

平成18年度学士論文

## 2値判別問題における有効制約法

東京工業大学 理学部 情報科学科  
学籍番号 03-2594-3

山下 寛隆

指導教員  
小島 政和 教授

2007年2月8日

# 目次

<b>1</b>	<b>はじめに</b>	<b>2</b>
<b>2</b>	<b>2 値判別問題と CGS model</b>	<b>3</b>
2.1	2 値判別問題 . . . . .	3
2.2	Support Vector Machine . . . . .	3
2.3	CGS model . . . . .	5
<b>3</b>	<b>凸 2 次計画問題に対する有効制約法の概要</b>	<b>9</b>
3.1	凸 2 次計画問題 . . . . .	9
3.2	有効制約法の考え方 . . . . .	9
3.3	勾配射影法 . . . . .	11
3.4	凸 2 次計画問題に対する有効制約法のアルゴリズム . . . . .	13
<b>4</b>	<b>凸 2 次計画問題に対する有効制約法の実装</b>	<b>15</b>
4.1	初期実行可能解の選び方 . . . . .	15
4.2	最適解情報からのリスタート . . . . .	16
4.3	逆行列を使わない計算方法 . . . . .	16
4.4	一部の変数を固定する . . . . .	18
<b>5</b>	<b>数値実験</b>	<b>21</b>
5.1	実験方法 . . . . .	21
5.2	実験結果 . . . . .	22
5.3	考察 . . . . .	23
<b>6</b>	<b>まとめと今後の課題</b>	<b>25</b>

# 1 はじめに

2 値判別問題とは、複数のデータが 2 つのグループに分かれている状態で新たな未知データが与えられた時に、そのデータがどちらのグループに属しているかを定める問題である。2 値判別問題は病気の発症パターンの研究などで頻繁に現れている。2 つのグループを判別する方法としてサポートベクターマシンと呼ばれるパターン認識手法が発達してきた。その中でも代表的な  $\beta$ -SVM の拡張として、近年 *J. Gotoh, A. Takeda* によって、CGS model(Conditional Geometric Score) という線形判別モデルが提案された。CGS model は最適値が負の場合は  $\beta$ -SVM と同値であり、変数  $\lambda$  の凸 2 次計画問題

$$\left\{ \begin{array}{l} \underset{\lambda}{\text{maximize}} \quad - \sum_{i=1}^m \sum_{j=1}^m y_i y_j x_i^T x_j \lambda_i \lambda_j \\ \text{subject to} \quad \sum_{i=1}^m \lambda_i y_i = 0 \\ \sum_{i=1}^m \lambda_i = 1 \\ 0 \leq \lambda_i \leq \frac{1}{(1-\beta)m} \quad i = 1, 2, \dots, m \end{array} \right.$$

に帰着される [2]。ただし、 $x_i$  はサンプルデータ、 $y_i$  は  $x_i$  に対応する +1 か -1 のラベル、 $m$  はデータ数、 $\beta$  は  $0 < \beta < 1$  の範囲で自分で設定するパラメータである。 $(x_i, y_i)$  は  $\lambda_i$  に対応している。最適解  $\lambda^*$  が求まると、2 つのグループを分ける最適な判別平面を構成出来るが、パラメータ  $\beta$  をどう定めれば判別率が最も良くなるかはわからないため、 $\beta$  を適当に変えて複数の最適化問題を解く必要がある。しかし、 $\beta$  を少し変化させても不等式制約の範囲が少し変化するだけで、問題は類似したままである。 $\beta_1 < \beta_2$  のとき、 $\lambda$  の定義域は大きくなる。つまり、 $\beta_1 < \beta_2$  のとき、 $\beta_1$  での最適解が  $\beta_2$  での実行可能解になっている。本研究はその特徴に着目し、 $\beta$  を動かしながら効率的に最適解を求める方法を提案する。

凸 2 次計画問題に適用できる最適化手法は様々なものがあるが、本研究の着目点から、最適解の情報を利用しやすい有効制約法を使用することにした。有効制約法は元問題の制約を等式制約と不等式制約に分けた部分問題を解きながら、元問題の最適解に近づいていく方法である。

有効制約法を MATLAB で実装し、数値実験を行った。得られた結果に対し、最適解の情報を利用することによる効果について考察をした。

本論文の構成は、

第 2 章 サポートベクターマシン、CGS model の紹介

第 3 章 有効制約法の説明、またその中で出てくる勾配射影法の説明

第 4 章 有効制約法を実装する上で計算の効率化を図るために施した点  
具体的には初期解の設定、一部分の変数を固定する方法など

第 5 章 MATLAB での数値実験、内点法との計算時間の比較など

となっている。

ちなみに文中では、記号  $x$  をデータとして扱い、記号  $\lambda$  を変数として扱うことに注意されたい。

## 2 2値判別問題とCGS model

### 2.1 2値判別問題

複数のデータが、グループごとに分けられている状態を考える。新たなデータが与えられた時に、そのデータがどのグループに属するかを決める問題を判別問題という。グループの数が2つの場合を2値判別問題、3つ以上の場合を多値判別問題とよぶ。

以下の例を考えよう。

No.	年齢	性別	体重	血圧	症状
1	30	男	72	98	糖尿病
2	45	男	75	110	糖尿病
3	35	男	64	92	健康
4	37	女	50	102	健康
5	28	男	62	86	糖尿病
6	51	男	68	107	健康
7	42	女	55	98	健康
:	:	:	:	:	:

表 1: 糖尿病患者のデータ

複数の人間の様々なデータを取り、このデータから判別基準を作成する。新しい人間のデータを調べ、判別基準と照らし合わせることでその人が糖尿病かどうかを判断することが出来る。判別基準が信頼出来るものであればある程、未知のデータの判別を間違える可能性は低くなる。あらかじめ与えてあるデータと属性の対応づけのことを教師データと呼び、教師データがあるような学習方法を教師あり学習と呼ぶ。

ここで判別基準が必要となるが、2つのグループを判別する方法として、サポートベクターマシン (Support Vector Machine, SVM) と呼ばれるパターン認識手法が挙げられる。

### 2.2 Support Vector Machine

サポートベクターマシンは、線形判別関数を用いて2つのグループを分ける手法であり、線形判別関数は

$$f(x) = \text{sign}(w^T x + b)$$

で表される。 $x$  は入力データ、 $w, b$  は判別関数を決定するパラメータである。ただし、

$$\text{sign}(u) = \begin{cases} 1 & u > 0 \\ -1 & u \leq 0 \end{cases}$$

つまり、2つのグループ(ここでは と )に対して、各グループのラベルを+1 と-1 に数値化する。データを誤りなく分けるだけでなく、未知のデータに対しても正しく判別するためには、2つのグループをなるべく余裕を持って分けるような判別平面が求められる。

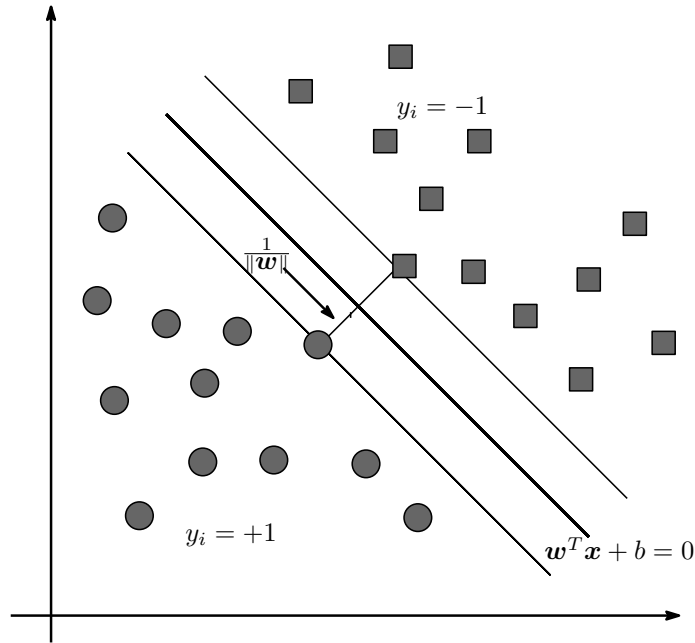


図 1：線形判別（ がラベル+1で、 がラベル-1。データ数は  $m$  とする。）

2次元で線形分離可能な場合を考えると、

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, m \quad (1)$$

を満たすようなパラメータ  $\mathbf{w}, b$  が存在する。これは、2直線  $\mathbf{w}^T \mathbf{x} + b = \pm 1$  でデータが完全に分離されていて、 $\mathbf{w}^T \mathbf{x} + b = 0$  との距離  $\frac{1}{\|\mathbf{w}\|}$  の中にはデータが1つも存在しないことを示している。つまり、制約 (1) の下で、距離  $\frac{1}{\|\mathbf{w}\|}$  を最大にするようなパラメータを求めればよい。等価な問題へ変形すれば、最適な判別平面を求める問題は、

$$\begin{cases} \text{minimize}_{\mathbf{w}, b} & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{subject to} & y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, m \end{cases}$$

となる。これを Hard Margin SVM という。

線形分離不可能な場合は上記の考え方は使えない。データ  $\mathbf{x}_i$  に対してスラック変数  $\xi_i \geq 0$  を導入し、その分だけ超平面をはみ出すことを許す。その代わりに、はみ出した分は罰金項を設けることにし、重み定数を  $C$  とすると、最適な判別平面を求める問題は

$$\begin{cases} \text{minimize}_{\mathbf{w}, b, \xi} & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \\ \text{subject to} & \xi_i \geq 1 - y_i(\mathbf{w}^T \mathbf{x}_i + b) \\ & \xi_i \geq 0 \quad i = 1, 2, \dots, m \end{cases}$$

となる。これを、Soft Margin SVM (または、C-SVM) という。

C-SVMの欠点として、モデルのイメージが難しいことが挙げられる。それに代わる新しいモデルとして  $\nu$ -SVM が提案された。超平面をはみ出すデータ数の上限値としてパラメータ  $\nu$  を指定し、

$$\begin{cases} \text{minimize}_{\mathbf{w}, b, \mathbf{z}, \rho} & \frac{1}{2} \|\mathbf{w}\|^2 - \nu \rho + \frac{1}{m} \sum_{i=1}^n z_i \\ \text{subject to} & z_i \geq \rho - y_i(\mathbf{w}^T \mathbf{x}_i + b) \quad i = 1, 2, \dots, m \\ & z_i \geq 0, \quad \rho \geq 0 \end{cases} \quad (2)$$

で定式化される。C-SVM で  $C = \frac{1}{\rho}$  に取れば、 $\rho$ -SVM と同じ結果が得られる。パラメータの詳しい説明については、[1] を参照して欲しい。

### 2.3 CGS model

近年、*J.Gotoh and A.Takeda* によって、CGS model(Conditional Geometric Score) という線形判別モデルが提案された。このモデルは C-SVM とは異なるリスク尺度を導入し、Hard Margin SVM や  $\rho$ -SVM の拡張になっている。ここではモデルの簡単な紹介と、凸 2 次計画問題への変形過程を述べることにする。

超平面  $\mathbf{w}^T \mathbf{x} + b = 0$  とデータ  $\mathbf{x}_i$  との距離は、

$$g(\mathbf{w}, b, \mathbf{x}_i, y_i) = \frac{y_i(\mathbf{w}^T \mathbf{x}_i + b)}{\|\mathbf{w}\|}$$

で表される。 $-g(\mathbf{w}, b, \mathbf{x}_i, y_i)$  を損失、 $-g(\mathbf{w}, b, \mathbf{x}_i, y_i)$  の分布を損失分布と定義する。損失分布を考えると、 $-g(\mathbf{w}, b, \mathbf{x}_i, y_i) > 0$  のデータ  $\mathbf{x}_i$  が正しく判別出来ていないデータに該当する。この損失分布の下側の確率が  $\beta$  になる点  $\alpha_\beta$  を超える損失についての平均  $\phi_\beta(\mathbf{w}, b)$  を最小にするように判別関数を決めることがこのモデルの根本的な考え方である。

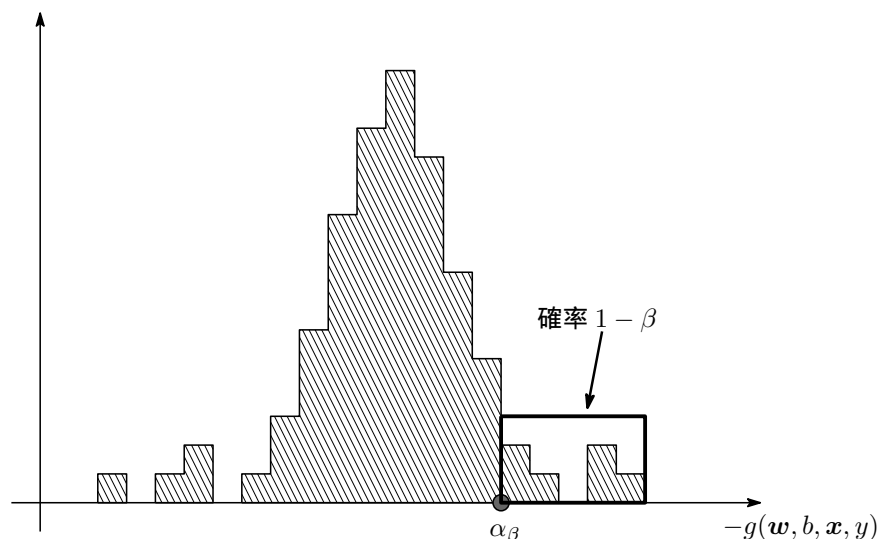


図 2 :  $-g(\mathbf{w}, b, \mathbf{x}, y)$  の分布

$\beta$  は自分で設定するパラメータである。 $\beta$  を大きく設定すると、誤判別の大きいデータのみを扱う。逆に  $\beta$  を小さく設定すると、正しく判別は出来ているが超平面に近いデータも含めて扱うことになる。 $\phi_\beta(\mathbf{w}, b)$  の最小化は *Rockafellar-Uryasev* によって以下のように書き換えられることがわかっている。

#### Rockafellar-Uryasev(1999)

$$\begin{aligned} \min_{\mathbf{w}, b} \phi_\beta(\mathbf{w}, b) &= \min_{\mathbf{w}, b, \alpha} F_\beta(\mathbf{w}, b, \alpha) \\ F_\beta(\mathbf{w}, b, \alpha) &:= \alpha + \frac{1}{1-\beta} E [[f(\mathbf{w}, b) - \alpha]^+] \\ [X]^+ &= \max\{X, 0\} \end{aligned}$$

この式を適用すると、

$$\min_{\mathbf{w}, b} \phi_\beta(\mathbf{w}, b) \implies \min_{\mathbf{w} \neq \mathbf{0}, b, \alpha} \alpha + \frac{1}{1-\beta} \sum_{i=1}^m \frac{1}{m} \left[ -\frac{y_i(\mathbf{w}^T \mathbf{x}_i + b)}{\|\mathbf{w}\|} - \alpha \right]^+ \quad (3)$$

と表現できる。m 個のデータは等確率  $\frac{1}{m}$  で出現するとした。ちなみに、 $\beta > 1 - \frac{1}{m}$  ならば、Hard Margin SVM と同じ結果が得られる。ここで、

$$\frac{y_i(\mathbf{w}^T \mathbf{x}_i + b)}{\|\mathbf{w}\|} = \frac{y_i(\eta \mathbf{w}^T \mathbf{x}_i + \eta b)}{\|\eta \mathbf{w}\|}$$

であり、 $\mathbf{w}, b$  を定数倍してもこの項の値は変わらないので、最適解を求める上では  $\|\mathbf{w}\|^2 = 1$  と固定して構わない。さらに、スラック変数  $z_i$  を導入して  $[\ ]^+$  を消すと、(3) は、同値な問題

$$\left\{ \begin{array}{l} \text{minimize}_{\mathbf{w}, b, \alpha, \mathbf{z}} \quad \alpha + \frac{1}{(1-\beta)} \sum_{i=1}^m \frac{1}{m} z_i \\ \text{subject to} \quad z_i + y_i(\mathbf{w}^T \mathbf{x}_i + b) + \alpha \geq 0 \quad i = 1, 2, \dots, m \\ z_i \geq 0 \quad i = 1, 2, \dots, m \\ \|\mathbf{w}\|^2 = 1 \quad \dots (*) \end{array} \right. \quad (4)$$

へと書き換えられる。最適値が負の場合は (\*) を  $\|\mathbf{w}\|^2 \leq 1$  に変換した緩和問題を解き、得られた最適解を  $\frac{1}{\|\mathbf{w}\|}$  倍することで式 (4) の最適解を得ることが出来る。

式 (4) のラグランジュ関数を用いて、双対問題を導出しよう。ラグランジュ関数は

$$L(\mathbf{w}, b, \alpha, \mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \delta) := \alpha + \frac{1}{1-\beta} \sum_{i=1}^m \frac{1}{m} z_i - \sum_{i=1}^m \lambda_i \{z_i + y_i(\mathbf{w}^T \mathbf{x}_i + b) + \alpha\} - \sum_{i=1}^m \mu_i z_i + \frac{1}{2} \delta (\|\mathbf{w}\|^2 - 1)$$

ラグランジュ双対問題は

$$\begin{array}{ll} \text{maximize} & \text{minimize} \\ \boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{\mu} \geq \mathbf{0}, \delta & \mathbf{w}, b, \alpha, \mathbf{z} \end{array} L(\mathbf{w}, b, \alpha, \mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \delta)$$

で表される。停留条件および  $\lambda_i \geq 0, \mu_i \geq 0$  より

$$\sum_{i=1}^m \lambda_i y_i \mathbf{x}_i = \delta \mathbf{w} \quad (5)$$

$$\sum_{i=1}^m y_i \lambda_i = 0 \quad (6)$$

$$\sum_{i=1}^m \lambda_i = 1 \quad (7)$$

$$0 \leq \lambda_i \leq \frac{1}{(1-\beta)m} \quad (8)$$

(5) ~ (7) を  $L$  に代入して、

$$L = -\frac{1}{2} \delta (\|\mathbf{w}\|^2 + 1)$$

が得られ、 $\delta \neq 0$  のとき、ラグランジュ双対問題

$$\text{maximize}_{\boldsymbol{\lambda}, \delta} \quad -\frac{1}{2} \left( \delta + \frac{1}{\delta} \left\| \sum_{i=1}^m y_i \lambda_i \mathbf{x}_i \right\|^2 \right) \quad \text{subject to (6) to (8)} \quad (9)$$

を得る。(4)の最適値が負の場合、(9)の最適値も負になる。よって少なくとも $\delta > 0$ でなければならず、 $\|w\|^2 = 1$ と合わせて、 $\|\sum_{i=1}^m y_i \lambda_i \mathbf{x}_i\| > 0$ と言える。相加相乗より、

$$-\frac{1}{2}\left(\delta + \frac{1}{\delta}\left\|\sum_{i=1}^m y_i \lambda_i \mathbf{x}_i\right\|^2\right) \leq -\left\|\sum_{i=1}^m y_i \lambda_i \mathbf{x}_i\right\| = -\sqrt{\sum_{i=1}^m \sum_{j=1}^m y_i y_j \mathbf{x}_i^T \mathbf{x}_j \lambda_i \lambda_j}$$

が成り立つ。(等号成立は $\delta = \sqrt{\sum_{i=1}^m \sum_{j=1}^m y_i y_j \mathbf{x}_i^T \mathbf{x}_j \lambda_i \lambda_j}$ のとき。)したがって、(4)の最適値が負の場合は、以下の双対問題が導かれる。

$$\left\{ \begin{array}{l} \text{maximize}_{\lambda} \quad -\sum_{i=1}^m \sum_{j=1}^m y_i y_j \mathbf{x}_i^T \mathbf{x}_j \lambda_i \lambda_j \\ \text{subject to} \quad \sum_{i=1}^m \lambda_i y_i = 0 \\ \quad \quad \quad \sum_{i=1}^m \lambda_i = 1 \\ \quad \quad \quad 0 \leq \lambda_i \leq \frac{1}{(1-\beta)m} \quad i = 1, 2, \dots, m \end{array} \right. \quad (10)$$

これは一般に凸2次計画問題と呼ばれる最適化問題である。本研究で扱う最適化問題はこの凸2次計画問題(10)である。最適解 $\lambda^*$ が求まると、2つのグループを分ける最適な判別平面を $\lambda^*$ から構成出来る。どのパラメータ $\beta$ に対して判別率が最も良くなるかは事前にわからないため、 $\beta$ を適当に変えて複数の最適化問題を解く必要がある。しかし、 $\beta$ を少し変化させても(10)は不等式制約の範囲が少し変化するだけで、問題は類似したままである。さらに $\beta_1 < \beta_2$ のとき、 $\beta_1$ での最適解が $\beta_2$ での実行可能解になっているという重要な特徴を持つ。本研究ではこの特徴に着目し、前の最適解の情報を利用してパラメータを動かしながら効率的に解を求める方法について提案する。

最後に、CGS modelが $\nu$ -SVMの拡張となっていることを示そう。 $\nu$ -SVM(2)の双対問題はCGS modelの双対問題と同様の導出を行うと

$$\left\{ \begin{array}{l} \text{maximize}_{\lambda} \quad -\sum_{i=1}^m \sum_{j=1}^m y_i y_j \mathbf{x}_i^T \mathbf{x}_j \lambda_i \lambda_j \\ \text{subject to} \quad \sum_{i=1}^m \lambda_i y_i = 0 \\ \quad \quad \quad \sum_{i=1}^m \lambda_i \geq \nu \quad \dots (**) \\ \quad \quad \quad 0 \leq \lambda_i \leq \frac{1}{m} \quad i = 1, 2, \dots, m \end{array} \right. \quad (11)$$

となる。 $\hat{\lambda} = \frac{1}{\nu} \lambda$ とおけば、(11)は不等式 $\sum_{i=1}^m \hat{\lambda}_i \geq 1$ を除き(10)と一致することがわかる。(\*\*)は式(2)の $\rho \geq 0$ に対応する項である。また、CGS modelと $\nu$ -SVMの間には、

$$\alpha = -\rho, \quad \beta = 1 - \nu$$

の関係がある。前に述べたように、CGS modelの目的関数は損失分布の上側 $1 - \beta$ 点を越える損失の平均であり、 $\alpha$ はそのしきい値であるので、最適値が負のとき、 $\alpha < 0$ であることは明らかである。よって、 $\rho > 0$ は常に成り立つのでこの制約を削除でき、対応する不等式は $\sum_{i=1}^m \hat{\lambda}_i = 1$ と見なせる。以上より、最適値が負の場合CGS modelと $\nu$ -SVMが同値であることが示された。CGS modelではStep 1としてまず



凸2次計画問題 (10) を解く。

最適値が非負の場合、(2) は  $(w, b, z, \rho) = 0$  のとき最適値 0 が常に成立するので、意味のない解を出力することになる。一方、CGS model は P.6 で示した (4) の緩和問題を解き、 $\|w\| = 0$  を出力した後は Step 2 へと移る。次の問題

$$\left\{ \begin{array}{l} \underset{w, b, \alpha, z}{\text{minimize}} \quad \alpha + \frac{1}{(1-\beta)} \sum_{i=1}^m \frac{1}{m} z_i \\ \text{subject to} \quad z_i + y_i(w^T x_i + b) + \alpha \geq 0 \quad i = 1, 2, \dots, m \\ z_i \geq 0 \quad i = 1, 2, \dots, m \\ \max_{i=1, 2, \dots, n} |w_i| = \frac{1}{\sqrt{n}} \end{array} \right.$$

を解き、得られた最適解を  $\frac{1}{\|w\|}$  倍することでまず (4) の実行可能解  $(\bar{w}, \bar{b}, \bar{\alpha}, \bar{z})$  を得る。次に  $\bar{w}$  を使用し、(4) の制約 (\*) を線形制約に変換した多面体集合

$$\mathcal{F}(\bar{w}) = \left\{ (w, b, \alpha, z) \left| \begin{array}{l} z_i + y_i(w^T x_i + b) + \alpha \geq 0 \quad i = 1, 2, \dots, m \\ z_i \geq 0 \quad i = 1, 2, \dots, m \\ \bar{w}^T w = 1 \end{array} \right. \right\}$$

を定義し、 $\mathcal{F}$  上で (4) の目的関数を最小化する線形計画問題

$$\underset{w, b, \alpha, z}{\text{minimize}} \quad q(z, \alpha) := \alpha + \frac{1}{1-\beta} \sum_{i=1}^m \frac{1}{m} z_i \quad \text{subject to } (w, b, \alpha, z) \in \mathcal{F}(\bar{w})$$

を考える。ピボット演算を1回することで現在の目的関数値  $q(\bar{z}, \bar{\alpha})$  より値が小さくなる点に移動するので、それを  $\frac{1}{\|\tilde{w}\|}$  倍することで、新しい実行可能解  $(\tilde{w}, \tilde{b}, \tilde{\alpha}, \tilde{z})$  を得られる。 $\tilde{w}$  から多面体集合  $\mathcal{F}(\tilde{w})$  を構成し、同様の操作を局所解に到達するまで繰り返す。得られる最適解は局所解であるが、構成された判別平面は最適値が負の場合と同等の判別率を保っている。

CGS model は -SVM と同値な問題を解き (Step 1)、得られた  $\|w\|$  の値によって Step 2 に移るかどうかを決める。この2段階のアルゴリズムが -SVM の拡張を実現している。詳しい説明は [2] を参照してほしい。

### 3 凸2次計画問題に対する有効制約法の概要

前章では本研究で扱う CGS model 内の凸2次計画問題の導出をしたが、本章ではまず凸2次計画問題の定式化をし、2次計画問題に対する手法として有名な有効制約法のアルゴリズムを紹介する。

#### 3.1 凸2次計画問題

2次計画問題(Quadratic Problem)とは、制約式が線形の等式と不等式であり、目的関数が2次関数であるような最適化問題のことである。

$$\begin{cases} \text{minimize} & c^T \lambda + \frac{1}{2} \lambda^T Q \lambda \\ \text{subject to} & a_i^T \lambda = b_i \quad i \in I_1 \\ & a_i^T \lambda \leq b_i \quad i \in I_2 \end{cases}$$

ここで、 $Q$  は  $n \times n$  行列であり、

$$\lambda^T Q \lambda = (\lambda^T Q \lambda)^T = \lambda^T Q^T \lambda = \frac{1}{2} \lambda^T (Q + Q^T) \lambda$$

となるので、任意の  $Q$  は  $\frac{1}{2}(Q + Q^T)$  に置き換えることが出来る。よって  $Q$  は対称行列としても一般性を失わない。さらに、任意の  $\lambda \in R^n$  に対して、 $\lambda^T Q \lambda \geq 0$  となるとき、 $Q$  は半正定値行列であるという。このとき、凸2次計画問題(Convex Quadratic Problem)とよぶ。関数  $f$  が凸関数であるとは、

$$f(\theta u + (1 - \theta)v) \leq \theta f(u) + (1 - \theta)f(v) \quad \forall u, v \in R^n, \theta \in [0, 1]$$

が成り立つことをいう。凸2次計画問題は非凸2次計画問題よりも解くのが容易であり、局所的最適解が大域的最適解になるという重要な性質を持つ。

2次計画問題を解く様々な手法が提案されているが、本研究の趣旨を考えると、初期解の選び方によって計算時間が変わるような手法が望ましい。そこで本研究では有効制約法を使用した。次節以降で有効制約法の考え方を述べるとともに、凸2次計画問題へ適用したアルゴリズムを記述する。

#### 3.2 有効制約法の考え方

まず、一般的な最適化問題で有効制約法の考え方を見ていくことにしよう。

$$\begin{cases} \text{minimize} & f(\lambda) \\ \text{subject to} & g(\lambda) \leq 0 \end{cases} \quad (12)$$

簡単のため、不等式制約だけとする。ラグランジュ乗数を  $\mu_i$  とすると、 $\lambda_k$  が式(12)の最適解であるならば、以下の条件を満たすような  $\mu$  が存在する。

$$\begin{aligned} \nabla f(\lambda_k) + \mu^T \nabla g(\lambda_k) &= 0 \\ g(\lambda_k) &\leq 0 \\ \mu^T g(\lambda_k) &= 0 \\ \mu &\geq 0 \end{aligned} \quad (13)$$

ただし、 $\nabla g_i(\lambda_k)$  は1次独立と仮定する。実行可能解  $\lambda$  に対して  $g_i(\lambda) = 0$  が成り立つとき、元の制約式  $g_i(\lambda) \leq 0$  は点  $\lambda$  で効いているといい、この制約式を  $\lambda$  で有効な制約式という。

有効な制約式の添字集合を  $W$  とおき、これを Working Set とよぶことにする。 $\lambda_k$  に対応する Working Set  $W_k$  を用いて (13) は、

$$\begin{aligned} \nabla f(\lambda_k) + \sum_{i \in W_k} \mu_i \nabla g_i(\lambda_k) &= 0 \\ g_i(\lambda_k) &= 0, & i \in W_k \\ g_i(\lambda_k) &< 0, & i \notin W_k \\ \mu_i &\geq 0, & i \in W_k \quad \dots (*) \\ \mu_i &= 0, & i \notin W_k \end{aligned} \tag{14}$$

と表される。ここで、 $\lambda_k$  での有効な制約式を等式に置き換えた (12) の部分問題

$$\begin{cases} \text{minimize} & f(\lambda) \\ \text{subject to} & g_i(\lambda) = 0 & i \in W_k \\ & g_i(\lambda) < 0 & i \notin W_k \end{cases} \tag{15}$$

を考えよう。この制約式が作る超平面を Working Set に対応して Working Surface と名づける。式 (15) の実行可能解が式 (12) の実行可能解になっていることは明らかであり、式 (15) の最適解とラグランジュ乗数は (14) の (\*) 以外を満たしている。さらに (\*) も満たすとき、それは元問題 (12) の停留点となる。(14) は最適解であるための必要条件であり、一般の最適化問題では最適解を求めることが困難なので、この停留点を求めることを目標とする場合も多い。しかし、本論文で扱うような凸 2 次計画問題では (14) は最適解であるための必要十分条件となるので、(14) を全て満たした実行可能解を最適解として構わない。

(12) の実行可能解から始めて (14) が満たされるまで、Working Set を更新し、部分問題 (15) を作り、その部分問題を解くというサイクルを繰り返す。Working Set を上手く更新することにより、サイズの小さい問題を解きつつ、元問題の最適解に近づいていけることが有効制約法の特長である。また、初期実行可能解が最適解に近い場合、反復回数が少なく済む利点がある。有効制約法の具体的なアルゴリズムについては次節で述べることにし、ここでは Working Set の変更の際の手順について述べる。Working Set を変更する必要があるのは、

- (i) 式 (15) の最適解に到達した場合
- (ii) 新しい制約境界に到達した場合

のどちらかである。

#### (i) の場合

全ての  $i \in W$  に対して  $\mu_i \geq 0$  となっていれば、終了。そうでないならば、 $\mu_i < 0$  に対応する  $i \in W$  の少なくとも 1 つを Working Set から外す。新しい Working Set で部分問題 (15) を作り、再び部分問題を解く。

#### (ii) の場合

式 (15) の最適解に到達する前に、有効でない制約の境界に達してしまうことがある。その場合にはその制約を有効な制約として扱う方が都合がいい。つまり、新しく有効になった制約  $g_j(x) = 0$  の添字  $j$  を Working Set に入れる。新しい Working Set で部分問題 (15) を作り、再び部分問題を解く。

Working Set が更新されるまでは、目的関数を減少させる降下方向  $d$  に沿って部分問題の実行可能解を更新していく。ただし、降下方向  $d$  は有効制約を保ってなければならない。降下方向の選び方として、本研究では勾配射影法を使用した。

### 3.3 勾配射影法

勾配射影法 (Gradient Projection Method) は制約なし最適化の最急降下法と考え方が似ている。制約式が線形で定義されている場合、アルゴリズムの記述が簡単になる。本研究で扱う凸 2 次計画問題も線形制約のみであるので、3.2 節よりも問題を少し具体化し、今後は制約式が線形の場合を考えることにする。式 (15) を線形制約の表現に書き換えると、

$$\begin{cases} \text{minimize} & f(\lambda) \\ \text{subject to} & \mathbf{a}_i^T \lambda = b_i \quad i \in W_k \\ & \mathbf{a}_i^T \lambda < b_i \quad i \notin W_k \end{cases} \quad (16)$$

いま、 $W_k$  の要素数が  $q$ 、つまり有効な制約式が  $q$  本あるものとする。 $\nabla f(\lambda)^T d < 0$  となるような  $d$  を求めれば、 $f$  を減少させる方向となる。ただし、有効な制約は保たれたままでなければならない、 $\forall i \in W_k$  で  $\mathbf{a}_i^T d = 0$  となる必要がある。具体的には、負の勾配  $-\nabla f(\lambda)$  を Working Set で定義された接平面上に射影することで  $d$  を求めることができる。

では  $d$  を計算しよう。 $W_k$  に対応する  $d$  を  $d_k$  とおく。 $\{i_1, i_2, \dots, i_q\} \in W_k$  に対して、

$$A_q = \begin{pmatrix} \mathbf{a}_{i_1}^T \\ \mathbf{a}_{i_2}^T \\ \vdots \\ \mathbf{a}_{i_q}^T \end{pmatrix}$$

と定義する。1 次独立性を仮定し、 $\text{rank } A_q = q$  とする。すると接空間  $M$  に垂直な部分空間  $N$  の任意のベクトル  $y$  は

$$y = A_q^T \mu_k \quad \text{for } \mu_k \in E^q$$

と表現出来る。負の勾配はこの 2 つの直交補空間のベクトルを用いて分解され、

$$-\nabla f(\lambda_k) = d_k + A_q^T \mu_k \quad (17)$$

と書ける。(ただし、 $d_k \in M, \mu_k \in E^q$ 。)  $A_q d_k = 0$  を利用して、

$$\begin{aligned} A_q d_k &= -A_q \nabla f(\lambda_k) - (A_q A_q^T) \mu_k = 0 \\ \therefore \mu_k &= -(A_q A_q^T)^{-1} A_q \nabla f(\lambda_k) \end{aligned}$$

式 (17) に代入し、

$$d_k = -[I - A_q^T (A_q A_q^T)^{-1} A_q] \nabla f(\lambda_k) = -P_k \nabla f(\lambda_k)$$

ここで、行列

$$P_k = [I - A_q^T (A_q A_q^T)^{-1} A_q]$$

を射影行列 (projection matrix) という。

$d_k \neq 0$  の時、 $d_k$  が降下方向となることは簡単に示せる。 $d_k$  と  $\nabla f(\lambda_k) + d_k$  は直交するので、

$$\nabla f(\lambda_k)^T d_k = (\nabla f(\lambda_k) + d_k^T - d_k^T) d_k = -|d_k|^2$$

となるからである。

降下方向が求まったら、ステップ幅を決める。まず実行可能領域をはみださない最大の  $\alpha_1$  を定め、その中で、 $f(\lambda + \alpha d)$  を最小化する  $\alpha$  を求めれば良い。もし  $\alpha = \alpha_1$  ならば、新しく (少なくとも1つの) 制約が有効になるので、その式に対応する添字を Working Set に追加する。そして新しい Working Set で (16) を解けばよい。

$d_k = 0$  の時、(17) は、

$$\nabla f(\lambda_k) + \mu_k^T A_q = 0 \quad (18)$$

となり、(14) の4番目の (\*) 以外を全て満たしていることになる。(18) で求まる  $\mu_k$  が全て非負であれば、(\*) も満たすことになり、 $\lambda_k$  は、凸2次計画問題の場合は元問題の最適解となる。非凸計画問題の場合は元問題の停留点になるが、一般に最適解を求めるのは困難なため、停留点を出力して反復終了となる。 $\mu_k$  の中に負の要素があれば、その中で最も値の小さい要素を探し、対応する制約の添字を Working Set から外す。そして新しい Working Set で (16) を解く。では新しく求まる  $d$  を  $\bar{d}_k$  とおくと、 $\bar{d}_k$  は

- 降下方向になっている
- 実行可能領域をはみ出さない

の2つを満たしているだろうか。

$\mu_{jk} = \min\{\mu_{1k}, \mu_{2k}, \dots, \mu_{qk}\}$  とする。つまり Working Set から  $j$  を外す場合を考える。 $\mu_k$  から  $\mu_{jk}$  を除いたものを  $\bar{\mu}_k$ 、 $A_q$  から  $a_j$  列を除いたものを  $A_{\bar{q}}$  と定義する。このとき、

$$-\nabla f(\lambda_k) = A_q^T \mu_k \quad (19)$$

$$-\nabla f(\lambda_k) = \bar{d}_k + A_{\bar{q}}^T \bar{\mu}_k \quad (20)$$

と書ける。 $\bar{d}_k = 0$  と仮定すると、(19)、(20) より、

$$\begin{aligned} A_q^T \mu_k &= A_{\bar{q}}^T \bar{\mu}_k \\ \sum_{\substack{i=1 \\ i \neq j}}^q \mathbf{a}_i (\mu_{ik} - \bar{\mu}_{ik}) + \mathbf{a}_j \mu_{jk} &= 0 \\ \Rightarrow \mu_{jk} &= 0 \quad (\because \text{rank } A_q = q) \end{aligned}$$

これは  $\mu_{jk} < 0$  に矛盾する。よって  $\bar{d}_k \neq 0$ 。このとき、降下方向になることは既に示した。また、(19) の転置の両辺に  $\bar{d}_k$  をかけ、 $A_{\bar{q}} \bar{d}_k = 0$  を利用すると、

$$0 > \nabla f(\lambda_k)^T \bar{d}_k = -\mu_{jk} \mathbf{a}_j^T \bar{d}_k$$

$\mu_{jk} < 0$  より、 $\mathbf{a}_j^T \bar{d}_k < 0$ 。よって  $\bar{d}_k$  は実行可能領域をはみ出さない。以上より  $\bar{d}_k$  は降下方向として適切であることが示せた。

降下方向の導出に勾配射影法を利用した場合の有効制約法のアルゴリズムは以下ようになる。

実行可能解  $\lambda$  が与えられているものとする。

**Step 1:**

Working Set  $W$ 、 $A_q$  を作る。

**Step 2:**

射影行列  $P = I - A_q^T(A_q A_q^T)^{-1}A_q$ 、 $d = -P\nabla f(\lambda)$  を計算する。

**Step 3:**

$d \neq 0$  ならば、それぞれ

$$\max\{\alpha : \lambda + \alpha d \text{ が実行可能}\}$$

$$\min\{f(\lambda + \alpha d) : 0 \leq \alpha \leq \alpha_1\}$$

を達成する  $\alpha_1$ 、 $\alpha_2$  を求め、解を  $\lambda \rightarrow \lambda + \alpha_2 d$  として Step 1 に戻る。 $\alpha_2 = \alpha_1$  ならば、新たに有効となる式に対応する添字を Working Set に追加する。

**Step 4:**

$d = 0$  ならば、 $\mu = -(A_q A_q^T)^{-1}A_q \nabla f(\lambda)$  を計算する。

a)  $\mu \geq 0$  であれば、反復終了。

b) そうでないならば、 $\mu$  の中で最も値の小さい要素に対応する列を  $A_q$  から外し、(同様に Working Set  $W$  からも対応する添字を外し、) Step 2 に戻る。

### 3.4 凸 2 次計画問題に対する有効制約法のアルゴリズム

3.3 節で述べた有効制約法を凸 2 次計画問題に適用する上で、特筆すべき点を挙げる。3.3 節よりも問題をさらに具体化し、目的関数が凸な 2 次関数という条件を付け加える。

$$\begin{cases} \text{minimize} & c^T \lambda + \frac{1}{2} \lambda^T Q \lambda \quad (Q : \text{半正定値行列}) \\ \text{subject to} & a_i^T \lambda = b_i \quad i \in W \quad \dots \textcircled{1} \\ & a_i^T \lambda \leq b_i \quad i \notin W \quad \dots \textcircled{2} \end{cases}$$

凸 2 次計画問題の性質から、降下方向として求めた  $d$  の方向に解を進めていけば、最適解に到達するまで常に目的関数が減少する。よってステップ幅  $\alpha$  を求める際、実行可能領域をはみださない最大の  $\alpha_1$  と、最適解に到達するようなステップ幅  $\alpha_2$  を比較し、より値の小さい方を選べばよい。 $\lambda' = \lambda + \alpha_1 d$  を ①、② に代入すると、

$$a_i^T (\lambda + \alpha_1 d) = b_i \quad i \in W$$

$$a_i^T (\lambda + \alpha_1 d) \leq b_i \quad i \notin W$$

$i \in W$  に関しては、 $a_i^T d = 0$  なので、① は成立。よって、実行可能領域をはみださない  $\alpha_1$  は

$$\alpha_1 = \min_{i \notin W} \left\{ \frac{b_i - a_i^T \lambda}{a_i^T d} \mid a_i^T d > 0 \right\}$$

で与えられる。また、 $\alpha_2$  は、

$$\alpha_2 = \arg \min_{\alpha} \{f(\lambda + \alpha d)\}$$

を解けばよい。 $\alpha_2$  の 2 次関数最小化なので、 $\frac{d}{d\alpha_2}f(\lambda + \alpha_2 d) = 0$  を解き、

$$\alpha_2 = -\frac{\lambda^T Q d}{d^T Q d} - \frac{c^T d}{d^T Q d} \quad (\text{ただし、} d^T Q d \neq 0)$$

が得られる。 $d^T Q d = 0$  のときは目的関数が  $\alpha$  の 1 次関数となる。 $\alpha > 0$  より、傾き  $= c^T \lambda + \lambda^T Q d < 0$ 。よって最小化を実現する  $\alpha$  は  $\alpha_2 = +\infty$  となるので、 $\alpha_1$  が採用される。また、本研究で扱う問題 (10) のように等式制約が存在している場合は、等式制約が有効な制約として常に保たれている必要がある。ただし、最適解での等式制約に対応するラグランジュ乗数は非負でなくてよいので、不等式制約に対応するラグランジュ乗数の中で負の要素があるかどうかをチェックすればよい。

これらを踏まえると、凸 2 次計画問題に対する有効制約法のアルゴリズムは以下のようになる。

### 凸 2 次計画問題に対する有効制約法のアルゴリズム

実行可能解  $\lambda$  が与えられているものとする。

**Step 1:**

Working Set  $W$ 、 $A_q$  を作る。

**Step 2:**

射影行列  $P = I - A_q^T (A_q A_q^T)^{-1} A_q$ 、 $d = -P(c + Q\lambda)$  を計算する。

**Step 3:**

$d \neq 0$  ならば、

$$\alpha_1 = \min_{i \notin W} \left\{ \frac{b_i - a_i^T \lambda}{a_i^T d} \mid a_i^T d > 0 \right\}$$

$$\alpha_2 = \begin{cases} -\frac{\lambda^T Q d}{d^T Q d} - \frac{c^T d}{d^T Q d} & (d^T Q d \neq 0) \\ +\infty & (d^T Q d = 0) \end{cases}$$

$\alpha = \min\{\alpha_1, \alpha_2\}$  と取り、 $\lambda \rightarrow \lambda + \alpha d$  として Step 1 に戻る。 $\alpha = \alpha_1$  ならば、新たに有効となる式に対応する添字を Working Set に追加する。

**Step 4:**

$d = 0$  ならば、 $\mu = -(A_q A_q^T)^{-1} A_q (c + Q\lambda)$  を計算する。

- a) 元問題の不等式制約に対応する  $\mu_i$  に対し、 $\mu_i \geq 0$  であれば、反復終了。 $\lambda$  は最適解。
- b) そうでないならば、元問題の不等式制約に対応する  $\mu_i$  の中で最も値の小さい要素に対応する列を  $A_q$  から外し、(同様に Working Set  $W$  から対応する添字を外し、) Step 2 に戻る。

## 4 凸2次計画問題に対する有効制約法の実装

前章で述べた有効制約法を MATLAB で実装した。本研究で扱う凸2次計画問題は様々な性質を持っている。例えば、 $\beta_1 < \beta_2$  のとき、 $\beta_1$  での最適解が  $\beta_2$  での実行可能解になっている。また、パラメータ  $\beta$  を動かしても2本の等式制約は変化せず、その係数は +1 または -1 のいずれかである。これらの性質を利用して、初期解をうまく選んだり、計算量を減らすことが可能である。本章では実装する上で工夫した点について述べる。なお (10) は、最小化問題に定式化しなおし、 $\lambda_i$  の範囲は上記のように  $2m$  本の不等式制約とみなして

$$\left\{ \begin{array}{l} \text{minimize} \\ \lambda \end{array} \sum_{i=1}^m \sum_{j=1}^m y_i y_j \mathbf{x}_i^T \mathbf{x}_j \lambda_i \lambda_j \right. \\ \left. \begin{array}{l} \text{subject to} \\ \sum_{i=1}^m \lambda_i = 1 \\ \sum_{i=1}^m \lambda_i y_i = 0 \\ -\lambda_i \leq 0, \lambda_i \leq \frac{1}{(1-\beta)m} \quad i = 1, 2, \dots, m \end{array} \right.$$

の形に変えてから実装した。

### 4.1 初期実行可能解の選び方

まず、初期実行可能解を自分で設定する必要がある。ここでは初期解の選び方として2つの方法を挙げた。パラメータを  $\beta = 0$  と仮定すると、制約式は

$$\begin{aligned} \sum_{i=1}^m \lambda_i y_i &= 0 \\ \sum_{i=1}^m \lambda_i &= 1 \\ 0 \leq \lambda_i &\leq \frac{1}{m} \quad i = 1, 2, \dots, m \end{aligned}$$

となる。 $+1, -1$  のラベルに対応する変数ベクトルを  $\lambda_+, \lambda_-$ 、データ数を  $m_+, m_-$  とすると、

$$\lambda_+ = \frac{e}{2m_+}, \lambda_- = \frac{e}{2m_-} \quad (21)$$

は任意の  $0 < \beta < 1$  について実行可能となるので、初期実行可能解である。これを初期解  $\textcircled{A}$  と呼ぶことにする。数値実験の際この初期解を使用する。

しかし、(21) を初期解として設定すると、有効な制約式が1本も存在しない状態から反復を開始するので、1反復ごとに制約境界に達し有効な制約を1本追加していくとしても、データ分の反復回数は必要になってしまう。そこで与えられたデータから、仮の判別平面  $w_0^T x + b_0 = 0$  を引き、それを基準にデータを分ける。具体的には、仮の判別平面は各クラスのデータの平均をちょうど真ん中を通る平面を考えた。



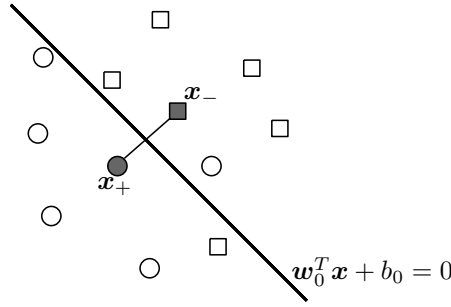


図 3 : 初期実行可能な判別平面  $w_0^T x + b_0 = 0$

ラベルが  $+1, -1$  のデータの平均を取り、それを

$$x_+ = \frac{1}{m_+} \sum x_i, \quad x_- = \frac{1}{m_-} \sum x_i$$

としよう。  $x_+, x_-$  を用いて  $w_0$  は

$$w_0 = \frac{x_+ - x_-}{\|x_+ - x_-\|}$$

と表される。  $x_+, x_-$  の中点はこの判別平面上を通るので、

$$w_0^T \frac{(x_+ + x_-)}{2} + b_0 = 0 \implies b_0 = -\frac{w_0^T (x_+ + x_-)}{2}$$

から  $b_0$  が求まる。この判別平面を基準として損失分布を考え、損失の大きいデータから順に  $\frac{1}{(1-\beta)m}$  を割り当てていく。(21) の 2 つの等式制約は + クラスと - クラスの変数の総和がそれぞれ  $\frac{1}{2}$  ずつであることを意味している。この制約を満たさなければならないので、まず、+ クラスのデータで損失の大きい順に  $\lfloor \frac{(1-\beta)m}{2} \rfloor$  番までの変数に  $\frac{1}{(1-\beta)m}$  を割り当て、その次の順番のデータには端数である  $\frac{1}{2} - \frac{1}{(1-\beta)m} \lfloor \frac{(1-\beta)m}{2} \rfloor$  を割り当てる。- クラスのデータにも同様の操作を施せば、制約を実現できる。このようにして得られた初期実行可能解を初期解 ㉓ と呼ぶことにする。

## 4.2 最適解情報からのリスタート

最初に設定したパラメータ  $\beta_1$  での最適解を求めてしまえば、その最適解をパラメータを  $\beta_2 (> \beta_1)$  に動かした問題に対して初期実行可能解として利用出来る。これを初期解 ㉔ と呼ぶことにする。CGS model の損失分布の考え方に戻ると、  $\alpha_\beta$  を少し大きくすることは、点  $\alpha_\beta$  を少しだけ右にずらし、その  $\alpha_\beta$  を超える損失の平均を最小化するということである。よって、最適な判別平面は大幅に変わらないことが予想され、  $\beta_1$  で正しく判別出来ていたデータ  $x_i$  はパラメータが  $\beta_2$  に変更されても正しく判別される可能性が高いと考えてよいだろう。また  $x_i$  が正しく判別されている場合、対応する変数は  $\lambda_i^* = 0$  となる可能性が高いことが CGS model の考え方から言える。そこで、  $\lambda_1^*$  のうち、値が 0 の要素は 0 のままにし、0 でない要素は  $\beta_1$  での最適な判別平面  $w_0^T x + b_0 = 0$  を基準として、初期解 ㉓ と同様の手順で  $\frac{1}{(1-\beta_2)m}$  を割り当てていくことを考えた。このようにして得られたパラメータ  $\beta_2$  での初期実行可能解を初期解 ㉕ と呼ぶことにする。

## 4.3 逆行列を使わない計算方法

有効制約法のアルゴリズムでも示した通り、降下方向  $d$  を求める際、  $A_q A_q^T$  の逆行列が必要になる。  $A_q A_q^T$  は  $q \times q$  行列であり、MATLAB の逆行列を求めるコマンド `inv` を使用すると、行列のサイズが大きいため

計算時間がかかってしまうので、逆行列を使わない計算方法を考える。

降下方向  $d$  およびラグランジュ乗数  $\mu$  は

$$\begin{aligned} d &= -[I - A_q^T(A_q A_q^T)^{-1}A_q](c + Q\lambda) \\ &= -(c + Q\lambda) + A_q^T(A_q A_q^T)^{-1}A_q(c + Q\lambda) \\ \mu &= -(A_q A_q^T)^{-1}A_q(c + Q\lambda) \end{aligned}$$

であり、

$$(A_q A_q^T)^{-1}A_q(c + Q\lambda) = z \implies (A_q A_q^T)z = A_q(c + Q\lambda)$$

ここで、 $\forall u \in R^q$  に対して、 $u^T A_q A_q^T u = \|A_q^T u\|^2 \geq 0$ 。rank  $A_q = q$  より、 $u \neq 0$  ならば、 $\|A_q^T u\|^2 > 0$ 。よって、 $A_q A_q^T$  は正定値行列である。行列が正定値ならばコレスキー分解が可能である。つまり、上三角行列  $L$  を用いて

$$\begin{aligned} (A_q A_q^T)z = A_q(c + Q\lambda) &\implies L^T L z = A_q(c + Q\lambda) \\ &\implies \begin{cases} L^T u = v (= A_q(c + Q\lambda)) \\ L z = u \end{cases} \end{aligned} \quad (22)$$

と分解出来るので、2段階で  $z$  を求めればよい。

(4) の不等式制約の係数行列  $A$  は

$$A = \begin{pmatrix} -1 & 0 & \cdots & 0 \\ 0 & -1 & & \\ \vdots & & \ddots & \\ 0 & & & -1 \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \end{pmatrix}$$

であり、 $q$  本の有効制約の係数行列  $A_q$  は  $A$  の中の  $q-2$  本と等式制約の2本で構成される。よって、

$$A_q A_q^T = \left( \begin{array}{c|c} I_{q-2} & B \\ \hline B^T & C \end{array} \right)$$

である。ただし、 $B = (b_{ij}) : (q-2) \times 2$  行列、 $b_{ij} \in \{\pm 1\}$ 、 $C : 2 \times 2$  行列。上三角行列  $L$  も  $L_1 = (l_{ij}^1) : (q-2) \times (q-2)$  行列、 $L_2 = (l_{ij}^2) : (q-2) \times 2$  行列、 $L_3 = (l_{ij}^3) : 2 \times 2$  行列を用いて

$$L = \left( \begin{array}{c|c} L_1 & L_2 \\ \hline \mathbf{0} & L_3 \end{array} \right)$$

と表され、 $A_q A_q^T = L^T L$  より、

$$\begin{aligned} \left( \begin{array}{c|c} I_{q-2} & B \\ \hline B^T & C \end{array} \right) &= \left( \begin{array}{c|c} L_1 & L_2 \\ \hline \mathbf{0} & L_3 \end{array} \right) \\ \therefore \begin{cases} L_1 & = I_{q-2} \\ L_2 & = B \\ L_3^T L_3 & = C - B^T B \end{cases} \end{aligned}$$

(22) に代入して整理すると、

$$\begin{pmatrix} L_1^T & \mathbf{0} \\ L_2^T & L_3^T \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_q \end{pmatrix} = \begin{pmatrix} v_1 \\ \vdots \\ v_q \end{pmatrix} \implies \begin{cases} (u_1, \dots, u_{q-2})^T = (v_1, \dots, v_{q-2})^T \\ l_{11}^3 u_{q-1} = v_{q-1} - \sum_{j=1}^{q-2} l_{j1}^2 u_j \\ l_{12}^3 u_{q-1} + l_{22}^3 u_{q-2} = v_q - \sum_{j=1}^{q-2} l_{j2}^2 u_j \end{cases}$$

$$\begin{pmatrix} L_1 & L_2 \\ \mathbf{0} & L_3 \end{pmatrix} \begin{pmatrix} z_1 \\ \vdots \\ z_q \end{pmatrix} = \begin{pmatrix} u_1 \\ \vdots \\ u_q \end{pmatrix} \implies \begin{cases} l_{22}^3 z_q = u_q \\ l_{11}^3 z_{q-1} + l_{12}^3 z_q = u_{q-1} \\ z_i + \sum_{j=1}^2 l_{ij}^2 u_{q-2+j} = u_i \quad (i = 1, 2, \dots, q-2) \end{cases}$$

の順で  $z_i$  を求めることが出来る。有効制約の本数が変わり  $A_q$  の行が追加または削除された時、 $L_1$  は単位行列のままであり、 $L_2 (= B)$  は  $A_q$  中の変化した行だけを見れば容易に求められる。 $L_3$  はすぐ形がわからないが、 $2 \times 2$  行列のコレスキー分解で求まることは変わらない。結局、毎反復  $2 \times 2$  行列のコレスキー分解と基本演算だけで  $d$  や  $\mu$  を求めることが可能である。

#### 4.4 一部の変数を固定する

前節では逆行列を使用しないことによって計算時間の短縮を図れることを示したが、それでもデータ数が多いときは行列のサイズが大きくなるので、1回の計算で時間はかかってしまう。そこで、一部の変数を固定して、サイズを小さくする方法を考える。

(10) の実行可能解  $\lambda^*$  が最適解であることは、以下を満たすような  $w^*$ 、 $b^*$ 、 $\alpha^*$  が存在することと必要十分である。

$$\lambda_i^* [z_i^* + y_i(w^{*T} \mathbf{x}_i + b^*) + \alpha^*] = 0 \quad (23)$$

$$z_i^* \left[ \frac{1}{(1-\beta)m} - \lambda_i^* \right] = 0 \quad (24)$$

$$\sum_{i=1}^m \lambda_i^* y_i = 0$$

$$\sum_{i=1}^m \lambda_i^* = 1$$

$$z_i^* \geq 0$$

$$0 \leq \lambda_i^* \leq \frac{1}{(1-\beta)m}$$

$$z_i^* + y_i(w^{*T} \mathbf{x}_i + b^*) + \alpha^* \geq 0$$

(23)、(24) 以外の式は実行可能解ならば必ず成立する条件なので、 $\lambda^*$  が最適解であることは、(23) かつ (24) を満たす  $w^*$ 、 $b^*$ 、 $\alpha^*$  が存在することと必要十分になる。

①  $\lambda_i^* = 0$  のとき

(24) より、 $z_i^* = 0$ 。このとき、

$$z_i^* + y_i(w^{*T} \mathbf{x}_i + b^*) + \alpha^* \geq 0 \iff \underline{y_i(w^{*T} \mathbf{x}_i + b^*) + \alpha^* \geq 0}$$

②  $\lambda_i^* = \frac{1}{(1-\beta)m}$  のとき

(24) より、

$$\begin{aligned} z_i^* + y_i(\mathbf{w}^{*T} \mathbf{x}_i + b^*) + \alpha^* = 0 &\iff z_i^* = -y_i(\mathbf{w}^{*T} \mathbf{x}_i + b^*) + \alpha^* \geq 0 \\ &\iff \underline{y_i(\mathbf{w}^{*T} \mathbf{x}_i + b^*) + \alpha^* \leq 0} \end{aligned}$$

③  $0 < \lambda_i^* < \frac{1}{(1-\beta)m}$  のとき

(23)、(24) から、

$$\begin{cases} z_i^* + y_i(\mathbf{w}^{*T} \mathbf{x}_i + b^*) + \alpha^* = 0 \\ z_i^* = 0 \end{cases} \iff \underline{y_i(\mathbf{w}^{*T} \mathbf{x}_i + b^*) + \alpha^* \leq 0}$$

まとめると、次の条件 (25) になる。

$$\begin{aligned} \textcircled{1} \lambda_i^* = 0 &\implies y_i(\mathbf{w}^{*T} \mathbf{x}_i + b^*) + \alpha^* \geq 0 \\ \textcircled{2} \lambda_i^* = \frac{1}{(1-\beta)m} &\implies y_i(\mathbf{w}^{*T} \mathbf{x}_i + b^*) + \alpha^* \leq 0 \\ \textcircled{3} 0 < \lambda_i^* < \frac{1}{(1-\beta)m} &\implies y_i(\mathbf{w}^{*T} \mathbf{x}_i + b^*) + \alpha^* = 0 \end{aligned} \quad (25)$$

また、(10) の最適解  $\lambda^*$  を用いて  $\mathbf{w}^*$ 、 $b^*$ 、 $\alpha^*$  を記述出来る。

$$\mathbf{w}^* = \frac{\sum_{i=1}^m \lambda_i^* y_i \mathbf{x}_i}{\delta} = \frac{\sum_{i=1}^m \lambda_i^* y_i \mathbf{x}_i}{\sqrt{\sum_{i=1}^m \sum_{j=1}^m y_i y_j \mathbf{x}_i^T \mathbf{x}_j \lambda_i^* \lambda_j^*}}$$

であり、 $0 < \lambda_i^* < \frac{1}{(1-\beta)m}$  であるような要素を各クラス  $s (\geq 1)$  個ずつ選び、連立方程式

$$\begin{aligned} \sum_{\substack{i=1 \\ \mathbf{x} \in S_+}}^s \mathbf{w}^{*T} \mathbf{x}_i + sb + s\alpha^* &= 0 \\ \sum_{\substack{j=1 \\ \mathbf{x} \in S_-}}^s \mathbf{w}^{*T} \mathbf{x}_j + sb - s\alpha^* &= 0 \end{aligned}$$

を解くことで  $b^*$ 、 $\alpha^*$  が求まる。

この関係式は最適解の場合しか成立しない。しかし、ある実行可能解  $\lambda$  から上の関係式を利用して  $w$ 、 $b$ 、 $\alpha$  を求め、それが (25) を満たす場合には  $\lambda, w, b, \alpha$  は最適解として保証される。そこで、実行可能解  $\lambda$  から  $w, b, \alpha$  を求め、 $\lambda$  を (25) を満たしている要素と満たしていない要素に分ける。満たしている変数は定数として固定し、満たしていない要素のみを変数として扱えば、問題のサイズを小さくすることが出来る。 $m$  個の要素のうち、 $l$  個が満たさなかったと仮定し、それを順番に  $(\lambda_1, \lambda_2, \dots, \lambda_l)$ 、 $\lambda_i$  に対応するデータとラベルを  $\mathbf{x}_i, y_i$  とする。このとき、(4) は、

$$\left\{ \begin{array}{l} \underset{\lambda}{\text{minimize}} \quad 2 \sum_{i=1}^l \sum_{j=l+1}^m y_i y_j \mathbf{x}_i^T \mathbf{x}_j \lambda_i \lambda_j + \sum_{i=1}^l \sum_{j=1}^l y_i y_j \mathbf{x}_i^T \mathbf{x}_j \lambda_i \lambda_j \\ \text{subject to} \quad \sum_{i=1}^l \lambda_i = 1 - \sum_{i=l+1}^m \lambda_i \\ \sum_{i=1}^l \lambda_i y_i = - \sum_{i=l+1}^m y_i \lambda_i \\ -\lambda_i \leq 0, \lambda_i \leq \frac{1}{(1-\beta)m} \quad i = 1, 2, \dots, l \end{array} \right. \quad (26)$$

と変数が  $l$  個の凸 2 次計画問題に書き換えられる。前章で述べた有効制約法のアルゴリズムを (26) に適用し、定数として固定していた解と得られた解を合わせて新たな  $\lambda$  を作る。そこからまた新たな  $w, b, \alpha$  を求め、同様の操作を条件を満たすまで繰り返す。降下方向を取って有効制約法のアルゴリズムを動かしている限り、必ず目的関数値は減少していくので、操作を続けていけば必ず (25) を満たすような最適解まで辿り着く。

—— 一部の変数を固定するアルゴリズム ——

実行可能解  $\lambda$  が得られているとする。

**Step 1:**

関係式から  $w, b, \alpha$  を復元する。

**Step 2:**

(25) を満たしていれば終了。そうでなければ満たしている要素と満たしていない要素に分ける。

**Step 3:**

満たしている要素は固定。満たしていない要素を変数とみなし、(26) を解く。

**Step 4:**

(26) の最適解と固定していた変数を合わせて新しい  $\lambda$  を作成し、Step 1 に戻る。

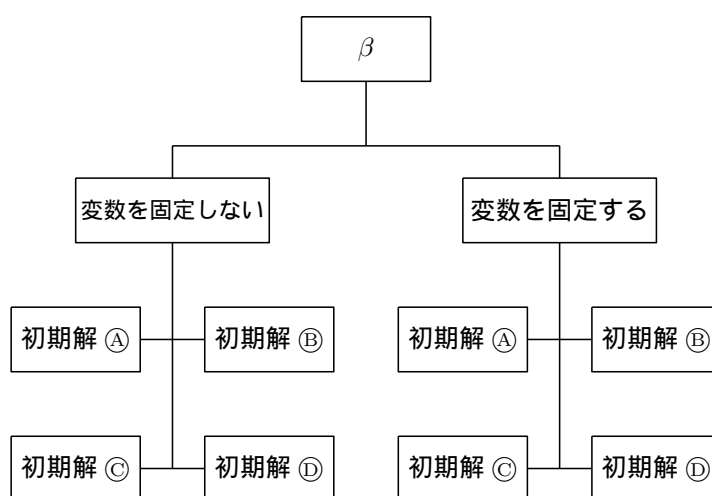
## 5 数値実験

有効制約法を MATLAB で実装し、(10) を解いた。数値実験を行った計算機環境は以下の通りである。

プログラミング言語 : MATLAB version 7.2  
CPU : dual Pentium 4 2.8GHz  
メモリ : 1GB

### 5.1 実験方法

初期実行可能解は、前章で述べた初期解 ①・初期解 ②・初期解 ③・初期解 ④ の 4 種類を使用する。それぞれの初期解に対し、一部の変数を固定する方法と固定しない方法を適用し、1 つのパラメータ  $\beta$  につき 8 つの場合で数値実験を行う。



$\beta$  の下限として、 $\beta_{min} = 1 - 2 \min\{\frac{m}{m_+}, \frac{m}{m_-}\}$  であることが知られている [2]。  $\beta_1 \geq \beta_{min}$  となる適当な初期パラメータ  $\beta_1$  を設定し、(10) の最適値 = 0 となるまで  $\beta$  を 0.5 ずつ増やしていく。 ③・④ は前の  $\beta$  での最適解が求まっていないと使用できないので、 $\beta = \beta_1$  に限り、初期解は ①・② の 2 種類とする。また、最適解の使いまわしによる効果を見るため、MATLAB 言語のソフトウェア SeDuMi を使用し、同様の問題を内点法でも解く。有効制約法で 2 次計画問題を解く MATLAB のコマンド quadprog があるが、比較にならないほどに計算時間がかかったので、実験結果は割愛する。  
計算時間は 10 cross-validation の平均を取る。

< n cross-validation >

データを  $n$  個のグループに分ける。その中から  $n-1$  個のグループのデータで最適解を出し、判別平面と判別率を求める。残り 1 つのグループのデータを判別平面で分類し、未知のデータに対する判別率を求める。この操作を  $n$  回繰り返し、平均を求める。判別平面を求めるために使用するデータを train data、未知のデータとして残されたデータを test data という。

実験用のデータセットは UCI repository から引用し、スケーリングを施して使用した。 $m$  はデータ数、 $n$  は特徴数である。例えば heart-scale は特徴数は 10 で、その内容は年齢、性別や血圧、血中コレステロール濃度などである。

## 5.2 実験結果

計算時間の単位は  $s$ (秒)、判別率は % である。以下のデータセットでは有効制約法と SeDuMi の判別率は全て一致した。

表：heart-scale( $m=270$ 、 $n=13$ )

$\beta$	変数を固定しない				一部の変数を固定する				SeDuMi	判別率 (train)	判別率 (test)
	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓐ	Ⓑ	Ⓒ	Ⓓ			
0.3	0.638	<b>0.411</b>			0.691	<b>0.384</b>			0.860	83.951	82.593
0.35	0.667	0.600	0.603	<b>0.309</b>	0.686	0.552	0.533	<b>0.319</b>	0.858	84.239	82.593
0.4	0.672	0.642	0.586	<b>0.421</b>	0.698	0.627	0.506	<b>0.353</b>	0.860	84.486	82.963
0.45	0.666	0.684	0.541	<b>0.497</b>	0.639	0.697	0.444	<b>0.411</b>	0.849	85.021	82.593
0.5	0.688	0.811	<b>0.633</b>	0.634	0.716	0.860	<b>0.534</b>	0.614	0.852	85.350	84.074
0.55	0.749	0.935	0.745	<b>0.558</b>	0.767	0.963	<b>0.636</b>	0.736	0.866	85.514	83.333
0.6	0.836	1.181	<b>0.906</b>	1.165	0.848	1.275	<b>0.747</b>	1.190	0.868	85.350	83.333

表：diabetes-scale( $m=768$ 、 $n=8$ )

$\beta$	変数を固定しない				一部の変数を固定する				SeDuMi	判別率 (train)	判別率 (test)
	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓐ	Ⓑ	Ⓒ	Ⓓ			
0.35	1.362	<b>1.014</b>			1.370	<b>0.834</b>			1.190	77.127	77.097
0.4	1.380	1.234	1.300	<b>1.191</b>	1.339	1.153	0.997	<b>0.963</b>	1.193	77.677	76.969

表：breast-cancer-scale( $m=680$ 、 $n=10$ )

$\beta$	変数を固定しない				一部の変数を固定する				SeDuMi	判別率 (train)	判別率 (test)
	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓐ	Ⓑ	Ⓒ	Ⓓ			
0.4	1.184	<b>1.513</b>			1.232	<b>1.180</b>			1.128	94.518	94.574
0.45	1.209	1.438	0.993	<b>0.645</b>	1.239	1.130	0.819	<b>0.533</b>	1.098	95.038	95.015
0.5	1.308	1.503	0.989	<b>0.775</b>	1.309	1.136	0.842	<b>0.606</b>	1.120	95.25	95.309
0.55	1.206	1.263	0.873	<b>0.672</b>	1.239	1.033	0.755	<b>0.500</b>	1.123	95.526	95.603
0.6	1.170	1.195	0.842	<b>0.654</b>	1.283	1.041	0.723	<b>0.498</b>	1.109	95.566	95.750
0.65	1.283	1.167	0.881	<b>0.678</b>	1.300	0.955	0.716	<b>0.463</b>	1.160	96.193	96.046
0.7	1.245	1.033	0.795	<b>0.684</b>	1.254	0.930	0.619	<b>0.466</b>	1.152	96.34	96.194
0.75	1.531	1.030	1.084	<b>0.686</b>	1.494	0.898	0.719	<b>0.467</b>	1.153	96.535	96.635
0.8	1.241	0.944	<b>0.723</b>	0.733	1.255	0.861	<b>0.445</b>	0.492	1.166	96.763	96.635
0.85	1.200	0.798	<b>0.645</b>	0.764	1.241	0.767	<b>0.297</b>	0.561	1.166	97.007	97.074
0.9	1.525	1.300	1.038	<b>0.978</b>	1.458	0.866	<b>0.530</b>	0.741	1.194	97.267	96.925

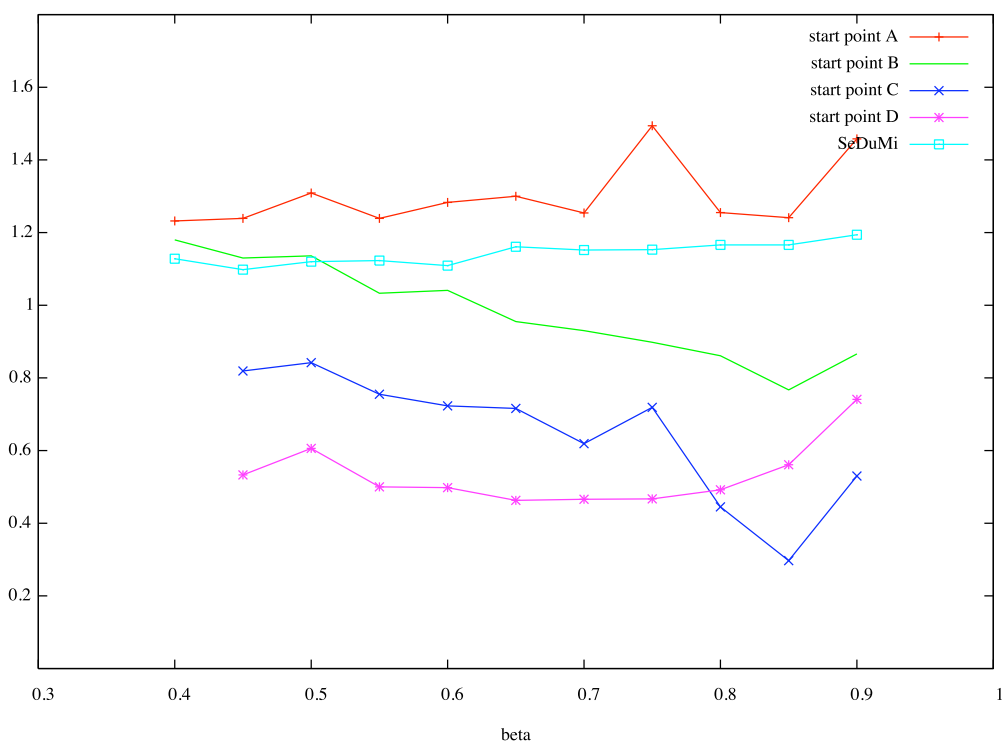
### 5.3 考察

5.2 節の実験結果から言えることに対して、考察を行う。

最適解の使いまわしをした方が計算時間は短くなる。

初期解 ④、⑤ を選ぶより、初期解 ③、⑥ を選んだ方が計算時間が短くなっていることが表から見て取れる。この結果は一部の 변수を固定する方法としない方法どちらに対しても言える。次のグラフは breast-cancer-scale で一部の 변수を固定した場合と SeDuMi の計算時間をまとめたものである。

一部の 변수を固定した場合の計算時間比較グラフ



内点法は、初期点を適当に定めパスを追跡して最適解まで辿り着く手法である。SeDuMiを使用する際、自分で初期解を与えていないので、SeDuMiで定めた内点から、各パラメータ  $\beta$  に対し類似した問題を1から解き直していることになる。計算時間が  $\beta$  を動かしてもほぼ変わらないのはそのためである。SeDuMiと比べて初期解 ③、⑥ とともに計算時間が減っているのは、前の最適解を利用することによる効果が確かめられたことになるだろう。

一方、初期解 ④、⑤ のどちらが適しているかを見比べることは出来なかった。一見すると初期解 ⑤の方が良さそうに思えるが、仮の判別平面を基準とした間違っただり付けがある場合、それらは1度有効な制約から外されない限り反復は終了しないので、もし間違っただり付けが多い場合には、有効制約が1本も存在しない状態からスタートする初期解 ④の方が、結果的に反復が早く終わると考えるからである。間違っただり付けが少なくなるような基準があれば、初期解 ⑤の方が適切かもしれない。



一部の変数を固定する方が計算時間が短くなっている。

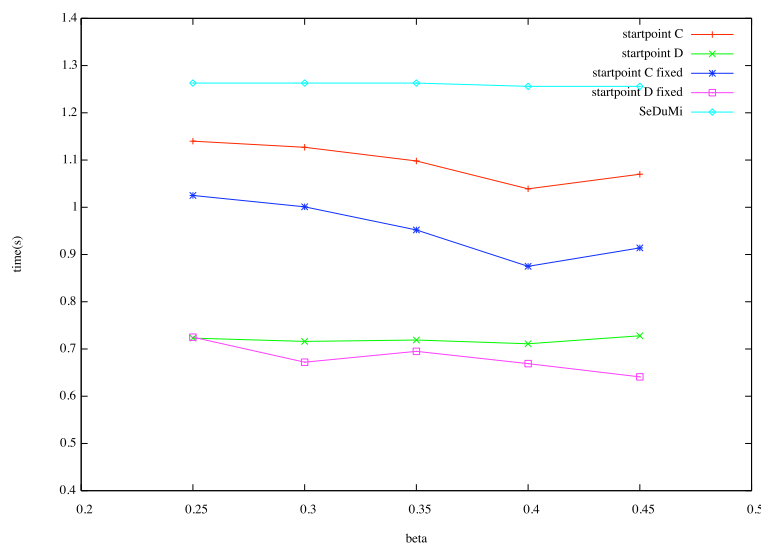
一部の変数を固定する方法と固定しない方法を同じパラメータ  $\beta$  で比較すると、変数を固定する方が計算時間が短くなっている。あるパラメータ  $\beta$  でデータ  $x_i$  が正しく判別される場合には、 $\beta$  を少し大きくしても  $x_i$  は正しく判別される可能性が高い。(25) を見ると、対応する変数は  $\lambda_i^* = 0$  となる可能性が高いことがわかる。よってこのような変数は定数として固定されやすく、問題のサイズが小さくなったと考えられる。しかし、最適解で  $\lambda_i^* = 0$  となる変数が P.20 のアルゴリズムでずっと定数に固定されているかはわからない。間違っただ変数を固定してしまうと、反復回数は多くなる。問題のサイズが小さくなり、1 反復あたりの計算時間が抑えられたことで結果的に総合の計算時間が短くなったケースもあったので、変数を固定する方法がどのような条件で効果的かを今後分析する必要がある。

実験結果を見ると、パラメータ  $\beta$  が小さければ初期解 ㉔ が優位で、 $\beta$  が大きければ初期解 ㉕ が優位になっている。毎反復での目的関数値の推移を追っていくと、㉕ は最適値付近まで値が更新されたあとほとんど値が変化せず、しばらく反復を続けたのちに従来の終了条件を満たして反復が終了していた。を大きくしていくと ㉕ が ㉔ より計算時間がかかってしまう原因としてそれが挙げられる。そこで目的関数値の差が  $10^{-10}$  以下になれば終了という条件を新たに追加して実験した。次の表とグラフは australian-scale に対しての実験結果である。今回は ㉔、㉕ の比較を見るため ㉖、㉗ の結果は省いている。

表：australian-scale(m=690、n=14)

$\beta$	変数を固定しない		変数を固定する		SeDuMi
	㉔	㉕	㉔	㉕	
0.25	1.140	<b>0.723</b>	1.025	<b>0.725</b>	1.263
0.3	1.127	<b>0.716</b>	1.001	<b>0.672</b>	1.263
0.35	1.098	<b>0.719</b>	0.952	<b>0.695</b>	1.263
0.4	1.039	<b>0.711</b>	0.875	<b>0.669</b>	1.256
0.45	1.070	<b>0.728</b>	0.914	<b>0.641</b>	1.256

計算時間比較



このデータセットに対しては、変数を固定した効果はあまり見られない。しかし、初期解 ㉔ より初期解 ㉕の方が計算時間が短くなっていることがわかる。P.22 で示したその他のデータセットに対しても終了条件を緩めて実験すると、 を大きくしても初期解 ㉕の方が計算時間が短くなった。この結果から、初期解 ㉕の方が少ない反復回数で最適解付近へ辿り着くことが言えるだろう。なお、終了条件を緩めても SeDuMi との判別率の違いは 0.1% 以下であった。

## 6 まとめと今後の課題

CGS model 内の凸 2 次計画問題に対して有効制約法を実装し、問題の性質や最適解を利用して、効率的に計算する方法を提案した。今回実験した 4 種類のデータセットに共通して見られた結果は以下の通りである。

- 前の での最適解を次の で利用することで計算時間を短縮できた
- 初期解 ㉕ が最適解へと速く近づいた
- 変数を固定する方が計算時間が短かった

最適解を使いまわす効果が確かめられた。CGS model の凸 2 次計画問題は -SVM と等価なので、この考え方は -SVM にも適用出来るものと思われる。しかし、データ数の大きい問題ではプログラムの挙動がおかしくなってしまう、検証が出来ていない。より多くのデータセットでの数値実験・検証は今後の課題にしたい。

*Chin-Chung Chang, Chin-Jen Lin* によって開発された `libsvm` という SVM を解くためのソフトウェアがある [7]。MATLAB と C 言語の違いはあるが、`libsvm` のほうがかなり速い計算結果が出ている。`libsvm` では初期解の設定が出来ず、今回は最適解の使いまわしによる効果を見ることに重点を置いていたため、自分で記述したプログラムで実験を行ったが、`libsvm` で使われているテクニックなども実装し、有効制約法のアルゴリズムを改善することも今後の課題である。

その他の課題としては

- 最適解情報だけでなく最適基底などの情報を利用した効率化
- 一部分の変数を固定する方法が効果的となるような条件の分析

などが挙げられる。

## 謝辞

本論文が完成するに至るまでに、たくさんの方々に助けていただきました。この場を借りて御礼申し上げます。小島政和先生は数理計画法の基礎から1年間みっちりとお指導下さいました。理解の遅い私ですが今後も御指導宜しくお願い致します。助手の武田さんには本当にお世話になりました。研究テーマを決定することは勿論のこと、やるべきことをすぐ見失う私に対して、根気よく丁寧に解説していただきました。深く感謝致します。何とか本論文を書き上げることが出来ました。また、博士課程の脇さんには様々なことを教わりました。特に計算時間を減らすプログラムの実装方法についてはかなりの時間を割いていただきました。ありがとうございました。

私の研究の支えになってくれた7階の研究室の皆様、1年間ありがとうございました。最後に小島研の中村道彦君とはお互いを鼓舞しながら1年間楽しく研究を続けることが出来ました。大学院に進学してもさらに研究に勤しみましょう。

## 参考文献

- [1] B.Scholkopf,A.J.Smola,R.C.Williamson,and P.L.Bartlett : New Support Vector Algorithms ,*Neural Computation*,12:1207-1245,2000
- [2] J.Gotoh and A.Takeda : A Linear Classification Model Based On Conditional Geometric Score,*Yokohama Publishers*,2005.
- [3] 栗田多喜男 : サポートベクターマシン入門, 2002.
- [4] 今野浩 : 線形計画法, 日科技連, 1987.
- [5] 田村明久、村松政和 : 最適化法, 共立出版, 2002.
- [6] David.G.Luenberger : Linear and Nonlinear Programming Second Edition, *Kluwer Academic Publishers* , 2003.
- [7] Chin-Chung Chang and Chin-Jen Lin : a Library for Support Vector Machines,2006.