数理解析研究所講究録
第1829巻 2013年 23-38

機械学習における非凸最適化問題に対するパラメトリック計画法を用いたアプローチ

名古屋工業大学・工学研究科・竹内　一郎
Ichiro Takeuchi
Department of Engineering,
Nagoya Institute of Technology
名古屋工業大学・工学研究科・小川　晃平
Kohei Ogawa
Department of Engineering,
Nagoya Institute of Technology
東京工業大学・大学院情報理工学研究科・杉山　将
Masashi Sugiyama
Department of Computer Science,
Tokyo Institute of Technology

概要

本研究ではさまざまな機械学習問題に共通して現れる非凸最適化問題を考察し, パラメトリック計画法を用いた最適化アルゴリズムを提案する. 提案アルゴリズムは問題に含まれるハイパーパラメータを連続的に変化させたときの局所最適解のパスを計算することができる. これらの問題は局所最適解のパスが不連続点を持つという特徴があるが, 局所最適解の性質を詳しく調べることによりこの問題に対処する. 本稿では半教師あり学習を例としてこのクラスの問題の性質を議論し, 数値実験により提案アルゴリズムの有効性を検証する.

1　はじめに

本稿では, 半教師あり学習やロバスト学習などのさまざまな機械学習問題に共通して現れる非凸最適化問題を考察する. これらの問題の学習アルゴリズムは,

$$\text{minimize} \quad (\text{convex function}) + \theta\ (\text{non-convex function}) \tag{1}$$

という形式の最適化問題として定式化される. ここで, $\theta \geq 0$ はモデル選択により決定されるハイパーパラメータである. 例えば, 半教師ありSVM [1] と呼ばれるアルゴリズムは, ラベルありデータに関する損失関数が凸, ラベルなしデータに関する損失関数が非凸であ

るため, (1) の形式となる. 半教師ありSVMの場合, ハイパーパラメータ θ はラベルなしインスタンスの重要度を決める役割を担い, 汎化性能が高くなるような値がモデル選択により決定される (詳しくは2節を参照).

これらの非凸最適化問題においては, よい局所最適解を求めることが現実的な目標となる. (1) の形式で表される問題では, $\theta = 0$ の場合に凸最適化問題となる. このため, まず, $\theta = 0$ の最適解を求め, θ を徐々に増やしながら局所最適解の系列を求めていくアプローチ [2] が有効である. このようなアプローチはアニーリング法 [2] の一種と捉えることができ, θ はアニーリングパラメータと呼ばれる.

本研究では, このクラスの問題に対し, パラメトリック計画法 [3] を用いた新しいアルゴリズムを提案する. パラメトリック計画法とは, パラメータ表現された最適化問題において最適解のパスを計算する方法である. 提案アルゴリズムを用いると, θ を0から連続的に増やしたときの局所最適解のパスを計算することができる. このプロセスは凸最適化問題 ($\theta = 0$) を非凸最適化問題へ連続的に変形しながら解の変化を追跡していくものであり, 無限小のステップ幅を持つアニーリング法と解釈できる.

機械学習では, 正則化パス追跡 [4,5] の文脈で凸パラメトリック計画法が使われている. 提案アルゴリズムはこれを非凸最適化問題に適用するために拡張したものであるが, 前者は大域最適解のパスを, 後者は局所最適解のパスを計算するという点で違いがある. 前者ではどんな最適化アルゴリズムを用いても同一の解が得られるが, 後者では一般に異なる局所最適解が得られる.

アニーリング法では, アニーリングパラメータのステップ幅を小さくするほどよい解を見つける可能性が高いと言われている [7]. ステップ幅を大きくしてしまうと離れた局所最適解へ移動する可能性が高まるためである. 提案アルゴリズムは無限小ステップ幅のアニーリング法であるため, 従来の最適化アルゴリズムよりもよい局所最適解を得られると期待できる. また, ハイパーパラメータ θ の刻み幅を小さくしてモデル選択を行えば, より正確なモデル選択を行えるという利点も持つ. 従来法ではパラメータのステップ幅と計算コストはトレードオフの関係にあったが, 提案法は無限小ステップ幅のアニーリング法を行うためこのトレードオフを解消することができる.

局所最適解パスを求めるアルゴリズムの構築に際して, 局所最適性条件 (局所最適解の必要十分条件) を導出する必要がある. この条件を詳しく調べると, このクラスの問題では, 局所最適解のパスが有限個の点で不連続となることが証明される. 本研究の技術的な貢献のひとつは, パスがどのような状況で不連続となるかを明らかにし, そのような場合に局所最適解を見つける方法を構築することである.

本稿では, 非凸最適化問題の例として, 主に, 半教師ありSVMを考察する. 第2節で半教師ありSVMとロバストSVM [6] を定式化し, 両者が同じような形式の非凸最適化問題として定式化されることを示す. 第3節では, 半教師ありSVMの局所最適性条件を求め, 局所最適解の性質を議論する. 第4節では, 局所最適性条件に基づいて局所最適解パスを計算するアルゴリズムを構築する. 第5節では, ベンチマークデータに対する数値実験を行い, 最適化性能, 汎化性能, 計算コストの観点から提案アルゴリズムの有用性を議論する.

本稿では以下の表記を利用する. $\mathbb{R}$ を実数集合とし, n 次元縦ベクトルを $\boldsymbol{v} \in \mathbb{R}^n$, $n \times m$ 行列を $M \in \mathbb{R}^{n \times m}$ などと表す. また, n までの自然数の集合を $\mathbb{N}_n := \{1, 2, \ldots, n\}$ と表す. $\mathbb{N}_n$ の部分集合を $\mathcal{A}$ としたとき, $\boldsymbol{v}_{\mathcal{A}}$ は $\boldsymbol{v} \in \mathbb{R}^n$ のうち $\mathcal{A}$ に含まれる要素のみを持つ部分

ベクトルを表す. 同様に, $M_{\mathcal{AB}}$ は, $\mathcal{A}$ に含まれる行と $\mathcal{B}$ に含まれる列を持つ M の部分行列を表す.

2 機械学習における非凸最適化問題

本節では, まず, 準備としてサポートベクトルマシン (SVM) を解説する. 続いて, 本稿で主に考察する半教師あり SVM を非凸最適化問題として定式化する. さらに, ロバスト SVM についても簡単に紹介し, 半教師あり SVM とほぼ同一の構造を持った非凸最適化問題となっていることを示す.

2.1 サポートベクトルマシン (SVM)

2 クラス分類問題を考え, 学習データを $\{(\boldsymbol{x}_i, y_i)\}_{i\in\mathbb{N}_n}$ とする. ここで, $\boldsymbol{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ は領域 $\mathcal{X}$ からの入力ベクトル, $y_i \in \{\pm 1\}$ は出力ラベルを表す. 識別関数を

$$f(\boldsymbol{x}) = w_0 + \boldsymbol{w}^\top \phi(\boldsymbol{x}) \tag{2}$$

と表す. ここで, $\phi : \mathcal{X} \to \mathcal{F}$ はカーネル関数 $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ によって定義される特徴写像を表し, 上付きの $^\top$ は行列やベクトルの転置を表す.

SVM は識別関数 (2) を学習するアルゴリズムであり, 以下のような凸最適化問題として定式化される:

$$\min_f \frac{1}{2}||\boldsymbol{w}||_2^2 + C\sum_{i=1}^{n}[1 - y_i f(\boldsymbol{x}_i)]_+. \tag{3}$$

ここで, $C > 0$ は正則化項と損失項のバランスと制御するための正則化パラメータ, $[z]_+ := \max(0, z)$ はヒンジ損失と呼ばれる損失関数である.

最適化問題 (3) の双対問題は

$$\max_{\{\alpha_i\}_{i=1}^n} \quad -\frac{1}{2}\sum_{i\in\mathbb{N}_n}\sum_{j\in\mathbb{N}_n}\alpha_i\alpha_j Q_{ij} + \sum_{i\in\mathbb{N}_n}\alpha_i \quad \text{subject to} \quad \sum_{i\in\mathbb{N}_n} y_i\alpha_i = 0,\ 0 \le \alpha_i \le C,\ i \in \mathbb{N}_n$$

と表される. ここで, $Q_{ij} = y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j), (i,j) \in \mathbb{N}_n \times \mathbb{N}_n$, である. 双対変数を用いると (2) は

$$f(\boldsymbol{x}) = w_0 + \sum_{i\in\mathbb{N}_n}\alpha_i y_i k(\boldsymbol{x}, \boldsymbol{x}_i) \tag{4}$$

と表される.

2.2 半教師あり SVM(S^{3}VM: semi-supervised SVM)

半教師あり学習ではラベルありデータに加えてラベルなしデータを学習に利用する. 前者のデータのインデックスの集合を $\mathcal{L}$, 後者のものを $\mathcal{U}$ とすると, ラベルありインスタンス $\{(\boldsymbol{x}_i, y_i)\}_{i\in\mathcal{L}}$ とラベルなしインスタンス $\{\boldsymbol{x}_i\}_{i\in\mathcal{U}}$ が学習データとして与えられることになる. ラベルなしインスタンスのラベルは未知であるので, 予測ラベル $\hat{y}_i \in \{\pm 1\}, i \in \mathcal{U}$ も同時に学習する必要がある. 以後, 半教師あり学習について述べる際には, $\ell = |\mathcal{L}|$, および, $u = |\mathcal{U}|$ とし, $\boldsymbol{x} \in \mathbb{R}^{\ell+u}$ の初めの ℓ 個の要素がラベルありインスタンス, 残りの u 個の要素がラベルなしインスタンスであるとする. また, $\boldsymbol{y} = [y_1, \ldots, y_\ell] \in \mathbb{R}^\ell$ をラベルありインスタンスのラベル, $\hat{\boldsymbol{y}} = [\hat{y}_{\ell+1}, \ldots, \hat{y}_{\ell+u}] \in \mathbb{R}^u$ をラベルなしインスタンスの予測ラベルとする. さらに, $\boldsymbol{y}$ と $\hat{\boldsymbol{y}}$ を並べたベクトルを $\tilde{\boldsymbol{y}} \in \{\pm 1\}^{\ell+u}$ とする.

SVM を半教師あり学習の枠組へ拡張したものとして半教師あり SVM(S^{3}VM: semi-supervised SVM) と呼ばれる手法が提案されている [1]. S^{3}VM の学習は以下の最適化問題として定式化される [7]:

$$\min_{f, \hat{\boldsymbol{y}}\in\{\pm 1\}^u} J(f, \hat{\boldsymbol{y}}) \equiv \frac{1}{2}||\boldsymbol{w}||_2^2 + C\sum_{i\in\mathcal{L}}[1 - y_i f(x_i)]_+ + \theta C\sum_{i\in\mathcal{U}}[1 - \hat{y}_i f(\boldsymbol{x}_i)]_+ \tag{5}$$

ここで, $\theta \in [0, 1]$ はラベルなしインスタンスの影響を制御するパラメータである. $\theta = 0$ のときはラベルなしインスタンスを無視するため, ラベルありインスタンスのみを用いた通常の SVM と等価なものとなる. 一方, $\theta = 1$ のときはラベルありインスタンスとラベルなしインスタンスが同程度の影響を持つことを意味する. 不確かさを持つラベルなしインスタンスの影響力はラベルありインスタンスよりも小さいか等しくあるべきなので, θ は $[0, 1]$ の範囲をとる.

分類境界 f が与えられているとき, 予測ラベルは

$$\hat{y}_i f(\boldsymbol{x}_i) \geq 0, \ \forall i \in \mathcal{U} \tag{6}$$

を満たすように決めることが妥当である. このため, S^{3}VM の学習は最適な $\hat{\boldsymbol{y}} \in \{\pm 1\}^u$ を求める組み合わせ最適化問題

$$\min_{\hat{\boldsymbol{y}}\in\{\pm 1\}^u} \left\{\min_f J(f, \hat{\boldsymbol{y}}) \ \text{ subject to } \ (6)\right\} \tag{7}$$

とみなすこともできる.

また, 制約 (6) のもとでは (5) の右辺第 3 項を $C\theta\sum_{i=1}^n[1 - |f(\boldsymbol{x}_i)|]_+$ と書けることに留意すると, S^{3}VM の学習は

$$\min_f \ \frac{1}{2}||\boldsymbol{w}||_2^2 + C\sum_{i\in\mathcal{L}}[1 - y_i f(\boldsymbol{x}_i)]_+ + C\theta\sum_{i\in\mathcal{U}}[1 - |f(\boldsymbol{x}_i)|]_+ \tag{8}$$

と書くこともできる. この目的関数の第 3 項は図 1(a) のような非凸関数であるので, S^{3}VM の学習は非凸最適化問題とみなすこともできる.

最適化問題 (5) はすべてのラベルなしインスタンスを同一のクラスへ分類する, すなわち, $f(\boldsymbol{x}_i) < 0, \forall i \in \mathcal{U}$, もしくは, $f(\boldsymbol{x}_i) > 0, \forall i \in \mathcal{U}$ のときに最適となる場合がある. この

ような不適当な解を回避するため, 通常, 予測ラベルのバランスに関する制約が導入される. 文献 [1] ではラベルなしインスタンスのクラス比率がラベルありインスタンスと等しいという制限が導入されている. その後, 文献 [8] ではこの制限を緩和して

$$\frac{1}{u}\sum_{i\in\mathcal{U}} f(\boldsymbol{x}_i) = 2r - 1 \tag{9}$$

を制約として加えることが提案されている. ここで, r はラベルありインスタンスのクラス比率で, $r = \frac{1}{\ell}\sum_{i\in\mathcal{L}} \max(0, y_i)$ と計算される. この制約はバランス制約と呼ばれ, (9) は線形制約であるため最適化問題へそのまま導入することができる. ここで, 全ラベルなしインスタンスを $\sum_{i\in\mathcal{U}} \boldsymbol{x}_i = \boldsymbol{0}$ と中心化すると,

$$\frac{1}{u}\sum_{i\in\mathcal{U}}(w_0 + \boldsymbol{w}^\top \boldsymbol{x}_i) = w_0 = 2r - 1, \tag{10}$$

となり, 識別関数の w_0 を固定することによってバランス制約を表現可能となる. 本稿でも S^3VM の学習において制約 (9) を課すこととする. したがって, (2) や (4) の識別関数 f の定数項 w_0 は (10) によって固定され, 最適化の対象から除外することができる.

2.3 ロバスト SVM (R-SVM: robust SVM)

ロバスト学習とは外れ値の影響を軽減するためのアプローチを指す. ロバストな SVM の学習法としてロバスト SVM [6] が提案されている. ロバスト学習を定式化するため, 各学習インスタンスが外れ値であるか否かを表すフラグ $o_i \in \{0,1\}, i \in \mathbb{N}_n$ を定義し, $o_i = 1$ であればインスタンス i が外れ値であることを表すものとする. ロバスト SVM 学習の目的関数は

$$\min_{f,\boldsymbol{o}} J(f,\boldsymbol{o}) \equiv \frac{1}{2}||\boldsymbol{w}||_2^2 + C\sum_{i:o_i=0}[1 - y_i f(\boldsymbol{x}_i)]_+ + (1-\theta)C\sum_{i:o_i=1}[1 - y_i f(\boldsymbol{x}_i)]_+$$

と表される. ここで, $\theta \in [0,1]$ は外れ値の影響を制御するパラメータである. $\theta = 0$ においては外れ値を正常値と同様に扱い, 通常の SVM と等価になる. 一方, $\theta = 1$ は外れ値の影響を完全に無視することを意味する.

分類境界 f が与えられているとき, マージン $y_i f(\boldsymbol{x}_i)$ の小さなインスタンスを外れ値とみなすことが妥当である. マージンがある定数 $s < 1$ よりも小さなものを外れ値と定義する, すなわち,

$$y_i f(x_i) \le s \;\Leftrightarrow\; o_i = 1, \tag{11}$$

とすると, ロバスト SVM 学習は最適な $\boldsymbol{o} \in \{0,1\}^n$ を求める組み合わせ最適化問題

$$\min_{\boldsymbol{o}\in\{0,1\}^n}\left\{\min_f J(f,\boldsymbol{o}) \text{ subject to } (11)\right\}$$

として表される. また, 制約 (11) のもとでは, ロバスト SVM の学習は,

$$\min_f \frac{1}{2}||\boldsymbol{w}||_2^2 + C\sum_{i=1}^{n}\mathrm{loss}_{\mathrm{robust}}(y_i f(\boldsymbol{x}_i)) \text{ where } \mathrm{loss}_{\mathrm{robust}}(z) = \begin{cases} [1-z]_+, & \text{if } z \ge s, \\ 1-s, & \text{if } z < s \end{cases}$$

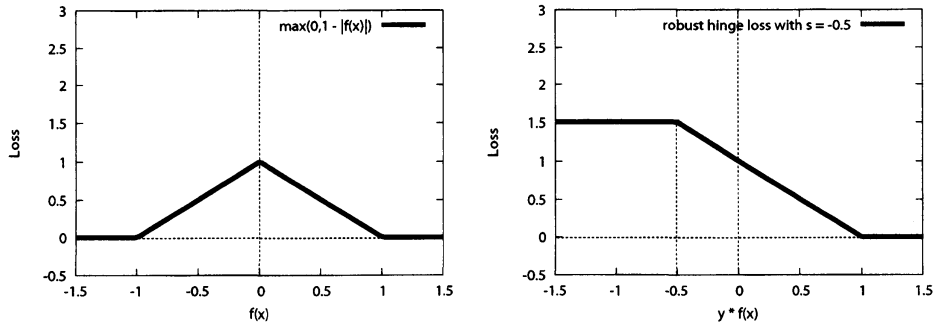


図 1: (左) 半教師あり SVM におけるラベルなしデータのための非凸損失関数, (右) ロバスト SVM のための非凸損失関数

と書くこともできる. 損失関数 $\mathrm{loss}_{\mathrm{robust}}$ は図 1(b) のような非凸関数であるので, ロバスト SVM の学習は非凸最適化問題とみなすこともできる.

3 S^3VM の局所最適解の性質

前節より半教師あり SVM(S^3VM) とロバスト SVM(R-SVM) の学習は同一の構造を持つことがわかる. どちらの問題においても, $\theta = 0$ とすると通常の SVM に一致し, θ を増やすにつれ非凸項の影響が大きくなる. 本研究の目的は, θ を 0 から 1 へ連続的に動かしたときに局所最適解のパスを計算することである. 本節以降では, S^3VM を例に話を進めるが R-SVM についても同様の議論が可能である.

本節では, 局所最適解パスを計算する準備として, S^3VM の局所最適解の必要十分条件を導出する. 凸計画問題では, よく知られた KKT(Karush-Kuhn-Tucker) 条件 [9] が最適解の必要十分条件となる. 一方, 非凸最適化問題では, KKT 条件が局所最適解の (必要条件であるが) 十分条件であるとは限らない. 本節で導出する必要十分条件は, 次節で述べるアルゴリズムの導出に重要な役割を果たす.

3.1 条件付最適解

S^3VM を組み合わせ最適化問題と解釈した定式化 (7) において, 内側の最適化問題は以下の定義のように凸計画問題となる:

定義 1 ラベルなしインスタンスの予測ラベル $\hat{\boldsymbol{y}} \in \{\pm 1\}^u$ が与えられたとき, 次の凸最適化問題:

$$f^*_{\hat{\boldsymbol{y}}} := \arg\min_f J(f, \hat{\boldsymbol{y}}) \text{ subject to (6)} \tag{12}$$

を $\hat{\boldsymbol{y}}$ が与えられたもとで条件付最適化問題と呼び, その最適解 $f^*_{\hat{\boldsymbol{y}}}$ を条件付最適解と呼ぶ. また, $\hat{\boldsymbol{y}}$ が与えられたもとでの条件 *(6)* を満たす f の集合は凸ポリトープ *(convex*

*polytope)*であり,

$$\mathrm{pol}_{\hat{\boldsymbol{y}}} := \{f | \hat{y}_i f(\boldsymbol{x}_i) \geq 0, \forall i \in \mathcal{U}\}$$

と表す.

予測ラベル $\hat{\boldsymbol{y}}$ は 2^u 通りあるので, それぞれに対応する条件付最適解が存在しうる[1]. これらの条件付最適解と S^{3}VM の局所最適解とは以下のような関係がある:

命題 2 問題 *(5)* のすべての局所最適解 f は, 条件 *(6)* を満たす予測ラベル $\hat{\boldsymbol{y}}$ のもとでの条件付最適解である.

(証明) ある局所最適解 f において条件 (6) を満たすように $\hat{\boldsymbol{y}}$ を決め, f が $\hat{\boldsymbol{y}}$ のもとでの条件付最適解でないとする. $\hat{\boldsymbol{y}}$ を固定した問題 (12) は凸計画問題であるので, $\mathrm{pol}_{\hat{\boldsymbol{y}}}$ 上に条件付最適解 $f^*_{\hat{\boldsymbol{y}}}$ 以外の局所最適解は存在せず, f が局所最適であることに矛盾する. したがって, すべての局所最適解が $\hat{\boldsymbol{y}}$ のもとでの条件付最適解である. (証明終)

一方, ある条件付最適解 $f^*_{\hat{\boldsymbol{y}}}$ は制約条件 (6) のもとで最適であるだけで, $\hat{\boldsymbol{y}}$ を変更すると近傍によりよい解が存在する場合がある. このため, すべての条件付最適解が局所最適解であるとは言えない.

条件付最適化問題の双対問題を考えると, 条件付最適解を

$$f(\boldsymbol{x}) = w_0 + \sum_{i \in \mathcal{L}} \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i) + \sum_{i \in \mathcal{U}} \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i) \tag{13}$$

の形式で表すことができる. 命題 2 より, S^{3}VM の任意の局所最適解も (13) の形式で表されることになる.

条件付最適化問題は凸計画問題であるため最適解の必要十分条件は以下のように表すことができる:

補題 3 ある予測ラベル $\hat{\boldsymbol{y}}$ が与えられたもとでの条件付最適解の必要十分条件は

$$y_i f(\boldsymbol{x}_i) > 1, i \in \mathcal{L} \;\Rightarrow\; y_i \alpha_i = 0, \tag{14a}$$
$$y_i f(\boldsymbol{x}_i) = 1, i \in \mathcal{L} \;\Rightarrow\; y_i \alpha_i \in [0, C], \tag{14b}$$
$$y_i f(\boldsymbol{x}_i) < 1, i \in \mathcal{L} \;\Rightarrow\; y_i \alpha_i = y_i C, \tag{14c}$$
$$\hat{y}_i f(\boldsymbol{x}_i) > 1, i \in \mathcal{U} \;\Rightarrow\; \hat{y}_i \alpha_i = 0, \tag{14d}$$
$$\hat{y}_i f(\boldsymbol{x}_i) = 1, i \in \mathcal{U} \;\Rightarrow\; \hat{y}_i \alpha_i \in [0, \theta C], \tag{14e}$$
$$0 < \hat{y}_i f(\boldsymbol{x}_i) < 1, i \in \mathcal{U} \;\Rightarrow\; \hat{y}_i \alpha_i = \theta C, \tag{14f}$$

$$\hat{y}_i f(\boldsymbol{x}_i) \geq 0, \forall i \in \mathcal{U}, \tag{15}$$

および,

$$\hat{y}_i f(\boldsymbol{x}_i) = 0 \;\Rightarrow\; \hat{y}_i \alpha_i \geq \theta C, i \in \mathcal{U} \tag{16}$$

である.

補題 3 は標準的な凸最適化理論 [9] を用いれば容易に導出が可能なため, 証明は省略する.

[1] 一部の $\hat{\boldsymbol{y}}$ においては制約条件 (6) を満たす実行可能解がないことがあり, その場合は条件付最適解が存在しない. また, 条件付最適化問題 (12) が狭義の凸計画問題でなければ, 一つの $\hat{\boldsymbol{y}}$ に対する条件付最適解が複数存在する.

3.2 局所最適解の必要十分条件

前小節にて, 命題 2 の逆は成り立たない, すなわち, 条件付最適解であっても局所最適解であるとは限らないことを述べた. どのような場合にそのような状況となるだろうか. 条件付最適解 $f^*_{\hat{\boldsymbol{y}}}$ が $\mathrm{pol}_{\hat{\boldsymbol{y}}}$ の境界上にある場合を考えてみる. このとき, 隣の凸ポリトープ側の近傍によりよい解が存在する可能性があるため, 条件付最適解 $f^*_{\hat{\boldsymbol{y}}}$ が局所最適解であると断言することはできない. 次の定理は隣の凸ポリトープ側に必ずよりよい解が存在することを保証するものである:

補題 4 ある予測ラベル $\hat{\boldsymbol{y}}$ が与えられたもとでの条件付最適解 $f^*_{\hat{\boldsymbol{y}}}$ が $\mathrm{pol}_{\hat{\boldsymbol{y}}}$ の境界上にある, すなわち, 集合

$$\mathcal{S} := \{i \in \mathcal{U} | \hat{y}_i f(\boldsymbol{x}_i) = 0\} \tag{17}$$

が空でないとする. ここで, 新たな予測ラベル $\hat{\boldsymbol{y}}'$ を

$$\hat{y}'_i := \begin{cases} -\hat{y}_i, & i \in \mathcal{S}, \\ \hat{y}_i, & i \notin \mathcal{S} \end{cases} \tag{18}$$

と定義する. このとき, 任意の $\theta \in (0, 1]$ において, $f^*_{\hat{\boldsymbol{y}}}$ は, $\hat{\boldsymbol{y}}'$ が与えられたもとでの条件付最適化問題に関して実行可能解であるが条件付最適解ではない. すなわち, $\mathrm{pol}_{\hat{\boldsymbol{y}}'}$ 上の条件付最適解 $f^*_{\hat{\boldsymbol{y}}'}$ は $\mathrm{pol}_{\hat{\boldsymbol{y}}}$ 上の条件付最適解 $f^*_{\hat{\boldsymbol{y}}}$ よりも厳密によい S^3VM の解であり,

$$J(f^*_{\hat{\boldsymbol{y}}'}, \hat{\boldsymbol{y}}') < J(f^*_{\hat{\boldsymbol{y}}}, \hat{\boldsymbol{y}}) \tag{19}$$

が成り立つ.

(証明) 本補題を証明するため, $\hat{\boldsymbol{y}}$, および, $\hat{\boldsymbol{y}}'$ が与えられたもとでの 2 つの条件付最適化問題を考え, 両者の最適性条件を比較する.

前者の条件付最適化問題は $\mathrm{pol}_{\hat{\boldsymbol{y}}}$ 上で定義される凸計画問題である. 補題 3 より, 条件付最適解 $f^*_{\hat{\boldsymbol{y}}}$ は (14) と (16) を満たしている. (18) によりラベルを反転すると, 最適性条件 (16) は, $y'_i = -y_i, i \in \mathcal{S}$, であるので,

$$\hat{y}'_i f^*_{\hat{\boldsymbol{y}}}(\boldsymbol{x}_i) = 0, i \in \mathcal{U} \;\Rightarrow\; \hat{y}'_i \alpha_i \leq -\theta C \tag{20}$$

と書ける.

続いて, $\mathrm{pol}_{\hat{\boldsymbol{y}}'}$ 上で定義される後者の条件付最適化問題を考える. 前者の条件付最適解 $f^*_{\hat{\boldsymbol{y}}}$ は $\mathrm{pol}_{\hat{\boldsymbol{y}}}$ と $\mathrm{pol}_{\hat{\boldsymbol{y}}'}$ の境界上にあるので, 後者の凸計画問題の実行可能解でもある. $f^*_{\hat{\boldsymbol{y}}}$ が後者の凸計画問題の最適解であるためには補題 3 を満たす必要があるが, (20) より, $\theta \in (0, 1]$ のとき,

$$\hat{y}'_i f^*_{\hat{\boldsymbol{y}}}(\boldsymbol{x}_i) = 0, i \in \mathcal{U} \;\Rightarrow\; \hat{y}'_i \alpha_i \geq \theta C \tag{21}$$

を満たすことができない. すなわち, $f^*_{\hat{\boldsymbol{y}}}$ は後者の凸計画問題の実行可能解であるが最適解ではないことになり, (19) が成り立つ. (証明終)

補題 4 より, 条件付最適解 $f^*_{\hat{\boldsymbol{y}}}$ が凸ポリトープ $\mathrm{pol}_{\hat{\boldsymbol{y}}}$ の境界上にある場合, 予測ラベルを (18) により反転した "隣" の凸ポリトープ上に必ず厳密によい解が存在することがわかった. 条件付最適解が凸ポリトープの内点にあるか境界にあるかにより次の補題が成り立つ:

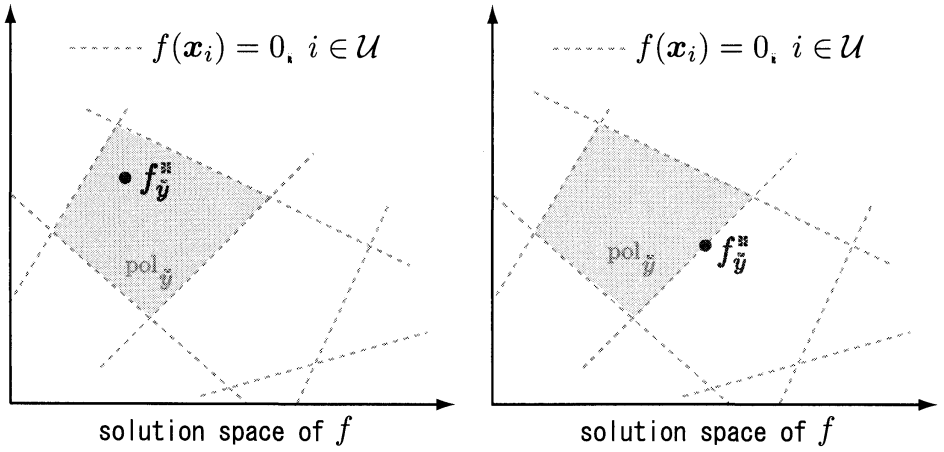


図 2: 条件付最適解 $f^*_{\hat{y}}$ と局所最適解の関係: (左) 条件付最適解 $f^*_{\hat{y}}$ が凸ポリトープ $\mathrm{pol}_{\hat{y}}$ の内点である場合は $\mathrm{S}^3\mathrm{VM}$ の局所最適解となるが, (右) 条件付最適解 $f^*_{\hat{y}}$ が凸ポリトープ $\mathrm{pol}_{\hat{y}}$ の境界上にある場合は $\mathrm{S}^3\mathrm{VM}$ の局所最適解とはならない.

補題 5 条件付最適解 $f^*_{\hat{y}}$ が $\mathrm{pol}_{\hat{y}}$ の内点にあれば *S^3 VM* の局所最適解であり, 境界にあれば局所最適解でない.

(証明) 解が $\mathrm{pol}_{\hat{y}}$ の内点にあればその近傍もすべて $\mathrm{pol}_{\hat{y}}$ に含まれるため, $\mathrm{pol}_{\hat{y}}$ 上の極小値である $f^*_{\hat{y}}$ は局所最適解である. 一方, 補題 4 より, 解が $\mathrm{pol}_{\hat{y}}$ 境界上にあれば, 予測ラベルを (18) により反転したものに対応する $\mathrm{pol}_{\hat{y}'}$ 側の近傍に必ず厳密によい解が存在するため局所最適解ではない. (証明終)

以上より, $\mathrm{S}^3\mathrm{VM}$ の局所最適解の必要十分条件は次の定理のようになる:

定理 6 f が *S^3 VM* の局所最適解であるための必要十分条件は, *(14)*, かつ,

$$\hat{y}_i f(\boldsymbol{x}_i) > 0, \forall i \in \mathcal{U} \tag{22}$$

である.

(証明) f が条件付最適解であるための必要十分条件は補題 3 の条件 (14), (15), かつ, (16) である. 一方, f が $\mathrm{pol}_{\hat{y}}$ の内点である条件は (22) である. 補題 5 より, 両者を合わせた (14) と (22) が局所最適解の必要十分条件となる. (証明終)

図 2 は $\mathrm{S}^3\mathrm{VM}$ の解空間を模式的に表したものであり, 条件付最適解が凸ポリトープの内点である場合 (左) は局所最適解であるが, 境界上にある場合 (右) は局所最適解でないことを例示している.

4 局所最適解パス追跡アルゴリズム

本節では, パラメータ θ を 0 から 1 へ連続的に変化させたときの局所最適解のパスを計算するアルゴリズム ($\mathrm{S}^3\mathrm{VM}^{\mathrm{path}}$ アルゴリズム) を導入する. このアルゴリズムの構築に際

Algorithm 1 S^3VM^{path}

1: **Input**: ラベルありインスタンス $\{(\boldsymbol{x}_i, y_i)\}_{i\in\mathcal{L}}$, ラベルなしインスタンス $\{\boldsymbol{x}_i\}_{i\in\mathcal{U}}$, 正則化パラメータ C, カーネル関数 k
2: ラベルありインスタンスのみを用いて学習した SVM を f とし, 予測ラベルを $\hat{y}_i \leftarrow \mathrm{sgn}(f(\boldsymbol{x}_i)), i \in \mathcal{U}$, とする;
3: $\theta \leftarrow 0$;
4: **while** $\theta \leq 1$ **do**
5: **CP-phase** を実行;
6: $\mathcal{S} \leftarrow \{i \in \mathcal{U} | \hat{y}_i f(\boldsymbol{x}_i) = 0\}$;
7: **DJ-phase** を実行;
8: **end while**
9: 出力: $\theta \in [0, 1]$ における局所最適解のパス

しては, 前節で導出した S^3VM の局所最適解の性質を利用する. 補題 5 より, 条件付最適解が凸ポリトープの内点である場合は局所最適であり, 境界上にある場合は局所最適でない. したがって, S^3VM^{path} アルゴリズムでもそれぞれの局面に応じた対処が必要となる.

まず, 条件付最適解が θ のある区間内において 1 つの凸ポリトープの内点である状況を考える. このとき, そのポリトープ上で定義される条件付最適化問題は凸であるので, 凸パラメトリック計画法を利用して局所最適解の変化を追跡することができる [10]. この局面を連続パスフェーズ *(continuous path phase: CP-phase)* と呼ぶことにする.

一方, 局所最適解のパスがある θ において凸ポリトープの境界に達したとすると, その瞬間にその解は局所最適でなくなってしまう. S^3VM^{path} アルゴリズムでは, 補題 4 を利用してその θ における局所最適解を求める. (18) による予測ラベル変換を行えば, 対応する凸ポリトープ上によりよい解を見つけることができる. この局面では局所最適解のパスが不連続にジャンプするため, 不連続ジャンプフェーズ *(discontinuos jump phase: DJ-phase)* と呼ぶ.

以下では, 連続パスフェーズと不連続ジャンプフェーズを, それぞれ, 4.1 節と 4.2 節で説明する. S^3VM^{path} アルゴリズムの概要をアルゴリズム 1 に示す.

4.1 連続パスフェーズ

連続パスフェーズでは, θ のある区間内において, 条件付最適解が凸ポリトープの内点である状況を考える. この状況では条件付最適解が S^3VM の局所最適解なので, θ の変化に対する条件付最適解の変化を追跡すればよい. 条件付最適化問題は凸二次計画問題であるので, ここではパラメトリック二次計画法 [11] を用いる. パラメトリック二次計画問題の最適解パスは, パラメータ θ の区分線形関数となるため効率的に計算できる.

最適解パスを導出するため次の集合を定義する:

$$\begin{aligned}\mathcal{O} &:= \{i \in \mathcal{L} \cup \mathcal{U} | \tilde{y}_i f(\boldsymbol{x}_i) > 1\},\\ \mathcal{M} &:= \{i \in \mathcal{L} \cup \mathcal{U} | \tilde{y}_i f(\boldsymbol{x}_i) = 1\},\\ \mathcal{I}_\ell &:= \{i \in \mathcal{L} | y_i f(\boldsymbol{x}_i) < 1\},\\ \mathcal{I}_u &:= \{i \in \mathcal{U} | \hat{y}_i f(\boldsymbol{x}_i) < 1\}.\end{aligned} \tag{23}$$

これらの集合が既知のとき, 最適性条件 (14) は

$$\begin{aligned}\boldsymbol{\alpha}_{\mathcal{O}} &= \mathbf{0},\\ \boldsymbol{\alpha}_{\mathcal{I}_\ell} &= \boldsymbol{y}_{\mathcal{I}_\ell} C,\\ \boldsymbol{\alpha}_{\mathcal{I}_u} &= \hat{\boldsymbol{y}}_{\mathcal{I}_\ell} C\theta,\\ K_{\mathcal{M}\mathcal{M}}\boldsymbol{\alpha}_{\mathcal{M}} &= \tilde{\boldsymbol{y}}_{\mathcal{M}} - \mathbf{1}w_0 - K_{\mathcal{M}\mathcal{I}_\ell}\mathbf{1}C - K_{\mathcal{M}\mathcal{I}_u}\mathbf{1}\theta C\end{aligned} \tag{24}$$

と表される. ここで, $K \in \mathbb{R}^{(\ell+u)\times(\ell+u)}$ は (i,j) 要素が $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ である行列, $\mathbf{0}, \mathbf{1}$ はすべての要素が, それぞれ, 0, 1 のベクトルを表す. 集合 $\mathcal{O}, \mathcal{M}, \mathcal{I}_\ell, \mathcal{I}_u$ の要素が変わらなければ, $\boldsymbol{\alpha}_{\mathcal{O}}, \boldsymbol{\alpha}_{\mathcal{I}_\ell}$ は定数, $\boldsymbol{\alpha}_{\mathcal{I}_u}$ は θ の線形関数である. さらに, $K_{\mathcal{M}\mathcal{M}}$ が正則ならば, $\boldsymbol{\alpha}_M$ も θ の線形関数となる.

集合 $\mathcal{O}, \mathcal{M}, \mathcal{I}_\ell, \mathcal{I}_u$ の要素が変わらない条件は

$$\begin{aligned} y_i f(\boldsymbol{x}_i) &\geq 0, \quad i \in \mathcal{L} \cap \mathcal{O},\\ \hat{y}_i f(\boldsymbol{x}_i) &\geq 0, \quad i \in \mathcal{U} \cap \mathcal{O},\\ 0 \leq y_i f(\boldsymbol{x}_i) &\leq 1, \quad i \in \mathcal{L} \cap \mathcal{M},\\ 0 \leq \hat{y}_i f(\boldsymbol{x}_i) &\leq 1, \quad i \in \mathcal{U} \cap \mathcal{M},\\ y_i f(\boldsymbol{x}_i) &\leq 1, \quad i \in \mathcal{I}_\ell,\\ \hat{y}_i f(\boldsymbol{x}_i) &\leq 1, \quad i \in \mathcal{I}_u\end{aligned} \tag{25}$$

と表される. これらの条件が満たされている間は, $\boldsymbol{\alpha}$ を θ の線形関数として表すことができ, (25) のいずれか 1 つの制約がアクティブになった瞬間に集合要素の入れ替えを行う. 集合要素の入れ替えを行う点はブレイクポイント (breakpoint) と呼ばれる.

Algorithm 2 連続パスフレーズ (CP-phase)

1: 入力: 問題パラメータ θ_{bgn}, θ_{bgn} における局所最適解 f, 予測ラベル $\hat{\boldsymbol{y}}$
2: $\theta \leftarrow \theta_{\text{bgn}}$;
3: **Step1**: (23) に基づいて $\mathcal{O}, \mathcal{M}, \mathcal{I}_\ell, \mathcal{I}_u$ を更新;
4: **Step2**: 線形方程式 (24) を解く (ランク 1 更新);
5: **if** (25) の制約条件の 1 つがアクティブになれば **then**
6: 　**Step1** へ戻る;
7: **end if**
8: **if** (6) の制約条件の 1 つがアクティブになる, もしくは, $\theta = 1$ ならば **then**
9: 　$\theta_{\text{end}} \leftarrow \theta$ とし, `CP-phase` を終了する;
10: **end if**
11: 出力: $\theta \in [\theta_{\text{bgn}}, \theta_{\text{end}}]$ における局所最適解のパス

パラメトリック二次計画法は, 線形方程式 (24) の計算とブレイクポイントの計算を繰り返すアルゴリズムである. パラメトリック二次計画法の計算コストはブレイクポイント数に依存する. ブレイクポイント数は最悪の場合に問題サイズの指数オーダとなるが通常は線形オーダであることが実験的に知られている. パラメトリック二次計画法の各ステップでは線形方程式 (24) のランク 1 更新の計算コストに大部分の計算資源が費やされるため, 計算量は $\mathcal{O}(|\mathcal{M}|^2)$ である. ブレイクポイント数が線形オーダであれば, 総計算コストは $\mathcal{O}((\ell+u)|\mathcal{M}|^2)$ となる. アルゴリズム 2 に CP フェーズの概要を示す.

4.2 不連続ジャンプフェーズ

連続パスフェーズ (CP-phase) において局所最適解のパスが凸ポリトープの境界に達した瞬間にその解は局所最適でなくなってしまう. そのため, $\mathrm{S^3VM^{path}}$ アルゴリズムは不連続ジャンプフェーズ (DJ-phase) に移行し, その時点の θ における局所最適解を探す局面に入る. DJ-phase では補題 4 を利用する.

境界上のラベルなしインスタンス $\mathcal{S} \equiv \{i \in \mathcal{U} | \hat{y}_i f(\boldsymbol{x}_i) = 0\}$ の予測ラベルを (18) によって反転すれば, 新たな予測ラベル $\hat{\boldsymbol{y}}'$ に対する新たな条件付最適化問題を考えることができる. 補題 4 より, 新たな条件付最適解 $f^*_{\hat{\boldsymbol{y}}'}$ は元の条件付最適解 $f^*_{\hat{\boldsymbol{y}}}$ よりも厳密によい解であることが保証される. もし, $f^*_{\hat{\boldsymbol{y}}'}$ が凸ポリトープ $\mathrm{pol}_{\hat{\boldsymbol{y}}'}$ の内点にあれば, $f^*_{\hat{\boldsymbol{y}}'}$ はそのときの θ における局所最適解となっている. 一方, $f^*_{\hat{\boldsymbol{y}}'}$ が $\mathrm{pol}_{\hat{\boldsymbol{y}}'}$ の境界上にあれば, $f^*_{\hat{\boldsymbol{y}}'}$ は局所最適解でない. そのような場合, 補題 4 の予測ラベル更新をもう一度適用し, さらに新しい条件付最適解を計算する. 凸ポリトープの内点となる条件付最適解をみつけるまでこの手順を繰り返すと, 局所最適解を見つけることができる.

予測ラベル $\hat{\boldsymbol{y}}$ の取りうる値は有限通りであり, 新たな条件付最適解を見つけるたびに $\mathrm{S^3VM}$ の解を厳密に改善できるので, DJ-phase は有限回で収束する. DJ-phase では複数の条件付最適化問題を解く必要があるが, 予測ラベルの更新前と更新後の最適解が近いことに留意すると, 前者から後者を容易に求めることができる. 予測ラベル更新前の解 $\{\alpha_i\}_{i \in \mathcal{L} \cup \mathcal{U}}$ のうち, $\mathcal{S}$ に対応するものは更新後の問題の最適性条件を満たしていないが, 残りの $\mathcal{L} \cup \bar{\mathcal{S}}$ に対応するものは既に更新後の問題の最適性条件を満たしている. アルゴリズム 3 に DJ-phase の概要を示す.

Algorithm 3 不連続ジャンプフェーズ (DJ-phase)

1: 入力: 問題パラメータ θ, θ における (局所最適でない) 解 f, 予測ラベル $\hat{\boldsymbol{y}}$, アクティブ集合 $\mathcal{S} \equiv \{i \in \mathcal{U} | \hat{y}_i f(\boldsymbol{x}_i) = 0\}$
2: **while** $\mathcal{S} \neq \emptyset$ **do**
3: (18) により $\hat{\boldsymbol{y}}'$ を計算し, 予測ラベルを $\hat{\boldsymbol{y}} \leftarrow \hat{\boldsymbol{y}}'$ と反転;
4: 条件付最適解を $f \leftarrow \arg\min_f J(f, \hat{\boldsymbol{y}})$ s.t. (6) と計算;
5: **end while**
6: 出力: θ における局所最適解 f, 予測ラベル $\hat{\boldsymbol{y}}$

5 数値実験

本節では前節で導入した S^3VM^{path} アルゴリズムの性能を調べるため数値実験を行う. S^3VM^{path} を教師あり SVM [12], および, 既存の S^3VM アルゴリズムである S^3VM^{light} [1], deterministic アニーリング (DA) を用いたアプローチ [13], convex concave 法 (CCCP) を用いたアプローチ [14] と比較する. 評価基準として, 最適化性能, 汎化性能, 計算コストに着目する. 比較実験に用いるベンチマークデータの概要を表 1 に示す. カーネル関数にはガウスカーネル $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\gamma||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2)$ を用いる. ここで, $\gamma > 0$ はガウスカーネルの幅を決めるパラメータである.

最適化性能 S^3VM の学習は非凸最適化問題であるので, 一般に, 各アルゴリズムが異なる解を出力する. CCCP と S^3VM^{path} は S^3VM の局所最適解を出力することが保証されるが, S^3VM^{light} と DA の解は必ずしも局所最適解とは限らない. 公平な比較をするため, 各アルゴリズムの出力した予測ラベル $\hat{\boldsymbol{y}}$ を用いて計算した (5) の目的関数値 $J(f, \hat{\boldsymbol{y}})$ を比較する.

図 3 に $C \in \{1, 10, 100, 1000\}$, $\gamma = 1$, $\theta = 1$ の場合の結果を示す. S^3VM^{path} は従来法に比べて目的関数値を小さくできており, よい局所最適解が得られていることを示唆している. これは, θ を徐々に大きくしながら局所最適解を追跡することによるアニーリング法の効果と推察できる. 通常, アニーリング法ではステップサイズを細かくする方がよいとされる. S^3VM^{path} が S^3VM^{light} よりも常によい最適化性能を示しているのは, S^3VM^{path} が無限小ステップ幅のアニーリング法を行なっているためと考えられる.

汎化性能 まず, 各アルゴリズムの汎化性能として, ラベルなしデータ, および, テストデータに対する誤分類率を比較する. モデル選択は, 評価データを用いて, $C \in \{1, 10, 100, 1000\}$, $\theta \in \{\frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1\}$, $\gamma \in \{\frac{1}{4d}, \frac{1}{2d}, \frac{1}{d}, \frac{2}{d}, \frac{4}{d}\}$ の候補から選択する. ここで, d は入力 $\boldsymbol{x}$ の次元数である. ただし, S^3VM^{light} は θ を徐々に大きくしながら学習を行うので, $2^{-9}, 2^{-8}, \ldots, 2^{-1}, 1$

表 1: 数値実験で用いるベンチマークデータの概要: d は入力次元, n は総インスタンス数, ℓ はラベルありインスタンス数, u はラベルなしインスタンス数, v は評価インスタンス数, t はテストインスタンス数を表す. 半教師あり学習は, ラベルありインスンタンスが少量, ラベルなしインスンタンスが大量にある場合に利用されるので, ℓ が u に比べて十分に小さくなるように選んでいる.

Data set	d	n	ℓ	u	v	t
DIGIT1 (#D1)	241	1500	30	1200	93	177
BreaseCancerDiagnostic (#D2)	30	569	15	300	30	224
Spambase (#D3)	57	4601	70	1000	100	3431
Musk (#D4)	166	6598	70	1000	100	5428
USPS2 (#D5)	241	1500	75	1000	60	365
ESET2 (#D6)	617	2700	60	1200	80	1360
PC-MAC (#D7)	7511	1946	90	1500	23	333

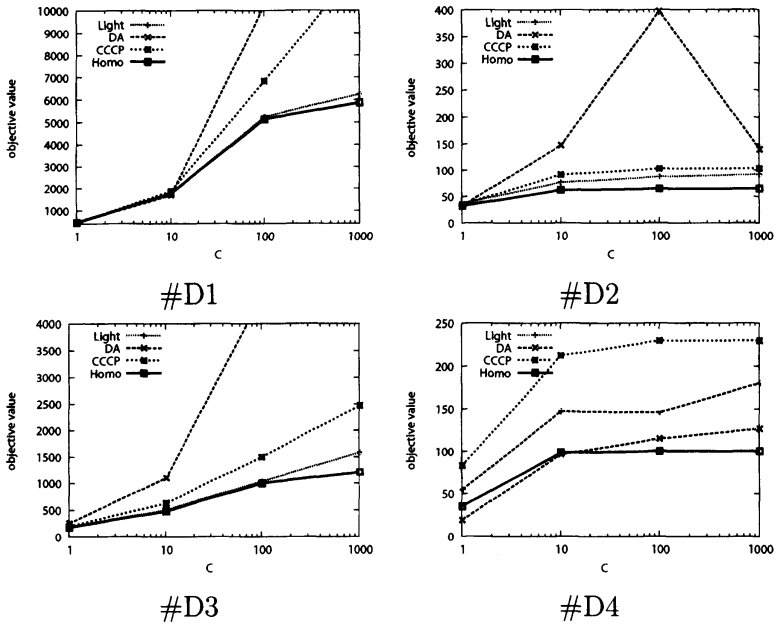


図 3: 各アルゴリズムの最適化性能の比較

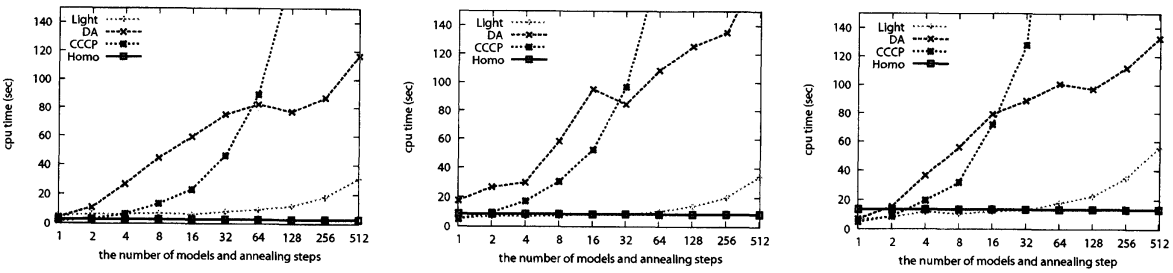
の候補から選択する. また, S^3VM^{path} はすべての $\theta \in [0,1]$ における解を計算するため, 評価データを最小にするものを選択する.

表 2 に実験結果を示す. 提案法である S^3VM^{path} は従来法と比べて誤判別率を低くする傾向があることがわかった.

計算コスト　最後に各アルゴリズムの計算コストを比較する. 図 4 は各アルゴリズムが複数の θ における解を求めるのに要した総計算コストを表している. 横軸は異なる θ における解の個数を表している. 従来法である DA, CCCP, S^3VM^{light} では解の個数が増えるに

表 2: データの分割を 10 通り行った際の平均誤判別率. "u" はラベルなしインスタンスに対する, "t" はテストインスタンスに対する平均誤判別率を表す. 数値が小さいほど汎化性能がよいことを示し, 太字は各設定において最良のものを表す.

	SVM	S^3VM^{light}	DA	CCCP	S^3VM^{path}
	u / t	u / t	u / t	u / t	u / t
D1	13.3 / 12.8	13.3 / 11.8	16.3 / 15.4	12.8 / 12.2	**12.0** / **11.0**
D2	9.7 / 9.2	11.6 / 10.3	11.6 / 11.1	8.8 / 8.2	**8.2** / **8.1**
D3	16.1 / 16.6	13.5 / 14.0	**11.7** / **11.9**	12.5 / 12.7	12.0 / 12.0
D4	11.4 / 11.7	11.5 / 11.4	9.2 / **9.2**	9.6 / 9.7	**9.0** / 9.2
D5	12.4 / 12.2	12.7 / 12.7	15.5 / 12.6	11.8 / **11.0**	**11.6** / 12.4
D6	17.0 / 17.3	12.4 / 12.9	11.8 / 12.5	11.8 / 11.9	**9.8** / **10.5**
D7	11.7 / 12.0	**7.0** / 7.0	8.5 / 7.1	8.5 / 7.0	7.3 / **6.6**



#D1 #D5 #D6

図 4: モデル選択における計算コスト

つれて計算コストも増えていることがわかる. 通常, モデル選択における候補数やアニーリング法のステップ数は多ければ多いほど好ましいが, これは計算コストとトレードオフの関係にあることがわかる. 一方, S^3VM^{path} ではすべての $\theta \in [0,1]$ における局所最適解を計算するため, 解の個数に限らず計算コストは一定となる. これは, 解の個数と計算コストのトレードオフを回避できていることを意味し, 従来法に対する提案法の有効性を示唆している.

6 まとめ

本研究では半教師ありSVMやロバストSVMなどに共通して現れる非凸最適化問題を考察した. このクラスの問題の局所最適解の性質を分析することにより, パラメトリック計画法に基づく最適化アルゴリズムを構築した. 本稿では, このクラスの非凸最適化問題の例として, 主に, 半教師ありSVMを考察した. 提案アルゴリズムを用いると, ラベルなしデータの影響を連続的に増加させていったときの局所最適解のパスを計算することができる. ベンチマークデータを用いた数値実験により, 既存のアルゴリズムよりも最適化性能, 汎化性能, 計算コストの観点で利点を持つこと示した. 今後は大規模なデータに適用するため coordinate-descent 法を用いた近似的なパラメトリック計画アルゴリズムを構築する.

参考文献

[1] T. Joachims. Transductive inference for text classification using support vector machines. *International Conference on Machine Learning*, 1999.

[2] J. Hromkovic. *Algorithmics for Hard Problems.* Springer, 2001.

[3] T. Gal. *Postoptimal Analysis, Parametric Programming, and Related Topics.* Walter de Gruyter, 1995.

[4] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

[5] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–415, 2004.

[6] X. Shen, G. Tseng, X. Zhang, and W. H. Wong. On ψ-learning. *Journal of the American Statistical Association*, 98(463):724–734, 2003.

[7] O. Chapelle, V. Sindhwani, and S. S. Keerthi. Optimization techniques for semi-supervised support vector machines. *J. Mach. Learning Res.*, 9:202–233, Feb. 2008.

[8] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. *Tenth International Workshop on Artificial Intelligence and Statistics*, 2005.

[9] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[10] E. L. Allgower and K. George. Continuation and path following. *Acta Numerica*, 2:1–63, 1993.

[11] M. J. Best. An algorithm for the solution of the parametric quadratic programming problem. *Applied Mathemetics and Parallel Computing*, pages 57–76, 1996.

[12] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1996.

[13] V. Sindhwani, S. Keerthi, and O. Chapelle. Deterministic annealing for semi-supervised kernel machines. *International Conference on Machine Learning*, 2006.

[14] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive SVMs. *Journal of Machine Learning Research*, 7:1687–1712, 2006.