha_j < B_j$ の時 $y_j \leq f(x_j)$ から自明

上記条件を満たさないサンプル (i, j) の組み合わせを τ -違反ペア (τ -violating pair) と呼ぶ. 従って, τ -違反ペア ($\tau > 0$) が存在する α は最適解ではない. 逆に, 最適解では, 任意のペア (i, j) が 0-違反とならない.

定義 2.1.9 (τ -違反ペア; τ -violating pair)

$$\alpha_i > A_i, \alpha_j < B_j, g_j - g_i > \tau$$

上記を満たす (i, j) を τ 違反ペアと呼ぶ.

τ 違反ペアが存在する時, b の存在範囲 $g_j \leq b \leq g_i$ は空集合となる.

定義 2.1.10 (最大違反ペア; the most violating pair)

$$(i, j) = \underset{\alpha_i > A_i, \alpha_j < B_j}{\operatorname{argmax}} (g_j - g_i) = (\underset{\alpha_i > A_i}{\operatorname{argmin}} g_i, \underset{\alpha_j < B_j}{\operatorname{argmax}} g_j) \quad (2.12)$$

上記の (i, j) を最大違反ペアと呼ぶ.

この時, 補題 2.1.2 は, 補題 2.1.3 に言い換えられる.

補題 2.1.3 (SMO の収束判定) 最大違反ペア (i, j) の勾配差 $\delta = g_j - g_i \leq 0$ の時かつその時に限り, α は最適解.

通常は $\delta = g_j - g_i = 0$ となり, 切片は $b = g_j = g_i$ と一意に定まる. ただし, $\delta < 0$ の場合も存在する. これは, 自由サポートベクターが存在せず, j, i が共に束縛サポートベクターの場合である. 例えば, $x_1 = x_2 = x, y_1 = -y_2 = 1$ の時, $\alpha_1 = -\alpha_2 = C$ で $f(x_1) = f(x_2) = b$ となり, 最大違反ペアは $(2, 1)$ で勾配差は $g_1 - g_2 = -2$ となる. $-1 \leq b \leq 1$ で b は不定である.

α が最適解でない場合, 次の補題により, $b = (g_1 + g_2)/2$ の時, 問題 2.1.1 (SVM 主問題) と問題 2.1.2 (SVM 双対問題) の目的関数の差 (双対ギャップ) を小さくできる.

補題 2.1.4 (SVM 双対ギャップの上限) 学習データ数を n , 最大違反ペア (i, j) の勾配差 $\delta = g_j - g_i$ とする時, 双対ギャップ Δ の上限は $\frac{1}{2} C n \max(\delta, 0)$ である.

$$\begin{aligned} \Delta &= \left(\frac{1}{2} R^2 w^2 + C \sum_k \max(1 - y_k f(x_k), 0) \right) \\ &\quad - \left(\sum_k y_k \alpha_k - \frac{1}{2} \sum_{k,l} \alpha_k \alpha_l K(x_k, x_l) / R^2 \right) \leq \frac{1}{2} C n \max(\delta, 0) \end{aligned}$$

等号は, $b = \frac{1}{2}(g_i + g_j)$ の時成立.

証明 双対問題の解が α の時, 主問題の $w = \sum_k \alpha_k \phi(x_k) / R^2$, $b = \frac{1}{2}(g_i + g_j) + \epsilon$ とする. また, $f(x_k) = w^\top \phi(x_k) + b$ と記す.

$\sum_k \alpha_k = 0$ に注意すると,

$$\begin{aligned} \Delta &= \sum_k \max((C y_k - \alpha_k)(y_k - f(x_k)), -\alpha_k(y_k - f(x_k))) \\ &= \sum_k \max((A_k - \alpha_k)(y_k - f(x_k)), (B_k - \alpha_k)(y_k - f(x_k))) \end{aligned}$$

最後の等式は, $A_k = \min(C y_k, 0)$, $B_k = \max(C y_k, 0)$ による.

g_i, g_j, b の定義から,

$$\begin{cases} y_k - f(x_k) \geq g_i - b = -\frac{\delta}{2} - \epsilon \geq -\frac{\max(\delta, 0)}{2} - \epsilon & (\alpha_k > A_k) \\ y_k - f(x_k) \leq g_j - b = \frac{\delta}{2} - \epsilon \leq \frac{\max(\delta, 0)}{2} - \epsilon & (\alpha_k < B_k) \end{cases}$$

$\delta^+ = \max(\delta, 0)$ と置くと,

$$\begin{cases} (A_k - \alpha_k)(y_k - f(x_k)) \leq C \left(\frac{\delta^+}{2} + \epsilon \right) \leq C \left(\frac{\delta^+}{2} + |\epsilon| \right) \\ (B_k - \alpha_k)(y_k - f(x_k)) \leq C \left(\frac{\delta^+}{2} - \epsilon \right) \leq C \left(\frac{\delta^+}{2} + |\epsilon| \right) \end{cases}$$

従って,

$$\Delta \leq \sum_{k=1}^n C \left(\frac{\delta^+}{2} + |\epsilon| \right) = n C \left(\frac{1}{2} \delta^+ + |\epsilon| \right)$$

右辺は $\epsilon = 0$ すなわち, $b = \frac{1}{2}(g_i + g_j)$ の時最小の $\frac{1}{2} n C \delta^+$ となる. \square

SMO 型アルゴリズムでは, 補題 2.1.3 に基づいて収束条件を設定する. ただし, 予め定めた微小な τ に対する τ -違反ペアがない, すなわち, 最大違

反ペアの勾配差 δ が τ 以下となった時点で, 十分な精度の (準) 最適化が得られたとして終了する. この時, $f(x)$ として補題 2.1.4 の上限を与える以下を用いる.

$$f(x) = \sum_k \alpha_k \frac{K(x_k, x)}{R^2} + \frac{1}{2}(g_i + g_j)$$

SMO アルゴリズム

```

 $\alpha \leftarrow 0$  : サンプル重み
 $g \leftarrow y$  : 勾配
loop
  [部分問題の決定]
  適当な  $\tau$ -違反ペア  $(i, j)$  を選択.
  [部分問題の最適化]
   $S(\alpha + \lambda(e_j - e_i))$  を最大化する  $\lambda$  を見つける.
  [サンプル重み・勾配の更新]
   $(\alpha_i, \alpha_j) \leftarrow (\alpha_i - \lambda, \alpha_j + \lambda)$ .
   $g_s \leftarrow g_s - \lambda(K(x_j, x_s) - K(x_i, x_s))/R^2$ .
  [終了判定]
  最大違反ペア  $(i, j)$  の勾配差  $g_j - g_i \leq \tau$  ならば終了.
end loop
切片  $b = (g_i + g_j)/2$ .

```

アルゴリズム 2.1.1: SMO アルゴリズム

SMO 型解法の基本アルゴリズムをアルゴリズム 2.1.1 に示す.

SMO 型解法では, まず, 制約 (2.9) を満たす α の初期解 (通常は, ベクトル 0) を与える. 次に, 適当な τ -違反ペア (i, j) を選択して部分問題を決定し, その最適化により解 α を改善することを, τ -違反ペアがなくなるまで繰り返す. $S(\alpha)$ は単調増加列であるため, サポートベクターマシンの最適解が存在する場合は, 有限回で収束する.

また, τ 違反ペアの判定や部分問題の解の計算に勾配 g が必要となるため, SMO 型アルゴリズムでは, 通常, サンプル重み α と共に勾配 $g = \frac{dS(\alpha)}{d\alpha}$ の更新・管理を行う.

なお, 能動サポートベクターマシンの高速化で用いられる逐次サポートベクターマシン (incremental SVM) では, α の初期値として 0 ではなく,

新サンプル i_t を学習データに追加する前の最適解 α^{t-1} を用いる。ただし、新サンプル i_t に対する重み $\alpha_{i_t} = 0$ とする。逐次学習では、 α^{t-1} が新たな最適解 α^t の近似となっているため、0 から始める場合より、一般に少ない繰り返し回数で最適解に至る。

ワーキングセット選択戦略

SMO 型アルゴリズムで部分問題を定める τ -違反ペアとして最もよく用いられるものは、最大違反ペアである。

$$(i, j) \leftarrow \left(\underset{\alpha_i > A_i}{\operatorname{argmin}} g_i, \underset{\alpha_j < B_j}{\operatorname{argmax}} g_j \right)$$

Fan らは、この選択戦略をワーキングセット選択戦略 wss1 と名付けている [11]。wss1 では、最大違反ペア (i, j) に対して簡単に解析解が得られる部分問題 (2.11) を解き α を改善する [20]。

Fan らは、より効率的な戦略として、以下のワーキングセット選択戦略 wss2 を提案している [11]。wss2 では、 λ のみではなく、wss1 で固定していた最大違反ペア (i, j) の一方を可変とした上で、(2.11) を最大とする λ を求め、 α を更新する。

$$\begin{aligned} j &\leftarrow \underset{\alpha_j < B_j}{\operatorname{argmax}} g_j \\ (\lambda, i) &\leftarrow \underset{\substack{\alpha + \lambda(\mathbf{e}_j - \mathbf{e}_i) \in [A, B] \\ \lambda > 0, i \in \mathcal{L}}}{\operatorname{argmax}} S(\alpha + \lambda(\mathbf{e}_j - \mathbf{e}_i), \beta). \end{aligned}$$

2.1.8 スパン上界

一抜き法により求めたサポートベクターマシンの汎化誤差を近似するために、Chapelle らにより、スパンの概念が導入された [6]。本節ではより一般的な形でスパンを定義する。

本節では、以下の記法を用いる。

$$K_{i,j} = K(x_i, x_j)$$

0_A : 集合 A の要素数 $|A|$ だけ 0 を並べた $|A| \times 1$ のベクトル。

1_A : 集合 A の要素数 $|A|$ だけ 0 を並べた $|A| \times 1$ のベクトル。

y_A : $i \in A$ について、 y_i を並べた $|A| \times 1$ のベクトル。

$K_{A,n}$: $i \in A$ について $K(x_n, x_j)$ を並べた $|A| \times 1$ のベクトル.

$K_{A,B}$: $i \in A, j \in B$ なる $K(x_i, x_j)$ を並べた $|A| \times |B|$ 行列.

$$H_{A,B} = \begin{pmatrix} 0 & \mathbf{1}_A^\top \\ \mathbf{1}_B & K_{A,B} \end{pmatrix}, \quad H_{A,n} = \begin{pmatrix} \mathbf{1}_A \\ K_{A,n} \end{pmatrix}$$

定義 2.1.11 (スパン) 特徴空間上の点群 $\{\phi(x_i) | i \in A\}$ とその内分点, 外分点の集合である超平面 $\Lambda(A)$ と, 点 $\phi(x_n)$ の距離を「スパン」と呼ぶ.

$$\text{span}(n, A) = \min_{z \in \Lambda(A)} \|\phi(x_n) - z\| \quad (2.13)$$

ただし,

$$\Lambda(A) = \left\{ \sum_{i \in A} \lambda_i \phi(x_i) \mid \sum_i \lambda_i = 1 \right\}$$

補題 2.1.5 (スパンの行列表現)

$$\text{span}(n, A)^2 = K_{n,n} - H_{A,n}^\top H_{A,A}^{-1} H_{A,n} \quad (2.14)$$

証明 $H_{A,A}^{-1} H_{A,n} = \begin{pmatrix} u \\ \lambda \end{pmatrix}$ と置くと, 定義から, $\mathbf{1}_A^\top \lambda = \sum_i \lambda_i = 1$, $\mathbf{1}_A u + K_{A,A} \lambda = K_{A,n}$ が成立する. さらに, $u = \lambda^\top \mathbf{1}_A u = \lambda^\top K_{A,n} - \lambda^\top K_{A,A} \lambda$.
この時,

$$\begin{aligned} \text{span}(n, A)^2 &= K_{n,n} - H_{A,n}^\top H_{A,A}^{-1} H_{A,n} = K_{n,n} - \begin{pmatrix} u \\ \lambda \end{pmatrix}^\top H_{A,A} \begin{pmatrix} u \\ \lambda \end{pmatrix} \\ &= K_{n,n} - (\lambda^\top K_{A,A} \lambda + 2\lambda^\top \mathbf{1}_A u) \\ &= K_{n,n} - 2\lambda^\top K_{A,n} + \lambda^\top K_{A,A} \lambda \\ &= \|\phi(x_n) - z\|^2 \quad \text{ただし, } z = \sum_{i \in A} \lambda_i \phi(x_i). \end{aligned}$$

$\mathbf{1}_A u + K_{A,A} \lambda = K_{A,n}$ より, 各 $i \in A$ について

$$\begin{aligned} 0 &= K_{i,n} - K_{i,A} \lambda - u \\ &= K_{i,n} - K_{i,A} \lambda - \lambda^\top K_{A,n} + \lambda^\top K_{A,A} \lambda \\ &= (\phi(x_i) - z)^\top (\phi(x_n) - z) \end{aligned}$$

従って, $z \in \Lambda(A)$ は, 任意の $i \in A$ について $\phi(x_i) - z$ と $\phi(x_n) - z$ が直交する点, すなわち, $\Lambda(A)$ 中で $\phi(x_n)$ と最も近い点で, $\text{span}(n, A)$ はその距離である. \square

定理 2.1.2 (追加学習による汎化誤差増加幅) ある学習済みサポートベクターマシンの自由サポートベクターの集合 $F = \{i | 0 < \alpha_i < C\}$, 束縛サポートベクターの集合 $B = \{i | \alpha_i = C y_i\}$, 判別関数を $f(x)$ とする.

新たなサンプル n を得て再学習した時, 自由サポートベクターの集合 $G = F \cup \{n\}$ となり, 束縛サポートベクターの集合は B で変化しないと仮定する.

この時, 新たなサポートベクターの重み $\bar{\alpha}$, 切片 \bar{b} は以下を満たす.

$$\begin{pmatrix} \bar{b} \\ \bar{\alpha}_G/R^2 \end{pmatrix} = H_{G,G}^{-1} \begin{pmatrix} -\mathbf{1}_B^\top \alpha_B/R^2 \\ y_G - K_{G,B} \alpha_B/R^2 \end{pmatrix}, \quad \bar{\alpha}_B = \alpha_B \quad (2.15)$$

かつ

$$A_i \leq \bar{\alpha}_i \leq B_i \quad (i \in G)$$

この時, 汎化誤差指標 $\max_{\alpha} S(\alpha)$ の増加幅は,

$$\Delta S_{n,\alpha} = \frac{R^2}{2} \left(\frac{y_n - f(x_n)}{\text{span}(n, F)} \right)^2 \quad (2.16)$$

証明 自由サポートベクター集合 F , 束縛サポートベクター集合 B に対する再学習前のサンプル重みのベクトルを α_F, α_B とする ($\alpha_B = C y_B$).

自由サポートベクター $i \in F$ について $f(x_i) = y_i$, $\sum_{i \in F \cup B} \alpha_i = 0$ から

$$\begin{pmatrix} 0 & \mathbf{1}_F^\top \\ \mathbf{1}_F & K_{F,F} \end{pmatrix} \begin{pmatrix} b \\ \alpha_F/R^2 \end{pmatrix} + \begin{pmatrix} \mathbf{1}_B^\top \alpha_B/R^2 \\ K_{F,B} \alpha_B/R^2 \end{pmatrix} = \begin{pmatrix} 0 \\ y_F \end{pmatrix}$$

$$u_F = \begin{pmatrix} b \\ \alpha_F/R^2 \end{pmatrix}, z_F = \begin{pmatrix} -\mathbf{1}_B^\top \alpha_B/R^2 \\ y_F - K_{F,B} \alpha_B/R^2 \end{pmatrix} \text{ とすると, } H_{F,F} u_F = z_F$$

より

$$u_F = H_{F,F}^{-1} z_F \quad (2.17)$$

同様の関係は, 追加学習後も成立するため, 式 (2.15) が得られる.

汎化誤差指標 $S(\alpha)$ は, 以下で得られる.

$$\begin{aligned} S(\alpha) &= ((y_F^\top - \alpha_B^\top K_{B,F}/R^2) \alpha_F - \frac{1}{2} \alpha_F^\top K_{F,F}/R^2 \alpha_F) \\ &\quad + (y_B^\top \alpha_B - \frac{1}{2} \alpha_B^\top K_{B,B}/R^2 \alpha_B) \\ &= R^2 (z_F^\top u_F - \frac{1}{2} u_F^\top H_{F,F} u_F) + (y_B^\top \alpha_B - \frac{1}{2} \alpha_B^\top K_{B,B}/R^2 \alpha_B) \\ &= \frac{R^2}{2} z_F^\top H_{F,F}^{-1} z_F + (y_B^\top \alpha_B - \frac{1}{2} \alpha_B^\top K_{B,B}/R^2 \alpha_B) \end{aligned} \quad (2.18)$$

追加学習後のサポートベクターの変化に関する仮定から

$$\bar{\alpha}_B = \alpha_B, \quad \bar{\alpha}_G = \begin{pmatrix} \bar{\alpha}_F \\ \bar{\alpha}_n \end{pmatrix}$$

$S(\bar{\alpha})$ は、式 (2.18) の第 1 項が $R^2 z_G^\top H_{G,G}^{-1} z_G$ となったものである。ただし、

$$H_{G,G} = \begin{pmatrix} H_{F,F} & H_{F,n} \\ H_{n,F} & K_{n,n} \end{pmatrix}, \quad z_G = \begin{pmatrix} z_F \\ z_n \end{pmatrix}, \quad z_n = y_n - K_{n,B} \alpha_B / R^2$$

Woodberry の公式から

$$\begin{aligned} \begin{pmatrix} H_{F,F} & H_{F,n} \\ H_{n,F} & K_{n,n} \end{pmatrix}^{-1} &= \begin{pmatrix} H_{F,F}^{-1} & 0_F \\ 0_F^\top & 0 \end{pmatrix} \\ &\quad + \frac{1}{\text{span}(n, F)^2} \begin{pmatrix} H_{F,F}^{-1} & H_{F,n} \\ & -1 \end{pmatrix} \begin{pmatrix} H_{F,F}^{-1} & H_{F,n} \\ & -1 \end{pmatrix}^\top \end{aligned}$$

従って、

$$\begin{aligned} z_G^\top H_{G,G}^{-1} z_G &= z_F^\top H_{F,F}^{-1} z_F + \frac{(z_n - z_F H_{F,F}^{-1} H_{F,n})^2}{\text{span}(n, F)^2} \\ &= z_F^\top H_{F,F}^{-1} z_F + \frac{(z_n - u_F H_{F,n})^2}{\text{span}(n, F)^2} \\ &= z_F^\top H_{F,F}^{-1} z_F + \frac{(y_n - f(x_n))^2}{\text{span}(n, F)^2} \end{aligned}$$

ただし、 $f(x_n) = (\alpha_F K_{F,n} + \alpha_B K_{B,n}) / R^2 + b$.

従って、

$$\Delta S_{n,\alpha} = S(\bar{\alpha}) - S(\alpha) \quad (2.19)$$

なお、 $\text{span}(n, F) = 0$ かつ $f(x_n) \neq y_n$ なるサンプル n に対しては、サポートベクターの構成を変えずには、整合的な判別面を形成できず、等式は成立しない。□

系 2.1.1 (能動学習時の汎化誤差指標の増加幅) 定理 2.1.2 で式 (2.15) が成立する時、追加学習サンプル n のラベルが不明の場合、汎化誤差指標 $\max_\alpha S(\alpha)$ の増加幅の最小値は、下記となる。

$$\min_{y_n = \pm 1} \Delta S_{n,\alpha} = \frac{R^2}{2} \left(\frac{\max(1 - |f(x_n)|, 0)}{\text{span}(n, F)} \right)^2 \quad (2.20)$$

2.2 サポートカーネルマシン (SKM)

2.2.1 概要

サポートカーネルマシンは、半正定値計画の一種である 2 次錘計画 (Second Order Conic Programming; SCOP) による Lanckriet らのカーネル学習の定式化 [15] をもとに、Bach ら [13] により 2 次錘計画問題として提案された代表的多カーネル学習 (Multi-Kernel Learning; MKL) 手法である。本章では、Radius-Margin 上界に基づくサポートカーネルマシンの導入を行った後、他の定式化との関係、バッチ解法について述べる。

サポートカーネルマシンは、予め M 個の特徴ベクトル $\phi_k(x)$ またはカーネル $K_k(x, z) = \phi_k(x)^\top \phi_k(z)$, $k = 1, \dots, M$ が与えられている時、これらのカーネルの線形結合で表される合成カーネル $K(x, z; \beta) = \sum_k \beta_k K_k(x, z)$ に対するサポートベクターマシンの中で、汎化誤差指標 (Radius-Margin 上界 + L1 ペナルティ) が最小となるものを見出す。ただし β は配分 (制約 $\sum_k \beta_k = 1, \beta_k \geq 0$ を満たすもの) に限定する。

得られる合成カーネルによるサポートカーネルマシンは、各カーネル $K_k(x, z)$ による単カーネルサポートベクターマシンより性能の高いものとなる可能性が高い。

サポートカーネルマシンは、さまざまな活用が可能である。例えば、各データ特徴 k に関するカーネル K_k を用意すれば、有効な特徴 (正確には、カーネルの組合せ β) の決定と、判別関数 $f(x) = \sum_i \alpha_i K(x_i, x; \beta) + b$ の決定を同時に行える。また、RBF カーネルの γ などのパラメータを変えたカーネルを複数用意することで、適切なカーネルとそのパラメータの選択と、判別関数の決定を同時に行うことができる。

なお、本論文では、合成カーネル $K(x, z; \beta)$ に対して個別のカーネル $K_k(x, z)$ を要素カーネルと呼ぶ。

2.2.2 Radius-Margin 上界に基づく定式化

訓練データ L に対する各要素特徴空間 $\phi_k(L)$ の最小超球の中心を c_k , 半径を R_k とする。

$$\|\phi_k(x) - c_k\| \leq R_k^2$$

この時, 合成特徴ベクトルを次のように与える.

$$\phi^+(x; \beta) = \left(\sqrt{\beta_1} \frac{\phi_1(x)^\top}{R_1}, \dots, \sqrt{\beta_M} \frac{\phi_M(x)^\top}{R_M} \right)^\top$$

対応する合成カーネルは下記となる.

$$\begin{aligned} K(x, z; \beta) &= \phi^+(x; \beta)^\top \phi^+(z; \beta) = \sum_k \beta_k \frac{\phi_k(x)^\top \phi_k(z)}{R_k^2} \\ &= \sum_k \beta_k \frac{K_k(x, z)}{R_k^2} \end{aligned}$$

$$c^+(\beta) = \left(\sqrt{\beta_1} \frac{c_1^\top}{R_1}, \dots, \sqrt{\beta_M} \frac{c_M^\top}{R_M} \right)^\top$$

とすると,

$$\|\phi^+(x; \beta) - c^+(\beta)\|^2 = \sum_k \beta_k \frac{\|\phi_k(x) - c_k\|^2}{R_k^2} \leq \sum_k \beta_k = 1$$

従って, 合成特徴空間 $\phi^+(L; \beta)$ は配分 β に関わらず $c^+(\beta)$ を中心とする半径 1 の超球に含まれる. また, $\beta = e_k$ の時は合成特徴空間 $\phi^+(L; \beta) = \phi_k(L)/R_k$ を含む最小超球の半径 R は 1 となる. このため, 合成特徴空間 $\phi^+(L; \beta)$ を含む最小超球の半径 R を β によらず 1 で近似する.

従って, 合成カーネル $K(x, z; \beta)$ に対する L1-サポートベクターマシンの目的関数 (双対問題) は下記となる.

$$S(\alpha; \beta) \equiv \sum_i \alpha_i y_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j; \beta) \quad (2.21)$$

合成カーネルに対する L1-サポートベクターマシンの汎化誤差指標は, 下記で与えられる (問題 2.1.2(SVM 双対問題) 参照).

$$\max_{\alpha} S(\alpha; \beta) \quad \text{subject to} \quad \sum \alpha_i = 0, \quad \alpha_i \in [A_i, B_i]$$

従って, 汎化誤差指標が最小となる配分 β は以下のサポートカーネルマシン問題の解として求められる.

問題 2.2.1 (SKM 問題)

$$\min_{\substack{\beta_k \geq 0 \\ \sum \beta_k = 1}} \max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} S(\alpha; \beta) \quad (2.22)$$

$$\begin{aligned} \text{where } S(\alpha; \beta) &= \sum_i \alpha_i y_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j \underbrace{\left(\sum_k \beta_k \frac{K_k(x_i, x_j)}{R_k^2} \right)}_{K(x_i, x_j; \beta)} \\ &= \sum_k \beta_k \underbrace{\left(\sum_i \alpha_i y_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j \frac{K_k(x_i, x_j)}{R_k^2} \right)}_{S_k(\alpha)} \end{aligned}$$

これが, サポートカーネルマシンの第 1 の定式化である.
 双対性が成立する時, 以下が成立する [13].

$$\min_{\substack{\beta \geq 0 \\ \sum_k \beta_k = 1}} \max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum_i \alpha_i = 0}} S(\alpha; \beta) \quad (2.23)$$

$$= \max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum_i \alpha_i = 0}} \min_{\substack{\beta \geq 0 \\ \sum_k \beta_k = 1}} S(\alpha; \beta) \quad (2.24)$$

従って, 問題 2.2.1(SKM 問題) は β に関する線形計画 ($\min \sum_k S_k(\alpha^*) \beta_k$) であり, $K(x_i, x_j; \beta^*)$ をカーネルとする α に関するサポートベクターマシン問題である. このため, 最適解 α^*, β^* は共に疎な解となり, 多数のデータ, カーネルの中で限られたデータ, カーネルのみが非零となる. α_i^* が非零のデータはサポートベクター, β_k^* が非零のカーネルはサポートカーネルと呼ばれる.

サポートカーネルマシンの判別関数は以下で与えられる.

$$f(x) = \sum_i \alpha_i^* K(x_i, x; \beta^*) + b^* = \sum_{i,k} \alpha_i^* \beta_k^* \frac{K_k(x_i, x)}{R_k^2} + b^*$$

ただし

$$\max_{\alpha_j^* < B_j} y_j - \sum_s \alpha_s^* K(x_s, x_j; \beta^*) \leq b^* \leq \min_{\alpha_i^* > A_i} y_i - \sum_s \alpha_s^* K(x_s, x_i; \beta^*)$$

自由サポートベクターが存在する時は,

$$b^* = \max_{\alpha_j^* < B_j} y_j - \sum_s \alpha_s^* K(x_s, x_j; \beta^*) = \max_{\alpha_i^* > A_i} y_i - \sum_s \alpha_s^* K(x_s, x_i; \beta^*)$$

なお, 線形計画の性質 ($\min_{\beta_k \geq 0, \sum \beta_k = 1} S_k(\alpha) \beta_k = \min_k S_k(\alpha)$) により, 以下が成立する.

$$\begin{cases} S_k(\alpha^*) = S(\alpha^*, \beta^*), \sqrt{\sum_{i,j} \alpha_i^* \alpha_j^* \frac{K_k(x_i, x_j)}{R_k^2}} = \gamma^* & (\beta^* > 0) \\ S_k(\alpha^*) > S(\alpha^*, \beta^*), \sqrt{\sum_{i,j} \alpha_i^* \alpha_j^* \frac{K_k(x_i, x_j)}{R_k^2}} < \gamma^* & (\beta^* = 0) \end{cases} \quad (2.25)$$

$$\gamma^* = \left\| \sum_i \alpha_i \phi^+(x_i; \beta) \right\| = \sqrt{\sum_{i,j} \alpha_i^* \alpha_j^* K(x_i, x_j; \beta)} \quad (2.26)$$

従って, サポートカーネル ($\beta_k > 0$) は, 要素カーネル単独による M 個のサポートベクターマシン中で $S_k(\alpha^*)$ が最小で, サポートカーネルマシンの $S(\alpha^*; \beta^*)$ に等しい. 同時に, マージン $\sqrt{\sum_{i,j} \alpha_i^* \alpha_j^* \frac{K_k(x_i, x_j)}{R_k^2}}^{-1}$ が最大で, サポートカーネルマシンのマージン $1/\gamma^*$ と等しい.

なお, $\max_{\alpha} S_k(\alpha) \geq S_k(\alpha^*) \geq S(\alpha; \beta)$ であるため, サポートカーネルマシンによる汎化誤差指標 $S(\alpha; \beta)$ は, 要素カーネル単独によるサポートベクターマシンの汎化誤差指標 $\max_{\alpha} S_k(\alpha)$ より小さい. このため, サポートカーネルマシンによる判別の方が要素カーネル単独によるサポートベクターマシンによる判別より判定精度が良いことが期待できる.

ただし, あくまでも汎化誤差自身ではなく, 汎化誤差指標の最小化に過ぎないため, 実際には, 単独の要素カーネルの方がサポートカーネルマシンより汎化誤差が小さい場合もある.

2.2.3 等価問題

Bach らによるサポートカーネルマシンの最初の定式化 [13] では, 合成特徴ベクトル $\phi(x)$ は前節とは異なる以下の形で与えられた.

$$\phi(x) = (\phi_1(x), \dots, \phi_M(x))^{\top}$$

さらに, 判別面の垂線方向 w は以下により与えられた.

$$w = (w_1, \dots, w_M)^{\top}$$

また, R_k は各要素カーネルに対する事前重み (priors) として扱われ, 問題 2.2.1(SKM 問題) と等価な以下の形で提示された.

問題 2.2.2 (SKM_ORIGINAL)

$$\begin{aligned} \min_{\substack{w_k, b \\ \xi_i \geq 0}} \quad & \frac{1}{2} \left(\sum_k R_k \|w_k\| \right)^2 + C \sum_i \xi_i & (2.27) \\ \text{subject to} \quad & y_i \left(\sum_k w_k^\top \phi_k(x_i) + b \right) \geq 1 - \xi_i, \quad i = 1, \dots, n \end{aligned}$$

問題 2.2.2 (SKM_ORIGINAL) では, 問題 2.1.1 (SVM 主問題) でのマージン $1/(R \|w\|)$ の代わりに, 各要素カーネルのマージンの調和平均 $1/(\sum_k R_k \|w_k\|)$ を用いる形になっている.

2 次錘 ($\|w_k\| \leq t_k$) を与える変数 t_k と, $u \geq \sum_k R_k t_k$ なる u を導入すると, 問題 2.2.2(SKM_ORIGINAL) は, 2 次錐計画 (SCOP; Second Order Conic Programming) となる.

問題 2.2.3 (SKM_SCOP)

$$\begin{aligned} \min_{\substack{t_k \geq \|w_k\| \\ u \geq \sum R_k}} \min_{\substack{w_k, b, \xi \\ \xi_i \geq 0}} \quad & \frac{1}{2} u^2 + C \sum_i \xi_i & (2.28) \\ \text{subject to} \quad & y_i \left(\sum_k w_k^\top \phi_k(x_i) + b \right) \geq 1 - \xi_i, \quad i = 1, \dots, n \end{aligned}$$

問題 2.2.1(SKM 問題) は, この他にも以下の 3 つの問題と等価である.

問題 2.2.4 (SKM_BACH)

$$\max_{\substack{\alpha_i \in [A_i, B_i], \sum \alpha_i = 0 \\ R_k \gamma \geq \|\sum_i \alpha_i y_i \phi_k(x_i)\|}} \sum_i \alpha_i y_i - \frac{1}{2} \gamma^2 \quad (2.29)$$

問題 2.2.5 (SKM_DIRECT)

$$\begin{aligned} \min_{\substack{\beta_k \geq 0 \\ \sum \beta_k = 1}} \min_{\substack{\tilde{w}_k, b \\ \xi_i \geq 0}} & \frac{1}{2} \sum_k \beta_k (R_k \|\tilde{w}_k\|)^2 + C \sum_i \xi_i \\ \text{subject to} & y_i \left(\sum_k \beta_k \tilde{w}_k^\top \phi_k(x_i) + b \right) \geq 1 - \xi_i, \quad i = 1, \dots, n \end{aligned} \quad (2.30)$$

問題 2.2.6 (SKM_BetaFree)

$$\max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} \min_k \left(\sum_i \alpha_i y_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j \frac{K_k(x_i, x_j)}{R_k^2} \right) \quad (2.31)$$

問題 2.2.4 (SKM_BACH) は, [13] において後述する Moreau-Yoshida-正則化による解法で用いられた。

問題 2.2.5 (SKM_DIRECT) は, 問題 2.2.1 (SKM 問題) において第 k 要素に対するサポートベクターマシン双対問題をサポートベクターマシン主問題に置き換えたものである

問題 2.2.6 (SKM_BetaFree) に対しては, 式 (2.25) で示したよう, 下記の弱双性は成立するが強双対性は成立しない。

$$\min_k \left(\max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} S_k(\alpha) \right) \geq \max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} \left(\min_k S_k(\alpha) \right) = S(\alpha^*; \beta^*)$$

定理 2.2.1 (SKM 等価問題)

問題 2.2.1 (SKM 問題), 問題 2.2.2 (SKM_ORIGINAL), 問題 2.2.3 (SKM_SCOP), 問題 2.2.5 (SKM_DIRECT), 問題 2.2.4 (SKM_BACH), 問題 2.2.6 (SKM_BetaFree) は等価である。

なお, 問題 2.2.2 の w_k は, 問題 2.2.5 の $\beta_k \tilde{w}_k$ に等しい。

証明

- (1) 問題 2.2.2 と問題 2.2.3 の等価性: 自明
- (2) 問題 2.2.3 と問題 2.2.4 の等価性

問題 2.2.3 を b, ξ_i について双対化する.

$$\begin{aligned}
 & \min_{\substack{t_k \geq \|w_k\| \\ u \geq \sum R_k}} \min_{\substack{w_k, b \\ \xi_i \geq 0}} \frac{1}{2} u^2 + C \sum_i \xi_i \\
 & - \min_{\alpha_i y_i \geq 0} \sum_i \alpha_i y_i (y_i (\sum_k w_k^\top \phi_k(x_i) + b) - 1 + \xi_i) \\
 & = \min_{\substack{w_k, t_k \geq \|w_k\| \\ u \geq \sum R_k t_k}} \max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} \frac{1}{2} u^2 + \sum_i \alpha_i y_i - \sum_k w_k^\top \underbrace{\sum_i \alpha_i \phi_k(x_i)}_{R_k z_k(\alpha)}
 \end{aligned}$$

2 次錘 $\|w_k\| \leq t_k$ の性質から, $\|\lambda_k\| \leq \mu_k$ なる任意の定数 μ_k とベクトル λ_k について $u_k t_k \geq \lambda_k^\top w_k$ (等号成立は, $t_k \lambda_k = \mu_k w_k$) が成立するので,

$$\begin{aligned}
 & \min_{\substack{w_k, t_k \geq \|w_k\| \\ u \geq \sum R_k t_k}} \max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} \frac{1}{2} u^2 + \sum_i \alpha_i y_i - \sum_k R_k z_k(\alpha)^\top w_k \\
 & = \min_{\substack{u, t_k, w_k \\ t_k \geq \|w_k\|}} \max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} \left\{ \frac{1}{2} u^2 + \sum_i \alpha_i y_i - \sum_k R_k z_k(\alpha)^\top w_k \right. \\
 & \quad \left. - \min_{\gamma \geq 0} \gamma (u - \sum R_k t_k) - \min_{\substack{\mu, \lambda \\ \mu_k \geq \|\lambda_k\|}} \sum_k (\mu_k t_k - \lambda_k^\top w_k) \right\} \\
 & = \min_{\substack{u, t_k, w_k \\ t_k \geq \|w_k\|}} \max_{\substack{\gamma \geq 0, \alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0, \mu_k \geq \|\lambda_k\|}} \left(\begin{aligned} & \left(\frac{1}{2} u^2 - \gamma u \right) + \sum (R_k \gamma - \mu_k) t_k \\ & + \sum_k (\lambda_k - R_k z_k(\alpha))^\top w_k + \sum_i \alpha_i y_i \end{aligned} \right) \\
 & = \max_{\substack{\gamma \geq 0, \alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0, \mu_k \geq \|\lambda_k\|}} \min_{\substack{u, t_k, w_k \\ t_k \geq \|w_k\|}} \left(\begin{aligned} & \left(\frac{1}{2} u^2 - \gamma u \right) + \sum (R_k \gamma - \mu_k) t_k \\ & + \sum_k (\lambda_k - R_k z_k(\alpha))^\top w_k + \sum_i \alpha_i y_i \end{aligned} \right) \\
 & = \max_{\substack{\gamma \geq 0, \alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} \max_{\substack{\lambda_k, \mu_k \\ \mu_k \geq \|\lambda_k\|}} -\frac{1}{2} \gamma^2 + \sum_i \alpha_i y_i \\
 & \quad \text{where } u = \gamma, \mu_k = R_k \gamma, \lambda_k = R_k z_k(\alpha) \\
 & = \max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} \max_{\gamma \geq \|z_k(\alpha)\|} -\frac{1}{2} \gamma^2 + \sum_i \alpha_i y_i
 \end{aligned}$$

(3) 問題 2.2.4 と問題 2.2.1 との等価性

$$\begin{aligned}
& \max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} \max_{\gamma} -\frac{1}{2}\gamma^2 + \sum_i \alpha_i y_i + \min_{\beta_k \geq 0} \sum_k \frac{1}{2}\beta_k (\gamma^2 - \|z_k(\alpha)\|^2) \\
&= \min_{\substack{\beta_k \geq 0 \\ \sum \beta_k = 1}} \max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} \sum_i \alpha_i y_i - \frac{1}{2} \sum_k \beta_k \|z_k(\alpha)\|^2 \\
&= \min_{\substack{\beta_k \geq 0 \\ \sum \beta_k = 1}} \max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} \underbrace{\sum_i \alpha_i y_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j \sum_k \beta_k \frac{K_k(x_i, x_j)}{R_k^2}}_{S(\alpha; \beta)}
\end{aligned}$$

(4) 問題 2.2.1 と問題 2.2.5 の等価性: 自明

(5) 問題 2.2.1 と問題 2.2.6 の等価性

線形計画の性質から

$$\min_k S_k(\alpha) = \min_{\substack{\beta_k \geq 0 \\ \sum \beta_k = 1}} \sum_k \beta_k S_k(\alpha)$$

従って,

$$\begin{aligned}
\min_{\substack{\beta_k \geq 0 \\ \sum \beta_k = 1}} \max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} \sum_k \beta_k S(\alpha) &= \max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} \min_{\substack{\beta_k \geq 0 \\ \sum \beta_k = 1}} \sum_k \beta_k S(\alpha) \\
&= \max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} \min_k S_k(\alpha)
\end{aligned}$$

(6) 問題 2.2.5 と問題 2.2.2 の対応:

問題 2.2.5 では以下が成立する.

$$\begin{aligned}
\tilde{w}_k &= \sum_i \alpha_i \frac{\phi_k(x_i)}{R_k^2}, \quad R_k \|\tilde{w}_k\| = \gamma^* \quad (\beta_k > 0) \\
\gamma^{*2} &= \sum_k \beta_k (R_k \|\tilde{w}_k\|)^2 = \left(\sum_k \beta_k R_k \|\tilde{w}_k\| \right)^2 \\
&= \left(\sum_k R_k \|\beta_k \tilde{w}_k\| \right)^2
\end{aligned}$$

このため, 問題 2.2.2 の w_k と, 問題 2.2.5 の $\beta_k \tilde{w}_k$ が対応する.

□

2.2.4 代表的バッチ解法

サポートカーネルマシンに対する (バッチ型) 解法としては, 2 次錐問題への半正定値問題用汎用パッケージの適用, Moreau-Yoshida 正則化に基づく近似解法 [13], 線形半無限計画に基づくサポートベクターマシンと線形計画の繰り返しによる解法 [23] などが知られている.

Moreau-Yoshida-正則化 SKM

Bach らは, サポートカーネルマシン問題を 2 次錐問題 (SKM-SOCP) として提示した. 本節の解法は, 通常の SCOP の求解に用いられる半正定値問題 (semi-definite programming; SDP) パッケージでは解きづらい大規模な問題に対しても解を得られる手法として Bach らにより提案された.

基本的考え方は, まず 2 パラメータ (α, β) のサポートカーネルマシン問題を, 1 パラメータ (α) の最適化問題で近似し, この近似解 α から対応する β の近似値を得る. 近似の精度は制御変数 a_k により制御し, 順次 1 パラメータ最適化問題の近似精度を高めて, 最終的にサポートカーネルマシン問題の解を得る.

問題 2.2.4(SKM_BACH) において, $\mu_k = R_k \gamma \geq R_k \|z_k(\alpha)\|$ となる点に着目し, 1 パラメータ問題として下記を用いる. $a_k \rightarrow 0$ ($\forall k$) とした時, 問題 2.2.4(SKM_BACH) と等価となる.

$$\max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} \underbrace{\max_{\mu_k \geq R_k \|z_k(\alpha)\|} \sum_i \alpha_i y_i - \frac{1}{2} \gamma^2 - \frac{1}{2} \sum_{k \in J} \frac{(\mu_k - \gamma R_k)^2}{a_k^2}}_{G(\alpha)} \quad (2.32)$$

$$\begin{aligned} G(\alpha) &= \max_{\mu_k \geq R_k \|z_k(\alpha)\|} \sum_i \alpha_i y_i - \frac{1}{2} \gamma^2 - \frac{1}{2} \sum_k \frac{(\mu_k - \gamma R_k)^2}{a_k^2} \\ &= \max_{\gamma, \mu_k} \sum_i \alpha_i y_i - \frac{1}{2} \gamma^2 - \frac{1}{2} \sum_k \frac{(\mu_k - \gamma R_k)^2}{a_k^2} \\ &\quad + \min_{\delta_k \geq 0} \sum_k \delta_k \frac{\mu_k - R_k \|z_k(\alpha)\|}{a_k^2} \end{aligned}$$

であるので, $G(\alpha)$ を与える (γ, μ_k) は,

$$\begin{aligned}\mu_k(\alpha) &= R_k \gamma(\alpha) + \delta_k = \begin{cases} R_k \gamma(\alpha) & (k \notin J) \\ R_k \|z_k(\alpha)\| & (k \in J) \end{cases} \\ \gamma(\alpha) &= \gamma(\alpha, J)\end{aligned}$$

ただし

$$J \equiv \{k | \delta_k > 0\}, \quad \gamma(\alpha, J) \equiv \frac{\sum_{k \in J} \frac{R_k^2}{a_k^2} \|z_k(\alpha)\|}{1 + \sum_{k \in J} \frac{R_k^2}{a_k^2}}$$

なお $\|z_k(\alpha)\|$ に関して降順に k_1, \dots, k_M として, 下記が成立する l^* を求めると, $J = J_{l^*}$ として $J, \gamma(\alpha), \mu_k(\alpha)$ が順次求まる.

$$\begin{aligned}J_l &= \{k | \|z_k(\alpha)\| > \|z_{k_l}(\alpha)\|\} \\ \gamma(\alpha, J_l) &\geq \|z_{k_l}(\alpha)\| \quad \text{for } l = 1, \dots, l^* \\ \gamma(\alpha, J_{l^*+1}) &< \|z_{k_{l^*+1}}(\alpha)\|\end{aligned}$$

更に, J の要素はサポートカーネルに相当し, β_k の推定値は

$$\hat{\beta}_k = \frac{R_k \delta_k}{a_k^2 \gamma(\alpha)} = \begin{cases} \frac{R_k^2}{a_k^2} \left(\frac{\|z_k(\alpha)\|}{\gamma(\alpha)} - 1 \right) & (k \in J) \\ 0 & (k \notin J) \end{cases}$$

問題 (2.32) の解 α を得るには, SMO 型のアルゴリズムにより部分問題

$$\max_{\lambda} G(\alpha^0 + \lambda(\mathbf{e}_i - \mathbf{e}_j))$$

の解を λ に関して線形探索して求めることを繰り返す.

サポートベクターマシンの場合の部分問題に解析解が存在するため, 高速に解を得られたが, 本解法では部分問題を解くために線形探索を用いるので, 必ずしも高速な解法とはなっていない.

SKM-ILP

問題 2.2.1(SKM 問題) は, 次の線形半無限計画 (semi-Infinite Linear Programming) と等価である.

問題 2.2.7 (SKM-ILP 問題)

$$\begin{aligned} & \min_{\substack{\theta, \beta \geq 0 \\ \sum_k \beta_k = 1}} \theta & (2.33) \\ & \text{subject to } \sum \beta_k S_k(\alpha) \leq \theta \quad \forall \alpha \in \{\alpha \mid \alpha_i \in [A_i, B_i], \sum_i \alpha_i = 0\} \end{aligned}$$

問題 2.2.7 の解はアルゴリズム 2.2.1 により得られる。本アルゴリズムでは、適当な α に対する制約 ($\sum_k \beta_k S_k(\alpha) \leq \theta$) を順次付加しながら β に関する線形計画を繰り返し解くことで解を改善する。

```

1:  $t \leftarrow 0$ 
2:  $\mathcal{CS} \leftarrow \{\beta_k \geq 0 \mid k = 1, \dots, M\} \cup \{\sum \beta_k = 1\}$ .
3:  $\alpha^0 \leftarrow 0$ .
4:  $\beta^0 \leftarrow (1/M, \dots, 1/M)$ .
5:  $\theta^0 \leftarrow S(\alpha^0; \beta^0)$ .
6: loop
7:    $t \leftarrow t + 1$ 
8:   if  $S(\alpha; \beta^{t-1}) > \theta^{t-1}$  なる  $\alpha$  が存在 then
9:      $\alpha^t \leftarrow \alpha$ . //  $\alpha^t$  の選択
10:  else
11:    exit loop // 収束
12:  end if
13:   $S_k^t \leftarrow S_k(\alpha^t), k = 1, \dots, M$ .
14:   $\theta_0^t \leftarrow \sum \beta_k^{t-1} S_k^t$ . //  $\theta_0^t = S(\alpha^t; \beta^{t-1})$ 
15:   $c_t \leftarrow [\sum_k \beta_k S_k^t \leq \theta]$ . //  $\alpha^t$  に対応する制約
16:   $\mathcal{CS} \leftarrow \mathcal{CS} \cup \{c_t\}$ . // 冗長な制約は削除.
17:   $(\theta^t, \beta^t) \leftarrow \operatorname{argmin}_{(\theta, \beta) \in \mathcal{CS}} \theta$  // LP 解  $\theta^t = S(\alpha^t; \beta^t)$ .
18: end loop
19: return  $(\alpha^{t-1}, \beta^{t-1})$ .

```

アルゴリズム 2.2.1: SKM-ILP の基本解法

t 回目の線形計画の解は,

$$(s_t, \beta) = \operatorname{argmin}_{\substack{\sum \beta_k = 1 \\ \beta_k \geq 0}} \max_{s \in \{1, \dots, t\}} S(\alpha^s; \beta)$$

とすると, $(\theta^t, \beta^t) = (S(\alpha^{s_t}, \beta^t), \beta^t)$ となる.

[22, 23] は, 本基本解法に基づくサポートカーネルマシンのバッチ解法である. β を固定した時, $\max_{\alpha} S(\alpha, \beta)$ が θ の下限を与える. そこで, [23] では, サポートベクターマシン問題 $\max_{\alpha} S(\alpha, \beta^{t-1})$ の解を, アルゴリズム 2.2.1 の α^t の選択 (9 行目) での α^t とする. また, [22] では, β が最適解と異なる可能性が高い初期段階においては, 全サンプルに対するサポートベクターマシンではなく, チャンクされたデータのみに対する部分サポートベクターマシン問題の解を α^t とすることで α^t の計算コストを抑えた chunked MKL を提案している.

線形計画解 (θ^t, β^t) は通常, 直前の線形計画のアクティブ制約から SIMPLEX 法を始める事で通常数回の繰り返して新たな解が得られる.

α^t の選択に関しては, 最適な β が β^{t-1} の近傍にある場合, $K(\cdot, \cdot; \beta^{t-1})$ をカーネルとするサポートベクターマシンの解が最適な制約を与える. サポートベクターマシンの計算コストは通常大きい, 逐次的なサポートベクターマシン・アルゴリズム (incremental SVM) を用いることで計算コストを削減することが可能である.

線形計画とサポートベクターマシン問題という大規模問題の解法を用いるため, カーネル数, データ数の多い大規模サポートベクターカーネル問題に適用可能な手法である.

2.3 能動学習とサンプリング戦略

入力データ x は容易に得られるが, 機械学習を行なうために必要なラベル y を得ることは容易でない場合がしばしば生じる.

例えば, 文書分類を行なう分類器を学習したい場合, 個々の文書 (入力 x) は容易に得られる. しかし, 各文書の分類結果 (ラベル y) は, 人間が判断して与える必要がある. 設備状態の診断基準を学習したい場合も同様に各種センサーによる測定値など (入力 x) は多数得られているが, 機器状態が正常か異常かなどの判定結果 (ラベル y) は, 専門家に判断をしてもらわないと得られない場合がしばしば生じる.

このように, 入力データ x は多数あるが, そのラベル y を得るコストが高い場合に機械学習を適用するには, 機械学習で得られる判別器の性能が高くなるようなサンプルを入力データの中から学習アルゴリズム自身が選び出して, これらに対してのみ専門家などにラベル付けをしてもらうことで, 少ないラベル付けで高い精度の判別器を得ることが考えられる. 能動学習は, このような目的のために提案された.

本節では, 能動学習の分類と, サポートベクターマシンで用いられる代表的サンプリング戦略について述べる.

2.3.1 能動学習

能動学習には, query learning と selective sampling という大きく 2 種類のアプローチがある. query learning [1] は, 学習アルゴリズムが新しいサンプルを生成する能力を持ち, そのラベルを求める. 一方, selective sampling [9] では, 学習アルゴリズムは新たなサンプルは生成できず, 一定のサンプル集合からラベル付け対象を選択する (pool-based モデル), または順次提示されるサンプル列の各サンプルに対してラベル付けするかどうかを決定する (stream-based モデル).

本研究では, pool-based の selective sampling による能動学習を対象とする

能動学習の基本アルゴリズムを次に示す.

- 1: 学習器 A , ラベル付けするサンプル数 N を指定.
- 2: ラベルなしデータ集合 \mathcal{U} と, 正例, 負例数個からなるラベル付きデータ集合 \mathcal{L} の初期値を設定.
- 3: **while** $|\mathcal{L}| \leq N$ **do**
- 4: ラベル付きデータ集合 \mathcal{L} に対して学習器 A を適用し, 判別器 M を得る.
- 5: M に基づくサンプリング戦略 SS に従いサンプル i を選択し, ラベル y_i を取得.
- 6: $(\mathcal{L}, \mathcal{U}) \leftarrow (\mathcal{L} \cup \{i\}, \mathcal{U} - \{i\})$.
- 7: **end while**

アルゴリズム 2.3.1: 能動学習基本アルゴリズム

学習器としては, ニューラルネットワーク [10], 確率的分類器 [16], Nearest Neighbor [17], 決定木など各種学習アルゴリズムが用いられてきた. ま

た, Query by Committee[21, 14] は複数の分類器による結果の多数決を分類結果とする分類器を学習する.

特に, 各種実問題でのサポートベクターマシンの成功を受けて能動サポートベクターマシンが複数の研究者 [4, 7, 25] により提案された. そして, 線形カーネルを用いた能動サポートベクターマシンによる文書分類の研究などが行われている [25].

2.3.2 能動学習におけるサンプリング戦略

能動学習における基本的なサンプリング戦略としては, 不確実性に基づくサンプリング (uncertainty sampling)[16], 情報利得に基づくサンプリング (information-based sampling)[18], 複数の学習器を用いる query by committee (QBC)[21, 14] が提案されている.

Lewis ら [16] は, 確率的分類器を学習する単一の学習器を用いる場合について, 判定結果が最も不確実な (確率が 0.5 近い) サンプルを選択する不確実性に基づくサンプリング (uncertainty sampling) を提案した. 本戦略は, サンプルに対する不確実性を最も減少させる (情報利得最大) ようにサンプルを選択するが, 分類器が持つパラメータの不確実性を最も減少させるものとは限らない.

Mackay[18] は, ラベル取得によるパラメータの不確実性を減少させる, すなわち, 情報利得が最大となるサンプルを選択する 情報量に基づくサンプリングを提案した.

Seung ら [21] は, 複数の分類器 (committee) の多数決により判定を行なう際に, 判別結果が一致しないサンプルを選択する戦略を用いる Query by Committee を提案した. Freund ら [14] は, ラベルリングによる情報利得の VC 次元に基づく分析により, データを分離可能な判別器が存在する場合には, Query By Committee ではラベル付け数の増加について指数的に汎化誤差が減少することを理論的に示した.

能動サポートベクターマシンでは, 判別面に最も近いサンプルを選択する MARGIN 戦略が一般的である. 本戦略は, uncertainty sampling の一種であり, Davis ら [10] により 1992 年にニューラルネットワーク (Perceptron) の学習に対して適用されている. 能動サポートベクターマシンでの MARGIN 戦略の使用は, 前述のように複数の研究者 [4, 7, 25] によりほぼ同時に提案された.

Campbell ら [4] らは, Freund らと同様に, VC 次元による分析から, サ

ポートベクターマシンにより分離可能な問題においては, MARGIN 戦略によるラベル付け数の増加につれて指数的に汎化誤差が減少することを, 理論的に示した. Tong ら [25] は, より直観的な以下の説明を与えている.

ハードマージン・サポートベクターマシンは, $(w^0, b^0) = \frac{1}{\|w\|}(w, b)$ と置くことにより, 下記と等価となる.

$$\max_{\|w^0\| \leq 1, b} \min_i y_i (\phi(x_i)^\top w^0 + b^0) \quad (2.34)$$

最適解が正でない場合は, 分離可能な問題ではない. 分離可能な場合には, 可能解 (w^0, b^0) は全サンプルで $(y_i \phi(x_i)^\top) w^0 + y_i b^0 > 0$ となる. 制約 $(y_i \phi(x_i)^\top) w^0 + y_i b^0 > 0$ は (w^0, b^0) に関する超半平面で, 全サンプルに関する制約は (w_0, b_0) に関する超多面体となる. 最も効率的なサンプリング法は, この超多面体 Λ と超円柱 $\{(w, b) \mid \|w\| = 1\}$ の共通部分である可能解領域の体積を 2 等分する超平面に対応するサンプルを選択することである. このような 2 分探索により指数的にパラメータ (w^0, b^0) のあいまい性は減少し, 汎化誤差も減少する.

今, RBF カーネルなど, $K(x_i, x_i) = \phi(x_i)^\top \phi(x_i) = R^2$ で一定, すなわち, $\phi(x_i)$ が超球面上に分布する場合を考える. この時, 点 (w^0, b^0) と超平面 $\phi(x_i)^\top w^0 + b^0 = 0$ の距離は, $|\phi(x_i)^\top w^{0*} + b^{0*}| / \sqrt{R^2 + 1}$ である. そして, 問題 2.1.1 (SVM 主問題) の最適解 (w^{0*}, b^{0*}) は, 可能解の中で, 最も近い超平面 $\phi(x_i)^\top w^{0*} + b^{0*} = 0$ との距離が最大のものである. 最も近い超平面は複数存在し, いずれも等距離で, 各サポートベクター $(\phi(x_i), y_i)$ に対応する. 従って, (w^{0*}, b^{0*}) は, 超多面体 Λ に内接する最大の超球の中心である. 最適判別面 $\phi(x)^\top w^{0*} + b^{0*} = 0$ 上のサンプル x_{new} に対応する制約 $(\phi(x_{new})^\top w^0 + b^0 = 0)$ は, 内接球の中心 (w^{0*}, b^{0*}) を通り, 内接球を 2 等分する. 外接する可能解領域もほぼ 2 等分されると期待されるので, 最適判別面に近いサンプルの選択は良いサンプリング戦略となる.

従って, 長期的な収束性の面からは優れているため, 能動サポートベクターマシンでは, MARGIN 戦略が一般的な戦略となっている.

2.3.3 サポートベクターマシンにおけるサンプリング戦略

能動サポートベクターマシンにおいて最も一般的な戦略は, MARGIN 戦略である. 本戦略は, サンプル数の増加につれて指数的に汎化誤差を減少させるため, 長期的に優れた戦略である. しかし, 能動学習の初期においては必ずしも最良の性能を持つものではない.

能動サポートベクターマシンにおける初期性能を考慮した代表的サンプリング戦略としては, MARGIN 以外に RANDOM, KFF, COMB, SHIFT の4つが挙げられる.

RANDOM はランダムにサンプルを選択する. MARGIN が判別面に最も近いサンプルを選択するのに対して, KFF は学習済みサンプルのいずれにも類似しないサンプルの選択を行う. COMB, SHIFT は, 共に, 初期戦略として有効な KFF から, 学習経過を見ながら, 最終フェーズで有効な MARGIN へと選択戦略を切り替えることを狙った戦略である.

以下では, 学習済みデータの集合を L , ラベルなしデータの集合を U と記す.

RANDOM

RANDOM は, ラベルなしデータ U 中からランダムにサンプルを選択する.

選択戦略の有効性の評価のためのベースラインとなるが, 必ずしも性能が低いとは限らない.

データの分布密度に比例して一定個を選択するため, 低密度領域のサンプルは選ばれにくく, 高密度領域のサンプルが選択され易い. このため, 学習結果は, 低密度領域での正解率は保障されないが, 高密度領域での正解率が高くなりやすい. このため, 比較的少ないサンプリングでも全体的な正解率が高くなり易い.

MARGIN 戦略

SIMPLE 戦略とも呼ばれる. ラベルなしデータ U 中から判別面に最も近いサンプルを選択する [4, 25]. 判別面の近傍探索 (exploration) の代表的手法である.

$$\operatorname{argmin}_{x \in U} |f(x)| = \operatorname{argmin}_{x \in U} \left| \sum_{s \in L} \alpha_i K(s, x) + b \right|$$

能動学習がある程度進んだ段階で (サポートベクターマシンに対する能動学習では, 現在の α が最適値に近い場合に), 効率的に正解率を向上させることができる手法として知られている. MARGIN 戦略では, 現在の判別境界の近くの点をサンプリングするので判別境界の詳細な形状を知ることができるためである.

判別面の方向 w が安定している時, MARGIN 戦略は切片 b の 2 分探索となる.

KFF (Kernel Farthest First) 戦略

MARGIN 戦略は, 能動学習が進んだ段階では効率が良い. しかし, 現在の α が最適解と大きく異なる場合, 実際の判別境界は, 現在の判別境界の近傍点ではなく, 別にある可能性が高い. このような状況では, より広い範囲の情報を収集することが重要である.

KFF 戦略は, 代表的な遠征戦略で, Baram らにより提案された [3]. 学習済みの訓練データのいずれとも最も似ていないサンプルを選択する.

$$\begin{aligned} & \operatorname{argmax}_{x \in U} \min_{s \in L} \|\phi(x) - \phi(s)\|^2 \\ & = \operatorname{argmax}_{x \in U} \min_{s \in L} (K(x, x) - 2K(x, s) + K(s, s)) \end{aligned}$$

KFF 戦略では, ラベルに寄らずにサンプリングが決定される. KFF 戦略では, サンプリングされた点が特徴空間 $\phi(X)$ 上でなす網の目の大きさを極力均等に保ちつつ, 順次細くなるようにサンプリングを進めていく.

XOR-型問題の初期フェーズなどでは, 有効であることが知られている. ただし, 学習が進んだ段階では性能が悪い.

COMB 戦略

KFF 戦略は能動学習の初期に, MARGIN 戦略は学習がある程度進んだ異なる局面で有効となる. このため, 判別面からの遠征 (expedition) である KFF 戦略から, 判別面の近傍探索 (exploration) である MARGIN への切り替えが重要となる.

両者の切り替え手法として, 戦略選択を multi-armed bandit 問題と捉える COMB [3], 次に述べる判別面の安定性に基づく切り替え確率の更新による SHIFT [19] とが提案されている.

COMB は, RANDOM, MARGIN, KFF などの個別戦略のサンプル選択を総合的に評価してサンプルを選択する. メタ戦略のオンライン学習手法 EXP4 [2] により, 選択サンプルによる判別性能の改善度に応じて, 総合的評価に用いる各戦略の有効性の評価値を毎回更新し, 各回での総合的評価を変化させる [3].

-
- 1: 各戦略 $k = 1, \dots, M$ による, 各ラベルなしサンプル i に対する有効性予想確率 $b_i^k(t)$, $\sum_i b_i^k = 1$ を計算.
 - 2: 全戦略 k での有効性予測確率 $b_i^k(t)$ が閾値以下のサンプル i は選択候補から除外 (残った候補数 n_e).
 - 3: 各戦略 k の有効性評価 w_k による加重平均により, 各サンプルに対する総合有効性予測確率 $p_i(t)$ を計算.

$$p_i(t) = (1 - \gamma) \sum_j w_j b_i^j(t) + \gamma/n_e \text{ where } \gamma = \sqrt{\frac{n_e \log M}{(e-1) g_{max}}}$$
 - 4: 総合有効性予測確率 $p_i(t)$ に従ってランダムにサンプル i_t を選択し, ラベル y_{i_t} を取得.
 - 5: 追加訓練データ i_t を用いて再学習を行い, 判別性能の改善度合い $r_{i_t}(t)$ を評価.
 - 6: 各戦略 k の有効性評価 w_k の更新.

$$w_k(t+1) \propto w_k(t) \exp(\gamma/n_e \cdot r_{i_t}(t)/p_{i_t} \cdot b_{i_t}^k(t)), \sum w_k = 1.$$

アルゴリズム 2.3.2: COMB

Require: U :ラベルなしデータ, L : ラベル付データ, O :オラクル

Require: λ, ϵ, A :学習アルゴリズム

```

 $p_0 \leftarrow 1.$ 
for  $t = 1, \dots$  do
   $r \leftarrow$  一様乱数  $\in [0, 1].$ 
  if  $r < p_{t-1}$  then
     $x \leftarrow$  KFF( $U, L$ ) //探索 (exploration)
  else
     $x \leftarrow$  MARGIN( $U, h_{t-1}$ ). //詳細化 (exploitation)
  end if
   $y \leftarrow O(x).$ 
   $L \leftarrow L \cup (x, y).$ 
   $U \leftarrow U - x.$ 
   $h_t \leftarrow A(L).$ 
   $p_t \leftarrow \max(\min(p_{t-1} \exp(d(h_t, h_{t-1})\lambda), 1 - \epsilon), \epsilon).$ 
end for

```

アルゴリズム 2.3.3: SHIFT (Exploration-Based Active Learner)

t 回目のサンプル選択では, まず, 各戦略 k による各ラベルなしサンプル i の有効性予測 $b_i^k(t)$ を得る. 次に, 各戦略 k の有効性評価 $w_k(t)$ による加重平均に基づいて, サンプル i の総合有効性予測 $p_i(t)$ を得る. そして, $p_i(t)$ に従って, ラベルなしサンプルをランダムに選択する.

得られた新たな訓練データ $i_t(t)$ に基づいてサポートベクターマシンの再学習をした後, 戦略有効性 w_k の更新を行う. 更新では, 判別性能の改善度 $r(t)$ が大きいほど, また, サンプル i_t に他の戦略より高い有効性予測 $b_i^k(t)$ を与えた戦略ほど強化する. 以下の乗法的更新式を用いるため, 強化する戦略の選択性が高い.

$$w_k(t+1) \propto w_k(t) \exp(\gamma r_{i_t}(t)/n_e b_{i_t}^k/p_{i_t})$$

SHIFT 戦略

Osugi らによる Exploration-Based Active Learner [19] で用いられた戦略である.

SHIFT では, 学習結果が安定しない状況では KFF を, 安定してからは MARGIN を用いるという考え方にに基づき, MARGIN と KFF の 2 戦略を確率的に切り替える.

このため, 戦略の選択確率 $p(t)$ を学習結果 $h(t)$ の安定性の評価, すなわち, $h(t)$ と $h(t+1)$ の距離 $d(h_1, h_2)$ に基づき更新する. 以下の乗法的更新則を用いる.

$$p(t+1) = \max(\min(p(t) \exp(\lambda d(h(t+1), h(t))), 1 - \epsilon), \epsilon)$$

学習結果 $h(t)$ は t 回目の選択で得られたサポートベクターマシンによる全サンプルに対する評価値のベクトル $(f_t(x_1), \dots, f_t(x_n))^T$ で表現し, 学習結果間の距離は, このベクトル間の角度を補正して求める.

$$d(h(t), h(t+1)) = 3 - 4 \frac{|h(t)^T h(t+1)|}{\|h(t)\| \cdot \|h(t+1)\|}$$

ただし, KFF が初期に有効な XOR-型問題に対しては有効であるが, 通常は, MARGIN 戦略の方が有効か同等で, 有効性は限定的である.

2.3.4 最適化の視点からの MARGIN 戦略の意味

本節では, MARGIN 戦略は, サポートベクターマシン最適化の視点からは学習済みの (ラベル付き) サンプルと未学習の (ラベルなし) サンプルの

ペアに制限した場合の最大違反ペアの選択となっていることを示す.

双対ギャップの上限は最大違反ペア (i, j) の勾配差 $\delta = g_j - g_i$ に比例する. このため, SMO アルゴリズムでは, 最大違反ペア (i, j) を選び出し, その勾配差 $g_j - g_i$ が最小となるように α を更新することを繰り返した.

最大違反ペアではなくても, (潜在的な最大違反ペアの候補である) 勾配差が大きい違反ペア (i, j) を選び出し, 勾配差 $g_j - g_i$ が小さくなるように α を更新することは, 双対ギャップを小さくする上で, 有効な戦略となる.

そこで, 未学習サンプルと学習済みサンプルのペアからなる違反ペア (i, j) で最大の勾配差 $g_j - g_i$ を持つ未学習サンプルを, サンプルング候補とする戦略, すなわち, 以下の最大化問題を考える.

$$\max_{l \in \mathcal{U}} \left(\max_{j \in \mathcal{L}, \alpha_j < B_j, \alpha_l > A_l} g_j - g_l, \max_{\alpha_l < B_l, i \in \mathcal{U}, \alpha_i > A_i} g_l - g_i \right) \quad (2.35)$$

ただし, 未学習サンプル l については, $\alpha_l = 0$ と解釈する. 従って, $\alpha_l < B_l$, $\alpha_l > A_l$ となるのは, 各々 $y_l = 1$, $y_l = -1$ の場合に限られる. また, $\max_{j \in \mathcal{L}, \alpha_j < B_j} g_j$, $\min_{i \in \mathcal{L}, \alpha_i > A_i} g_i$ は, 学習済みサンプル \mathcal{L} 内での最大違反ペア (i_*, j_*) の勾配 g_{j_*} , g_{i_*} に対応する. 従って, $y_l = 1$ の時, (l, j_*) , $y_l = -1$ の時 (i_*, l) のペアを考えればよい.

切片 $b = \frac{1}{2}(g_{j_*} + g_{i_*})$, 勾配差 $\delta = g_{j_*} - g_{i_*}$ とすると, $g_{j_*} = b + \delta/2$, $g_{i_*} = b - \delta/2$ となる.

この時, 問題 (2.35) は下記と等価である.

$$\max_{l \in \mathcal{U}} \max \left(1 + \frac{\delta}{2} - y_l f(x_l), 0 \right)$$

未学習サンプル l のラベル $y_l = \pm 1$ が不明の時の min-max 戦略は

$$\max_{l \in \mathcal{U}} \min_y \max \left(1 + \frac{\delta}{2} - y_l f(x_l), 0 \right) = \max \left(1 + \frac{\delta}{2} - \min_{l \in \mathcal{U}} |f(x_l)|, 0 \right)$$

これは, MARGIN 戦略に一致する.

この時, 定理 2.1.2 に類似する以下の補題が成立する.

定理 2.3.1 (SMO 1 回による追加汎化誤差増加幅) 最大違反ペア (i_*, j_*) の勾配差 $\delta = g_{j_*} - g_{i_*}$ の時, 追加学習サンプル l の $y_l = 1$ の時違反ペア $(i, j) = (l, j_*)$, $y_l = -1$ の時 $(i, j) = (i_*, l)$ を選び, 部分問題 (2.11) の解により, α を更新する.

i, j が共に自由サポートベクターとなる時, 汎化誤差指標 $\max_{\alpha} S(\alpha)$ の増加幅は, 下記となる.

$$\Delta S_{i,\alpha} = \frac{R^2}{2D_{i,j}^2} (g_j - g_i)^2 = \frac{R^2}{2D_{i,j}^2} \max \left(1 + \frac{\delta}{2} - |f(x_i)|, 0 \right)^2 \quad (2.36)$$

ただし, $D_{i,j}^2 = \|\phi(x_j) - \phi(x_i)\|^2 = K(x_i, x_i) + K(x_j, x_j) - 2K(x_i, x_j)$.

2.4 サポートベクターマシンにおけるカーネル選択問題

2.4.1 バッチ版サポートベクターマシンでのカーネル選択

サポートベクターマシンは, 予め与えられたカーネル $K(\cdot, \cdot)$ に対して, 汎化誤差指標 (2.6) が最小の判別面を求めることができる. しかし, 適切なカーネルを選択しないと, 必ずしも高精度の判別を実現できない.

通常は, 候補となるカーネルを複数個準備し, 各カーネルに対する (何らかの) 汎化誤差に対する指標が最小のものを最適なカーネルとする.

実用的な汎化誤差の指標としては, n -fold クロスバリデーションが用いられることが多い. n -fold クロスバリデーションでは, 元データ L を n 群 L_1, \dots, L_n ($L_1 \cup \dots \cup L_n = L, L_1 \cap \dots \cap L_n = \phi$) に分割し, 元データ A から第 i 群をのぞいたデータ $L - L_i$ を訓練データ, 第 i 群 L_i をテストデータとして用いて, 学習結果によるテストデータの判定精度 p_i を求める. 各群 L_i に対する判定精度 p_i の平均を, 汎化誤差の指標とする.

一抜き法 (LOO; leave-one-out) では, 元データ L から i 番目のデータを除いたデータ $L - \{i\}$ を訓練データとして, i 番目のデータをテストデータとして, 学習した判別関数 $f_i(x)$ によるテストデータの判定 $f_i(x_i)$ の正否を求める. この操作を, 元データ中の各データ $i \in L$ に対して行い, 平均正答率を汎化誤差の指標とする.

サポートベクターマシンでは, 非サポートベクター ($\alpha_i = 0$) の有無は判別関数に影響しない. このため, 非サポートベクターがテストデータの場合には必ず正解となる. 従って, サポートベクター数が n_{sv} の時, LOO による平均誤答率は最大 $n_{sv}/|L|$ となる.

LOO による平均誤答率のより正確な近似として, スパン上界 (span-bound)[6] が提案されている (2.1.8 節参照).

Chapelleらは、スパン $S_p = \text{span}(p, SV - \{p\})$ を、特徴ベクトル空間で、 p 以外のサポートベクター $SV - \{p\}$ の内分点、外分点の集合と、 $\phi(x_p)$ との距離とした。この時、スパン上界は $\sum_{p \in SV} \alpha_p S_p^2$ により得られる。

更に、サポートベクター間のカーネル $K(x_i, x_j)$, $i, j \in SV$ を並べたカーネル行列を K_{SV} とする時、下記の H の逆行列の対角成分 $M_{i,i}^{-1}$ を S_i^2 の近似とする手法やその修正版などが提案されている [6].

$$H = \begin{bmatrix} K_{SV} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix}$$

ただし、カーネル候補数が多い時、一抜き法やクロスバリデーションは計算量が多大となる。

例えば、 m 個のデータ特徴から判別に有効な特徴集合を見つける問題は、 2^m 個ある入力特徴選択子 $\sigma \in \{0, 1\}^m$ 中で汎化誤差が最小の集合を見つける問題で、各入力特徴選択子 σ に対応して、特徴ベクトル $\phi(\sigma(x))$ 、カーネル $K(\sigma(x), \sigma(z))$ が存在するため、考慮するカーネル候補は 2^m 個となる。

また、RBF カーネルのパラメータ γ の最適値を求める問題は、 γ が正の実数上を変化できるので、カーネル候補は無数に存在する。

このような場合には、Radius-Margin 上界、または、その変形を用いる手法が提案されている。

特徴選択問題を扱った [27] では、Radius-Margin 上界を用いて以下を満たす特徴選択子 $\sigma \in \{0, 1\}^m$ を求める。

$$\min_{\sigma \in \{0, 1\}^\dagger} R^2(\sigma(L)) W^2(\sigma(L))$$

上記の解を得るため、 σ_i を総和が m の正の実数に緩和して、 $\lambda \rightarrow \infty$, $p \rightarrow 0$ と制御しながら、最急降下法により、下記に対する最適な σ を求めることを繰り返す。

$$\min_{\sigma \in \mathcal{R}^\dagger} R^2(\sigma(L)) W^2(\sigma(L)) + \lambda \sum_i \sigma_i^p \text{ subject to } \sum_i \sigma_i = m, \sigma_i \geq 0$$

また、最適な RBF のパラメータ γ の決定を扱った [8] では、微分可能な上界として、以下を提案し、最急降下法により最適な γ (および学習パラメータ C) を求める手法を提案している。

$$\begin{aligned} & 2(R_\gamma^2 + 1/C) \left(\sum_i y_i \alpha_i - \frac{1}{2} \alpha_i \alpha_j K_\gamma(x_i, x_j) \right) \\ & = 2(R_\gamma^2 + 1/C) \left(\frac{1}{2} \|w^2\| + C \sum_i \xi_i \right) \end{aligned}$$

これらの解法は、データ量が多いと計算に時間を要する。また、解が局所最適解となるため、完全な最適性を保障できないなどの問題がある。

2.4.2 能動サポートベクターマシンでのカーネル事前選択問題

対話的な文書検索などの能動サポートベクターマシンの代表的な適用分野では、通常、線形カーネルが有効とあることが予め知られている。

しかし、一般には、現在の対象データにどのカーネルが有効であるという事前知識はない。また、RBF カーネルが有効であるという事前知識がある場合でも、適切なカーネルパラメータ (γ 等) の値は不明であり、事前知識に基づいて対象問題を能動学習するために適したカーネルを 1 つ選択することは難しい。

また、前節で述べたサポートベクターマシンのカーネル選択手法は、いずれも、データのラベル y_i が既知の場合の手法であり、学習を始めるまではデータのラベル y_i が不明な能動学習には適さない。

従って、能動サポートベクターマシンで用いる適切なカーネルを能動学習開始前に特定することは困難である。

しかし、次の例題に見るように、能動サポートベクターマシンでは用いるカーネルの選択によっては、途中で判別境界が存在しなくなるなど、ラベル付けと学習が効率的に進まない事態が生じ得る。

このため、能動サポートベクターマシンを行なううえで、適切なカーネルの選択は非常に重要な問題となる。

例題

入力領域 $X = [-0.5, 0.5] \times [-0.5, 0.5]$ で、真の判別関数が、中心 $(0.0, -0.1)$ 、長軸半径 0.43 、長軸方向 $(0.24, 0.36)$ 、短軸半径 0.17 、短軸方向 $(0.12, 0.12)$ なる楕円の内側で $y = +1$ 、外側で $y = -1$ なる場合を考える。

学習用データは、ランダムに選択した X 内の 1000 点とした。

この時、いくつかのカーネルを用いた能動学習でのラベル付きデータ数の推移に伴う誤答率の変化を図 2.5 に、最終的に得られた判別面を図 2.4 に示す。誤答率は、 X の各辺を 40 等分した格子点 (計 $40 \times 40 = 1600$ 点) での判別結果の誤答率である。

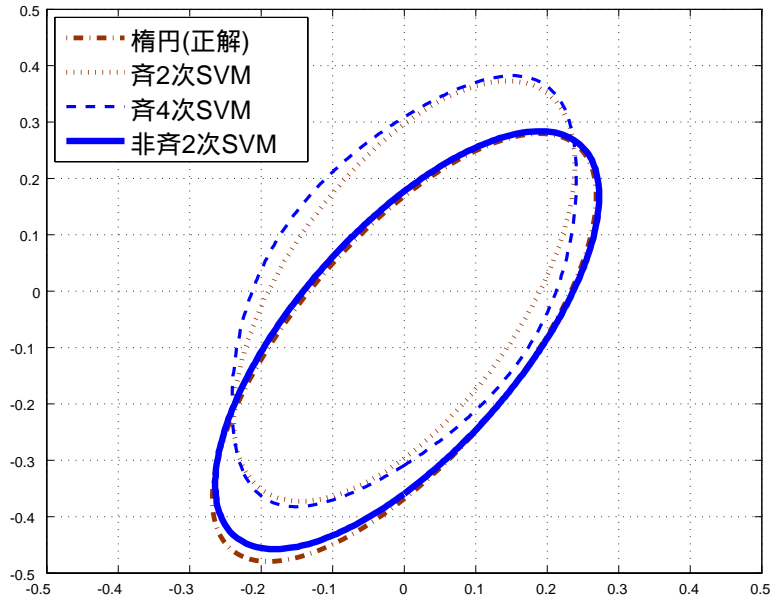


図 2.4: 楕円境界に対する斉次多項式能動 SVM の判別面

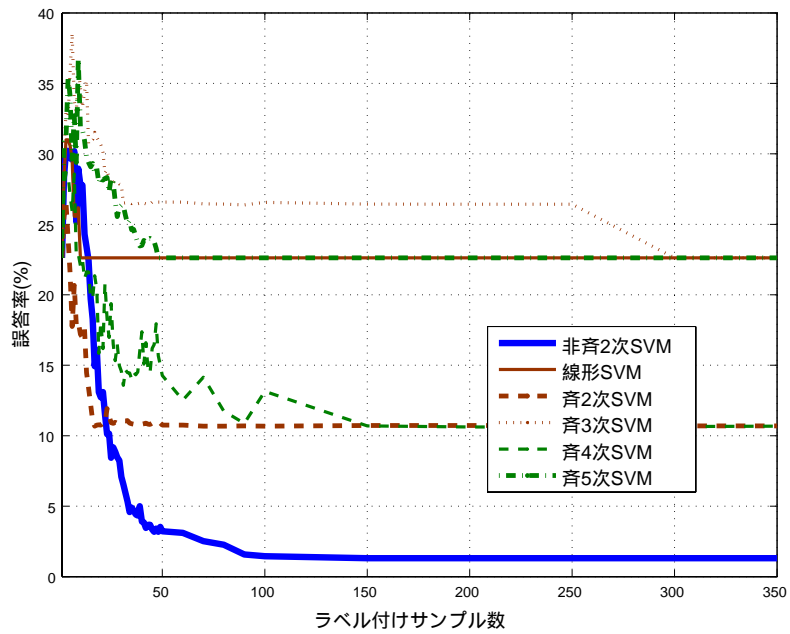


図 2.5: 楕円境界に対する斉次多項式能動 SVM の性能差

本例題の場合には, 非斉 2 次多項式カーネル $(x_i^\top x_j + 1)^2$ により, ほぼ正確に楕円を学習できる (図 2.4). この時, 能動学習の進展に伴う誤答率の低下は大きく, 効率的にラベル付けと判別関数の改良が進む (図 2.5).

しかし, 能動学習開始時点では, 入力データ分布 $\{x_i\}$ は分かっても, ラベル分布 $\{y_i\}$ は未知なので, 上記の適切なカーネルの選択は難しい.

そこで, 分析者がある斉 k 次多項式カーネル $(x^\top x)^k$ を選択したと仮定する. $k = 1, 2, 3, 4, 5$ の時, 能動サポートベクターマシンを行った結果を図 2.5 に示す.

k が偶数次の時は, 斉次多項式カーネルによる判別面は原点を中心とする凸曲線で, 楕円に近い形状であるが, 真の楕円の中心が原点でないため, 10% 以上の誤差が残り, 効率的なラベル付け, 学習が行えない.

k が奇数次 (線形, 3 次など) の時は, 斉次多項式カーネルによる判別関数は, $f(-x) - b = -(f(x) - b)$ という非対称性を持ち, 学習データとは整合しない. このため, 能動学習が進展すると途中で判別境界が存在しなくなり ($f(x) = -1, \alpha = 0$), 学習が止まる. このため, 図 2.4 には判別面を表示できない. その後のラベル付け事例の選択はランダムとなる.

このように, サポートベクターマシンは, 適切なカーネルを選択すれば, 非常に高い判別精度を実現できる学習手法であり, 能動サポートベクターマシンにおいても効率的なラベル付けが可能となる. しかし, カーネルが適切でないと, 学習とラベル付けが進まず, 能動学習の効率が大きく低下し, 得られる判別関数の判別性能も低いものとなる.

次章以降では, この能動サポートベクターマシンにおけるカーネル事前選択の困難を解決する方策について検討を行なう.

2.5 本章のまとめ

本章では, まず, 本研究の基礎となる諸技術「サポートベクターマシン」, 「サポートカーネルマシン」, 「能動学習」とそのための「サンプリング戦略」についての既存研究を紹介した. さらに, 「サポートベクターマシン」による「能動学習」を行なう場合に問題となる, 対象問題に適したカーネルを能動学習開始前には適切に選択できないという「カーネル事前選択問題」を指摘し, このカーネル事前選択がうまく行なえない場合には, 学習が途中で止まり効率的な能動学習を行なえないことを, 簡単な例により示した.

第3章 サポートカーネルマシンに基づく能動学習の提案

本章では、まず、前節で述べた「能動サポートベクターマシンにおけるカーネル事前選択」の困難を解決するため、代表的な多カーネル学習であるサポートカーネルマシンを能動化した「単純能動サポートカーネルマシン」を導入する。さらに、単純能動サポートカーネルマシンを、専門家とのインタラクティブなラベル付け作業に用いる際の問題点である、応答性の悪さを解決するため、「2段階能動サポートカーネルマシン」を提案する。

本章で重視する適用上の制約は以下の2つである。

制約 1:インタラクティブ応答性 専門家を拘束してインタラクティブにラベル付けを行うことを想定する。このため、一定時間以内に学習結果やラベル付け対象事例を選択して専門家にフィードバックできる応答性の高さが重要となる。

制約 2:大量ラベルなしデータに対する対応 ラベルなしデータは大量に存在するため、画像処理における各種画像特徴など、候補カーネルで用いるすべての特徴値を全データに対して事前に計算して、保持することは、容量制約や時間制約から、困難な場合も少なくない。このような未処理データに対するデータ特徴値の取得に要する時間も考慮して、ラベル付け対象事例の選択などを行う必要がある。

3.1 カーネル事前選択問題解決のアプローチ

能動サポートベクターマシンでは、ラベルの分布が不明であるため、能動学習を始める前には、対象データに適した適切なカーネルを決定することはできない。しかし、対象問題では色情報が重要であるとか、RBF カー

ネルが一般に有効であるといった事前知識を元に, 有用そうな複数個のカーネル候補を $K_k(\cdot, \cdot)$ を考えることは可能である.

さらに, 能動学習の途中で一部のラベルが判明すれば, それに基づいてカーネル候補群から適したカーネルを絞り込むができる.

このための手法としては, 2.4 節で述べたサポートベクターマシンでのカーネル選択手法がひとつの候補となる. しかし, これらの手法は, 最適性が保障されない, 大規模問題を解く際に計算量が多すぎる, 適用が同一特徴を用いた RBF カーネルの半径決定に限定されるなどの問題がある. 一方, 2.2 節で述べたサポートカーネルマシンは, 要素カーネルの線形和である合成カーネル集合に含まれるカーネルの中で汎化誤差指標を最小とするものを求める. 汎化誤差指標最小という意味での大域的最適性は保障されており, また, サポートベクターマシンと線形計画の繰り返しにより解が得られるため, データ数や要素カーネル数が多い場合でも解法の工夫により効率的に解を得られる可能性が高い.

そこで, 本論文では, サポートカーネルマシンを能動化することで, ラベルなしサンプルのラベル付けにより, 効率的に適したカーネルの絞り込みと判別関数の改良とを同時に行なう方策を検討する.

3.2 単純能動サポートカーネルマシンの概要

3.2.1 能動サポートカーネルマシンの基本アルゴリズム

前節での考え方に基づく能動サポートカーネルマシンの基本アルゴリズムをアルゴリズム 3.2.1 に示す. 次節で述べる 2 段階能動サポートカーネルマシンと区別するため, 本アルゴリズムを単純能動サポートカーネルマシンと呼ぶこととする.

本アルゴリズムは, 第 1 行で候補カーネル群 $\{K_k(\cdot, \cdot)\}_{k=1, \dots, M}$ を指定する点, 第 3 行でサポートカーネルマシンの最適解 (α^*, β^*) を求めて判別境界 $f(x) = 0$ を形成する点のみが, 能動サポートベクターマシンと異なる.

能動サポートカーネルマシンでは, 能動学習の進展に応じて, 可能な合成カーネルの中から現時点の学習データに合うカーネル $K(\cdot, \cdot; \beta^*)$ を見出して, 判別境界を形成する.

- 1: 候補カーネル群 $\{K_k(\cdot, \cdot)\}_{k=1, \dots, M}$ を指定.
- 2: 正例, 負例数個からなる初期既知ラベル集合 \mathcal{L} と, 初期未知ラベル集合 \mathcal{U} を設定.
- 3: **while** $|\mathcal{L}| \leq N$ **do**
- 4: \mathcal{L} に対する SKM の最適解 (α^*, β^*) を求め, 判別境界 $f(x) = 0$ を形成.
- 5: $f(x)$ に基づくサンプリング戦略 SS に従いサンプル i を選択し, ラベル y_i を取得.
- 6: $(\mathcal{L}, \mathcal{U}) \leftarrow (\mathcal{L} \cup \{i\}, \mathcal{U} - \{i\})$.
- 7: **end while**

アルゴリズム 3.2.1: 能動サポートカーネルマシン

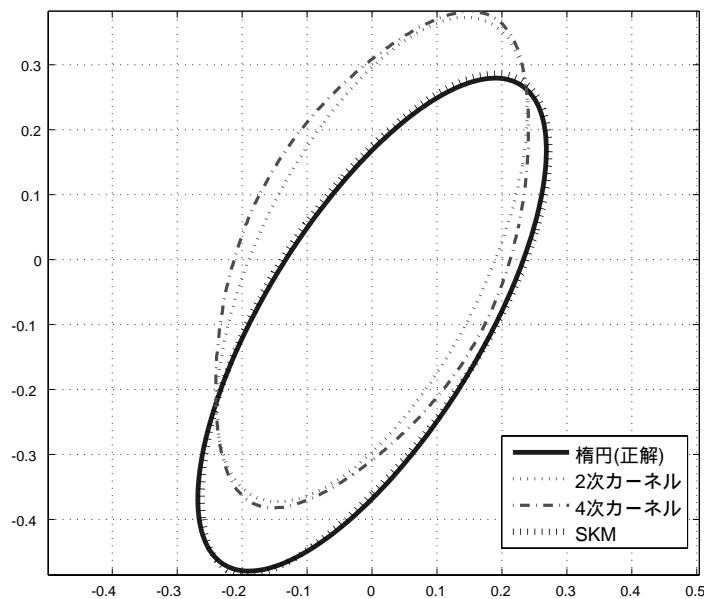


図 3.1: 楕円境界に対する能動 SVM と能動 SKM の判別面の比較

第 3 章 サポートカーネルマシンに基づく能動学習の提案

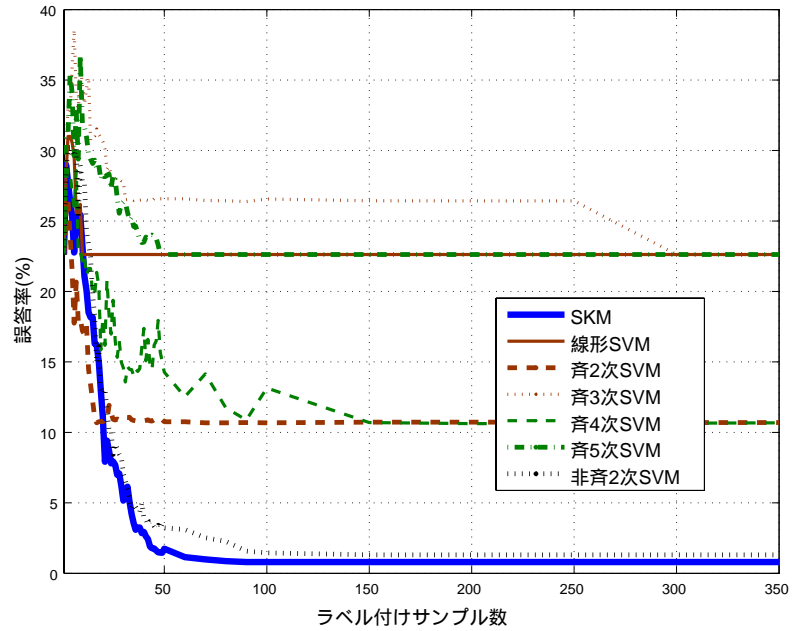


図 3.2: 楕円境界に対する能動 SVM と能動 SKM の誤答率の比較

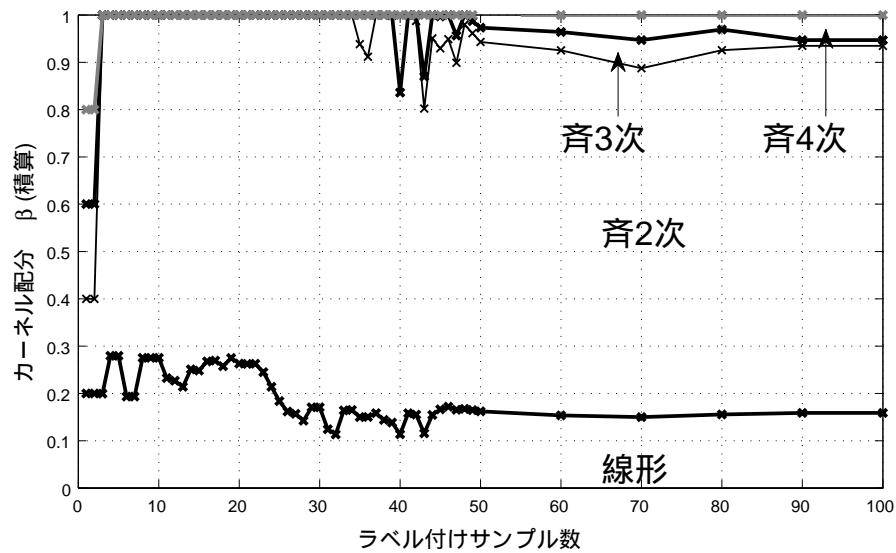


図 3.3: 楕円境界に対する能動 SKM の β の変化

3.2.2 簡単な適用例

2.4 節での例題に, $k = 1, \dots, 5$ の斉 k 次多項式カーネルを要素カーネルとする単純能動サポートカーネルマシン, 各要素カーネルによる能動サポートベクターマシン, 及び, 非斉次 2 次多項式カーネルによる能動サポートベクターマシンを適用した結果を, 図 3.1, 図 3.2, に示す.

サポートカーネルマシンが表現できる判別関数は, (非斉次)5 次多項式であるので, 要素カーネルを個別に用いた場合と異なり, 能動サポートカーネルマシンでは, ほぼ正確な判別境界 (楕円) を形成できる (図 3.1).

更に, サポートベクター数に近い約 50 個のラベル付けで誤答率がほぼ 0 となることから (図 3.2), 能動学習過程で適切なカーネル配分 β が得られ, ラベル付けと判別関数の学習が効率的に進んでいることが判る.

非斉次 2 次多項式を用いた単カーネル能動サポートベクターマシンと比較しても, 能動サポートカーネルマシンの方が早期に誤答率が減少し, かつ, 最終的誤答率がやや小さくなっている.

図 3.3 に, 能動学習の進展に伴う配分 β の変化を示す. 初期サンプルの学習後, β の更新を始めた直後から, 主要なカーネルである斉 2 次カーネルと線形カーネルに大きな配分がなされ, 斉 5 次カーネルの配分は 0 となっている. ラベル付きサンプル数が 25 個頃から線形カーネルに対する重みは, ほぼ 0.16 で安定し, 90 個程度から斉 2 次, 3 次, 4 次カーネルの配分が, 各々 0.78, 0.01, 0.05 程度で安定する.

なお, 誤差は完全に 0 とならない. 判別面近傍に訓練データが存在しない領域があるため, 真の判別面 (楕円) は学習データを正しく判別できる判別面の 1 つではあるが, 学習データに対するマージンが最大の判別面とはならないため, 学習結果は楕円とは完全には一致しない. このため, 未学習データに対する微小な誤差が残る.

また, 能動サポートカーネルマシンでも, 能動サポートベクターマシンと同様, 途中で判別境界が存在しなくなる ($S(\alpha^*; \beta^*) = \infty$) 可能性がある. ただし, この場合, 式 (2.25) から, 要素カーネルによるサポートベクターマシンはいずれも判別境界を形成できない ($\max_{\alpha} S_k(\alpha) \geq S_k(\alpha) = \infty$). このため, 能動サポートベクターマシンに比べ, 能動サポートカーネルマシンが, 途中で判別面が形成できなくなる可能性は小さい.

単純能動サポートカーネルマシンの問題点としては, 以下が挙げられる.

2.2.4 節で述べたようにサポートカーネルマシンの解法は, サポートベクターマシンと LP の繰り返しによる収束計算であるため, 最適解 (α^*, β^*) を得るまでに時間を要する. また, 合成カーネル $K(\cdot, \cdot; \beta)$ は M 個の要素

カーネルを持つため、サポートベクターマシンでのカーネル計算の M 倍の時間を要する。このため、上記基本アルゴリズムでは、ラベル付けの後次の事例選択がなされるまでの時間 (応答時間) が掛かりすぎ、対話的環境で能動学習を進めるには適さない。

3.3 2 段階能動サポートカーネルマシン

3.3.1 サポートカーネルマシン能動化のアプローチ

一般に、サポートカーネルマシンが最適解 α^* , β^* に収束するには、計算時間を要する。このため、インタラクティブな環境下で能動学習を行おうとする場合、次のラベル付けをするサンプルの選択に必要な最新の判別関数を得るまでの時間、すなわち、新たなラベル付きサンプルの追加によるサポートカーネルマシンの学習時間が掛かりすぎる。

そこで、提案アルゴリズムでは、能動学習における応答性と最終的な判別精度の両者を確保するため、以下のアプローチを用いる。

1. サポートカーネルマシンの解法は、カーネル数、データ数が多い大規模問題にも適用可能とするため、2.2.4 節の SKM-ILP に基づくアルゴリズム 2.2.1 を基本に能動学習化する。
2. 能動学習中 (能動学習段階) と能動学習終了後 (完全最適化段階) は異なる設計とする。

能動学習段階では、専門家のラベル付けに対する応答性を重視し、ラベル付けサンプルの選択を高速に行う。このため、サポートカーネルマシンの最適解ではなく、高速に得られる近似解 (α, β) による判別関数 $f(x)$ を用いて、ラベル付けをするサンプルを選択する。

能動学習が終了した段階 (完全最適化段階) では得られた判別関数の判別精度が重要となるので、サポートカーネルマシンを完全に最適化して、厳密解 (α^*, β^*) を得る。

3. 多数のラベルなしサンプルからラベル付けサンプルを選択する際に必要な $f(x)$ の評価回数を、確率的最小化の考え方にに基づき減少させる。

アプローチ 3 は、本章冒頭で述べた「制約 2:大量ラベルなしデータに対する対応」にもなっている。

3.3.2 アルゴリズムの詳細

本節では, 対話的な能動学習に適した 2 段階能動サポートカーネルマシンを提案する. アルゴリズムを図 3.3.1, 図 3.3.2, に示す.

トップレベル関数 *two-phased-SKM* では以下の指定が必要である.

- サポートカーネルマシンのパラメータ
 - 要素カーネル $K_k(\cdot, \cdot)$, 事前重み R_k , $k = 1, \dots, M$.
 - 例外 ξ に対するコスト C .
- データ
 - データベース $\mathcal{D} = \{(x_i, y_i) \mid i = 1, \dots, N_0\}$.
 - ラベル付きサンプルの初期集合 \mathcal{L}_0 .
 - ラベルなしサンプルの初期集合 \mathcal{U}_0 .
- サンプリングパラメータ
 - 最大ラベル付け数 N .
 - サンプリング戦略 SS .
 - 事前サンプリング数 n .
- 制御モード, 閾値
 - サポートベクターマシンの解 α の収束判定の閾値 τ .
 - カーネル重み β の収束判定のための閾値 ϵ_0, ϵ_1 .
 - モード α -mode, β -mode.

ラベルなしサンプル ($i \in \mathcal{U}$) は, データベース \mathcal{D} 中のサンプルデータ (x_i, y_i) の入力 x_i のみにアクセスが可能である. このため, 初期化段階で, 各カーネルに対して, $\{\phi_k(x_i) \mid i \in \mathcal{U}\}$ を囲む最小超球半径を求めることができ, R_k をこの値に設定する. サンプル選択関数 $\text{select}(\alpha, \beta, b, n, \mathcal{U}; SS)$ により選択されると, 専門家によるラベル付けが行われ, ラベル付きサンプル ($i \in \mathcal{L}$) となり, 入力 x_i , ラベル y_i の両者がアクセスが可能となる.

サンプル選択でのサンプリング戦略 SS は, 任意の戦略が可能である. ただし, 本章では, α, β, b により定まる判別関数 $f(x)$ による MARGIN 戦略

を基本としたものとする。他の戦略についての検討は 4 章で行なう。事前サンプリング数 n については、MARGIN 戦略の高速化の節で説明する。

制御モード α -mode は、 α の近似値の求め方、すなわち、サポートベクターマシンの近似解法を指定する。一方、 β -mode は、カーネル重み β の近似値の求め方を指定する。update 関数に関する説明でより詳しく述べる。また、閾値 ϵ_0, ϵ_1 については、トップレベル関数に関する説明で詳しく述べる。

確率的最小化による MARGIN 戦略の高速化

図 3.3.1:13 行目の select で MARGIN 戦略を用いて最適な事例を選択するには、全てのラベルなしサンプル $x_i \in \mathcal{U}$ に対して $f(x_i)$ を評価する必要がある。しかし、ラベルなしサンプル数 $|\mathcal{U}|$ が多いと、この評価だけで非常に時間がかかる。

応答性を確保するため、全ラベルなしサンプル中で $|f(x_i)|$ が最小のサンプルではなく、最小値に近いサンプルを選択することとする。

具体的には、 \mathcal{U} からランダムに選択した n サンプル中で $|f(x_i)|$ が最小のサンプルを選択する。すなわち、「乱数による事前選択」後に「マージン戦略」を適用する。

n 事例中の最小値が、 $\{f(x_i)\}_{i \in \mathcal{U}}$ の $100 \times p\%$ 点より小さい確率は、 $1 - (1 - p)^n$ である。例えば $n = 50$ の時、最小値が 5% 点より小さい確率は約 92% ($= 1 - 0.95^{50}$)、10% 点より小さい確率は約 99.5% ($= 1 - 0.90^{50}$) となる。 $n = 100$ の時は、5% 点より小さい確率は約 99.5%、10% 点より小さい確率はほぼ 100% となる。

このため、サンプル選択は、ラベルなし事例数 $|\mathcal{U}|$ に依らず、事前選択の個数 n により定まる一定時間で、最適性の高いサンプル選択を行える。

なお、確実に全サンプルが事前選択の対象となるように、一度、事前選択されたサンプルは、マージン戦略によりラベル付けサンプルとして選択をされなくても、全てのラベルなしサンプルが事前選択されるまで、再度、事前選択の候補とはならないようにしている。

なお、ランダムな事前選択では、基本的にデータ数の多い領域のサンプルを選択しやすい。このため、データ数の多い領域の正解率を優先的に上げる傾向を持つ。

本手法は、本章冒頭で述べた「制約 2: 大量ラベルなしデータに対する対応」となっている。

通常、全数チェックを行わずに効率的に最小の $|f(x_i)|$ を探索するには、事前に大量の入力データに対する索引を用意して、それに基づいて探索領

```

function two-phased-SKM( $\mathcal{L}, \mathcal{U}, \{(x_i, y_i)\}$ )
//初期化: $t$ , 制約集合  $CS$ , 重み  $\alpha, \beta$ 
 $t \leftarrow 0$ .  $CS \leftarrow \{\beta_k \geq 0, \sum \beta_k = 1\}$ .  $\beta^0 \leftarrow (1/M, \dots, 1/M)$ .
for all  $s \in \mathcal{L}$  do
     $\mathcal{L}0 \leftarrow \mathcal{L}0 \cup \{s\}$ .  $\alpha_s^t \leftarrow 0$ .
     $g_{[k]s}^0 \leftarrow y_s - \sum_{i \in \mathcal{L}0} \alpha_i^0 K(x_i, x_s; \beta^0)$ ,  $g_s^0 \leftarrow \sum_k \beta_k g_{[k]s}^0$ .
     $(\alpha^0, g^0, g_{[k]}^0) \leftarrow \text{update}(\alpha, g^0, g_{[k]}^0, \beta^0, \alpha\text{-mode}, \mathcal{L}0)$ .
end for
//能動学習段階
repeat
     $b^t \leftarrow \text{intercept}(g^t, \mathcal{L})$ 
     $s^t \leftarrow \text{select}(\alpha^t, \beta^t, b^t, \mathcal{U}; SS)$ . //サンプリング戦略
     $(\mathcal{L}, \mathcal{U}) \leftarrow (\mathcal{L} \cup \{s^t\}, \mathcal{U} - \{s^t\})$ ,  $\alpha_{s^t} \leftarrow 0$ .
     $g_{[k]s^t} \leftarrow y_{s^t} - \sum_{i \in \mathcal{L}} \alpha_i K(x_i, x_{s^t}; \beta^t)$ ,  $g_{s^t} \leftarrow \sum_k \beta_k g_{[k]s^t}$ .
    repeat
         $t \leftarrow t + 1$ .
        //SVM 近似解  $\alpha^t$ 
         $(\alpha^t, g^t, g_{[k]}^t) \leftarrow \text{update}(\alpha, g^{t-1}, g_{[k]}^{t-1}, \beta^{t-1}, \alpha\text{-mode}, \mathcal{L})$ .
         $S_k^t \leftarrow S_k(\alpha^t)$ .  $\theta_0^t \leftarrow \sum \beta_k^{t-1} S_k^t$ . // $\theta_0^t = S(\alpha^t; \beta^{t-1})$ 
         $CS \leftarrow CS \cup \{\sum_k \beta_k S_k^t \leq \theta\}$ .
        //LP 解  $\theta^t = S(\alpha^t; \beta^t) \leq \theta_0^t$ 
         $(\theta^t, \beta^t) \leftarrow \text{argmin}\{\theta | (\theta, \beta) \in CS\}$ ,  $g^t \leftarrow \sum_k \beta^t g_{[k]}^t$ .
    until ( $\beta\text{-mode} = \text{once}$ ) or  $(\theta_0^t > 0$  and  $|1 - \theta^t / \theta_0^t| \leq \epsilon_1)$ 
until  $|\mathcal{L}| \geq N$ 
//完全最適化段階
repeat
     $t \leftarrow t + 1$ .
    //SVM 解  $\alpha^t = \text{argmax} S(\alpha; \beta^{t-1})$ 
     $(\alpha^t, g^t, g_{[k]}^t) \leftarrow \text{update}(\alpha, g^{t-1}, g_{[k]}^{t-1}, \beta^{t-1}, \text{full}, \mathcal{L})$ .
     $S_k^t \leftarrow S_k(\alpha^t)$ .  $\theta_0^t \leftarrow \sum \beta_k^{t-1} S_k^t$ . // $\theta_0^t = S(\alpha^t; \beta^{t-1})$ 
     $CS \leftarrow CS \cup \{\sum_k \beta_k S_k^t \leq \theta\}$ .
    //LP 解  $\theta^t = S(\alpha^t; \beta^t)$ 
     $(\theta^t, \beta^t) \leftarrow \text{argmin}\{\theta | (\theta, \beta) \in CS\}$ ,  $g^t \leftarrow \sum_k \beta^t g_{[k]}^t$ .
until  $\theta_0^t > 0$  and  $|1 - \theta^t / \theta_0^t| \leq \epsilon_0$ 

```

アルゴリズム 3.3.1: 2段階能動サポートカーネルマシン (toplevel)

```

function update( $\alpha, g, g_{[k]}, \beta, \alpha\text{-mode}, \mathcal{L}$ )
if  $\alpha\text{-mode} = \text{full}$  then
    //SVM 解
    repeat
        ( $\alpha, g, g_{[k]}$ )  $\leftarrow$  update( $\alpha, g, g_{[k]}, \beta, \text{wss2}, \mathcal{L}$ ).
        ( $i, j$ )  $\leftarrow$  (  $\underset{\alpha_s > A_s, s \in \mathcal{L}}{\text{argmin}} g_s, \underset{\alpha_s < B_s, s \in \mathcal{L}}{\text{argmax}} g_s$  )
    until ( $g_j - g_i \leq \tau$ )
else
    ( $i, j$ )  $\leftarrow$  (  $\underset{\alpha_s > A_s, s \in \mathcal{L}}{\text{argmin}} g_s, \underset{\alpha_s < B_s, s \in \mathcal{L}}{\text{argmax}} g_s$  )
    if  $g_j - g_i > \tau$  then
        if  $\alpha\text{-mode} = \text{wss1}$  then
             $\lambda \leftarrow \underset{\substack{\alpha + \lambda(\mathbf{e}_j - \mathbf{e}_i) \in [A, B] \\ \lambda > 0}}{\text{argmax}} S(\alpha + \lambda(\mathbf{e}_j - \mathbf{e}_i), \beta).$ 
        else if  $\alpha\text{-mode} = \text{wss2}$  then
            ( $\lambda, j$ )  $\leftarrow \underset{\substack{\alpha + \lambda(\mathbf{e}_j - \mathbf{e}_i) \in [A, B] \\ \lambda > 0, i \in \mathcal{L}}}{\text{argmax}} S(\alpha + \lambda(\mathbf{e}_j - \mathbf{e}_i), \beta).$ 
        end if
         $\alpha \leftarrow \alpha + \lambda(\mathbf{e}_j - \mathbf{e}_i).$ 
         $g_{[k]s} \leftarrow g_{[k]s} - \lambda(K_k(x_j, x_s) - K_k(x_i, x_s))/R_k^2, k = 1, \dots, M, \forall s \in \mathcal{L}.$ 
         $g_s \leftarrow \sum_k \beta_k g_{[k]s}, \forall s \in \mathcal{L}.$ 
        ( $i, j$ )  $\leftarrow$  (  $\underset{\alpha_s > A_s, s \in \mathcal{L}}{\text{argmin}} g_s, \underset{\alpha_s < B_s, s \in \mathcal{L}}{\text{argmax}} g_s$  )
    end if
end if
return ( $\alpha, g, g_{[k]}$ ).

```

アルゴリズム 3.3.2: 2 段階能動サポートカーネルマシン (update)

```

function intercept( $g, \mathcal{L}$ )
    ( $i, j$ )  $\leftarrow$  (  $\underset{\alpha_s > A_s, s \in \mathcal{L}}{\text{argmin}} g_s, \underset{\alpha_s < B_s, s \in \mathcal{L}}{\text{argmax}} g_s$  )
     $b \leftarrow (g_i + g_j)/2.$ 
return  $b.$ 

```

アルゴリズム 3.3.3: 2 段階能動サポートカーネルマシン (intercept)

域を絞り込む必要がある。特に、能動サポートカーネルマシンでは、カーネル重み β が変化するため、各要素カーネルに基づく複数の索引を用いるなど、索引の構造も複雑になりやすい。

これに対して、本サンプリング手法では、得られるサンプルはあくまで準最小値を与えるものであるが、限られた数 (n 個) のサンプルをランダムに読み込み、その場でサポートベクター x_{sv} とのカーネル値 $K_k(x_{sv}, x_i)$ の計算を行い、判別関数値 $f(x_i)$ を求めれば良い。事前の索引作成などのデータ処理等が不要で、極めて軽いサンプリング手法となっている。

トップレベル関数

2 段階能動サポートカーネルマシン (図 3.3.1) は、前述のように基本的に SKM-ILP の解法に従うが、初期化の後、能動学習段階と完全最適化段階の 2 段階で能動学習を行なう。

能動学習段階では制御変数 (α -mode, β -mode, n) に従って、 $S(\alpha^t; \beta^{t-1}) > S(\alpha^{t-1}; \beta^{t-1})$ を満たすサポートベクターマシン近似解 α^t を得る。

n はサンプル選択での事前選択の個数、 α -mode は α に対する近似法を指定する。値としては、近似を行わない full、と近似法 wss1, wss2 とを取りえる。詳細は次の update 関数の説明で述べる。

β -mode は、サポートカーネルマシンとしての収束条件を満たすまで α , β の更新を繰り返すか (full)、各々 1 回の更新とするか (once) を制御する。

完全最適化段階では、サポートカーネルマシンを完全に最適化する (α -mode=full, β -mode=full に相当)。

モード (α -mode, β -mode, n) と既存解法の間係を表 3.1 に示す。

モード (α -mode, β -mode, n) =(full, full, ∞) の場合が 3.2 節の単純能動サポートカーネルマシンで、能動学習段階でも毎回サポートカーネルマシン最適化を行う。

Sonnenburg らは、SKM-ILP に基づくサポートカーネルマシンのバッチ解法として、部分的データに対するサポートベクターマシンの最適解を α^t とする chunked MKL を提案している [22, 23]。これは、全ラベルが既知の場合の解法のため、サンプリング戦略 SS が異なるが、モード (full, once, ∞) に近い。

これに対して、提案する 2 段階能動サポートカーネルマシンでは、能動学習での応答性に重点を置き、サポートベクターマシン最適解のより粗い近似を与える (wss1, once, n)、または、(wss2, once, n) を用いる。すなわち、能動学習段階では、 α^{t-1} の wss1 や wss2 による更新、線形計画 (LP) によ

表 3.1: 2 段階 SKM のモードと既存解法の関係

α -mode	β -mode	n	解法
full	full	∞	単純能動 SKM
full	once	∞	\approx chunked MKL
wss1/wss2	once	$n < \infty$	提案手法
full	none	∞	能動 SVM

る β^{t-1} の更新を各々 1 回行い, 判別境界を更新し, ランダムに選択した n 個のラベルなし事例中で, この判別境界に最も近い事例のラベルを得ることを繰り返す.

なお, β -mode=full の時の能動学習時のサポートカーネルマシン収束条件 ϵ_1 を完全最適化段階の収束条件 ϵ_0 と同じにすると, 能動学習時の応答時間が掛かり過ぎるため, 異なる設定が可能としている.

サポートベクターマシン近似解—update 関数

update 関数 (図 3.3.2) は, 近似モード α -mode に応じて, $S(\alpha^{new}; \beta) > S(\alpha; \beta)$ なるサポートベクターマシン問題 $\max_{\alpha} S(\alpha; \beta)$ の近似解 α^{new} を求める. 近似モードには, full (完全最適化), wss1, wss2 の 3 種類を設けた. 能動学習段階では wss1 や wss2 を, 完全最適化段階では full を用いる. また, full の解の準最適性を τ により制御する.

$S(\alpha; \beta), S_k(\alpha)$ の α_i に関する勾配を $g_i, g_{[k]i}$ と記す.

$$g_i = \frac{\partial}{\partial \alpha_i} S(\alpha; \beta) = y_i - \sum_{j \in \mathcal{L}} \alpha_j K(x_j, x_i; \beta)$$

$$g_{[k]i} = \frac{\partial}{\partial \alpha_i} S_k(\alpha) = y_i - \sum_{j \in \mathcal{L}} \alpha_j \frac{K_k(x_j, x_i)}{R^2}$$

また, 以下の (i, j) を最大違反ペアと呼ぶ.

$$(i, j) = \underset{\substack{\alpha_j < B_j \\ \alpha_i > A_i}}{\operatorname{argmax}} (g_j - g_i) = (\underset{\alpha_i > A_i}{\operatorname{argmin}} g_i, \underset{\alpha_j < B_j}{\operatorname{argmax}} g_j)$$

この時, 判別関数の切片 b の存在範囲は, $g_j \leq b \leq g_i$ となる.

最適解 (厳密解) では, 最大違反ペアの勾配差 $\delta = g_j - g_i \leq 0$ となり, $g_j \leq b \leq g_i$ となる. 通常は, 自由サポートベクターが存在し ($g_j - g_i = 0$), 切片 $b = g_j = g_i$ に一意に確定する.

wss1[20] では, 最大違反ペア (i, j) に対して簡単に解析解が得られる部分問題 (2.11) を解き α を改善する

wss2 では, λ のみではなく, wss1 で固定していた最大違反ペア (i, j) の一方を可変とした上で, (2.11) を最大とする λ を求め, α を更新する.

full では, SMO 型アルゴリズム [20] と同様, 最大違反ペアの勾配差 δ が微小な正数 τ 以下となるまで wss1 または wss2 による α の更新を繰り返し, (準) 最適解 α^* を求める.

3.3.3 2 段階サポートカーネルマシンの収束性

$$\theta^t = \min_{\substack{\beta_k \leq 0 \\ \sum \beta_k = 1}} \max_{s \leq t} S(\alpha^s; \beta), \theta^{t-1} \leq S(\alpha^t; \beta^{t-1}) \text{ とする.}$$

補題 3.3.1 (2 段階 SKM の収束性 (1))

$$\theta^{t-1} = S(\alpha^{t-1}; \beta^{t-1}) \leq \theta_0^{t-1} = \theta^t \leq S(\alpha^t; \beta^{t-1})$$

証明 θ^t, θ^{t-1} を与える β を β^t, β^{t-1} とする.

この時任意の β ($\beta_k \leq 0, \sum \beta_k = 1$) について,

$$\begin{aligned} \theta^t &= \max_{s \leq t} S(\alpha^s; \beta^t) \leq \max_{s \leq t} S(\alpha^s; \beta) \\ \theta^{t-1} &= \max_{s < t} S(\alpha^s; \beta^{t-1}) \leq \max_{s < t} S(\alpha^s; \beta) \end{aligned}$$

であるから,

$$\begin{aligned} \theta^t &= \max_{s \leq t} S(\alpha^s; \beta^t) \geq \max_{s < t} S(\alpha^s; \beta^t) \geq \theta^{t-1} \\ \theta^t &\leq \max_{s \leq t} S(\alpha^s; \beta^{t-1}) = \max(\theta^{t-1}, S(\alpha^t; \beta^{t-1})) = S(\alpha^t; \beta^{t-1}). \end{aligned}$$

従って,

$$\theta^{t-1} = S(\alpha^{t-1}; \beta^{t-1}) \leq \theta^t \leq S(\alpha^t; \beta^{t-1}) \quad \square$$

補題 3.3.2 (2 段階 SKM の収束性 (2)) $\theta^t = S(\alpha^t; \beta^t)$

証明

$$S(\alpha^t; \beta^{t-1}) \geq \theta^t = \max(\max_{s < t} S(\alpha^s; \beta^t), S(\alpha^t; \beta^t)) \geq S(\alpha^t; \beta^t).$$

従って,

$$\gamma S(\alpha^t; \beta^t) + (1 - \gamma)S(\alpha^t; \beta^{t-1}) = \theta^t$$

なる γ ($0 \leq \gamma \leq 1$) が存在する. この時 $\beta^+ = \gamma \beta^t + (1 - \gamma) \beta^{t-1}$ と定義すると,

$$\begin{aligned} S(\alpha^t; \beta^+) &= \gamma S(\alpha^t; \beta^t) + (1 - \gamma)S(\alpha^t; \beta^{t-1}) = \theta^t \\ \max_{s < t} S(\alpha^s; \beta^+) &\leq \gamma \max_{s < t} S(\alpha^s; \beta^t) + (1 - \gamma) \max_{s < t} S(\alpha^s; \beta^{t-1}) \\ &= \gamma \max_{s < t} S(\alpha^s; \beta^t) + (1 - \gamma)\theta^{t-1} \\ &\leq \max_{s < t} S(\alpha^s; \beta^t) \quad (\because \theta^{t-1} \leq \max_{s < t} S(\alpha^s; \beta^t)) \\ &\leq \theta^t \\ \max_{s \leq t} S(\alpha^s; \beta^+) &= \max(\max_{s < t} S(\alpha^s; \beta^+), S(\alpha^t; \beta^+)) \leq \theta^t \end{aligned}$$

θ^t の β に関する最小性より, $\max_{s \leq t} S(\alpha^s; \beta^+) = \theta^t = S(\alpha^t; \beta^+)$.

従って β^+ を β^t と定義し直すと $\theta^t = S(\alpha^t; \beta^t)$. \square

最適解は,

$$(\alpha^*, \beta^*) = \operatorname{argmin}_{\substack{\sum \beta_k = 1 \\ \beta_k \geq 0}} \max_{\substack{\alpha_i \in [A_i, B_i] \\ \sum \alpha_i = 0}} S(\alpha; \beta), \quad \theta^* = S(\alpha^*; \beta^*)$$

t 回目の線形計画の解は,

$$(\alpha^t, \beta^t) = \operatorname{argmin}_{\substack{\sum \beta_k = 1 \\ \beta_k \geq 0}} \max_{s \in \{1, \dots, t\}} S(\alpha^s; \beta), \quad \theta^t = S(\alpha^t; \beta^t) \leq \theta^*$$

従って, $\theta^t = S(\alpha^t; \beta^t)$ は単調増加であり, ラベル付きサンプルの追加 (t の増加) に伴い, 解 α^t, β^t は最適解に近づく.

最終的な解の最適性は, 能動学習終了後, サポートカーネルマシンとして厳密解を求めることにより保証される.

3.3.4 実装

MATLAB 言語により記述した版と, C++ による実装の 2 種類を開発した. MATLAB では, 最適化ツールボックスの `linprog` を線形計画に用いた. また, C++ 版では CLP (COIN-OR Linear Program Solver)[12] を線

形計画ソルバーとして用い、前回の線形計画での有効制約を、初期の有効制約とすることで、線形計画の高速化を図っている。

さらに、制約 c_t を制約集合 CS に追加する際には、 $S_k(\alpha^s) \leq S_k(\alpha^t) (\forall k)$ なる s に対する制約 c_s は冗長となるため、 CS から削除し、制約群 CS の冗長性を少なくしている。

また、サポートベクターマシンの収束速度を上げるため、update 関数で α -mode が full の場合には、SMO 解法で一般的な shrink, unshrink 操作を行なっている。

サポートベクターマシン、サポートカーネルマシンにおいては、カーネル $K(\cdot, \cdot)$ へのアクセスが頻繁に起こるため、カーネルの効率的なキャッシュ操作が重要である。

MATLAB 版では、その時点の各サポートベクターの集合 $\{i | \alpha_i \neq 0\}$ を SV 、ラベル付きサンプル集合を \mathcal{L} とする時、 \mathcal{L} のサイズが一定数に達するまでは、各サンプル $j \in \mathcal{L}$ に対して、各サポートベクター $i \in SV$ に対する合成カーネル $K(x_i, x_j; \beta)$ 、要素カーネル $K_k(x_i, x_j)$ を全てキャッシュする。一定数を越えた場合は、LRU (Least Recently Used) 方式でキャッシュからはずすデータ $j \in \mathcal{L}$ を決め、キャッシュされるカーネルの量を一定に保つ。

C++ 版では、ラベル付きサンプル数ではなく、メモリ量を指標としているが、同様のキャッシュ管理を行なっている。

β を更新した場合は、キャッシュされた $K_k(x_i, x_j)$ から $K(x_i, x_j) = \sum_k \beta_k K(x_i, x_j)$ により再計算する。

大規模データベースへの適用時には、ラベル付きサンプル集合がメモリに保持しきれない場合が生じる。このため、サポートベクターとなる可能性が低い以下のサンプル集合 $\bar{\mathcal{S}}$ を除いたサポートベクター候補集合 $\mathcal{S} = \mathcal{L} - \bar{\mathcal{S}}$ を \mathcal{L} の代わりにワーキングメモリに保持するオプションも用意した。

$$\begin{aligned} \bar{\mathcal{S}} = \{l \in \mathcal{L} | & ((\alpha_l = 0) \text{ and } (g_l \geq g_j) \text{ and } (\alpha_l = B_l)) \\ & \text{ or } ((\alpha_l = 0) \text{ and } (g_l \leq g_i) \text{ and } (\alpha_l = A_l)) \\ & \text{ただし, } (i, j) \text{ は最大違反ペア} \} \end{aligned}$$

3.3.5 計算量

要素カーネル数 M 、ラベルなしサンプル数 N_U 、ラベル付きサンプル数 N_L 、事前選択サンプル数 n サポートベクター数 N_s 、線形計画の制約集合

第3章 サポートカーネルマシンに基づく能動学習の提案

の個数 $N_C = \#|CS|$ とする. この時, サンプル選択, および, wss1, wss2 による α の更新, 線形計画による β の更新, それぞれの計算量は以下の通りである.

「ランダム事前選択 + マージン戦略」でのカーネル計算回数

ランダム事前選択を行わない場合には, 選択に必要な要素カーネルの計算回数は $N_U \times N_s \times M$ である. これに対して, 提案手法では $n \times N_s \times M$ に低減する.

能動学習段階での計算量

ラベル付きサンプル数 N_L の時に, 1 サンプルを追加した時の計算量は下記となる.

- カーネル計算の回数:

ラベル付けサンプル間のカーネル値はキャッシュすると仮定する. この時, 1 回のサンプル追加による要素カーネルの計算回数は $(N_L + 1) \times M$ である (逐次学習型のサポートベクターマシン共通).

- α 更新の計算量:

wss1 を用いる場合は,

$$\begin{aligned} & \text{勾配 } g_{[k]i}, g_i \text{ の計算} \times (N_L + 1) \times (M + 1) \\ & + \text{部分問題の解計算} \times 1 \\ & + \text{勾配更新} \times (N_L + 1) \times (M + 1) \end{aligned}$$

wss2 を用いる場合は,

$$\begin{aligned} & \text{勾配 } g_{[k]i}, g_i \text{ の計算} \times (N_L + 1) \times (M + 1) \\ & + \text{部分問題の解計算} \times N_L \\ & + \text{勾配更新} \times (N_L + 1) \times (M + 1) \end{aligned}$$

β 更新の計算量

SMO により得られた制約 $\sum_k S_k(\alpha^t) \beta_k \leq \theta$ を制約集合 CS に追加して, 線形計画を 1 回解く.

制約の本数が N_C 個の時に制約が 1 個追加される時,

-
- 制約集合の冗長性 ($S_k(\alpha^s) < S_k(\alpha^t) (\forall k)$) のため, 制約の係数の比較 $2 \times N_C \times M$ 回
 - 制約集合が更新された時, 制約がなす多面体の頂点数は, 最大で2項係数 $\binom{N_C}{M}$ 個増加する. 従って, 線形計画内部 (SIMPLEX法) では, 最悪 $O(N_C^M)$ のオーダーの β の更新が必要となる. ただし, 通常は数回の更新で線形計画解が得られる.

3.4 実験

本節では, 能動サポートベクターマシンと能動サポートカーネルマシンの性能差, 能動サポートカーネルマシンの各種モードの誤答率, 応答時間に与える影響を調べ, 提案アルゴリズムの有用性を検証する.

3.4.1 実験システム

実験では, MATLAB で記述したコードを用いた. LP 解は MATLAB 組み込みの `linprog` により求めた. ラベル付けされた事例に対する各サポートベクターとのカーネル $[K(x_i, x_j)]_{i \in SV}$, $[K_k(x_i, x_j)]_{i \in SV}$ は最大 1000 事例分をキャッシュする. 1000 事例を超えた場合は, LRU 方式で管理する. β を更新した場合は, キャッシュされた $[K_k(x_i, x_j)]$ から $[K(x_i, x_j)]$ を再計算する. ラベル付きサンプル集合 \mathcal{L} をサポートベクター集合 S で代替するオプションは用いない.

全学習データ ($\mathcal{L} \cup \mathcal{U}$) に対する各特徴ベクトル $\phi_k(x_i)$ を囲む最小超球の半径 R_k は, 2次計画問題を解き, 能動学習の前に求める (2.1.5 節 参照).

3.4.2 実験データと評価項目

実験には, 表 3.2 に示す Raetsch らによる 13 個のベンチマーク用データベース¹ を用いた. 各データベースには, 学習データとテストデータのペアが 100 組あるが, 本実験では, 各データベースの第 1 組から第 20 組を用いた. 各組の学習データとテストデータを併せると同一のデータ集合となる.

¹<http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm>. 表 3.2 の tuned サポートベクターマシンの誤答率は, 同サイトの `errors` ファイルから計算.

表 3.2: 実験データベースと能動 SKM の最終誤答率

DB 名	データ数		誤答率 (%)		p 値
	学習	テスト	tuned SVM	能動 SKM	
banana	400	4900	11.5 ± 0.7	11.1 ± 0.6*	(0.97)
breast-cancer	200	77	26.9 ± 4.8	28.2 ± 5.5	(0.21)
diabetis	468	300	23.8 ± 1.8	23.9 ± 2.0	(0.41)
flare-solar	666	400	32.2 ± 1.9*	35.8 ± 1.8	(0.00)
german	700	300	23.0 ± 1.8*	24.4 ± 2.7	(0.03)
heart	170	100	16.2 ± 3.1	16.9 ± 2.6	(0.22)
image	1300	1010	3.0 ± 0.6*	3.4 ± 0.6	(0.02)
ringnorm	400	7000	1.7 ± 0.1	1.7 ± 0.1	(0.49)
splice	1000	2175	10.9 ± 0.7	11.1 ± 0.8	(0.22)
thyroid	140	75	4.8 ± 2.4	3.5 ± 2.6*	(0.95)
titanic	150	2051	22.2 ± 0.6	22.2 ± 0.6	(0.38)
twonorm	400	7000	3.0 ± 0.2	2.7 ± 0.2*	(1.00)
waveform	400	4600	9.8 ± 0.3	9.6 ± 0.3*	(0.98)

13 個のベンチマークに対して使用したカーネルは共通で、半径の異なる 15 個の RBF カーネル $\exp(-\gamma \|x_i - x_j\|^2)$, $\gamma = (2, 5, 10) \times (1e-2, 1e-1, 1, 10, 100)$ を用いた。

実験では、各組の学習データに対して能動サポートベクターマシン、能動サポートカーネルマシンを行い、ラベル付データ数が 1, 2, ..., 49, 50 (以上, 1 刻み), 60, ..., 90, 100 (以上, 10 刻み), 150, ..., 450, 500 (以上, 50 刻み), 600, ..., 900, 1000 (以上, 100 刻み) の各時点での、テストデータに対する誤答率、及び、計算時間を調べた。

なお、能動サポートカーネルマシンのラベル付けデータ数 k 個に対する誤答率は、その時点でサポートカーネルマシンの完全最適化をした時の誤答率である。具体的には、各データ数に達した時点で、能動サポートベクターマシンの学習状態を保存し、完全に最適化を行って誤答率を求めた後、保存された学習状態を回復して、能動学習を続けることを繰り返した。この際、計算時間を正しく求めるため経過時間の補正も行う。

以下の評価では、特に断らない限り、第 1 から第 20 組の誤答率及び計算

時間から求めた平均応答時間の平均値を用いる.

なお, 使用マシンは, Celron M (1.40GHz), 1.5GB である. コスト (式 (2.6), (2.24)) を $C=1000$, サポートベクターマシンの収束条件を $\tau=1e-3$, サポートカーネルマシンの収束条件を $\epsilon_1=1e-2$ (能動学習段階), $\epsilon_0=1e-6$ (完全学習段階) とした.

3.4.3 能動サポートベクターマシン v.s. 能動サポートカーネルマシン

実験では, まず, 以下の検討を行った.

- カーネルによる能動サポートベクターマシンの誤答率の違い
- 能動サポートベクターマシンと単純能動サポートカーネルマシンの誤答率

ただし, RBF カーネルは 15 種類あるため, 各カーネルを用いたサポートベクターマシンは 15 個あり, これを 13 個のベンチマークの 20 組のデータ全てに対して各サンプル数で完全最適化を行い誤答率を求めると, 計算時間がかかる.

このため, 第 1 段階では, 各データベースの第 1 組から第 5 組を用いて, 各カーネルを用いた 15 個の能動サポートベクターマシンと単純能動サポートカーネルマシンの合計 16 個の平均誤答率のラベル付けの進行に伴う変化を調べた.

そして, 第 1 組から第 5 組での最終誤答率の平均によって, 各ベンチマークに対してサポートベクターマシンとして最良のカーネルを特定した.

第 2 段階では, 各データベースの第 1 組から第 20 組に対して, 特定された最良のカーネルを用いた能動サポートベクターマシンと, 単純能動サポートカーネルマシンを適用し, ラベル付けの進行に伴う平均誤答率の変化を調べた.

最良 vs. 最悪能動サポートベクターマシン

第 1 組から第 5 組を用いた第 1 段階の実験結果を図 3.4, 図 3.5, 図 3.6 に示す. 鎖線は最終平均誤答率が最大の最悪能動サポートベクターマシン, 点線は最終平均誤答率が最小の最良能動サポートベクターマシン, 実線は能動サポートカーネルマシンの結果である.

第 3 章 サポートカーネルマシンに基づく能動学習の提案

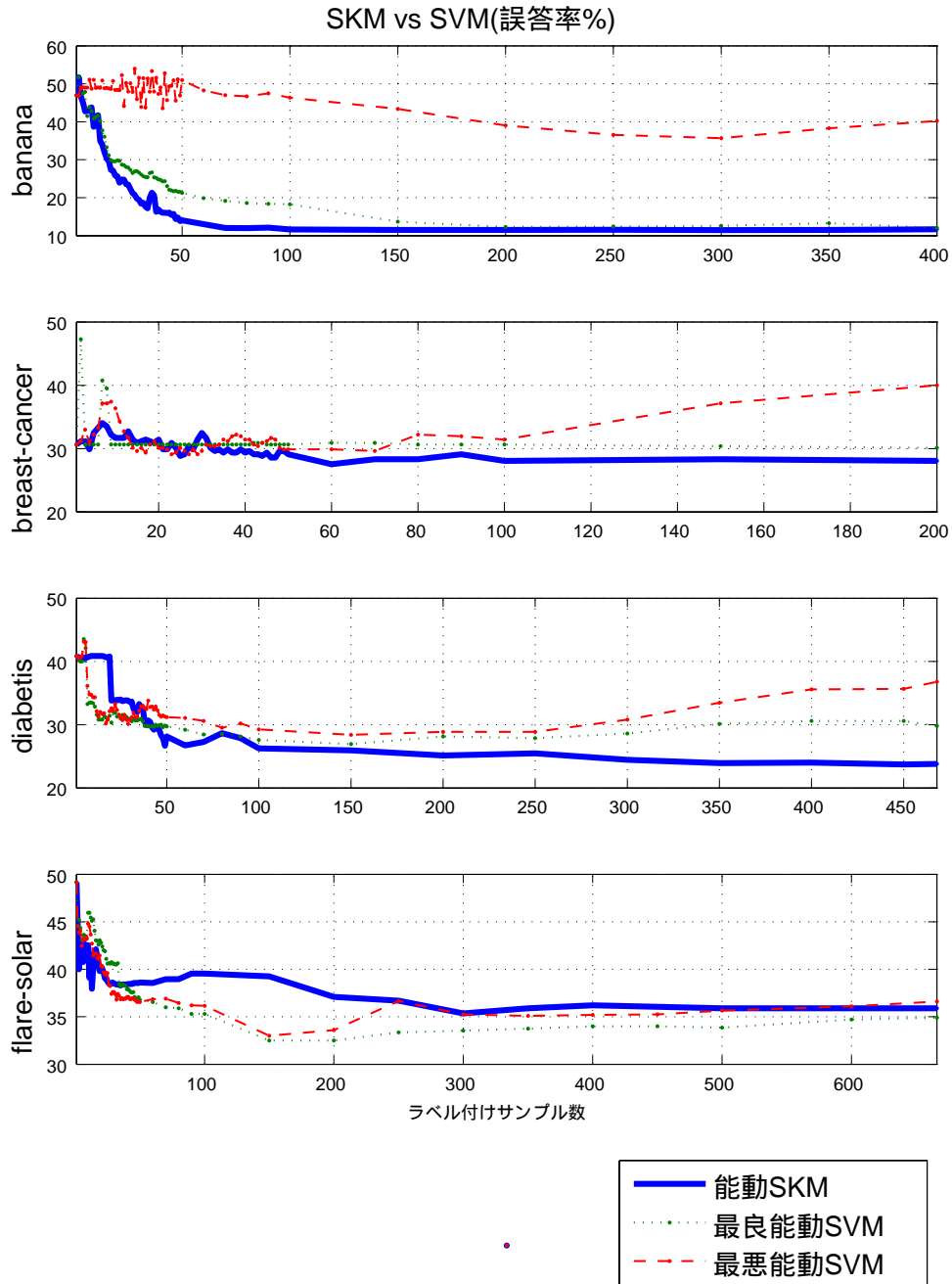


図 3.4: ベンチマーク結果 [5 データベース] (1)

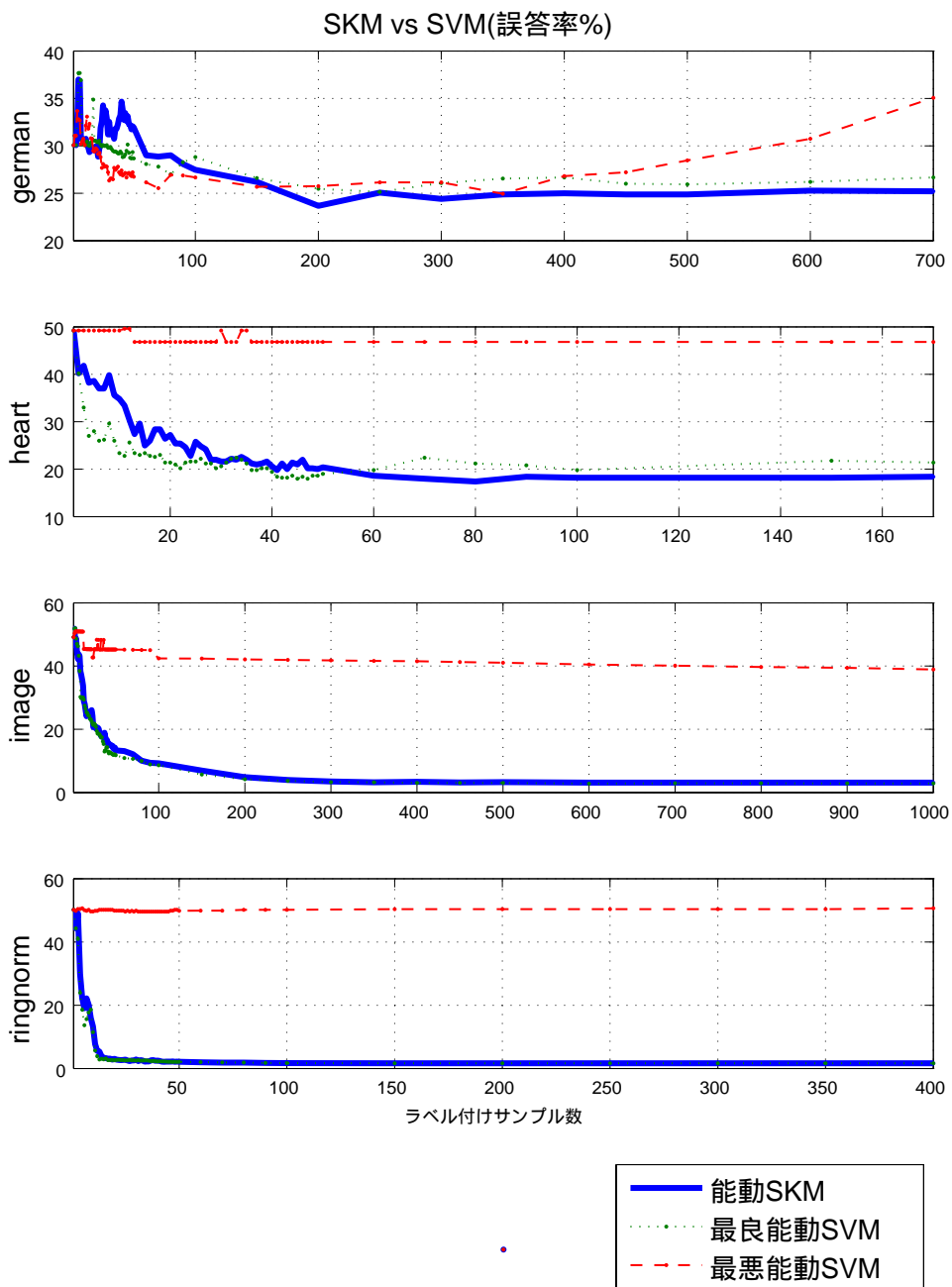


図 3.5: ベンチマーク結果 [5 データベース] (2)

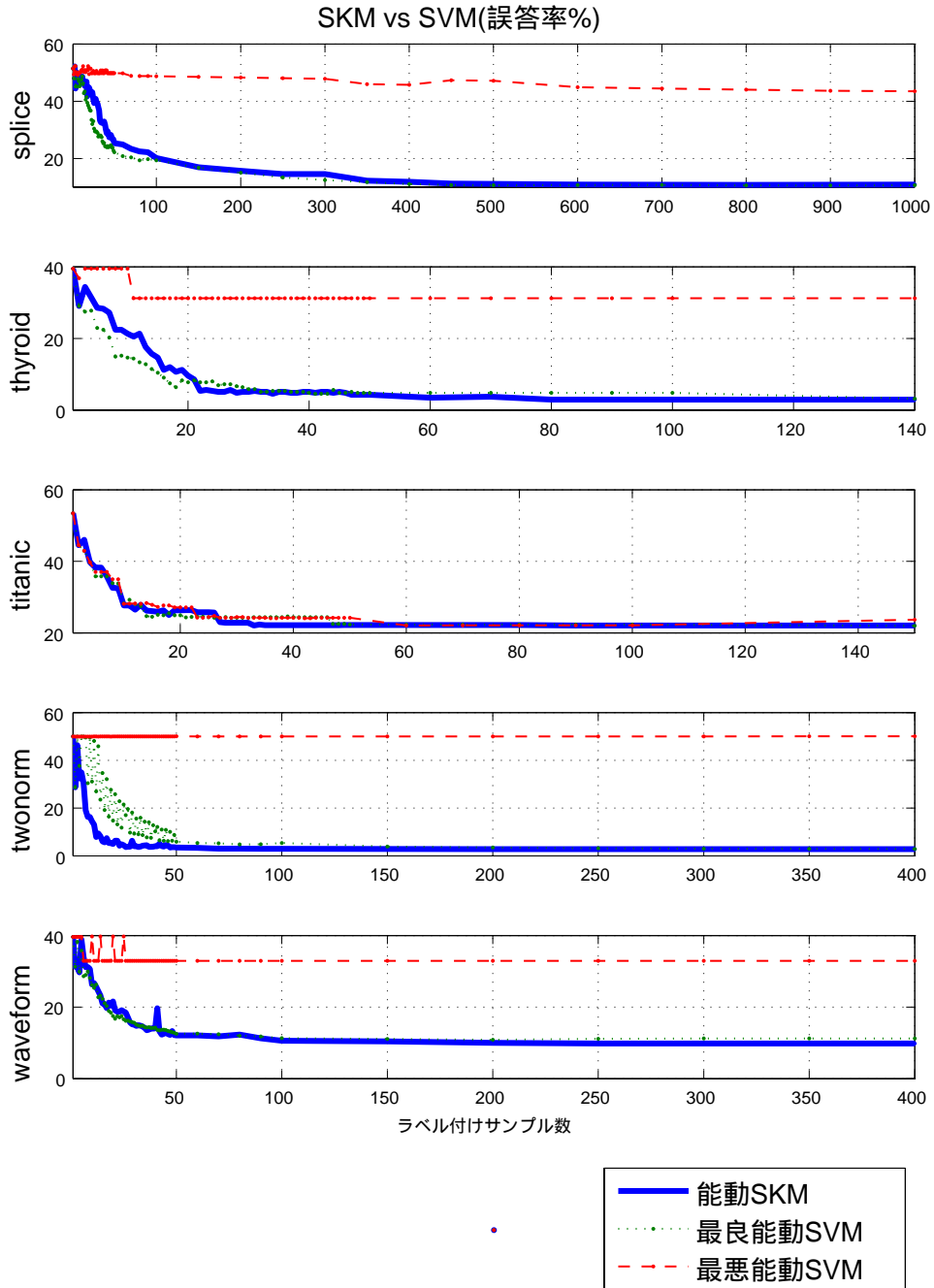


図 3.6: ベンチマーク結果 [5 データベース] (3)

banana, heart, image, ringnorm, splice, thyroid, twonorm, waveform の 8 個のデータベースでは, 最良能動サポートベクターマシンと最悪能動サポートベクターマシンで誤答率が 20% 以上異なり, カーネル選択が能動サポートベクターマシンの性能に大きな影響を与えていることがわかる.

特に, 最悪能動サポートベクターマシンでは, 誤答率が高いベンチマークデータ (banana, breast-cancer,diabetis, german) で誤答率が, ラベル付けの進行につれて途中から増加している. これは, 能動学習の前半では正例と負例の境界となっている真の判別面近傍のサンプルが選択されるため, 真の判別面に近い判別面が得られているが, 後半では, 判別面から離れた例外データが学習対象となるため, 例外コストを減少させるように, 判別面が真の判別面からずれていく結果である.

一方, 最良能動サポートベクターマシンでは,diabetis を除き, このような傾向は生じていない. 従って, 適切に判別面が形成されていることがわかる.

なお, 能動サポートカーネルマシンは, flare-solar を除く, 12 個のベンチマークデータで, 最良能動サポートベクターマシンとほぼ同等の性能となっている. (誤答率の差は 2 % 以内).

以下では, 能動サポートカーネルマシンと最良の能動サポートベクターマシンとのより詳細な分析を行う.

能動サポートカーネルマシン vs. 最良サポートベクターマシン

第 1 から第 20 組を用いた第 2 段階の実験結果を図 3.7, 図 3.8, 図 3.9, 表 3.3 に示す. グラフの実線は単純能動サポートカーネルマシンを, 点線は第 1 段階で特定した最良カーネルを用いた能動サポートベクターマシンの結果を示す. なお, 太い線は誤答率の “平均” を, 細い線は “平均 ± 標準偏差” である.

表 3.3 は, ラベル付けサンプル数が, 50 点, 100 点の時点での平均誤答率と標準偏差を示す. 表 3.4 は, 最終時点 (全学習データ学習時) の平均誤答率と標準偏差である. また, * は, 能動サポートカーネルマシンと最良能動サポートカーネルマシンを比べた時, 平均誤答率が有意に低いことを示す (有意水準 5%). 有意水準は, Welch 検定を用いて求めた.

ラベル付けサンプル数が, 50 点, 100 点の時点での, 平均誤答率の差を比べると, splice で有意に最良サポートベクターマシンの方が能動サポートカーネルマシンよりも誤答が少ない以外は, 両者の誤答率はほぼ同等か, 能動サポートカーネルマシンの方が誤答が少ない.

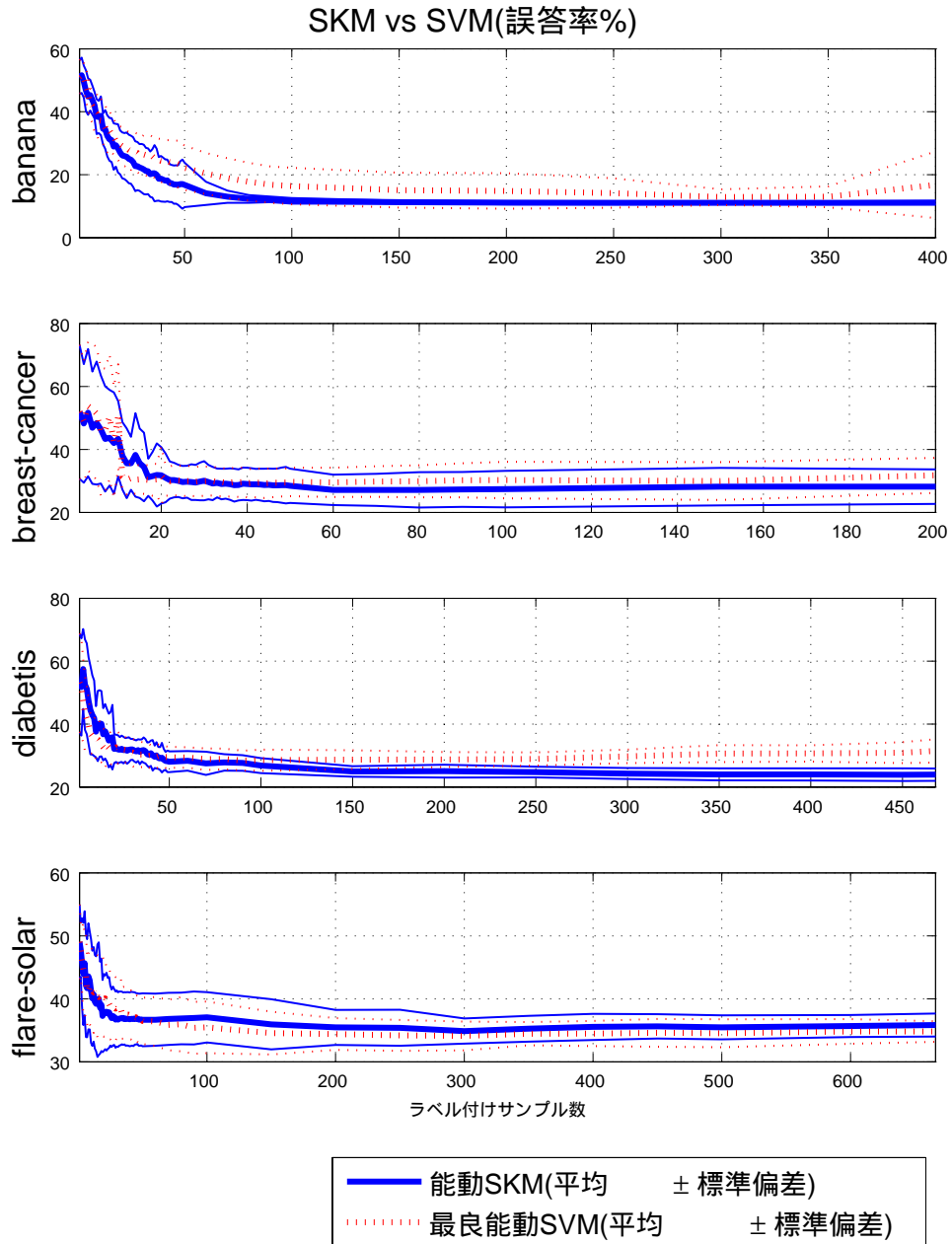


図 3.7: ベンチマーク結果 [20 データベース] (1-1)
(能動 SKM vs 能動 SVM)

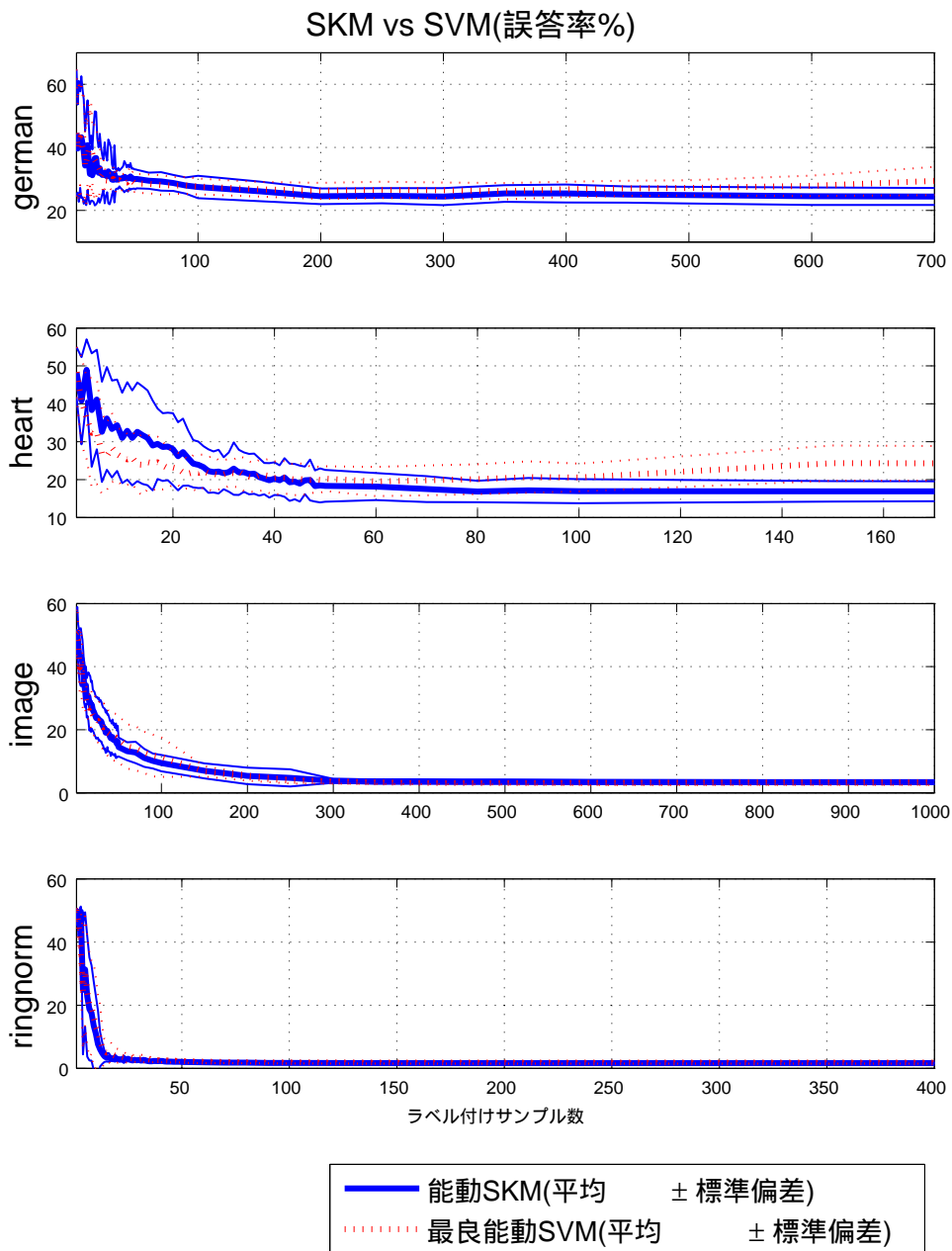


図 3.8: ベンチマーク結果 [20 データベース] (1-2)
(能動 SKM vs 能動 SVM)

第 3 章 サポートカーネルマシンに基づく能動学習の提案

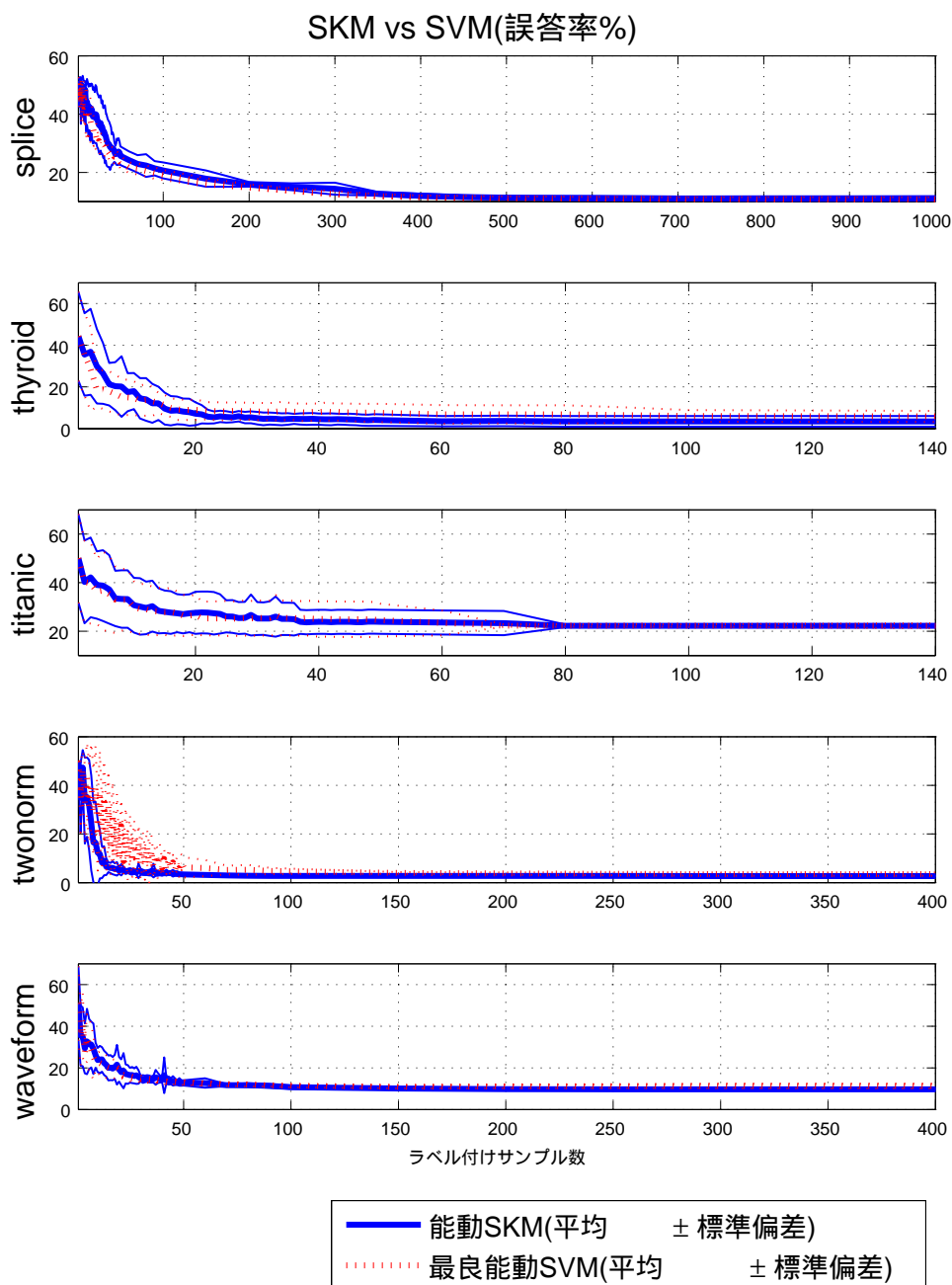


図 3.9: ベンチマーク結果 [20 データベース] (1-3)
(能動 SKM vs 能動 SVM)

表 3.3: 能動 SKM と最良能動 SVM の誤答率の比較 (中間)

ベンチマーク	SKM(50)	SVM(50)	p 値
banana	16.8 ± 7.1*	22.2 ± 7.1	(0.01)
breast-cancer	28.4 ± 5.4	29.7 ± 4.5	(0.22)
diabetis	28.6 ± 3.2	29.4 ± 3.3	(0.22)
flare-solar	36.7 ± 4.2	36.5 ± 3.7	(0.56)
german	30.0 ± 3.0	28.6 ± 3.0	(0.92)
heart	18.4 ± 4.2	20.3 ± 3.3	(0.06)
image	14.5 ± 3.0	17.0 ± 7.4	(0.09)
ringnorm	2.0 ± 0.2*	2.3 ± 0.5	(0.02)
splice	25.7 ± 3.2	23.5 ± 2.2*	(0.99)
thyroid	4.1 ± 2.8*	7.5 ± 4.4	(0.00)
titanic	23.9 ± 4.9	24.6 ± 6.8	(0.36)
twonorm	3.3 ± 0.4*	6.3 ± 4.2	(0.00)
waveform	12.8 ± 1.2	13.0 ± 1.1	(0.34)
ベンチマーク	SKM(100)	SVM(100)	p 値
banana	11.7 ± 0.9*	16.4 ± 5.8	(0.00)
breast-cancer	27.4 ± 5.8	30.3 ± 5.8	(0.06)
diabetis	26.8 ± 2.4*	28.5 ± 3.4	(0.03)
flare-solar	37.1 ± 4.0	35.5 ± 4.1	(0.89)
german	27.5 ± 3.6	27.6 ± 2.5	(0.44)
heart	16.9 ± 3.2*	20.6 ± 3.6	(0.00)
image	9.4 ± 2.6	11.3 ± 6.2	(0.11)
ringnorm	1.7 ± 0.1*	1.9 ± 0.4	(0.01)
splice	20.7 ± 2.7	19.1 ± 1.7*	(0.98)
thyroid	3.5 ± 2.6*	6.1 ± 2.8	(0.00)
titanic	22.2 ± 0.6	22.3 ± 0.5	(0.29)
twonorm	2.8 ± 0.2*	4.6 ± 1.1	(0.00)
waveform	10.7 ± 0.7*	11.4 ± 0.7	(0.00)

なお, 各欄は平均 ± 標準偏差, *は有意水準 5%で誤答率が小さいことを示す.

表 3.4: 能動 SKM と最良能動 SVM の誤答率の比較 (最終)

ベンチマーク	SKM(最終)	SVM(最終)	p 値
banana	11.1 ± 0.6*	16.8 ± 10.6	(0.01)
breast-cancer	28.2 ± 5.5*	31.8 ± 5.5	(0.02)
diabetis	23.9 ± 2.0*	31.5 ± 3.7	(0.00)
flare-solar	35.8 ± 1.8	34.8 ± 1.6*	(0.96)
german	24.4 ± 2.7*	29.3 ± 4.5	(0.00)
heart	16.9 ± 2.6*	24.3 ± 4.6	(0.00)
image	3.4 ± 0.6	3.2 ± 0.7	(0.82)
ringnorm	1.7 ± 0.1*	1.9 ± 0.4	(0.01)
splice	11.1 ± 0.8	10.8 ± 0.7	(0.86)
thyroid	3.5 ± 2.6*	5.9 ± 2.6	(0.00)
titanic	22.2 ± 0.6	22.3 ± 1.0	(0.46)
twonorm	2.7 ± 0.2*	3.5 ± 0.7	(0.00)
waveform	9.6 ± 0.3*	11.2 ± 1.0	(0.00)

なお、各欄は平均 ± 標準偏差、*は有意水準 5%で誤答率が小さいことを示す。

能動サポートカーネルマシンと最良能動サポートベクターマシンの最終的な誤答率を比較すると、flare-solar, image, splice, titanic を除き、能動サポートカーネルマシンの方が有意に少ない (有意水準 5%)。

さらに、表 3.2 に示した、Raetsch らが C, γ の両者を最適化したサポートベクターマシン (tuned SVM) の第 1 組から第 20 組の平均誤答率と、能動サポートカーネルマシンの第 1 組から 20 組に対する最終誤答率の平均を比べると、flare-solar, german, image を除き、能動サポートカーネルマシンは tuned SVM と同等か、有意に低い誤答率となっている。有意差がある german, image でも差は 2%以内、flare-solar で 4% 以内であった。

本実験では、能動サポートカーネルマシンのパラメータ C を 1000 に固定し、さらに、要素カーネル集合も全てのデータベースに対して同一のものを使用している。それにも関わらず、全学習データを学習した段階で、 C, γ の両者を最適化した tuned SVM とほぼ同等の性能を持つことから、サポートカーネルマシンはバッチ学習においても高い判別性能を実現しやすいことが分かる。

従って、能動サポートカーネルマシンは、同一の要素カーネル群で 13 デー

表 3.5: 2 段階能動 SKM のモード

名称	α -mode	β -mode	n
basic	full	full	∞
fullfull	full	full	50
full	full	once	50
wss1	wss1	once	50
wss2	wss2	once	50

タベースほとんど全てに対して、個別に最適化されたサポートベクターマシンとほぼ同等の性能を示し、能動学習過程でも最良の性能を持つ要素カーネルによるサポートベクターマシンとほぼ同等の性能を持つことがわかる。

以上から、能動サポートカーネルマシンは汎用性の高い、強力な能動学習手法であることが示された。

3.4.4 能動サポートカーネルマシンにおけるモードの影響

本節では、単純能動サポートカーネルマシンの能動学習時の応答性能を向上させた 2 段階能動サポートカーネルマシンの各モード (表 3.5) の誤答率及び応答時間 (ラベル付けと次のラベル付け事例提示の間の時間) を比較する。

なお、学習モード basic は、単純能動サポートカーネルと同一である。また、誤答率、応答時間は、共に、各ベンチマークデータの第 1 組から 20 組に対する結果に基づく。

full と fullfull、及び、wss1 と wss2 はほぼ同じ結果であったため、basic (単純能動サポートカーネルマシン)、full、wss2 の 3 つについて、誤答率と応答時間 (ラベル付け後新たなラベル付けサンプルを選択するまでの時間) を、図 3.10、図 3.11、図 3.12 に示す。

まず、誤答率 (各図の第 1 列) については、各モードでほとんど差がなかった。表 3.6 にラベル付けサンプル数が 50、100 の時点、および、表 3.7 に最終段階での basic、wss2 での誤答率の平均、標準偏差を示す。ラベル付きサンプル数 50 の時の image を除き、誤答率に関して有意差は生じていない。

第 3 章 サポートカーネルマシンに基づく能動学習の提案

一方, 応答時間 (各図の第 2 列) は, basic, fullfull, full が場合によっては数秒 ~ 数 10 秒を要するのに対して, wss1, wss2 は瞬時であった。

ラベル付けサンプル数が 100 個の時点でも, basic では, 平均応答時間の全ベンチマーク平均は 2.0 秒, 90% 点の全ベンチマーク平均は 3.7 秒で, 毎回数秒を要する。これに対して, wss2 では, 平均 0.05 秒, 90% 点平均でも 0.10 秒でリアルタイムな応答が可能である。

なお, breast-cancer を除き全てのデータベースで, 一定個数のラベル付けの後, basic, full の応答時間が増加傾向から減少傾向に転じる。特に, image, rignnorm, splice, thyroid, twonorm, waveform では, 最終的に, ほぼ瞬時に応答するようになる。これは, α, β がほぼ最適値に収束し, かつ, 未学習の例外的事例がほとんどなくなり, α, β の更新が生じなくなったためである。

さらに, 図 3.13 図 3.14 図 3.15 に, 以下の時間の変化を示す。

- (1) basic モードでの各時点での応答時間
- (2) wss2 モードでの完全最適化段階での計算時間

図から以下が分かる。

- basic モードの各時点の応答時間は増加の後減少する傾向がある。
- basic モードでの応答時間が増加傾向を示す間は, 各回の応答時間は, wss2 での完全最適化とほぼ同じ計算時間である。

なお, 各モードでのトータルな計算時間は以下で見積もれる。

- basic モード: 各サンプル数での応答時間の累積
- wss2 モード: 完全最適化時間 (能動学習段階の応答時間の累積は無視可能)

従って, wss2 モードによる 2 段階能動サポートカーネルマシンでは, 単純能動サポートカーネルマシン (basic モード) に比べて, トータルな計算量の大幅な削減が可能となる。

以上から, wss1, wss2 モードによる 2 段階能動サポートカーネルマシンは, 誤答率に影響を与えずに, 専門家との対話的能動学習に適した高い応答性を持つことが確認できた。

表 3.6: モード basic と wss2 の誤答率の比較 (中間)

ベンチマーク	basic(50)	wss2(50)	p 値
banana	16.8 ± 7.1	16.2 ± 6.5	(0.60)
breast-cancer	28.4 ± 5.4	28.8 ± 5.3	(0.41)
diabetis	28.6 ± 3.2	30.6 ± 3.3	(0.08)
flare-solar	36.7 ± 4.2	36.1 ± 4.3	(0.66)
german	30.0 ± 3.0	29.3 ± 2.7	(0.76)
heart	18.4 ± 4.2	18.6 ± 3.3	(0.40)
image	14.5 ± 3.0*	17.9 ± 5.7	(0.01)
ringnorm	2.0 ± 0.2	2.1 ± 0.2	(0.14)
splice	25.7 ± 3.2	25.4 ± 3.6	(0.59)
thyroid	4.1 ± 2.8	4.4 ± 2.6	(0.35)
titanic	23.9 ± 4.9	22.7 ± 1.6	(0.83)
twonorm	3.3 ± 0.4	3.4 ± 0.4	(0.31)
waveform	12.8 ± 1.2	12.9 ± 1.3	(0.48)
ベンチマーク	basic(100)	wss2(100)	p 値
banana	11.7 ± 0.9	11.7 ± 0.6	(0.41)
breast-cancer	27.4 ± 5.8	28.4 ± 5.5	(0.28)
diabetis	26.8 ± 2.4	27.4 ± 3.6	(0.26)
flare-solar	37.1 ± 4.0	37.0 ± 4.6	(0.51)
german	27.5 ± 3.6	28.0 ± 2.6	(0.29)
heart	16.9 ± 3.2	17.0 ± 3.5	(0.48)
image	9.4 ± 2.6	9.8 ± 1.4	(0.28)
ringnorm	1.7 ± 0.1	1.7 ± 0.1	(0.34)
splice	20.7 ± 2.7	19.6 ± 1.5	(0.92)
thyroid	3.5 ± 2.6	3.5 ± 2.6	(0.50)
titanic	22.2 ± 0.6	22.2 ± 0.6	(0.54)
twonorm	2.8 ± 0.2	2.8 ± 0.2	(0.40)
waveform	10.7 ± 0.7	10.9 ± 0.7	(0.20)

なお, 各欄は平均 ± 標準偏差, *は有意水準 5%で誤答率が小さいことを示す.

表 3.7: モード basic と wss2 の誤答率の比較 (最終)

ベンチマーク	basic(最終)	wss2(最終)	p 値
banana	11.1 ± 0.6	11.1 ± 0.6	(0.50)
breast-cancer	28.2 ± 5.5	28.2 ± 5.5	(0.50)
diabetis	23.9 ± 2.0	23.9 ± 2.0	(0.50)
flare-solar	35.8 ± 1.8	35.8 ± 1.8	(0.50)
german	24.4 ± 2.7	24.4 ± 2.7	(0.50)
heart	16.9 ± 2.6	16.9 ± 2.6	(0.52)
image	3.4 ± 0.6	3.3 ± 0.6	(0.64)
ringnorm	1.7 ± 0.1	1.7 ± 0.1	(0.51)
splice	11.1 ± 0.8	11.0 ± 0.8	(0.51)
thyroid	3.5 ± 2.6	3.5 ± 2.6	(0.50)
titanic	22.2 ± 0.6	22.2 ± 0.6	(0.50)
twonorm	2.7 ± 0.2	2.7 ± 0.2	(0.47)
waveform	9.6 ± 0.3	9.8 ± 0.7	(0.21)

なお、各欄は平均 ± 標準偏差、*は有意水準 5%で誤答率が小さいことを示す。

3.4.5 考察

前節での実験から以下が明らかとなった。

- サポートカーネルマシンは、同一の要素カーネル集合、学習パラメータ C で、13 データベースほとんど全てに対して、個別に最適化されたサポートベクターマシンとほぼ同等の性能を示した。この結果、バッチ学習時にも、高い判別性能を実現しやすい。

能動学習過程でも最良の性能を持つ要素カーネルによるサポートベクターマシンとほぼ同等の性能を持つ。

- 2段階能動サポートカーネルマシン (学習モード wss2) は、毎回完全な最適化を行う単純能動サポートカーネルマシンと、能動学習過程においても、完全最適化後は、ほぼ同等の誤答率を持つ。

しかし、応答時間は、ラベル付きサンプル数 100 個程度でも、単純能動サポートカーネルマシンが平均的に数秒を要するのに、2段階能動サポートカーネルマシンでは平均 0.1 秒以内で瞬時に応答する。

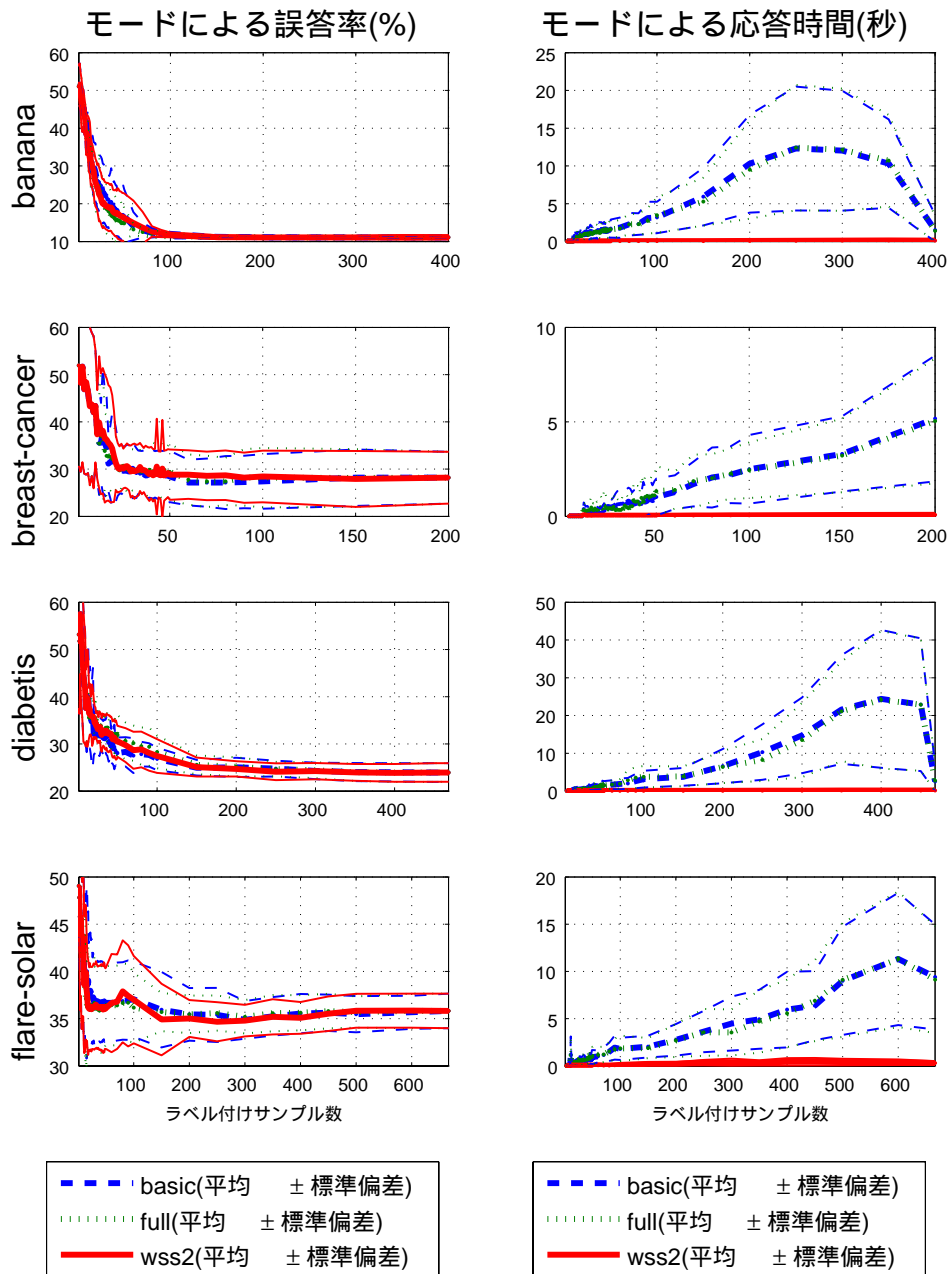


図 3.10: ベンチマーク結果 [20 データベース] (2-1)
(2 段階能動 SKM のモードによる違い)

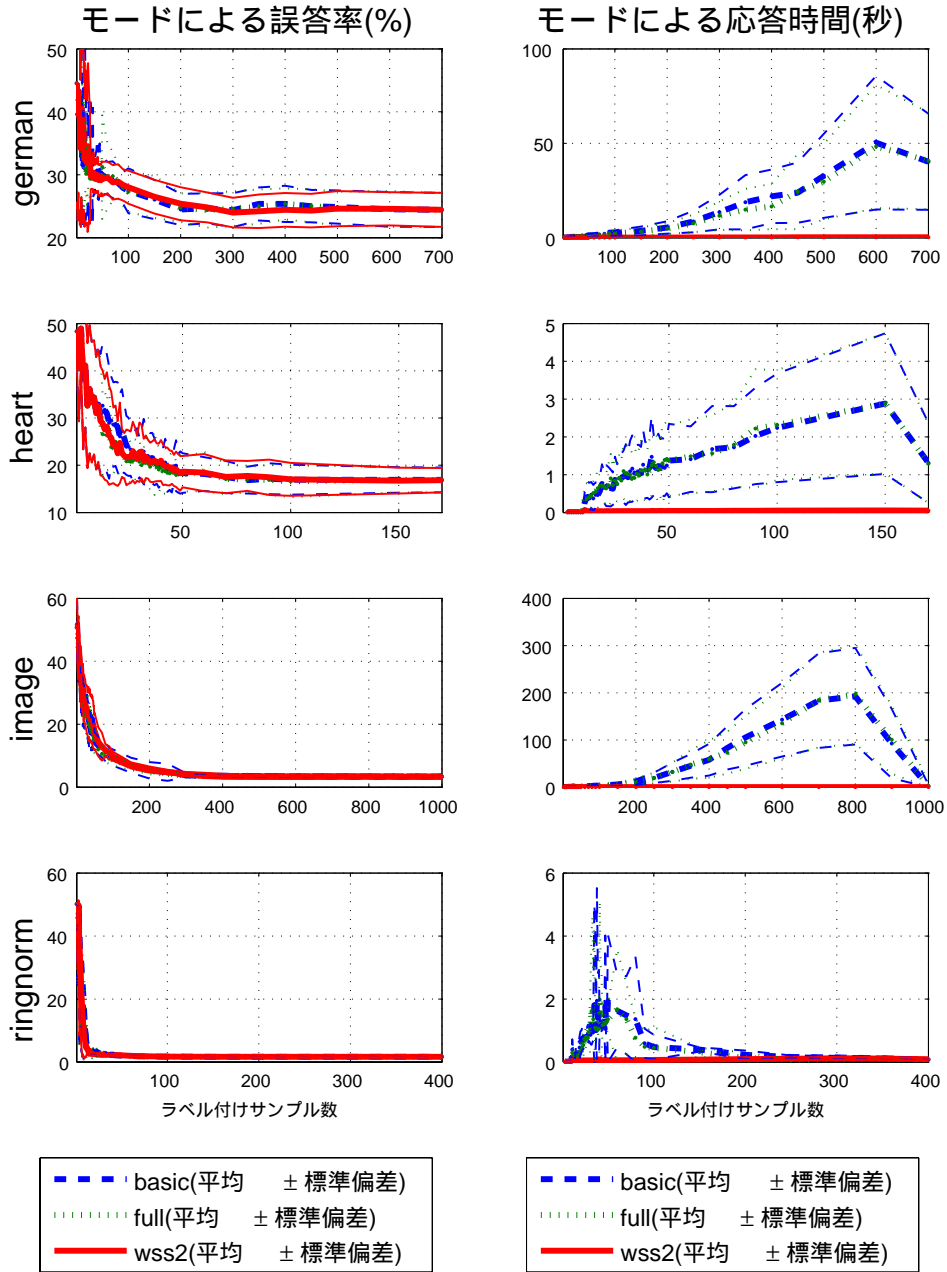


図 3.11: ベンチマーク結果 [20 データベース] (2-2)
(2 段階能動 SKM のモードによる違い)

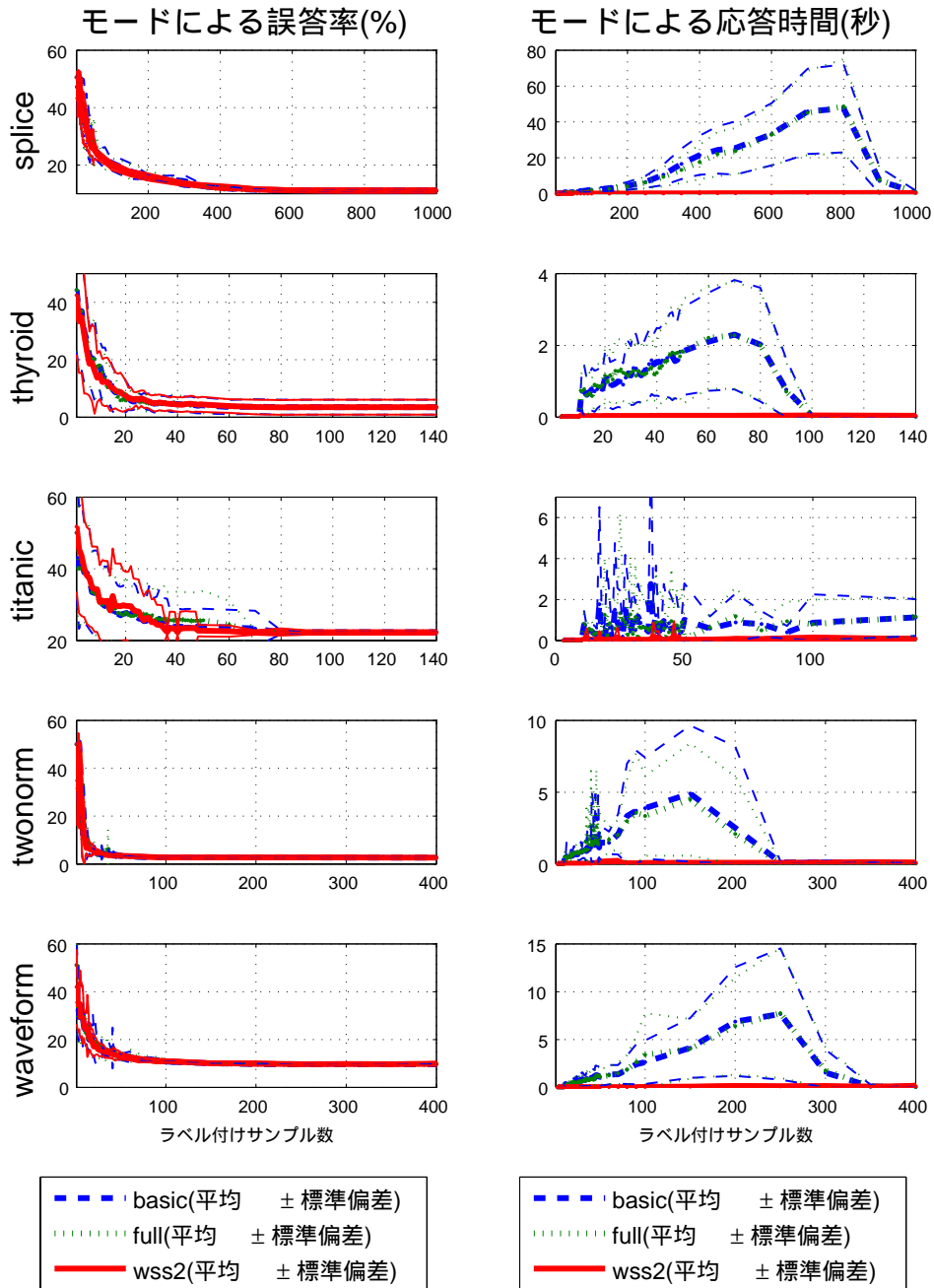


図 3.12: ベンチマーク結果 [20 データベース] (2-3)
(2 段階能動 SKM のモードによる違い)

第 3 章 サポートカーネルマシンに基づく能動学習の提案

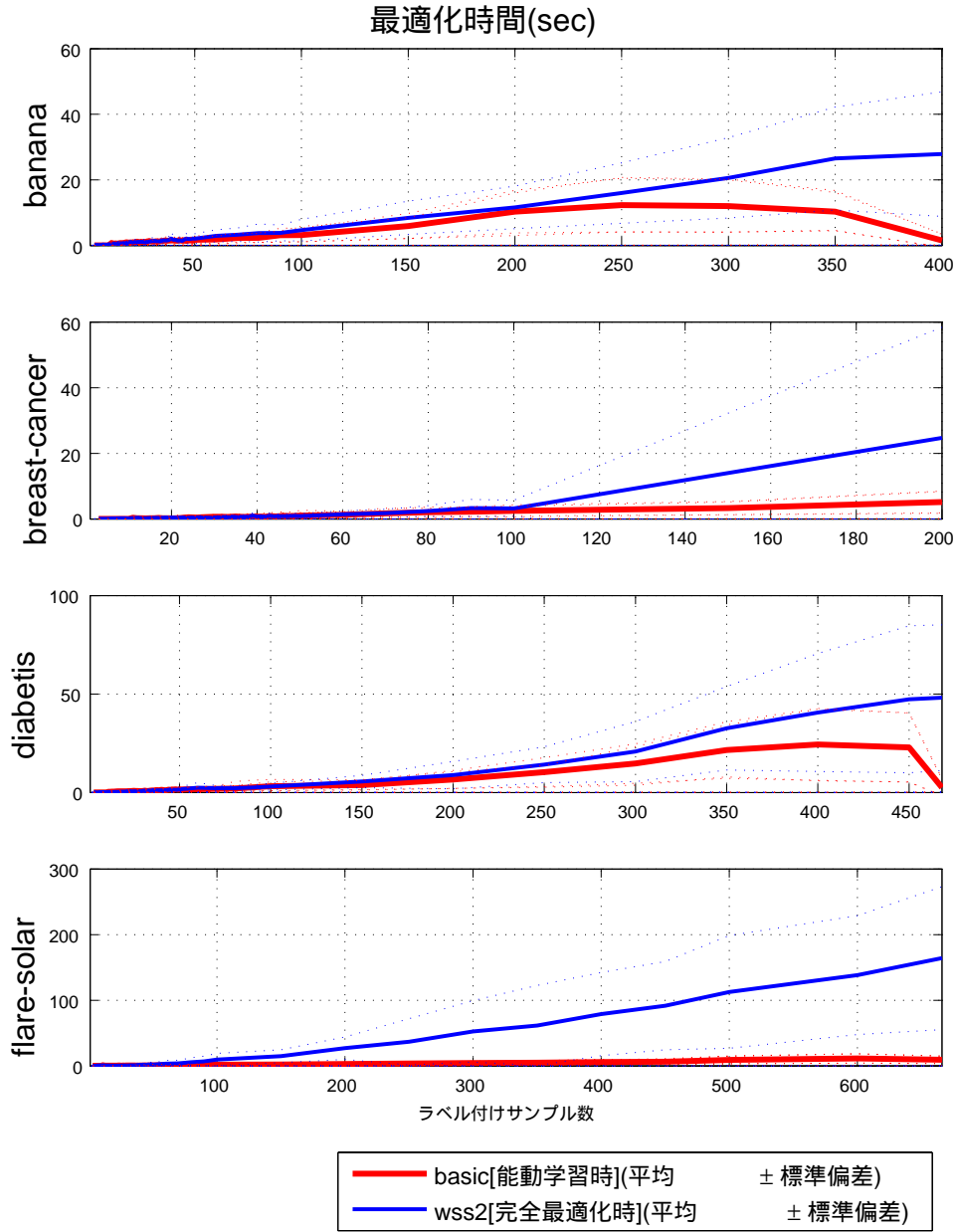


図 3.13: ベンチマーク結果 [20 データベース] (3-1)
(最適化時間の対比)

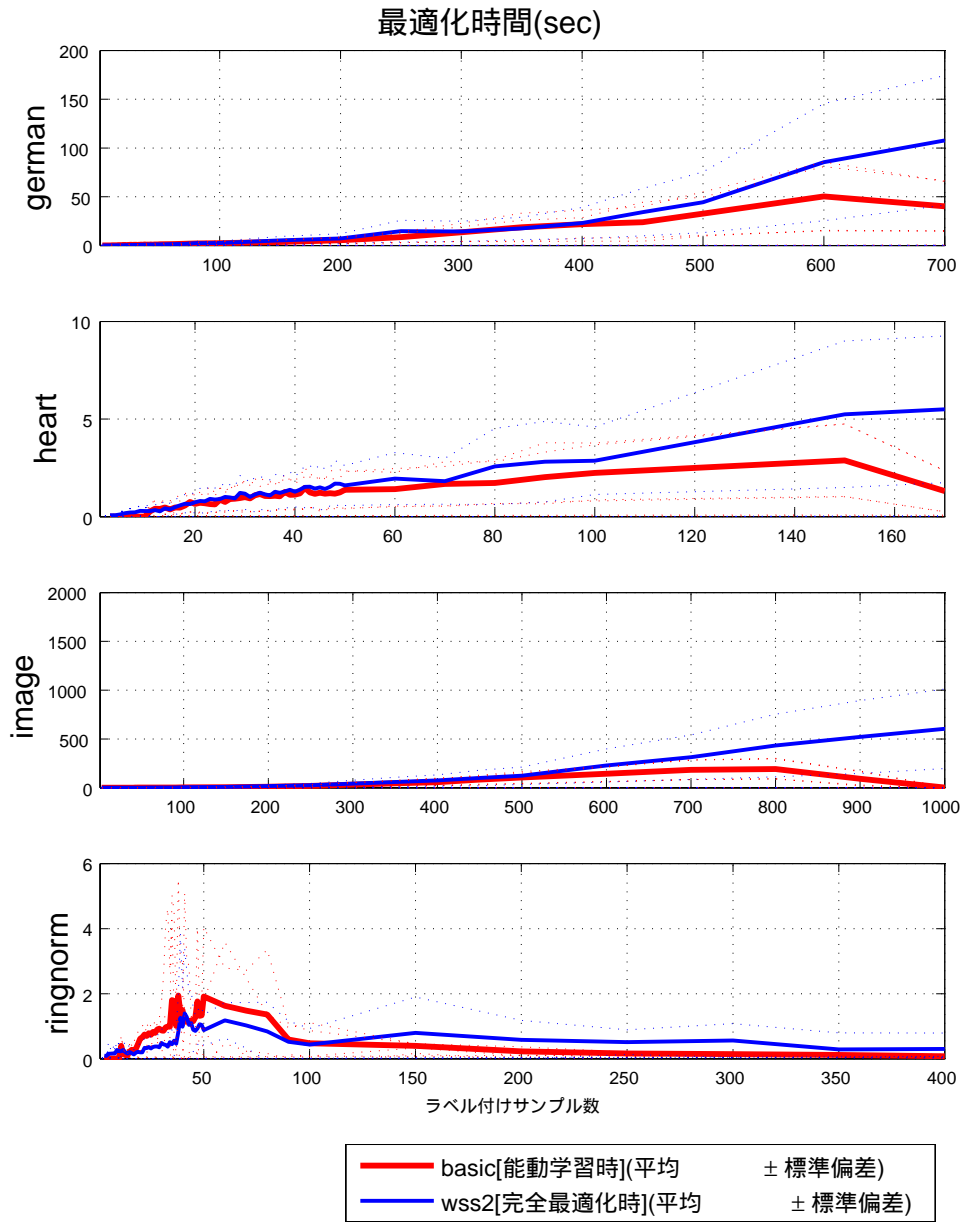


図 3.14: ベンチマーク結果 [20 データベース] (3-2)
(最適化時間の対比)

第3章 サポートカーネルマシンに基づく能動学習の提案

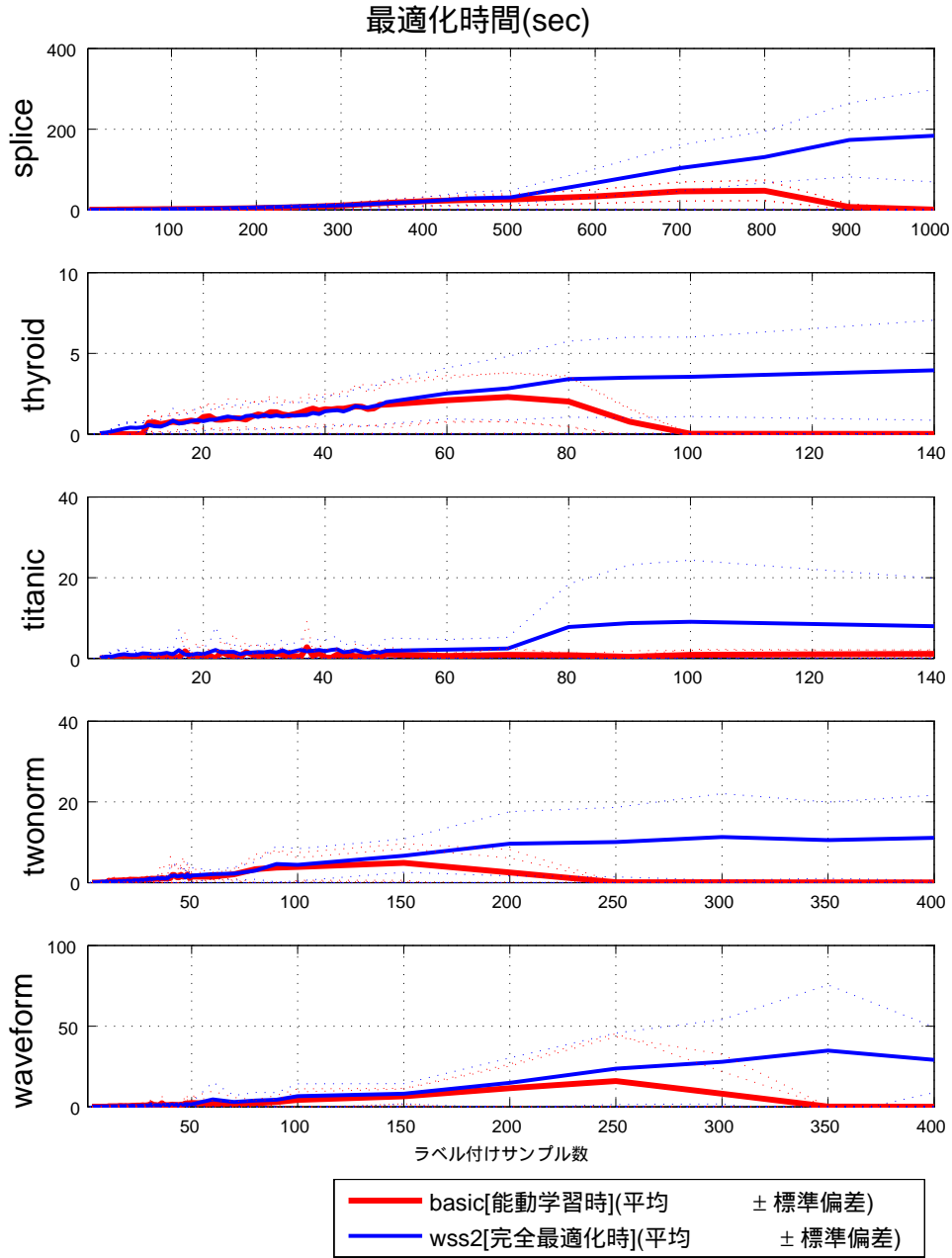


図 3.15: ベンチマーク結果 [20 データベース] (3-3)
(最適化時間の対比)

さらに、同一ラベル付けサンプル数での最適判別関数を得るまでのトータルな計算時間は、単純能動サポートカーネルマシンより2段階能動サポートカーネルマシンの方が短い。

このことから、提案した2段階能動サポートカーネルマシンは汎用性が高く、かつ、高い判別精度を実現できると同時に、インタラクティブに専門家とやり取りをする必要がある能動学習時に高い応答性を持つ、強力な能動学習手法であることが示された。

本節では、以下の2点について考察する。

- 2段階能動サポートカーネルマシンが、単純能動サポートカーネルマシンより最適判別関数のトータルな計算時間が短い理由
- 2段階能動サポートカーネルマシンで、毎回サポートカーネルマシンの完全な最適化を行わずに、サンプル選択を行っても、判別精度があまり悪くならない理由

最適判別関数のトータルな計算時間短縮の理由

各要素カーネルの目的関数値 $S_k(\alpha)$ は、一定数の学習サンプルが得られるまでは、ラベル付きサンプルの追加により、大きく増加する。このため、新たな制約 $\sum_k S_k(\alpha) \beta_k \leq \theta$ は、それまでの多くの線形制約 $\sum_k S_k^t(\alpha) \beta_k \leq \theta$ より厳しく、従来の制約が無効となり易い。このため、前回の α, β の最適値は α, β の新たな最適値の計算のための良い初期値とはならない。従って、単純能動サポートカーネルマシンでの各回の計算時間(応答時間)は、2段階サポートカーネルマシンで最後(完全最適化段階)に1回のみ完全な最適化を行なう際の計算時間とほぼ等しくなる。

このため、最後に1回のみ完全最適化を行う2段階能動サポートカーネルマシンの方が、毎回完全な最適化を行う単純能動サポートカーネルマシンより、トータルな計算時間が早くなりやすい。

2段階能動サポートカーネルマシンでの近似が機能する理由

2段階能動サポートカーネルマシンでは、以下の2つの近似を用いて能動学習時のサンプル選択の高速化を行っている。

- 「ランダム選択+マージン戦略」によるマージン戦略の近似

- サポートカーネルマシンの近似解による最適判別面の近似

以下では、この 2 つのそれぞれについて影響を検討する。

ランダム選択 + マージン戦略による近似の影響

学習用サンプルとテスト用サンプルの分布は類似していると仮定する。

ランダム選択はサンプルの分布密度に比例した選択を行なう。このため、データ量が多い高密度領域のサンプルほど選択され易い。ランダム選択の後にマージン選択を行なうことで、高密度領域かつ判別面近傍のサンプルがラベル付け候補として選択されやすい。このため、本戦略はデータ量の多い領域での正解率を（一定のレベルまで）効率的に向上させる効果がある。ただし、ランダム選択の段階で、判別面更新効果が大きいマージン内領域 ($|f(x)| \leq 1$) のサンプルが 1 つも選択されなくなると、精度向上は進まなくなる。このような状況が生じるのは「全ラベルなしサンプル数」に占める「マージン内領域のラベルなしサンプル数」の比率が小さい場合である（例えば、「比率」 $p = 10\%, 5\%, 1\%$ の場合 50 個中 1 個も選択されない確率 $(1-p)^{50}$ は各々 $0.5\%, 7\%, 60\%$ ）。例外データが存在しない場合には誤答率はこの「比率」とほぼ等しい。しかし、全サンプル数が少ないと、ラベル付けの進行に伴うラベルなしサンプル数の減少により「比率」は大きくなり易いために、また、例外データが多い（誤答率が高い）場合には誤答率に占めるマージン領域の影響が小さいために、本戦略の影響は小さいと予想される。

表 3.6 で、多サンプルかつ正解率が高い唯一のベンチマーク image のラベル付けサンプル数 50 個時点でのみ、wss2 (2 段階能動サポートカーネルマシン) の誤答率が単純能動サポートカーネルマシン (basic) より有意に大きくなったのは、この性質によると考えられる。

近似解に基づくサンプル選択の影響

ラベル付きサンプル数 k の時点で以下の 3 つでの誤答率を調べた。

1. 2 段階能動サポートカーネルマシン (wss2) で得られている近似判別面の最近傍 50 点を一括してラベル付けした後、得られた $k + 50$ 個のサンプルでサポートカーネルマシンを完全最適化

-
2. 得られた k 個のサンプルで完全に最適化したサポートカーネルマシンの判別面の最近傍 50 点を一括してラベル付けした後, 得られた $k + 50$ 個のサンプルでサポートカーネルマシンを完全最適化
 3. 2 段階能動サポートカーネルマシン (wss2) に従って判別面の近似更新を行ないながら 50 点を追加した後, 得られた $k + 50$ 個のサンプルでサポートカーネルマシンを完全最適化

1 と 2 の比較により, 誤答率の視点から見た近似判別面と最適な判別面の違いを, 1 と 3 の比較により, 適応的な判別面の逐次更新の効果を調べることができる。

図 3.16, 図 3.16 に結果を示す. 実験データでは 50 サンプル以降は 1 と 2 の差はほとんど生じない. また, 通常, 1 と 2 の差より 1 と 3 の差が大きく, 1 と 2 の差がなくなった後も, 1 と 3 の差が残った. ただし, 誤答率が安定すると 1 と 3 の差はほぼなくなる。

以上から, 近似判別面は比較的早い段階で各時点での最適判別面の良い近似となった点および, 逐次学習による適応的な判別面の更新の効果はある点が実験的に確認された。

3.5 本章のまとめ

本章では, まず, 能動サポートベクターマシンにおけるカーネル事前選択の困難を解決するための方法として, 単純能動サポートカーネルマシンを導入した。

更に, 単純能動サポートカーネルマシンの問題点である応答の遅さを解決する方策として, 高い応答性を持つ形でサポートカーネルマシンを能動化した 2 段階能動サポートカーネルマシンを提案した. その基本的な考え方は, 専門家に対してラベルを知りたい事例を迅速に提示して「ラベル」を蓄積する「能動学習段階」と, 一定量のラベルの蓄積後, 最適なカーネル組み合わせの選択と判定基準の構築を行う「完全最適化段階」との 2 段階を考え, 能動学習段階においては, サポートカーネルマシンの近似解を用いた応答性を重視したアルゴリズム設計を行い, 完全最適化段階においては解の最適性を重視したアルゴリズム設計を行うというものである。

さらに, 複数の単カーネルサポートベクターマシン, 単純能動サポートカーネルマシン, 2 段階能動サポートカーネルマシンを, 特性の異なる 13 個のデータベースに適用した実験により, 以下を明らかにした。

第3章 サポートカーネルマシンに基づく能動学習の提案

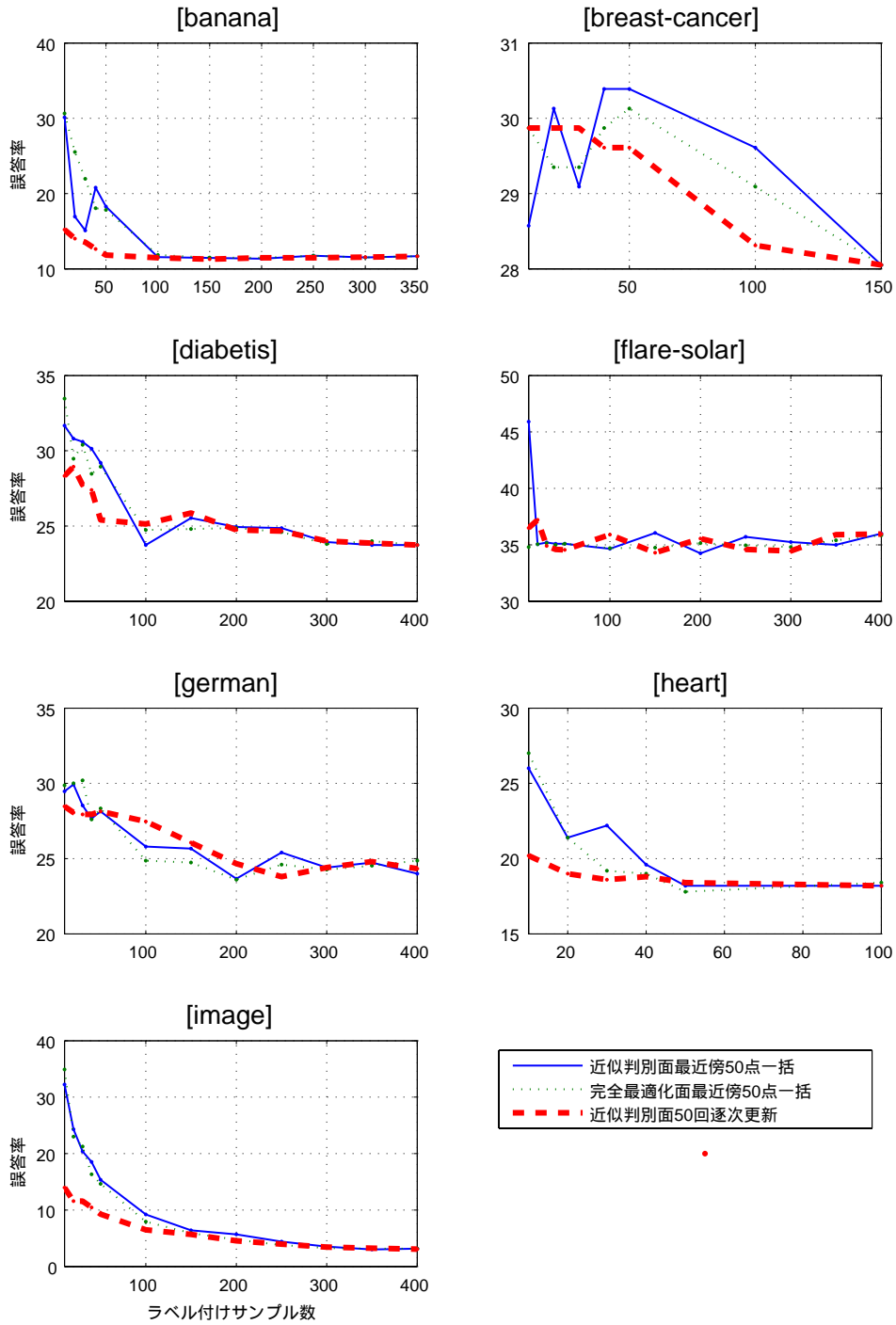


図 3.16: 近似による逐次更新と最近傍選択 (1)

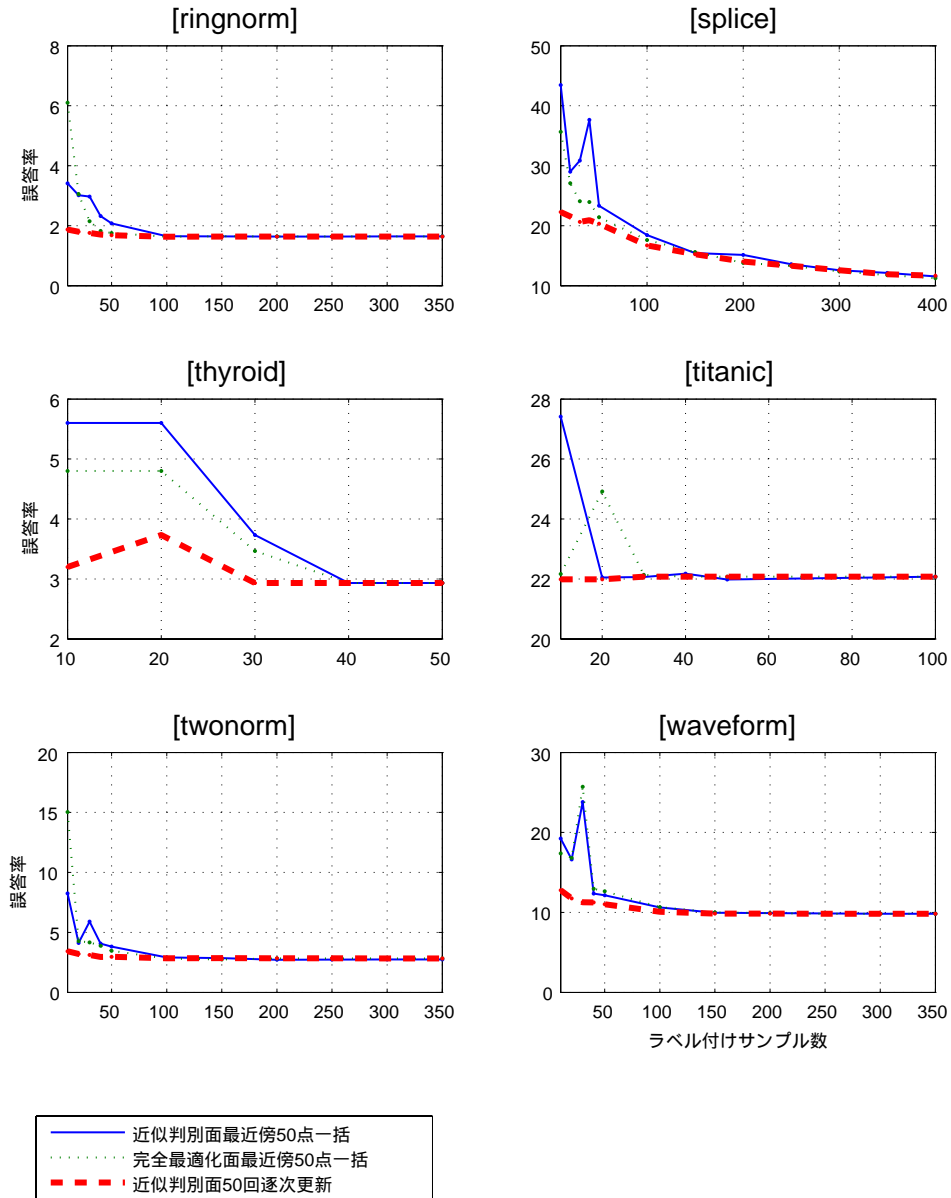


図 3.17: 近似による逐次更新と最近傍選択 (2)

第 3 章 サポートカーネルマシンに基づく能動学習の提案

- 単カーネルサポートベクターマシンを用いた場合、カーネルの選択により能動学習の効率、および、判定精度に大きな差が生じる。
- 同一の要素カーネル集合を用いた能動サポートカーネルマシンは、多様なデータベースに対して、個々のデータベースに対して最適化した単カーネルサポートベクターマシンと同等の学習精度を持つ。これは、能動サポートカーネルマシンが、能動学習の過程で、各データベースに適した合成カーネルを学習するためである。
- 単純能動サポートカーネルマシンが 1 件数 10 秒の応答時間を要する場合でも、2 段階能動サポートカーネルマシンは 1 秒以内の高速な応答が可能である。

さらに、同一ラベル付けサンプル数での最適判別関数を得るまでのトータルな計算時間は、単純能動サポートカーネルマシンより 2 段階能動サポートカーネルマシンの方が短い。

このことから、提案した 2 段階能動サポートカーネルマシンは汎用性が高く、かつ、高い判別精度を実現できると同時に、インタラクティブに専門家とやり取りをする必要がある能動学習時に高い応答性を持つ、強力な能動学習手法であることが示された。

第4章 能動サポートカーネルマシン向きサンプリング戦略

4.1 サンプリング戦略 SKM-SHIFT の提案

本章では, 効率的な多カーネル能動学習を実現する第2の要素としてサンプリング戦略について検討する. 特に, RBF カーネルはサポートベクターマシンで有効性が知られているため, 異なる半径の RBF カーネルを用いる能動サポートカーネルマシンについて検討を行う.

4.1.1 アプローチ

2.3 節で述べたように, サポートベクターマシンにおける代表的サンプリング戦略として以下の3つが挙げられる.

- MARGIN[4, 25]: 判別面に最も近い事例を選択 ($\operatorname{argmin}_x |f(x)|$).
- KFF (Kernel Farthest First)[3]: 既知事例との類似性が最小の事例を選択 ($\operatorname{argmax}_x \min_{i \in L} \|\phi(x) - \phi(x_i)\|$).
- SHIFT[19]: 学習結果の安定性に応じて確率的に KFF または MARGIN 戦略を取る.

MARGIN 戦略は, 判別面の方向・位置の精緻化 (exploitation) 戦略手法である. 一方, KFF 戦略は, 判別面の位置が不明な段階で有効な探索 (exploration) 戦略とされる. SHIFT は, 初期段階は探索型の KFF 戦略を, 学習結果が安定すると精緻化戦略である MARGIN 戦略へと移行を行うものであった. ただし, サポートベクターマシンにおいては, MARGIN 戦略が有効な場合が多く, 最も一般的な戦略である. 学習結果の安定性に応じて

第 4 章 能動サポートカーネルマシン向きサンプリング戦略

戦略を切り替える SHFT 戦略は, XOR 型問題の初期段階などでの有効が示されているが限られた局面である. これは, サポートベクターマシンにおける場合, 判別面の垂線ベクトル $w = \sum_i \alpha_i \phi(x_i)/R^2$ は, 比較的少数のサポートベクター ($\alpha_i \neq 0$) により定められるなど, w の曖昧性は比較的少数のサンプルにより解消されやすい傾向があるためだと考えられる.

しかし, サポートカーネルマシンでは, 判別面の垂線方向は

$$w = (\beta_1 \sum_i \alpha_i \phi_1(x_i)^\top / R_1^2, \dots, \beta_M \sum_i \alpha_i \phi_M(x_i)^\top / R_M^2)^\top$$

であり, 学習過程においては α のみではなく β も変動し, w の方向に大きく影響を及ぼす. このため α のみ変動するサポートベクターマシンに比べその判別面は学習で大きく変動しやすい. 従ってサポートカーネルマシンの能動学習においてはサポートベクターマシンの場合よりも判別面の変動を考慮したサンプリング戦略が重要となると考えられる.

本章では, この考え方に基づいて, サポートベクターマシンのサンプリング戦略 SHIFT 戦略をサポートカーネルマシン用に拡張したサンプリング戦略を提案する.

4.1.2 SKM-SHIFT 戦略

サポートカーネルマシンのための SHIFT 戦略 SKM-SHIFT を図 4.1.1 に示す.

MARGIN 戦略は判別面の方向 w が安定している場合の 2 分探索であるから, SKM-SHIFT 戦略ではサポートカーネルマシンにおける判別面の方向 w_β の安定性に応じて KFF 戦略から MARGIN 戦略へ移行する.

式 (2.25) より任意の要素カーネル K_k で $\|\sum_i \alpha_i \phi_k(x_i)/R_k\| \rightarrow \gamma^*$ であるから,

$$w_0^t = (\beta_1^t \sum_i \alpha_i^t \phi_1(x_i)^\top / R_1, \dots, \beta_M^t \sum_i \alpha_i^t \phi_M(x_i)^\top / R_M)^\top$$

に着目し, 判別面の安定性を下記に定める w_0^t と w_0^{t-1} の角度の余弦 $\text{corr}(\alpha^t, \beta^t, \alpha^{t-1}, \beta^{t-1})$ により測る.

$$\text{corr}(\alpha^t, \beta^t, \alpha^{t-1}, \beta^{t-1}) = \frac{\langle w_\beta^{0,t}, w_\beta^{0,t-1} \rangle}{\sqrt{\langle w_\beta^{0,t}, w_\beta^{0,t} \rangle \langle w_\beta^{0,t-1}, w_\beta^{0,t-1} \rangle}}$$

$$\text{ただし, } \langle w_\beta^{0,u}, w_\beta^{0,v} \rangle = \sum_{i,j,k} \beta_k^u \beta_k^v \alpha_i^u \alpha_j^v K_k(x_i, x_j)$$

入力

U :ラベルなしデータ, O :オラクル
 λ :半減期, ϵ :上限確率

変数

L :ラベル付データ
 $p_0 \leftarrow 1, \alpha^0 \leftarrow 0, \beta^0 \leftarrow (1/k, \dots, 1/k)$.

```
1: if  $U$  が空 then
2:   return empty
3: end if
4:  $\phi \leftarrow \text{angle}(\alpha, \beta, \alpha^0, \beta^0)$ .
5:  $p \leftarrow \max(\min(p_0 e^{-\lambda\phi}, 1 - \epsilon), \epsilon)$ .
6:  $r \leftarrow$  一様乱数  $\in [0, 1]$ .
7: if  $r < p$  then
8:    $x \leftarrow \text{KFF}(U, \alpha, \beta)$  //探索
9: else
10:   $x \leftarrow \text{MARGIN}(U, \alpha, \beta)$ . //詳細化
11: end if
12:  $y \leftarrow O(x)$ .
13:  $L \leftarrow L \cup (x, y)$ .
14:  $U \leftarrow U - x$ .
15:  $p_0 \leftarrow p, \alpha^0 \leftarrow \alpha, \beta^0 \leftarrow \beta$ .
16: return  $(x, y)$ 
```

アルゴリズム 4.1.1: SKM-SHIFT 戦略

4.1.3 サンプリング対象の絞込みによる高速化

SKM-SHIFT で用いる KFF は、全ラベル付きサンプルに対する最小距離 $\min_{i \in L} \|\phi(x) - \phi(x_i)\|$ ではなくワーキングメモリー上に保持された \mathcal{L} またはサポートベクター候補 S に対する最小距離 $\min_{i \in S} \|\phi(x) - \phi(x_i)\|$ の最大化を行う。

また、MARGIN 戦略や SKM-SHIFT 戦略を用いて最適なサンプルを選択するには、基本的には、ラベルなしサンプルすべてを評価する必要がある。しかしデータ数が多い場合全数評価は非常に時間がかかる。そこで、2 段階能動サポートカーネルマシンの設計でも述べたように、応答性を確保するため、全ラベルなしサンプル中で最適なサンプルではなく、ランダムに選択された一定数のサンプル中で最適なものを選択する方式を用いる。

この時、MARGIN および KFF ではランダムサンプルとサポートベクター候補 S 中の各サンプルの間に対して M 個のカーネル計算が必要で、サポートベクター候補数 $M \times |S|$ に比例する評価時間を要する。SKM-SHIFT 戦略では、MARGIN または KFF の計算に加えて $\text{corr}(\alpha^t, \beta^t, \alpha^{t-1}, \beta^{t-1})$ の計算が必要となる。ただし、 $\langle w_\beta^{0,t-1}, w_\beta^{0,t-1} \rangle$ を記憶している場合、 $\langle w_\beta^{0,t}, w_\beta^{0,s} \rangle = \sum_{j,k} \alpha_j^s \beta_k^t \beta_k^s (\sum_i \alpha_i^t K_k(x_i, x_j))$ で、2 段階能動サポートカーネルマシンで時刻 t のサポートベクター候補 j の勾配 $g_{[k]j} = y_j - \sum_i \alpha_i^t K_k(x_i, x_j)$ が計算済みであるため、新たなカーネル計算を要するのは、時刻 $t - 1$ ではサポートベクター候補で時刻 t でサポートベクター候補から除外された事例 j (通常高々数個) に対する $\sum_i \alpha_i^t K_k(x_i, x_j)$ の計算のみである (\mathcal{L} を用いる場合は新たなカーネル計算は不要)。このため、SKM-SHIFT 戦略は MARGIN 戦略とほぼ同じ時間で選択が可能となる。

4.2 実験

実験では、C++版のシステムを用いた。

本実験では代表的な手書き数字認識用ベンチマークである USPS データ¹を用いた。各画像は 16×16 ピクセルの画像で、平均 0、分散 1 に正規化した。実験では、学習用データ 7329 画像、テスト用データ 1969 画像からなる 10 個のデータセットを用いた。

実験は $\gamma = 1, 2, 5, 10, 20, 50$ の 6 種類の RBF カーネル ($\exp(-\gamma \|x_i - x_j\|^2)$) を用意し、各々の単カーネルサポートベクターマシン、全カーネル

¹<ftp://ftp.kyb.tuebingen.mpg.de/pub/bs/data/>

を用いたサポートカーネルマシンに対して、各データセットに対する画像認識を行わせた。サポートベクターマシン、サポートカーネルマシンの定式化におけるコスト C は 1000, 半径 $R_k = 1$ とした。収束条件は、サポートベクターマシン、サポートカーネルマシン共に $\tau = 1^{-3}$, サポートカーネルマシンの $\Delta\theta/\theta$ の上限 $\epsilon = 1^{-6}$, とした。

サンプリング戦略は RANDOM, MARGIN, SKM-SHIFT の 3 つを使用した。KFF は初期を除き明らかに性能が悪いため除外した。SKM-SHIFT(アルゴリズム 4.1.1) のパラメータは $\lambda = 0.5$, $\phi_0 = 0.3$, $\epsilon = 0.95$, サンプリングで用いるランダム抽出数 $M = 50$ とした。

訓練データの全数学習では、サポートベクターマシンについては libsvm [5] を用い、サポートカーネルマシンは C++ 版の 2 段階サポートカーネルマシンを用いた。

能動学習の実験では C++ 版 2 段階サポートカーネルマシンを用いた。能動サポートベクターマシンは、単一カーネルを要素カーネルとした C++ 版 2 段階サポートカーネルマシンを用いた。データ量が多いため、ラベル付きサンプル集合 \mathcal{L} の代わりにサポートベクター候補集合 S を用いるオプションを使用している。

同一パラメータ、同一戦略に対して 20 試行を行い、実験結果はその平均を示す。各試行ではランダムに選択した正例、負例各 1 個計 2 事例を初期データとした。正解率、カーネル配分 β の分析は、事例数が 12, 14, 16, ..., 28, 30, 40, 50, ..., 90, 100, 200, ..., 800, 900 となった各時点での最適解に基づく。

4.2.1 単カーネルサポートベクターマシンとサポートカーネルマシンの比較

全訓練データを用いた場合の各単カーネルサポートベクターマシンの最適解とサポートカーネルマシンの最適解との比較を行った。

データセット 0 に対しては、 $\gamma=1, 2, 5, 10, 20, 50$ の各サポートベクターマシンの汎化誤差指標が 1334.1, 467.6, 175.6, 163.3, 331.4, 1082.4 であるのに対して、最適サポートカーネルマシンの汎化誤差指標 $S(\alpha^*; \beta^*)$ は 149.8 であった。 $\gamma=5, 10, 20$ に対応する β_k^* のみが非零で、各々 0.5, 0.2, 0.3 となった。単独での汎化誤差指標が最小の 3 カーネルのみがサポートカーネルとして選択され、汎化誤差指標がより小さい線形合成カーネルが構成された。テストセットに対する最適サポートカーネルマシンの正解

表 4.1: SKM における最適配分 β^*

データセット	RBF パラメータ γ					
	1	2	5	10	20	50
0	0.00	0.00	0.52	0.19	0.29	0.00
1	0.00	0.00	0.48	0.24	0.22	0.06
2	0.00	0.00	0.48	0.28	0.13	0.11
3	0.00	0.00	0.53	0.17	0.29	0.01
4	0.00	0.00	0.44	0.36	0.11	0.09
5	0.00	0.00	0.46	0.29	0.15	0.10
6	0.00	0.00	0.47	0.21	0.32	0.00
7	0.00	0.00	0.52	0.20	0.24	0.04
8	0.00	0.00	0.44	0.37	0.03	0.16
9	0.00	0.00	0.42	0.38	0.16	0.05

率は 99.54% で, $\gamma = 10$ のサポートベクターマシン (99.64%) に次ぐ 2 位であった . 各サポートカーネルに対する $S_k(\alpha^*)$ はいずれも同一値 149.8 ($=S(\alpha^*; \beta^*)$) で, どの非サポートカーネルの $S_k(\alpha^*)$ より小さい .

データセット 0 以外の各データセットも同様に, 構成されたサポートカーネルマシンは汎化誤差指標は最小, 正解率は 1 位または 2 位であった . 各データセットに対する最適配分 β^* を表 4.1 に示す .

いずれのデータセットに対しても $\gamma = 5, 10, 20$ がサポートカーネルで, この内 $\gamma = 5$ の配分が最も高い . 最適サポートカーネルマシンは, 半径の大きい $\gamma = 5$ の RBF カーネルで判別面の概要を記述し, 半径の小さい $\gamma = 10$ や 20 の RBF カーネルで補正する構成となっている . 一方, 単カーネルサポートベクターマシンでは $\gamma = 10$ が汎化誤差指標が最小であった .

4.2.2 能動サポートカーネルマシンでのサンプリング戦略の比較

応答性

各事例数での応答時間(点線)とサポートベクター候補数 $|S|$ (実線)の平均を戦略別に図 4.1 に示す。ここで応答時間とは、実験に用いた Pentium 4 CPU 2.6GHz の Windows XP マシンで、ラベル付け後システムが次のラベル付け候補を提示するまでの時間である。

学習済み事例数 900 の時でも、RANDOM 戦略は約 0.02 秒、MARGIN、SHIFT 戦略は約 0.9 秒で応答する。

RANDOM 戦略は乱数発生のみで選択が行えるため応答は非常に迅速であるが、有効なサポートベクター候補を選択できずサポートベクター候補数が増加しない。MARGIN と SHIFT の応答時間はサポートベクター候補数に比例して増加し、両者ともほぼ同じ応答時間である。

なお、学習済み事例数 900 に対する再学習に要した時間は MARGIN 戦略、SHIFT 戦略とも約 30 秒であった。

正解率の推移

能動サポートカーネルマシンでの事例数増加に伴う正解率の推移について述べる。

実験では各サンプリング戦略の正解率はデータセットによらずほぼ同じ推移を示した。各戦略について各データセット 20 試行計 200 試行の平均正解率の推移を実線で、各データセットでの 20 試行平均を点で図 4.2 に示す。

RANDOM 戦略は明らかに劣るが、MARGIN 戦略と SHIFT 戦略は正解率については同等の性能を示す。正確にはデータセット 3, 4 を除くすべてのデータセットで 30 ~ 80 事例まで(データセット 3, 4 は 200 事例まで)、MARGIN の方が有意(有意水準 10%)に正解率が高く、その後有意差がなくなる。ただしどのデータセットも 50 事例で正解率の差は 0.5% 以内となった。

また、500 事例で MARGIN、SHIFT とともに全訓練データを用いた最良のサポートベクターマシン($\gamma = 10$)との正解率の差が 0.1% 以内となり、ほぼ同性能に達した。

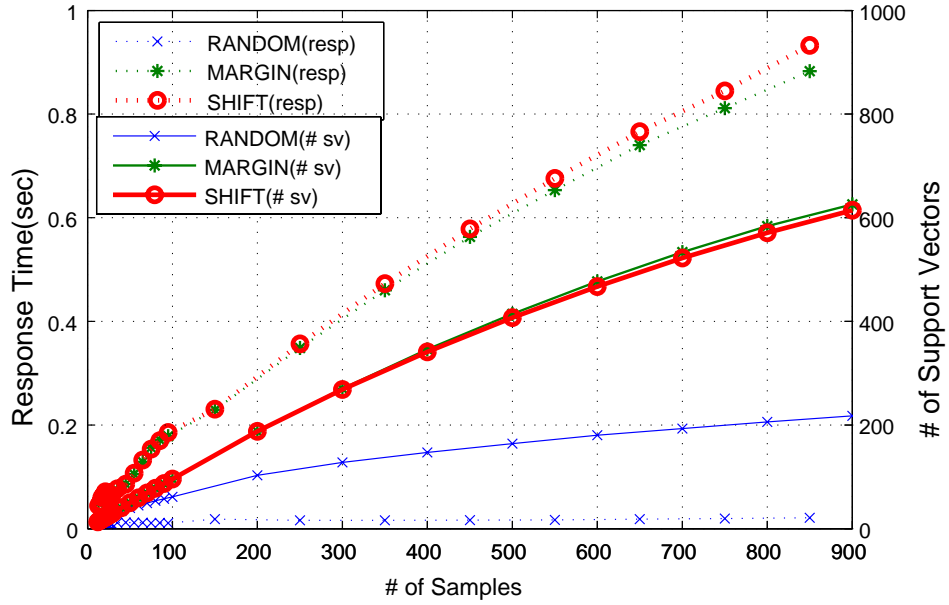


図 4.1: 能動 SKM の応答時間とサポートベクター数の推移

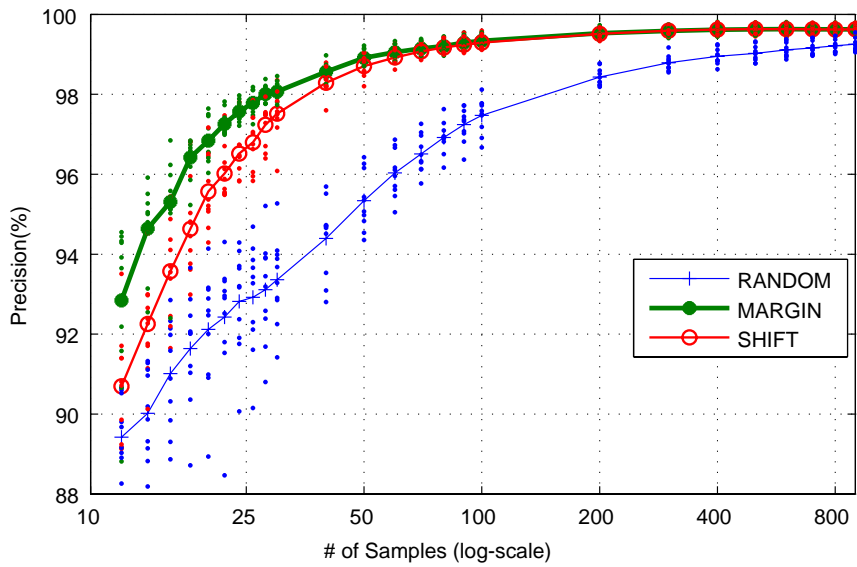


図 4.2: 能動 SKM での正解率の推移

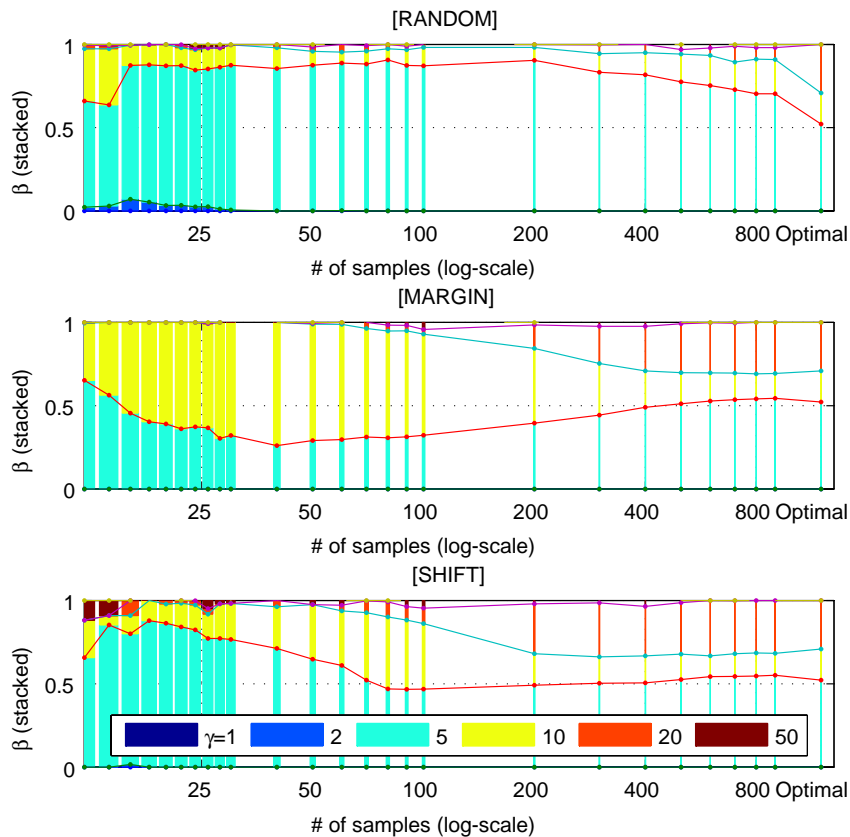


図 4.3: 能動 SKM でのカーネル配分 β の平均的推移

なお、単カーネルサポートベクターマシンによる能動学習においてもデータセット、 γ に関わりなく、RANDOM 戦略は性能が悪く、MARGIN、SHIFT 戦略は同等の性能を示した。

従って、本実験ではサポートカーネルマシン、サポートベクターマシン共に正解率の向上に関しては SHIFT 戦略の優位性が見られなかった。

4.2.3 カーネル配分の推移

能動サポートカーネルマシンでのカーネル配分 β の推移は、MARGIN 戦略と SHIFT 戦略で性能差が生じた。

図 4.3 にデータセット 0 に対する能動学習でのカーネル配分 β の平均

第 4 章 能動サポートカーネルマシン向きサンプリング戦略

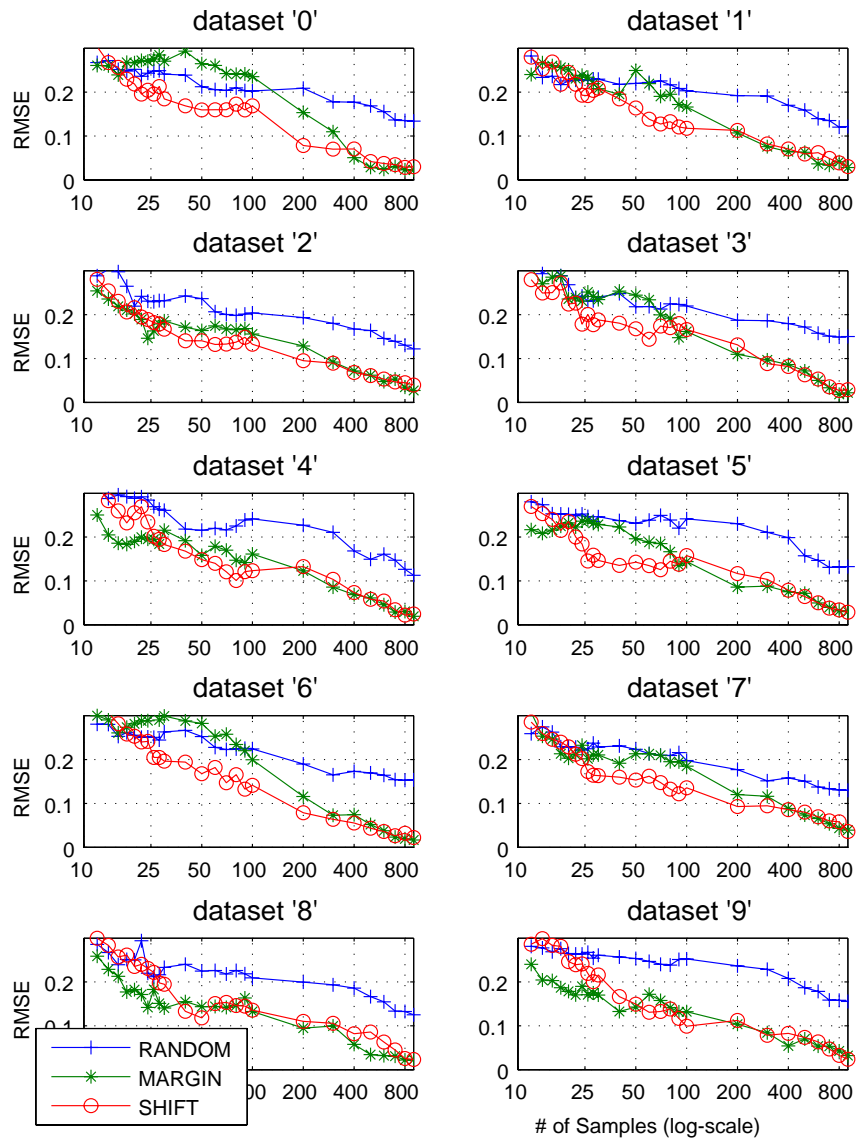


図 4.4: 能動 SKM でのカーネル配分誤差 $\|\beta - \beta^*\|$ の推移

的推移を示す(20 試行平均)．データセット 0 に対する全訓練データを用いた最適配分 β^* は $\gamma = 5, 10, 20$ に対して 0.5, 0.2, 0.3 である(表 4.1)．MARGIN 戦略では 200 事例程度まで最適カーネル配分 0.2 の $\gamma = 10$ に大きな配分がなされ易く, 全訓練データを用いた最適カーネル配分 β^* (図右端)に近い配分に達するには 400 事例以上が必要であったが, SHIFT 戦略では 200 事例程度でほぼ β^* に達した．同様の傾向は他のデータセットでも生じた．

MARGIN 戦略の初期段階では, データセットにかかわらず単カーネルサポートベクターマシンとして汎化誤差指標が最小の $\gamma = 10$ が選択されやすかった(20 事例程度で $\gamma = 10$ に対する β の配分は平均的に 0.5 から 0.6 程度)．また, 最適 β^* への収束については SHIFT 戦略の方がより早く近づく傾向がみられた．図 4.4 に, 各データセット, 各戦略について最適カーネル配分 β^* と各時点での β との RMSE(Root Mean Square Errors)の推移を示す．データセット 4, 8, 9 では 30 事例程度まで MARGIN 戦略の方が有意に β^* に近いが, それ以外のデータセットでは 100 事例近くまで SHIFT 戦略の方が有意に β^* に近い配分を得ている(有意水準 10%)．

4.2.4 考察

提案した能動学習方式は学習済み事例数が 1000 近くでも専門家がラベル付け後 1 秒以内にシステムがラベル付け対象を選択する．このため, インタラクティブな能動学習に耐え得る応答性をもつと考えられる．

また, 実験から, SHIFT 戦略は MARGIN 戦略に比べて, 精度面では必ずしも優位ではないが, カーネル選択すなわち最適なカーネル配分の推定においては, 一部の事例を除いては有効性が高いことが確認された．

一部例外が生じた理由は次のように考えられる．

前述のように MARGIN 戦略では早い段階で $\gamma = 10$ のカーネルに高い配分が与えられた．データセット 4, 8, 9 は最適配分 β^* での $\gamma = 10$ の比率が 0.35 から 0.38 で, 他のデータセットに比べて高い(表 4.1)．このため, 最適判別面と MARGIN 戦略による判別面の類似性が高いので, MARGIN 戦略によっても比較的早く適切な β を得ることが可能であったと考えられる．この結果, データセット 4, 8, 9 で SHIFT 戦略の優位性が生じにくかったと推察される．

しかし一般には, 単カーネルとして最適なものがサポートカーネルマシンでも高い配分を得られるとは限らない．このため, 単カーネルとして最

適の $\gamma = 10$ の最適配分 β_k^* があまり大きくない他のデータセットでは, 初期段階で現在の判別面の近傍に限らず, より広く事例を集める SHIFT 戦略が有効に機能したと推察される.

ただし本章では, 要素カーネルとして RBF カーネルのみを用いた検討を行った. $h(x) = \sum_k \beta_k \exp(-g_k x)$ とすると, 合成カーネル $K(x_i, x_j; \beta) = h(\|x_i - x_j\|^2)$, KFF 戦略で用いる平方距離 $\|\phi(x_i) - \phi(x_j)\|^2 = 2(1 - K(x_i, x_j; \beta)) = 2(1 - h(\|x_i - x_j\|^2))$ である. $h(x)$ は狭義単調増加であるため, $\|\phi(x_i) - \phi(x_j)\|$ の大小順は $\|x_i - x_j\|$ の大小順と同一である. 従って, 合成 RBF カーネルに対する KFF 戦略による選択は, x の空間での Farthest First (FF) 戦略と同一である. また, KFF (または FF) 戦略は, 選択されたデータ点がなす網の目を順次狭くしていく操作であり, MARGIN 戦略で選択される比較的データ間隔の短いサンプルと KFF 戦略によるデータ間隔が長いサンプルが混在していることは, 完全最適化段階で適切な RBF 半径を決定する際に, 有効な情報となっている可能性も高い. このため, RBF 以外のカーネルでの SKM-SHIFT の効果についてはより詳細な検討が必要である.

4.3 本章のまとめ

本章では, 最適なカーネルの組み合わせを早期に見出す能動サポートカーネルマシンのためのサンプリング手法 SKM-SHIFT を提案した. その基本的考え方は, 最適なカーネルの組み合わせが絞り込めていない初期段階では極力多様な事例に対するラベルを得てカーネルの組み合わせの絞込みを重視するのに対して, カーネルの組み合わせが最適値に近づいた後半では判定精度向上に重点を置くというものである.

そして, 異なる半径の RBF カーネルを要素カーネルに用いる場合について, 標準的ベンチマーク問題である米国郵政公社 (USPS) の手書き数字認識問題への適用を通じて, 2段階能動サポートカーネルマシンと SKM-SHIFT を組み合わせた能動学習手法では, MARGIN 戦略を組み合わせた場合に比べて, 精度面では必ずしも優位ではないが, カーネル選択すなわち最適なカーネル配分の推定においては, 一部の事例を除いては, より早く収束し易いことを示した.

第5章 配電柱腕金の再利用判定への適用

5.1 背景

電力会社では、配電資機材の有効活用に関する取り組みの一環として、撤去資機材の再利用拡大を進めている。再利用判定基準の明確化や、簡易修理、リサイクルの推進、劣化診断技術の導入などを検討しており、既存設備の有効利用によるコスト削減、リサイクルによる環境への影響の軽減を目指している。このような取り組みのひとつとして、配電線の架線に使用する腕金の再利用によるコスト削減、廃棄物の削減による環境への影響の軽減の試みがある。

現在は、腕金に発生した錆の色と錆の進行程度との分析結果に基づき、「再利用」、「廃棄」の各腕金錆の写真の判定見本を作成して、この判定見本を参考に人間が目視により判定をする形を取っている。しかしながら、工事会社の拠点ごとに集められた撤去腕金は、各拠点で異なる人間によって、判定見本に基づき判定されるため、光の当たり具合、判定する人の色に対する感じ方等に左右され、判定結果にばらつきが生じる。また、再利用の向上を目指した判定基準の微妙な変更が、現場での判定に反映されるまでに時間がかかるといった懸念がある。

一方、最近の情報技術の発展により、安価で解像度の高いデジタル画像データの収集が簡単に行えるようになり、大量の設備・機器画像データを収集できる環境を構築しやすくなりつつある。つまり、精度の高い均質な画像データの収集が簡単にできるようになってきている。

そこで、中部電力(株)配電部と(財)電力中央研究所システム技術研究所では、デジタル画像データと、サポートベクターマシンを用いた腕金錆画像に基づく再利用判定システムの開発を行った[28]。

本章では、上記研究での画像データを用いて、配電柱腕金の再利用判定に開発手法を適用した結果について述べる。

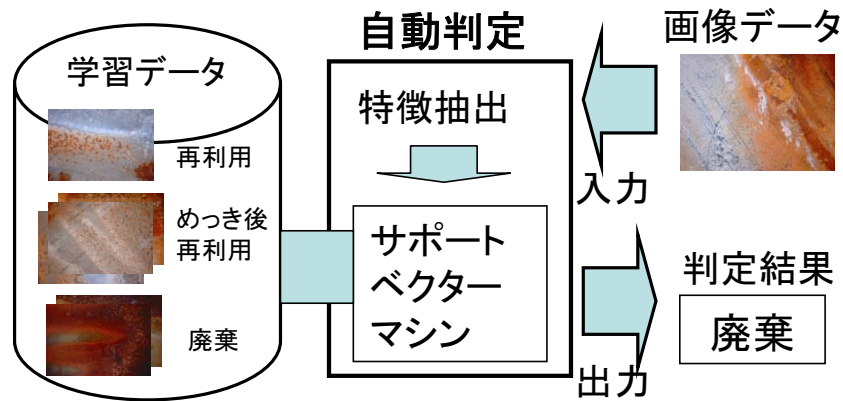


図 5.1: 腕金錆画像に基づく判定システム [28] の概念図

5.2 配電柱腕金再利用判定システムの概要

本節では,[28] で開発されたサポートベクターマシンによる配電柱腕金再利用判定システムの概要と, 検討対象となった画像特徴について述べる.

5.2.1 システムの概要

図 5.1 にサポートベクターマシンによる配電柱腕金再利用判定システムの概念図を示す. 判定システムは, 腕金に生じた錆による腐食状況を画像データとして取得, 計算機に入力して, 再利用可否を自動的に判定し, その結果を出力する.

計算機での再利用判定では, まず, 入力された画像データから基本的な画像特徴を抽出し (特徴抽出の詳細は次節参照), 抽出された画像特徴に基づいて, サポートベクターマシンにより再利用の可否を自動的に判定する.

サポートベクターマシンには, 図 5.1 の左部分に示した事前に与えられたサンプル画像データと対応するラベルの組により, 再利用の可否の判別関数を学習させておく. 判別関数は, ある画像データを入力とし, そのラ

ベル(「再利用」,「めっき後再利用」または「廃棄」)を出力とする。事前に一度だけ学習を行い判別関数を作成しておけば,以後は学習する必要はない。判別関数を作成した計算機は,新たな画像データが入力されると,作成されている判別関数に基づいて判定結果を出力する。

ただし,学習用のラベル付きデータベースが必要で,判定システムでは735枚の腕金錆画像を撮影し,専門家が各々の撮影画像を目視して再利用判定を行い,データベースを作成した。本データベース構築は非常に手間と時間を要した。

5.3 画像認識用特徴抽出

本節では,配電柱腕金再利用判定システムにおいて使用する画像特徴の決定に際して,検討された画像特徴について述べる。

再利用判定システムで取得できる腕金錆画像は 640×480 画素のフルカラー画像である。各画素は, R(Red), G(Green), B(Blue)の各色について0から255までの画素値を持つ。

得られた全画素のR,G,Bの各プレーンをそのままデータとして用いると,データの特徴数は, $640 \times 480 \times 3 = 921,600$ となる。今回用いたデータの総数は735枚であり,データ数に比べ特徴数の方がかなり大きい。一般に,パターン判別手法では,データの総数に比べて特徴数が大きすぎると,判別面の信頼性が低くなる。

そこで,[28]では,学習するデータ数に比べて特徴数が大きくなりすぎないように,画像データを加工している。

加工方法としては,粗視化と色ヒストグラム化の2種類が検討された。腕金の再利用判定は,錆の色に基づいて行われるものであり,錆の発生位置は判定に影響を与えてはならない。このため,最終的には,色ヒストグラム化が用いられた。

5.3.1 画像データの粗視化

画像データの粗視化では,学習するデータ数に比べて特徴数が大きくなりすぎないように,また,錆の発生位置の影響を小さくするように, 640×480 の画像サイズを 4×3 ,またはその2倍,5倍,8倍に小さくする。

例えば画像データサイズを 4×3 にする場合, 640×480 を図5.2のように,縦に4分割し,横に3分割し,分割したブロックの内部の画素値の平均

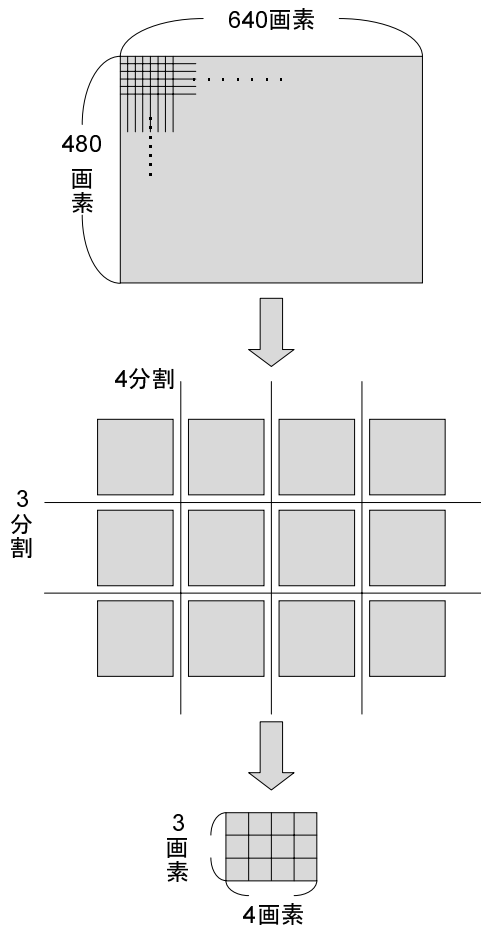


図 5.2: 画像データの粗視化

をとる. このように各ブロック内で画素を平均化することにより, 色だけではなく位置の情報も平均化される.

分析には, この粗視化した画像データの各成分の平均が 0, 分散が 1 となるように正規化したデータを用いた.

5.3.2 色ヒストグラム

色ヒストグラムにおける色表現としては, RGB と HSV の 2 つの色表示が検討された.

RGB による色ヒストグラムでは, R が各レベルにある画像数, G が各レベルにある画素数, B が各レベルにある画素数の順に頻度を並べる. デー

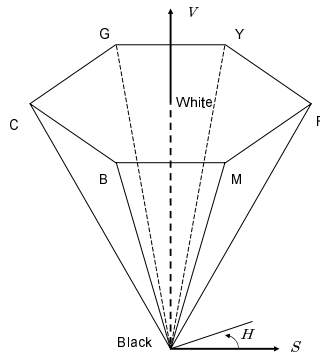


図 5.3: HSV 六角錐カラーモデル

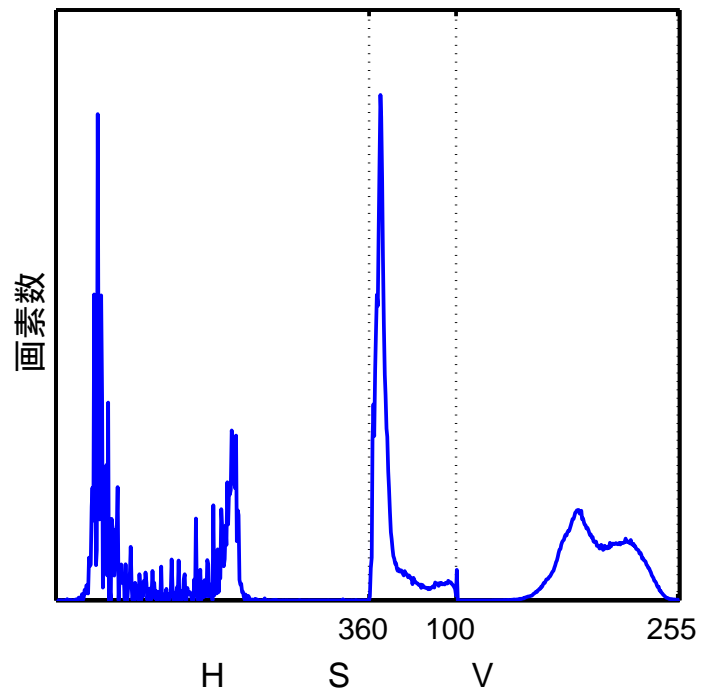


図 5.4: HSV ヒストグラム

タの特徴数は $256 \times 3 = 768$ と小さくなり、また、錆の発生位置に関する情報がなくなる。粗視化の時と同様、ヒストグラムの各成分の平均が 0、標準偏差 1 となるように正規化したデータを分析に用いた。

一方、HSV 表現によるヒストグラムでは、通常用いられる R, G, B の 3 色の代わりに、これを色相 (H : hue)、彩度 (S : saturation)、明度 (V : value) の HSV 色空間に変換することで、より人間の感覚に近い形で腕金錆画像の判別を行おうとする。

HSV 色空間は、図 5.3 に示す HSV 六角錐カラーモデルに基づいている。

明度 V 明度は色の明るさを表わすもので、黒の明度を 0 とし、白の明度を 255 とする。

彩度 S 彩度は色のあざやかさを表わすものであり、無彩色 (灰色) を彩度 0 と定義し、最もあざやかな色 (純色) (彩度 100) までを等分して表わす。六角形の中心 (灰色) から対象色までの距離と、中心から外周 (純色) までの距離の比で表わす。

色相 H 色相は赤、青、緑などの色の違いを表わすものである。六角錐カラーモデルを用いた HSV 色空間では、RGB とその補色の C (シアン)、M (マゼンダ)、Y (イエロー) を R, Y, G, C, B, M の順で正六角形の頂点におき、R からの角度で色相 H を表わす。0 から 359 の値を取る。

HSV ヒストグラムは、原画像の RGB データを HSV データに変換した上で、HSV データを H, S, V の順に各レベルにある画素数を並べたものである (図 5.4)。分析では、ヒストグラムの各成分の平均が 0、標準偏差 1 となるように正規化したデータを用いた。

データの特徴数は $360 + 100 + 256 = 716$ と小さくなり、また、錆の発生位置に関する情報が無い表現となっている。

腕金再利用判定システム [28] では、認識精度が高く、かつ、位置によらない色表現であるため、以上の各種画像特徴の中で HSV による色ヒストグラムが最終的に採用された。

5.4 実験

本節では、開発した 2 段階能動サポートカーネルマシンにより、前節で述べた各種画像特徴を用いたサポートカーネルマシンの能動学習を行う。

表 5.1: 腕金再利用判定におけるカーネル配分

画像サイズ	γ	β
3 × 4	0.01	0.34
6 × 8	0.05	0.31
3 × 4	0.05	0.20
15 × 20	0.05	0.14

より具体的には、要素カーネルとして、画像サイズ 3 × 4, 6 × 8, 15 × 20, 24 × 32 に縮小した RGB 画像, RGB 表現による色ヒストグラム, HSV 表現による色ヒストグラムの 6 種類の基本特徴に対して、各々 $\gamma = 0.01, 0.05, 0.1, 0.5, 1.0$ の RBF カーネル、合計 30 カーネルを用意した。

図 5.5 に、サンプリング戦略 RANDOM, MARGIN, SHIFT による配分 β の収束状況、および、正解率の推移を示す。

これらは、4 章における実験結果と同様の傾向を持つ。

正解率の改善に関しては、MARGIN 戦略、SKM-SHIFT 戦略、RANDOM 戦略の順となっている。MARGIN と SHIFT 共にラベル付けデータ数が 100 個付近で正解率 99% となり、両者の差はほとんどない。

サンプル重み β の配分は、SKM-SHIFT を用いる場合が最も早く最終的なカーネル配分に近づいた。

また、30 カーネル中最終的にサポートカーネルとなったのは、表 5.1 に示す 4 つのカーネルであった。明暗差への感応度が最も低く ($\gamma = 0.01$)、解像度も最も低い画像 (3 × 4) で基本的な判定を行い、これを感応度をより上げて ($\gamma = 0.05$) 解像度もやや高い画像に基づいて修正する形の合成カーネルとなっている。ただし、いずれも、粗視化画像に基づくカーネルで、ヒストグラムに基づくカーネルはサポートカーネルとならなかった。

5.5 考察

上記実験では、用意した 4 種類の粗視化画像、RGB 色ヒストグラム、HSV 色ヒストグラムの内、最終的にサポートカーネルとなったのは、粗視化画像に基づくカーネルのみで、ヒストグラムに基づくカーネルはサポートカーネルとならなかった。

第5章 配電柱腕金の再利用判定への適用

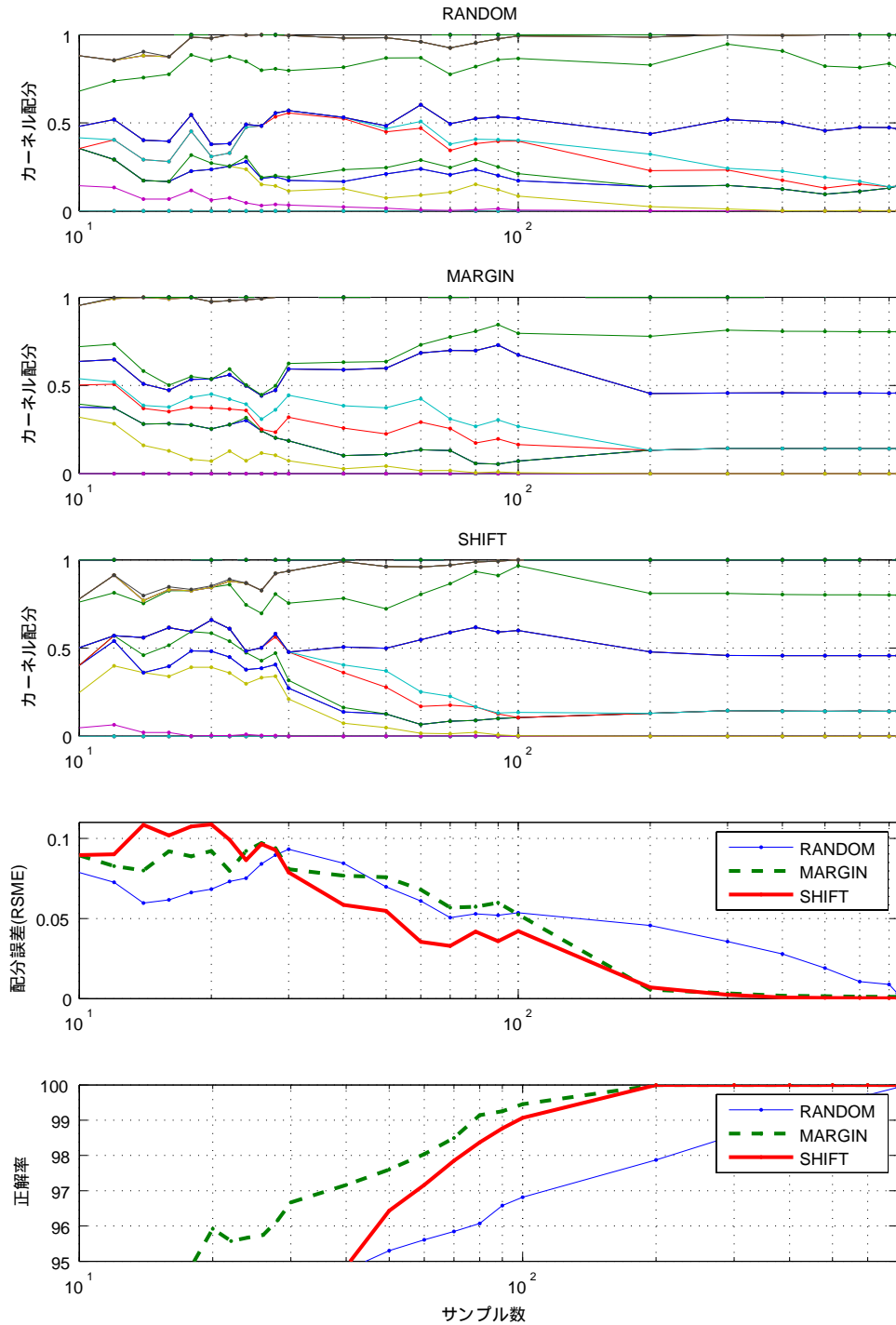


図 5.5: 腕金再利用判定におけるカーネル配分, 正解率の推移

本来、腕金再利用判定は色のみに基づいて判定されるが、実験結果は、純粹に色のみによる判定よりも、色の位置を考慮した方が判定精度が良かったことを意味する。実験に用いた画像データでは錆の位置に偏りがあったこととなる。

実際、実験に用いた画像では、腕金の配置（撮影方向）に一定の向きがあり、錆の位置に偏りがあった。そこで [28] では、最終的に、上下反転した画像を含めた学習を行ない、最終的な画像表現の判断を行っている。

SKM-SHIFT を用いた能動サポートカーネルマシンでは、100 個のサンプリング以降 β が安定することから、この比較的早い段階で学習データの偏りに気づくことが可能である。このため、上下反転データを学習データに加えるなどの対策を早期に行なえる。

これは、判別関数だけでなく、最適なカーネルの組み合わせも学習する能動サポートカーネルマシンの利点であり、提案手法には、単に、精度の高い判別関数を得るだけでなく、データの偏りの早期発見などの副次的効果があることが確認された。

5.6 本章のまとめ

本章では、具体的な設備診断問題への適用例として、配電柱につけられた腕金と呼ばれる金属部材の表面写真に基づく再利用判定システムへの適用を行なった。

本適用では、要素カーネルとして、画像中の各位置での色の類似度に関するカーネルと、画像全体での色の出現頻度に関するカーネルの 2 種類を用いた。この結果、ラベル付けデータ数が 100 個付近で正解率 99% という高い認識精度が得られた。ただし、本来、再利用判定は色の種類・頻度のみにより位置に影響されないとされているのに、画像中の各位置での色の類似度に関するカーネルのみがサポートカーネルとなった。特に、カーネル重みの収束性が高い SKM-SHIFT 戦略を用いた場合、100 サンプル程度でこの傾向が顕著となった。このため、画像データに位置に関する偏りがあることが早期に発見可能であった。

なお、この偏りは転置画像も学習データとすることで解消できるが、このようにデータの偏りを早期に発見することができるのは、判別関数のみでなく、カーネルも同時に学習する本手法の特性によるものである。

第6章 結論

本論文では、ラベルが不明の機器状態データなどラベル(ラベル)付けがなされていない大量のデータが存在する時に、システムが能動的にラベルなしデータを選択して専門家にラベル付けを行ってもらうことを繰り返すことで、高精度の診断を行うために有効な特徴やカーネルを早期に見極めると同時に、高精度の診断基準を効率的に得るための能動学習手法の検討を行った。

3章では、高い応答性を持つ形でサポートカーネルマシンを能動化した2段階能動サポートカーネルマシンを提案した。その基本的な考え方は、専門家に対してラベルを知りたい事例を迅速に提示して「ラベル」を蓄積する「能動学習段階」と、一定量のラベルの蓄積後、最適なカーネル組み合わせの選択と判定基準の構築を行う「完全最適化段階」の2段階を考え、能動学習段階においては応答性を重視したアルゴリズム設計を、完全最適化段階においては解の最適性を重視したアルゴリズム設計を行うというものである。

さらに、代表的な13個のベンチマークデータを用いた実験により、提案した能動サポートカーネル学習アルゴリズムは、同一の要素カーネル群でほとんど全てのデータベースに対して、最終的に、個別に最適化されたサポートベクトルマシンとほぼ同等の判別精度を得られる点、さらに、能動学習の最中の特性も、最良の性能を持つ要素カーネルによるサポートベクトルマシンとほぼ同等で、非常に強力な能動学習手法である点を示した。

第4章では、最適なカーネルの組み合わせを早期に見出す多カーネル能動学習のためのサンプリング手法SKM-SHIFTを提案した。その基本的な考え方は、最適なカーネルの組み合わせが絞り込めていない初期段階では極力多様な事例に対するラベルを得てカーネルの組み合わせの絞り込みを重視するのに対して、カーネルの組み合わせが最適値に近づいた後半では判定精度向上に重点を置くというものである。

第 6 章 結論

そして、異なる半径の RBF カーネルを要素カーネルとして用いる場合について、2 段階能動サポートカーネルマシンと SKM-SHIFT を組み合わせた能動学習手法の有効性を、標準的ベンチマーク問題である米国郵政公社 (USPS) の手書き数字認識問題への適用を通じて示した。

第 5 章では、具体的な設備診断問題への適用例として、配電柱腕金の表面写真に基づく再利用判定への適用を行い、カーネルの学習も同時に行なう能動サポートカーネルマシンの利点として、SKM-SHIFT により学習されたカーネルの分析により、能動学習の早い段階で、入力データの偏りを発見することが可能であることを示した。

本報告の主な成果は、代表的な多カーネル学習であるサポートカーネルマシンの枠組みを用いることで、従来は能動学習を始める前に既知とされていた判別に用いるカーネルが確定していない状態から能動学習を開始して、最適なカーネルと判別関数を同時に獲得する能動学習手法を提案したことにある。本手法は、(1) 能動学習時の高い応答性を持つ、(2) 早期に最適な特徴 (カーネルの組み合わせ) に達することができる、(3) 能動学習終了後は、最良のカーネルを用いたとほぼ同等の高い判別性能を持つ点が特徴である。

謝辞

本論文は、様々な方々のお力添えのもとに完成いたしました。

はじめに、本論文審査委員会の主査であり、主任指導教官でもありました高須淳宏教授におかれましては、本研究全般に関して多大な御指導と御鞭撻を賜りました。夕刻以降に度々研究打合せをしていただくなど様々な配慮をしていただきました。また、本研究をまとめるにあたり幾度となく激励していただくとともに我慢強くご支援くださったことなど、先生の御指導なくしては本研究は成し得ないものでありました。心より厚く御礼申し上げます。

本論文をまとめるにあたり、論文審査委員の先生方からも多大な御指導と御鞭撻を賜りました。北本朝展准教授には、研究内容の至らない点について親切的御指摘、御指導を賜りまして、深く感謝いたします。佐藤真一教授には、画像処理に関する深い知見に基づいた御指摘とともに、本質を捉えた御指導を賜りましたこと、深く感謝いたします。佐藤健教授には、非常に幅広く深い見識に基づいて、本研究遂行に関する激励を戴きましたことを深く感謝いたします。山田誠二教授には、機械学習、インタラクションに関する深い見識に基づいたご指摘と共に、実験方法の示唆など、研究論文をまとめるにあたり鋭く重要な御指摘、御指導をいただきましたことを深く感謝いたします。皆様の御指導、御鞭撻により本論文をまとめ、より質の高いものとすることができました。重ねて御礼申し上げます。

また、5章で述べた配電柱腕金の再利用判定システムの開発を進め、開発手法適用の機会を与えてくださった、中部電力配電部の方々、および、(財)電力中央研究所の小野田崇上席研究員、村田博士主任研究員、山名美智子元特別契約研究員(現(独)産業技術総合研究所 生命情報科学研究センター)に感謝いたします。

さらに、総合研究大学院大学での研究にご理解をいただきました(財)電力中央研究所システム技術研究所の方々に感謝いたします。谷口治人前所長、栗原郁夫所長、松井正一領域リーダーに感謝いたします。また、同僚の方々、特に、小野田上席研究員には機械学習の専門家として、また、中島

慶人上席研究員には博士課程の先輩として数々の助言をいただいたことに感謝いたします。

最後に、これまでの研究と本論文の作成は、いずれも家族の支援の上で成し遂げた成果であり、心の支えとなり元気づけてくれた家族に感謝します。

参考文献

- [1] ANGLUIN, D. Queries and concept learning, *Journal of Machine Learning*, **2** (1988), 319–342.
- [2] AUER, P., CESA-BIANCHI, N., FREUND, Y. and SCHAPIRE, R. E. The Nonstochastic Multiarmed Bandit Problem, *SIAM Journal on Computing*, **32**, 1 (2003), 48–77.
- [3] BARAM, Y., EL-YANIV, R. and LUZ, K. Online Choice of Active Learning Algorithms, *Journal of Machine Learning Research*, **5** (2004), 255–291.
- [4] CAMPBELL, C., CRISTIANINI, N. and SMOLA, A. J. Query Learning with Large Margin Classifiers, ICML '00: Proceedings of the 17th International Conference on Machine Learning, San Francisco, CA, USA (2000), Morgan Kaufmann Publishers Inc.
- [5] CHANG, C.-C. and LIN, C.-J. *LIBSVM: a library for support vector machines* (2001), Software at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [6] CHAPELLE, O., VAPNIK, V., BOUSQUET, O. and MUKHERJEE, S. Choosing Multiple Parameters for Support Vector Machines, *Machine Learning*, **46**, 1–3 (2002), 131–159.
- [7] CHEN, H. and KARGER, D. R. Less is more: probabilistic models for retrieving fewer relevant documents, SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA (2006), ACM.
- [8] CHUNG, K.-M., KAO, W.-C., SUN, C.-L., WANG, L.-L. and LIN, C.-J. Radius margin bounds for support vector machines with the RBF kernel, *Neural Computation*, **15**, 11 (2003), 2643–2681.

-
- [9] COHN, D. A., GHAHRAMANI, Z. and JORDAN, M. I. Active Learning with Statistical Models, *Journal of Artificial Intelligence Research*, **4** (1996), 129–145.
- [10] DAVIS, D. T. and HWANG, J. Attentional focus training by boundary region data selection, Proceedings of the 1992 International Joint Conference on Neural Networks (June 1992).
- [11] FAN, R.-E., CHEN, P.-H. and LIN, C.-J. Working Set Selection Using Second Order Information for Training Support Vector Machines, *Journal of Machine Learning Research*, **6** (2005), 1889–1918.
- [12] FORREST, J., NUEZ, DE LA D. and LOUGEE-HEIMER, R. *CLP User Guide* (2004), Software available at <http://www.coin-or.org/Clp/index.html>.
- [13] FRANCIS R. BACH, M. I. J., GERT R. G. LANCKRIET Multiple Kernel Learning, Conic Duality, and the SMO Algorithm, Proceedings of the 21st International Conference on Machine Learning (2004).
- [14] FREUND, Y., SEUNG, H. S., SHAMIR, E. and TISHBY, N. Information, Prediction, and Query by Committee, *Advances in Neural Information Processing Systems* **5** (1992).
- [15] LANCKRIET, G. R. G., CRISTIANINI, N., BARTLETT, P., GHAOUI, L. E. and JORDAN, M. I. Learning the Kernel Matrix with Semidefinite Programming, *Journal of Machine Learning Research*, **5** (2004), 27–72.
- [16] LEWIS, D. D. and GALE, W. A. A sequential algorithm for training text classifiers, Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval (1994).
- [17] LINDENBAUM, M., MARKOVITCH, S. and RUSAKOV, D. Selective Sampling for Nearest Neighbor Classifiers, Proceedings of the 16th National Conference on Artificial Intelligence (1999).

-
- [18] MACKAY, D. J. C. Information-Based Objective Functions for Active Data Selection, *Neural Computation*, **4**, 4 (1992), 590–604.
- [19] OSUGI, T., KUN, D. and SCOTT, S. Balancing Exploration and Exploitation: A New Algorithm for Active Machine Learning, ICDM '05: Proceedings of the 5th IEEE International Conference on Data Mining (2005).
- [20] PLATT, J. C. Fast training of support vector machines using sequential minimal optimization, *Advances in kernel methods: support vector learning*, MIT Press (1999).
- [21] SEUNG, H. S., OPPER, M. and SOMPOLINSKY, H. Query by committee, COLT '92: Proceedings of the 5th annual workshop on Computational learning theory (1992).
- [22] SONNENBURG, S., RAETSCH, G., SCHAEFER, C. and SCHÖLKOPF, B. Large Scale Multiple Kernel Learning, *Journal of Machine Learning Research*, **7** (2006), 1531–1565.
- [23] SONNENBURG, S., RATSCH, G. and SCHAFER, C. A General and Efficient Multiple Kernel Learning Algorithm, *Advances in Neural Information Processing Systems 18* (eds. Weiss, Y., Schölkopf, B. and Platt, J. C.), Cambridge, MA (2006).
- [24] TAX, D. M. J. and DUIN, R. P. W. Support vector domain description, *Pattern Recogn. Lett.*, **20**, 11-13 (1999), 1191–1199.
- [25] TONG, S. and KOLLER, D. Support vector machine active learning with applications to text classification, *Journal of Machine Learning Research*, **2** (2002), 45–66.
- [26] VAPNIK, V. *Statistical Learning Theory*, John Wiley and Sons Inc. (1998).
- [27] WESTON, J., MUKHERJEE, S., CHAPELLE, O., PONTIL, M., POGGIO, T. and VAPNIK, V. Feature Selection for SVMs, *Advances in Neural Information Processing Systems 13* (2001).

-
- [28] 山名美智子, 村田博士, 小野田崇, 大橋徹, 加藤誠二パターン識別手法を用いた錆画像による腕金再利用判定法の性能評価, 電気学会論文誌 C, **125**, 7 (2005), 1049–1054.