SVM 型機械学習に付随する二次計画問題

知識工学部　阿部裕介

2012 年 10 月 10 日

1　概要

機械学習は計算面で見れば，何らかの最適化問題を解くことにしばしば帰着する．SVM(Support Vector Machine) は分類・回帰における代表的な機械学習アルゴリズムのひとつであり，その計算面での中心的課題は訓練データ集合から定まる二次計画問題を効率的に解くソルバーを設計することである．これについては，Platt の提案した SMO(Sequential Minimal Optimization) タイプのアルゴリズムが，目下標準的に用いられている．

本レポートでは，現在広く使われているフリーの SVM ライブラリである LIBSVM の開発者の資料をもとに，分類 (C-SVC)・回帰 (ϵ-SVR)・外れ値検出 (One-Class SVM) の 3 種の SVM 型機械学習アルゴリズムが，あるひとつの一般的な二次計画問題に帰着することを示し，その実装上の留意点について記した．

2　C-SVC の定式化

ラベルの付された訓練データ $(\mathbf{x}_i, y_i),\ y_i \in \{1, -1\},\ i = 1, \cdots, l$ が与えられているとき，2 クラス分類を行う C-SVC は，以下の二次計画問題の解を求めることで得られる．

$$\begin{aligned} \min_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}) &= \frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \mathbf{e}^T \boldsymbol{\alpha} \\ \text{subject to} \quad & 0 \le \alpha_i \le C, \quad i = 1, \cdots, l, \\ & \mathbf{y}^T \boldsymbol{\alpha} = 0. \end{aligned} \tag{1}$$

ここで $\mathbf{e}$ はすべての要素が 1 であるベクトル，C は全変数 α_i の上界，Q は $l \times l$ 行列で，その各要素は K をカーネル関数としたときに，$Q_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ となるものである．

3　ϵ-SVR の定式化

回帰用の訓練データ $(\mathbf{x}_i, z_i),\ z_i \in \mathbb{R},\ i = 1, \cdots, l$ が与えられているとき，回帰を行う ϵ-SVR は，以下の二次計画問題の解を求めることで得られる．

$$\min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} f(\boldsymbol{\alpha},\boldsymbol{\alpha}^*) = \frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)^T Q(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*) + \epsilon\sum_{i=1}^{l}(\alpha_i+\alpha_i^*) - \sum_{i=1}^{l} z_i(\alpha_i-\alpha_i^*) \tag{2}$$
$$\text{subject to} \quad 0 \le \alpha_i, \alpha_i^* \le C, \quad i = 1,\cdots,l,$$
$$\sum_{i=1}^{l}(\alpha_i-\alpha_i^*) = 0.$$

ここで ϵ は ϵ チューブの幅，C は全変数 α_i, α_i^* の上界，Q は $l \times l$ 行列で，その各要素は K をカーネル関数としたときに，$Q_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ となるものである．

4 One-Class SVM の定式化

外れ値を検出したいデータ $\mathbf{x}_i \in \mathbb{R}^n, i = 1,\cdots,l$ が与えられているとき，One-Class SVM は以下の二次計画問題の解を求めることで得られる．

$$\min_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} \tag{3}$$
$$\text{subject to} \quad 0 \le \alpha_i \le 1, \quad i = 1,\cdots,l,$$
$$\mathbf{e}^T\boldsymbol{\alpha} = \nu l.$$

ここで $\mathbf{e}$ はすべての要素が 1 であるベクトル，Q は $l \times l$ 行列で，$Q_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, また $\nu \in (0, 1]$ はサポートベクトルの個数の下限を制御するパラメタである．

5 一般形

C-SVC，ϵ-SVR，One-Class SVM に付随する二次計画問題 (1),(2),(3) は，以下の一般形に帰着できる．

$$\min_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} + \mathbf{p}^T\boldsymbol{\alpha} \tag{4}$$
$$\text{subject to} \quad 0 \le \alpha_i \le C, \quad i = 1,\cdots,l,$$
$$\mathbf{y}^T\boldsymbol{\alpha} = \Delta.$$

ここで $y_i = \pm 1,\ i = 1,\cdots,l$, Q は $l \times l$ 行列で，その各要素は K をカーネル関数としたときに，$Q_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ となるものを想定している．C-SVC と One-Class SVM に付随する二次計画問題 (1),(3) はまさにこの形である．わかりづらいのは ϵ-SVR の (2) だが，それについては以下のように (2) を書き直すことで確認できる．

$$\min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} f(\boldsymbol{\alpha},\boldsymbol{\alpha}^*) = \frac{1}{2}[\boldsymbol{\alpha}^T\ \boldsymbol{\alpha}^{*T}]\begin{bmatrix} K & -K \\ -K & K \end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha} \\ \boldsymbol{\alpha}^*\end{bmatrix} + [\epsilon\mathbf{e}^T - \mathbf{z}^T, \epsilon\mathbf{e}^T + \mathbf{z}^T]\begin{bmatrix}\boldsymbol{\alpha} \\ \boldsymbol{\alpha}^*\end{bmatrix} \tag{2'}$$
$$\text{subject to} \quad 0 \le \alpha_i, \alpha_i^* \le C, \quad i = 1,\cdots,l,$$
$$\mathbf{y}^T\begin{bmatrix}\boldsymbol{\alpha} \\ \boldsymbol{\alpha}^*\end{bmatrix} = 0.$$

ここで K は $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ なる $l \times l$ 行列であり，$\mathbf{y}$ は長さが $2l$ で，$y_i = 1,\ i = 1, \cdots, l,\ y_i = -1,\ i = l+1, \cdots, 2l$ となるベクトルである．

6 SMO および Working Set Selection 3

SVM 型の機械学習においては，(4) で示されたある特定の型の二次計画問題を解き，その解を用いて識別関数・回帰関数を構成すればよい．問題は，訓練データ集合から定まる極めて大規模な二次計画問題を効率的に解く必要があることで，汎用の QP ソルバーでは限界がある．特に問題となるのが，(4) においてあらわれる行列 Q が，訓練データサイズの二乗の大きさの密行列であることである．Q を陽に保持して計算を行おうとするのは，空間計算量の面から得策ではない．

そこで提案されたのが，J. C. Platt が 1998 年に発表した SMO(Sequential Minimal Optimization) という手法である [4]．その名が示すとおり，SMO の基本的なアイディアは，最小の部分二次計画問題を逐次的に解いて解を更新していくことで，元の二次計画問題の解を求めることにある．ここで (4) の制約条件の 2 本目の等式制約から，最小の部分二次計画問題のサイズは 2 であることがわかる．なぜなら 1 つの変数の値を更新したら，等式制約を満たすために，最低でももう 1 つ変数を動かさなければならないからである．実際には SMO では「何らかの規準」にしたがって 2 つの注目すべき変数を選び出し (これが Working Set Selection である)，それによって定まる部分二次計画を逐次的に解いていく．ここで 2 変数であることの利点は，逐次的に解くべき部分二次計画問題が「解析的に」解けてしまう点である．以上の特長から，SMO では行列 Q 全体をメモリ上に保持することなく，計算を遂行できる．

一方で SMO の問題点は，Working Set Selection の不明瞭さにある．アルゴリズム的には選択のために必要な計算が軽く，かつ全体の収束が速い (解の反復更新回数が少ない) ものが望ましい．Platt のオリジナル SMO は，ランダム性を組み込んだ Working Set Selection を行っていたが，実行毎に収束までの時間に差がでたり，得られた解が多少とはいえ異なるのはユーザからすると好ましくない．

現在[*1] もっとも標準的に使われているフリーの SVM ライブラリである LIBSVM では，Working Set Selection に関して改良を重ねており，現行の Version 2.91 では，Working Set Selection 3 という手法が用いられている [1,3]．これは Platt のオリジナル SMO に比較すると，収束に至るまでの反復更新回数が少なく，ランダム性を用いていないので，同一の訓練データ集合に対して同一の解が得られて使い勝手がよい．

7 実装上の留意点

Working Set Selection 3 の詳細や理論面については文献 [2,3] を参照していただくとして，以下では SMO 型アルゴリズムによる QP ソルバーを実装する際の留意点について述べる．

7.1 フローチャート

二次計画問題 (4) を Working Set Selection 3 を用いた SMO 型アルゴリズムで解く際のフローチャートを次に示す．

[*1] 2010 年 8 月現在.

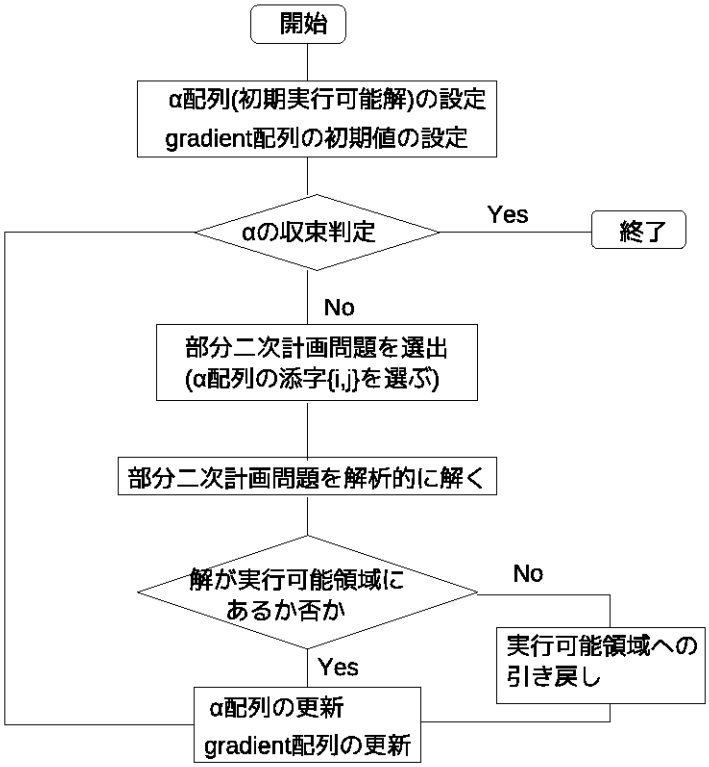

図 1: Working Set Selection 3 を用いた SMO 型 QP ソルバーのフローチャート

7.2 初期値の設定

$\boldsymbol{\alpha}$ 配列の初期値として，(4) の制約条件を満たす初期実行可能解を設定する必要がある．C-SVC(1)，ϵ-SVR(2) の場合にはすべて 0 を入れればよい．LIBSVM では One-Class SVM(3) の初期実行可能解として，$\boldsymbol{\alpha}$ 配列の最初の $[\nu l]$ 個は 1 を，その次には $(\nu l - [\nu l])$ を，それから後の残りにはすべて 0 をセットした配列を用いている．

gradient 配列 $\nabla f(\boldsymbol{\alpha}) = Q\boldsymbol{\alpha} + \mathbf{p}$ は，Working Set の選出においても，部分二次計画問題の解を求める際にも参照される重要な情報源である．その初期値は $\boldsymbol{\alpha}$ 配列の初期値を定めれば，定義から直ちに計算できる．例えば，C-SVC(1) では $\nabla f(\boldsymbol{\alpha}) = Q\boldsymbol{\alpha} - \mathbf{e}$ であるから，gradient 配列の初期値としては，すべて -1 をセットすればよい．$\boldsymbol{\alpha}$ 配列の更新に伴う gradient 配列の更新も同様に計算できる．

7.3 α の収束判定

KKT 条件で判定するが，gradient 配列 $\nabla f(\boldsymbol{\alpha})$ を利用した等価な条件に書き換えることで，判定に要する計算を軽くする工夫が LIBSVM ではなされている [1].

7.4 部分二次計画問題の解析的な解

簡単のため，K が正定値の場合のみを示す．このとき，$i \neq j$ ならば $K_{ii} + K_{jj} - 2K_{ij} > 0$ である．現実の場面では K が正定値でないカーネル関数[*2]が用いられる場合もあり，$i \neq j$ でも $K_{ii} + K_{jj} - 2K_{ij} \leq 0$ となりうる．こうしたケースでは解くべき部分二次計画問題に付加項を加えて別途対処する [3].

Working Set Selection 3 によって選出された $\boldsymbol{\alpha}$ 配列の添字集合を $B = \{i, j\}$ とし，$N = \{1, 2, \cdots, l\} \setminus B$ とおくと，解くべき部分二次計画問題は次のようになる．

$$\begin{aligned} &\min_{\alpha_i, \alpha_j} f(\alpha_i, \alpha_j) = \frac{1}{2}[\alpha_i \ \alpha_j] \begin{bmatrix} Q_{ii} & Q_{ij} \\ Q_{ij} & Q_{jj} \end{bmatrix} \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix} + (\mathbf{p}_B + Q_{BN}\boldsymbol{\alpha}_N)^T \begin{bmatrix} \alpha_i \\ \alpha_j \end{bmatrix} \\ &\text{subject to} \quad y_i\alpha_i + y_j\alpha_j = \Delta - \mathbf{y}_N^T\boldsymbol{\alpha}_N (= y_i\alpha_i^{\text{old}} + y_j\alpha_j^{\text{old}}). \end{aligned} \tag{5}$$

ここで本来ならば，下の不等式制約 (6) も加味して考えなければならないが，まず (5) を解き，(6) を満たしていなかったら実行可能領域に引き戻す処理を行う．

$$0 \leq \alpha_i, \alpha_j \leq C. \tag{6}$$

(5) の解は，$a = K_{ii} + K_{jj} - 2K_{ij}$, $b = -y_i\nabla f(\boldsymbol{\alpha})_i + y_j\nabla f(\boldsymbol{\alpha})_j$ とおいたとき，次のようになる．

$$\begin{cases} \alpha_i = \alpha_i^{\text{old}} + \frac{y_i b}{a} \\ \\ \alpha_j = \alpha_j^{\text{old}} - \frac{y_j b}{a}. \end{cases} \tag{7}$$

7.5 実行可能領域への引き戻し

(7) で求めた (5) の解 α_i または α_j が不等式制約 (6) を満たさないケースでは，以下の手順で実行可能領域に引き戻す．これは clipping と呼ばれる．ここでは α_i が (6) を満たさない場合について述べる．α_j が (6) を満たさない場合でも同様.

(6) を満たすように，新しく $\alpha_i^{\text{clipped}}$ を次のように定める．

$$\alpha_i^{\text{clipped}} = \begin{cases} C & \text{if} \quad \alpha_i > C \\ \alpha_i & \text{if} \quad 0 \leq \alpha_i \leq C \\ 0 & \text{if} \quad \alpha_i < 0. \end{cases} \tag{8}$$

これに伴い，(5) の等式制約から $\alpha_j^{\text{clipped}}$ は以下の式 (9) で求められる．

$$\alpha_j^{\text{clipped}} = \alpha_j + y_i y_j (\alpha_i - \alpha_i^{\text{clipped}}). \tag{9}$$

[*2] 例えば，シグモイドカーネル $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i^T \mathbf{x}_j + r)$.

8 まとめ

SVM の解説では，2 クラス分類を行う C-SVC を二次計画問題に落とし込むところまで解説されているものが多い．こうした解説に接して，回帰や外れ値検出を行う SVM 型機械学習との関係はどうなっているのか，またその二次計画問題は具体的にはどうやって解けばよいのか，という疑問を持つ読者は（自分以外にも）少なからずいるように思われる．本レポートではこれらの疑問に対してすべて答えるとまではいかないまでも，大まかな考え方の枠組みと参照すべきポインタを提示することを目指した．とりわけ，SVM を使うのではなく実装してみたい人にとっては，すぐに目につきやすい Platt の SMO よりも，参考文献に挙げた LIBSVM の開発資料を読むことをすすめたい．SVM を用いてデータ解析を行う人にとっても，ブラックボックスとして扱われがちな SVM の内部機構について，少しでも理解と興味を持っていただければ幸いである．

参考文献

[1] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines, 2010.

[2] P.-H. Chen, R.-E. Fan and C.-J. Lin. A study on SMO-type decomposition methods for support vector machines, 2006.

[3] R.-E. Fan, P.-H. Chen and C.-J. Lin. Working set selection using second order information for training support vector machines, 2005.

[4] J. C. Platt. Fast training of support vector machines using sequential minimal optimization, 1998.