

スパース推定のための QUBO 形式の導出

指導教員 大久保 潤 准教授

平成 21 年 2 月 12 日提出

工学部情報システム工学科

15TI015

氏 名 横田 知大

埼玉大学 理工学研究科・工学部
大久保研究室
埼玉県さいたま市桜区下大久保 255

概要

近年ではアニーリングの専用マシンが開発され、最適化問題や機械学習などの様々な分野でその利用方法が考えられている。アニーリングの専用マシンでは扱う問題を QUBO (Quadratic Unconstrained Binary Optimization) 形式で表現し、それをハードウェアとして実装する必要がある。しかし、QUBO 形式へ書き換える一般的な方法は知られておらず、個別に導出を行わなければならない。本研究ではスパース推定で用いられる絶対値関数に対する QUBO 形式の導出を行う。実験の結果、絶対値関数を Legendre 変換および Wolfe の双対定理を用いて導出を行なったところ、数値実験の結果から変数を一つ減らせることが予測できたので、定式化を見直すことで実際に変数を一つ減らすことができた。また、人工データを用いて数値実験をおこない、スパースな解を得られることが確認できた。

謝辞

研究ならびに生活面においてご指導を賜りました大久保潤准教授に深く感謝致します。

また，先輩としていつもよきアドバイスをくださいました，佐藤豪氏，中村崇人氏をはじめとする研究室の皆様，そして同期学生の皆様，並びに私を暖かく見守って頂いた両親はじめとする周囲のすべての皆様に深く感謝致します。

目次

概要	i
謝辞	ii
図目次	v
第1章 はじめに	1
1.1 背景	1
1.2 目的	1
1.3 本論文の構成	2
第2章 序論	3
2.1 スパース推定	3
2.1.1 最小二乗法	3
2.1.2 L_2 正則化 (Ridge)	4
2.1.3 L_1 正則化 (Lasso)	4
2.2 シミュレーテッド・アニーリング	5
2.3 イジング模型	6
2.3.1 色彩問題のイジングの定式化	6
2.3.2 QUBO 形式	6
第3章 q -loss 関数の QUBO 形式での導出例	8

3.1	q -loss 関数の定義	8
3.2	Legendre 変換	9
3.3	q -loss 関数への Legendre 変換の適用	10
第 4 章	絶対値関数に対する QUBO 形式の導出	15
4.1	Legendre 変換での導出	15
4.2	Wolfe の双対定理	17
4.3	Wolfe の双対定理を利用した双対問題への変換	17
4.4	シミュレーテッドアニーリングを用いた数値実験	18
4.5	変数 t の消去	19
第 5 章	シミュレーテッドアニーリングを利用したスパース性の検証	22
5.1	人工データ	22
5.2	数値実験	22
第 6 章	おわりに	24
6.1	まとめ	24
6.2	今後の課題	24
	参考文献	25

目 次

3.1	q -loss 関数の概形	9
3.2	$f(m) = L_q(m) - m^2$ の概形	11
3.3	$g(\eta)$ の概形	12
3.4	$\tilde{g}(\eta)$ の概形	13
4.1	絶対値関数の概形	15
4.2	コストの概形	20
4.3	t の概形	20
4.4	z_1 の概形	20
4.5	z_2 の概形	20
4.6	t を取り除いたコストの概形	21
4.7	t を取り除いた z_1 の概形	21
4.8	t を取り除いた z_2 の概形	21

第1章 はじめに

1.1 背景

近年、D-Wave System の D-Wave[1], 富士通のデジタルアニーラ [2], 日立の CMOS アニーリングマシン [3] などアニーリング専用マシンが開発され, 様々な分野での利用方法が考えられている. アニーリングマシンではイジングモデルのパラメータを入力として受け付けている. しかし, イジングモデルの系統的な導出方法は示されていないため表現がまだされていない関数が多くある.

先行研究 [4] ではロバスト性のある q -loss 関数を Legendre 変換を用いて QUBO 形式での導出を行い分類問題を例にその性能を評価している. また, 最近発表された ReLU 関数の QUBO 形式での導出に関する論文 [5] では Legendre 変換だけでは不十分であり, Wolfe の双対定理を利用することで導出を行っていた.

一方, スパース推定は画像処理や機械学習の分野で利用されており, データ数の削減や高次元で複雑なデータから関連するデータを選択し単純な形にするなど, とても重要な研究である.

1.2 目的

本研究の目的は, ReLU 関数の QUBO 形式での導出に関する論文 [5] での導出方法を用いて, スパース推定で利用される Lasso の正則化項である絶対値関数を QUBO 形式で導出を行うことである. さらに, 導出結果が正しいかをシュミレーテッドアニーリングを用いて評価を行う. また, Lasso(座標降下法) と Lasso(アニーリング)での結果を比較して性能の評価を行い, アニーリングでは推定値が 0 に収束するかを検証する.

1.3 本論文の構成

まず第2章で本研究の前提知識について説明する. 第3章では先行研究における q -loss 関数の QUBO 形式の導出例を紹介する. 第4章では絶対値関数の QUBO 形式での導出を行い正しく導出されているか評価する. 第5章では Lasso の正則化項を第4章で導出した式に置き換えシミュレーテッドアニーリングを用いて用いて解いた Lasso(アニーリング) と Lasso(座標降下法) での結果を比較し, Lasso(アニーリング) の評価を行う. そして, アニーリングではスパースな解が得られるかの検証を行う. 第6章では実験での結果をまとめ, 今後の課題について述べている.

第2章 序論

本章では, 研究内容に関する議論を行うにあたって必要となる基本的事項について説明する.

2.1 スパース推定

スパース推定 [7] とは, 多くのパラメータのうちほとんどを 0 でごく一部のみ非 0 の値を推定する方法である. スパース性を持つ推定方法の代表例として Lasso [8] がある. 回帰問題での回帰係数の推定方法として次の三つについて説明を行う.

2.1.1 最小二乗法

最小二乗法とは, 次の式をコスト関数としてコストが最小となるような回帰係数ベクトル β を推定する方法である. コスト関数は次のように表される.

$$\begin{aligned} J(\beta) &= \|y - X\beta\|_2^2 \\ &= (y - X\beta)^T (y - X\beta) \end{aligned} \quad (2.1)$$

ここで, y は n 次元の観測ベクトル, X は $n \times p$ 次元の計画行列, β は p 次元の回帰係数ベクトルである.

また, 最小二乗法によって求められる回帰係数ベクトルは次の式で求められる.

$$\begin{aligned} \frac{\partial J(\beta)}{\partial \beta} &= -2X^T(y - X\beta) = 0 \\ \beta &= (X^T X)^{-1} X^T y \end{aligned} \quad (2.2)$$

しかし, $X^T X$ が特異行列もしくはそれに近い場合, 推定をうまく行うことができない. そこで, コスト関数の式 (2.1) に回帰係数ベクトルに関する制約項 (正則化項) を追加することで対応する. その例が次に説明する Ridge と Lasso である.

2.1.2 L_2 正則化 (Ridge)

Ridge とは最小二乗法のコスト関数に L_2 正則化項 (二乗関数) を加えたものをコスト関数としたものである. L_2 正則化項を加えることにより, 最小二乗法と比較して, 過学習を抑え汎化能力を高めることができる. Ridge のコスト関数の式は次のようにあらわされる.

$$\begin{aligned} S_\lambda(\beta) &= \|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2 \\ &= (y - X\beta)^T(y - X\beta) + \lambda\beta^T\beta \end{aligned} \quad (2.3)$$

ここで, λ は正則化パラメータであり, $\lambda > 0$ である.

また, Ridge によって求められる回帰係数ベクトルは次の式で求められる.

$$\begin{aligned} \frac{\partial S_\lambda(\beta)}{\partial \beta} &= -2X^T(y - X\beta) - 2\lambda\beta = 0 \\ (X^T X + \lambda I_p)\beta &= X^T y \\ \beta &= (X^T X + \lambda I_p)^{-1} X^T y \end{aligned} \quad (2.4)$$

2.1.3 L_1 正則化 (Lasso)

Lasso (Least Absolute Shrinkage and Selection operator) [6] とは L_1 正則化項 (絶対値関数) を利用することで回帰係数をスパースに推定できる手法である. Lasso のコスト関数の式は下のよう表される.

$$\begin{aligned} S_\lambda(\beta) &= \|y - X\beta\|_2^2 + \lambda\|\beta\|_1 \\ &= (y - X\beta)^T(y - X\beta) + \lambda(|\beta_1| + |\beta_2| + \cdots + |\beta_p|) \end{aligned} \quad (2.5)$$

ここで, Lasso の正則化項は連続ではないので, 最小二乗法や Ridge のように β について偏微分することによって更新式を得ることができない. そこで, 座標降下法を用いて β の更新式を得る方法について説明する.

座標降下法 (Coordinate Descent) とは, p 個あるパラメータのうち $(p - 1)$ 個の要素の値を固定し, 残る 1 個のパラメータのみを更新するという手順を全ての要素で繰り返すことで更新式を得るアルゴリズムである.

ここでは, Lasso の推定量をわかりやすく表現するために計画行列 X が $X^T X = nI_p$ を満たすと仮定する. β_j 以外のパラメータを固定したベクトルを $\tilde{\beta}_k (k \neq j)$, 行列

X の k 列目を $x_{(k)}$, 第 j 変数に対応する項を取り除いた偏残差ベクトルを $r^{(j)} = y - \sum_{k \neq j} x_{(k)} \tilde{\beta}_k$ と置くと, $\beta_j (j = 1, \dots, p)$ で微分した式は次のようになる.

$$\begin{aligned} \frac{\partial S_\lambda(\beta)}{\partial \beta_j} &= -2x_{(j)}^T(y - X\beta) + \lambda \text{sign}(\beta_j) = 0 \\ x_{(j)}^T(r^{(j)} - x_{(j)}\beta_j) &= \frac{1}{2}\lambda \text{sign}(\beta_j) \\ \beta_j &= S\left(\frac{r^{(j)T}x_j}{n}, \frac{1}{2n}\lambda\right) \end{aligned} \quad (2.6)$$

ここで $S(x, \lambda)$ は軟閾値作用素 (soft-thresholding operator) と呼ばれ, 次の式で表される.

$$S(x, \lambda) = \begin{cases} x - \lambda & (x > \lambda) \\ 0 & (|x| \leq \lambda) \\ x + \lambda & (x < -\lambda) \end{cases} \quad (2.7)$$

2.2 シミュレーテッド・アニーリング

シミュレーテッドアニーリングとは, 温度 T というパラメータを新たに追加し, T を利用して状態を変化させその時のコストを繰り返し求めなおすことで最適化問題の近似解を求める方法である. D-WAVE のような量子アニーリングでは量子効果を利用するが, 温度による効果を使うことで古典的な計算機でも最適化問題を解くことが可能になる. 本研究ではシミュレーテッドアニーリングを利用して実験を行っている. アルゴリズムは次のようになる.

メトロポリス・ヘイスティングス法を利用した
シミュレーテッドアニーリングのアルゴリズム

- 1: 温度 T を初期化, 適当な状態 s でのコスト C_1 とする
 - 2: 温度 T を利用して s の近くの状態 \tilde{s} を選び, その時のコストを C_2 とする
 - 3: C_1 と C_2 を比較して
 C_1 の方が小さい場合, 確率 $\exp(-\frac{C_2 - C_1}{T})$ で \tilde{s} を s にする
 C_2 の方が小さい場合, \tilde{s} を s にする
 - 4: 温度 T を小さくする
 - 5: 手順 2 に戻る
-

2.3 イジング模型

量子アニーリングマシンではイジング模型という物理モデルがハードウェアとして実装されており、アニーリングを高速で実行し近似最適解を出力する。イジング模型は次のエネルギー関数で表される。

$$H = \sum_{i \neq j} J_{ij} \sigma_i \sigma_j + \sum_i h_i \sigma_i \quad (\sigma_i = \pm 1, \quad i = 1, 2, \dots) \quad (2.8)$$

ここで、 σ_i は入力変数であり、 $\sigma \in \{-1, +1\}$ 、 J_{ij} は二体相互作用パラメータ、 h_i は一体パラメータである。アニーリングマシンへの入力として J_{ij}, h_i を与えることでエネルギーが最小となる $\sigma = \{\sigma_i | i = 1, 2, \dots\}$ の組み合わせの近似最適解を出力する。

イジングでの定式化がすでにされている問題を例に説明を行う。

2.3.1 色彩問題のイジングの定式化

ここでは、色彩問題のイジングでの定式化 [9] についての説明を行う。無向グラフを $G = (V, E)$ とし、グラフの各頂点を n 色で染色する。この時、辺が同じ色の二つの頂点を結ばないようにする。この問題をイジング模型で表すと次の式で表される。

$$H = A \sum_v \left(1 - \sum_{i=1}^n x_{v,i} \right)^2 + A \sum_{(uv) \in E} \sum_{i=1}^n x_{u,i} x_{v,i} \quad (2.9)$$

ただし、 A は正の定数、 $x_{v,i}$ は頂点 v が色 i で染色されている時 1、そうでない時は 0 である二値変数である。第一項は各頂点が厳密に一色であることを強制し、これに違反するたびにペナルティを加える。また、第二項は辺が同じ色の二つの頂点を結ぶたびにペナルティを加える。

2.3.2 QUBO 形式

最適化問題から二値変数 $\sigma \in \{-1, +1\}$ を用いた二体イジング模型の形式に直接変形すること難しい。そこで最適化問題に対して二進展開などを行い、二値変数

$q \in \{0, 1\}$ を用いた形式にする. これを QUBO(Quadratic Unconstrained Binary Optimization) と呼び, QUBO は次のエネルギー関数で表される.

$$H = \sum_{i \neq j} \tilde{J}_{ij} q_i q_j + \sum_i \tilde{h}_i q_i \quad (q_i = 0 \text{ or } 1, \quad i = 1, 2, \dots) \quad (2.10)$$

このエネルギー関数は $q = \frac{\sigma+1}{2}$ を用いて変数変換することでイジング模型の形式に変換することができる. また, \tilde{J}_{ij} と \tilde{h}_i は式 (2.8) と同様にパラメータだが, $J_{ij} \neq \tilde{J}_{ij}, h_i \neq \tilde{h}_i$ である.

第3章 q -loss関数のQUBO形式での導出例

本章では先行研究 [4] で行われている q -loss 関数の QUBO 形式の導出について紹介する.

3.1 q -loss関数の定義

q -loss 関数は式 (3.1) と定義され, グラフの概形は図 3.1 のようになる.

$$L_q(m) = \min [(1 - q)^2, (\max [0, 1 - m])^2] \quad (3.1)$$

ここで, $q \in (-\infty, 0]$ はパラメータであり, m は実変数である.

論文では二値分類器を例として説明を行なっている. 通常の二乗誤差関数の場合, 正しく分類できている場合はペナルティを与えないようにし, 逆に間違った分類をしている場合には誤差の二乗のペナルティを与えることで分類を行う. しかし, 識別境界から遠い点にラベルノイズが存在する時, ペナルティが大きくなり識別境界が大きく変化してしまう. 一方 q -loss 関数の場合, 間違った分類をしている時のペナルティを一定にすることでラベルノイズによって識別境界が大きく変化することを防ぎロバスト性を確保していることを説明している.

しかし, 式 (3.1) には \min 関数や \max 関数が含まれているため二次形式ではなく, QUBO 形式ではない. そこで, Legendre 変換を用いて, 式 (3.1) の変換を行う. 次の節では Legendre 変換についての説明を行う.

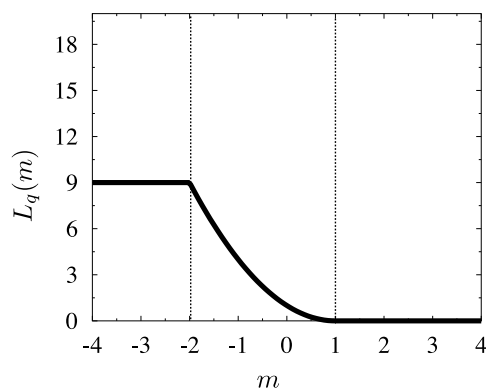


図 3.1: q -loss 関数の概形

3.2 Legendre 変換

この節では, Legendre 変換について簡単に説明する. 通常は下に凸の関数に対する定義を与えることが多いが, ここでは q -loss 関数への適用を考え上に凸の関数に対する変換を考える.

一変数関数 $f(x)$ をもとにして, 次のような関数 $f^*(\eta)$ を考える.

$$\begin{aligned} f^*(\eta) &= \min_x \{ \eta x - f(x) \} \\ &= - \max_x \{ f(x) - \eta x \} \end{aligned} \quad (3.2)$$

このような $f(x)$ から $f^*(\eta)$ への変換を Legendre 変換と呼び, $f^*(\eta)$ のことを共役関数 (conjugate function) と呼ぶ.

この変換は “傾き η の直線が関数 $f(x)$ が交わるという条件下での切片の最大値に -1 を掛けた値を η の関数として捉えること” に対応している. では, この変換の意味について考えてみる. 点 $(x_0, f(x_0))$ を通る傾き η の直線は

$$y = \eta(x - x_0) + f(x_0) \quad (3.3)$$

であり, $x = 0$ のとき切片 y_0 は

$$y_0 = f(x_0) - \eta x_0 \quad (3.4)$$

と求まる. 式 (3.2) はこれの最大値を求めよ, という事なので, 傾きを固定したまま直線を動かして, 切片が最大となるところを探せ, という事と同値となる.

実際に $f(x) = -x^2$ を Legendre 変換した時のことを考えてみる. $f(x)$ の共役関数 $f^*(\eta)$ は次のように求められる.

$$\begin{aligned} f^*(\eta) = \min_x \{ \eta x - (-x^2) \} &= \min_x \left\{ \left(x + \frac{1}{2}\eta \right)^2 - \frac{1}{4}\eta^2 \right\} \\ &= -\frac{1}{4}\eta^2 \end{aligned} \quad (3.5)$$

なぜこのように共役関数を考えるかと言うと, “凸性がある場合, 共役関数の共役関数は元の関数に戻る” という性質があるからである. すなわち $f^{**}(x) = f(x)$ となる. 次の式は $f^*(\eta)$ の共役関数 $f^{**}(x)$ を導出したものである.

$$\begin{aligned} f^{**}(x) = \min_{\eta} \{ x\eta - (-\frac{1}{4}\eta^2) \} &= \min_{\eta} \left\{ \frac{1}{4}(\eta + 2x)^2 - x^2 \right\} \\ &= -x^2 \\ &= f(x) \end{aligned} \quad (3.6)$$

この性質を使えば, $f^*(\eta)$ が扱いやすければ η を変数とする空間で式変形を行い, そのあとでその共役関数を考えて元に戻すことが可能になる.

また, 上に凸の関数の場合の Legendre 変換は式 (3.2) に示したが, 下に凸の関数の場合は次の式で求められる.

$$\begin{aligned} f^*(\eta) &= \max_x \{ \eta x - f(x) \} \\ &= -\min_x \{ f(x) - \eta x \} \end{aligned} \quad (3.7)$$

3.3 q -loss 関数への Legendre 変換の適用

q -loss 関数は凸関数ではないので, 代わりに $f(m) = L_q(m) - m^2$ と置く. この時, $f(m)$ は区間毎に次のように表され, 概形は図 3.2 のようになる.

$$f(m) = \begin{cases} -m^2 + (1-q)^2 & (m \leq q) \\ -2m + 1 & (q < m < 1) \\ -m^2 & (1 \leq m) \end{cases} \quad (3.8)$$

すると, $f(m)$ は上に凸な関数なので Legendre 変換を行うと次のように共役関数を求められる.

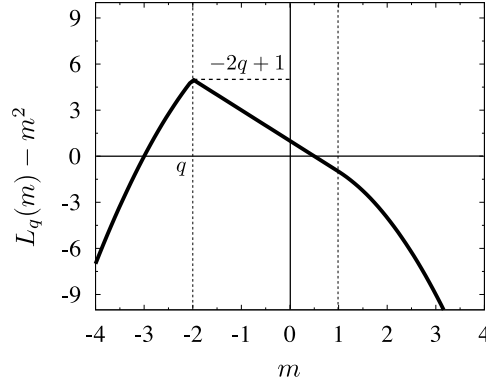


図 3.2: $f(m) = L_q(m) - m^2$ の概形

(a) $m \leq q$ の場合:

$$\begin{aligned} \min_m \{ \eta m - (-m^2 + (1-q)^2) \} &= \min_m \left\{ \left(m + \frac{1}{2}\eta \right)^2 - \frac{1}{4}\eta^2 - (1-q)^2 \right\} \\ &= -\frac{1}{4}\eta^2 - (1-q)^2 \end{aligned} \quad (3.9)$$

共役関数は $g(\eta) = -\frac{1}{4}\eta^2 - (1-q)^2$ となり, η は接線の傾きに対応するので, η の取りうる範囲は負の方向から $\eta \in [-2q, +\infty)$ である.

(b) $q < m < 1$ の場合: 点 $(q, -2q + 1)$ を通る直線を考える. $f(m) = -2m + 1$ は傾き -2 の直線なので, η の取りうる範囲は負の方向から $\eta \in (-2, -2q)$ であり, 対応する切片は, $2q^2 - 2q + 1$ から $+1$ まで線形に変化する. この切片の符号を変えたものが $g(\eta)$ となるので, $g(\eta)$ は二点 $(-2q, -(2q^2 - 2q + 1))$, $(-2, -1)$ を結ぶ直線 $g(\eta) = q\eta + (2q - 1)$ となる.

(c) $1 \leq m$ の場合:

$$\begin{aligned} \min_m \{ \eta m - (-m^2) \} &= \min_m \left\{ \left(m + \frac{1}{2}\eta \right)^2 - \frac{1}{4}\eta^2 \right\} \\ &= -\frac{1}{4}\eta^2 \end{aligned} \quad (3.10)$$

共役関数は $g(\eta) = -\frac{1}{4}\eta^2$ となり, η は接線の傾きに対応するので, η の取りうる範囲は負の方向から $\eta \in (-\infty, -2]$ である.

以上より, 共役関数 $g(\eta)$ は次のように表される. また, この関数の概形は図 3.3 のようになる.

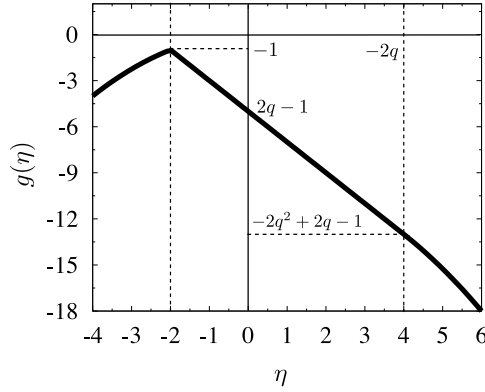


図 3.3: $g(\eta)$ の概形

$$g(\eta) = \begin{cases} -\frac{1}{4}\eta^2 & (\eta \leq -2) \\ q\eta + (2q - 1) & (-2 < \eta < -2q) \\ -\frac{1}{4}\eta^2 - (1 - q)^2 & (-2q \leq \eta) \end{cases} \quad (3.11)$$

$g(\eta)$ は上に凸な関数であり, $g(\eta)$ の共役関数を求めれば, $f(m) = L_q(m) - m^2$ が再現される. $g(\eta)$ の共役関数を $g^*(m)$ と置き, $g^*(m)$ の導出をすると次のようになる.

(a) $\eta \leq -2$ の場合:

$$\begin{aligned} \min_{\eta} \left\{ m\eta - \left(-\frac{1}{4}\eta^2 \right) \right\} &= \min_{\eta} \left\{ \frac{1}{4}(\eta + 2m)^2 - m^2 \right\} \\ &= -m^2 \end{aligned} \quad (3.12)$$

共役関数は $g^*(m) = -m^2$ となり, m は接線の傾きに対応するので, m の取りうる範囲は負の方向から $m \in (+\infty, 1]$ である.

(b) $-2 < \eta < -2q$ の場合: 点 $(-2, -1)$ を通る直線を考える. $g(\eta) = q\eta + (2q - 1)$ は傾き q の直線なので, m の取りうる範囲は負の方向から $m \in (q, 1)$ であり, 対応する切片は, $+1$ から $2q - 1$ まで線形に変化する. この切片の符号を変えたものが $g^*(m)$ となるので, $g^*(m)$ は二点 $(1, -1), (q, -(2q - 1))$ を結ぶ直線 $g^*(m) = -2m + 1$ となる.

(c) $-2q \leq \eta$ の場合:

$$\begin{aligned} \min_{\eta} \left\{ m\eta - \left(-\frac{1}{4}\eta^2 - (1 - q)^2 \right) \right\} &= \min_{\eta} \left\{ \frac{1}{4}(\eta + 2m)^2 - m^2 + (1 - q)^2 \right\} \\ &= -m^2 + (1 - q)^2 \end{aligned} \quad (3.13)$$

共役関数は $g^*(m) = -m^2 + (1-q)^2$ となり, m は接線の傾きに対応するので, m の取りうる範囲は負の方向から $m \in [q, -\infty)$ である.

以上のことから, $g(\eta)$ の共役関数は $f(m) = L_q(m) - m^2$ を再現していることがわかる.

しかし, 条件分岐が多い $g(\eta)$ はイジング模型で扱いにくい. そこで, 次のような関数 $\tilde{g}(\eta)$ を導入する.

$$\tilde{g}(\eta) = -\frac{\eta^2}{4} - (1-q)^2 \frac{(1 - \text{sign}(-\frac{\eta}{2} - 1))}{2} \quad (3.14)$$

この関数を区間ごとに表すと次のようになり, 関数の概形は図 3.4 のようになる.

$$\tilde{g}(\eta) = \begin{cases} -\frac{1}{4}\eta^2 & (\eta \leq -2) \\ -\frac{1}{4}\eta^2 - (1-q)^2 & (-2 < \eta < -2q) \\ -\frac{1}{4}\eta^2 - (1-q)^2 & (-2q \leq \eta) \end{cases} \quad (3.15)$$

この関数は明らかに凸関数ではなく, $-2 < \eta < -2q$ の範囲で $g(\eta) \neq \tilde{g}(\eta)$ である. しかし, $\tilde{g}(\eta)$ の共役関数を求めると $f(m) = L_q(m) - m^2$ となることがわかる. そのことについての説明を次に行う.

傾きが m の接線を考える. η が大きい方から考えると接線の傾きは $-\infty$ から大きくなっていき, 傾きが $m = q$ となった時左側の二次関数の上端にぶつかる. ここで, 共役関数の定義に立ち返ると, 今度はそのぶつかった左側の上端を軸として直線を考えることになり, $-2 < \eta < -2q$ の部分の形状は関係しない. よって, 共役関数を考える分には $g(\eta)$ と $\tilde{g}(\eta)$ には違いはない.

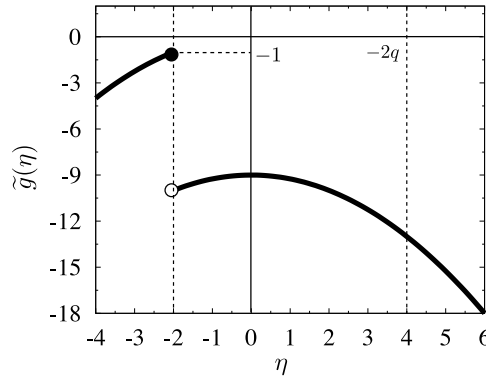


図 3.4: $\tilde{g}(\eta)$ の概形

変数変換 $t = -\frac{\eta}{2}$ を用いて, $\tilde{g}(\eta)$ の共役関数を求めると次のようになる.

$$\begin{aligned}
f(m) &= \min_{\eta} \{ \eta m - \tilde{g}(\eta) \} \\
&= \min_{\eta} \left\{ \eta m + \frac{\eta^2}{4} + (1-q)^2 \frac{(1 - \text{sign}(-\frac{\eta}{2} - 1))}{2} \right\} \\
&= \min_t \left\{ (m - t)^2 - m^2 + (1-q)^2 \frac{(1 - \text{sign}(t - 1))}{2} \right\} \quad (3.16)
\end{aligned}$$

$f(m) = L_q(m) - m^2$ なので, q -loss 関数 $L_q(m)$ は,

$$\begin{aligned}
L_q(m) &= f(m) + m^2 \\
&= \min_t \left\{ (m - t)^2 - m^2 + (1-q)^2 \frac{(1 - \text{sign}(t - 1))}{2} \right\} + m^2 \\
&= \min_t \left\{ (m - t)^2 + (1-q)^2 \frac{(1 - \text{sign}(t - 1))}{2} \right\} \quad (3.17)
\end{aligned}$$

と求められる.

式 (3.17) ならば, t, m について二次で書かれており, sign 関数については二進数展開時に t の最上位ビットを符号に対応させることで書き表すことができるので QUBO 形式で表現できている.

第4章 絶対値関数に対する QUBO 形式の導出

4.1 Legendre 変換での導出

次の式で表される関数 $f(m)$ を用いる.

$$f(m) = -\min(-m, m) \quad (4.1)$$

図 4.1 に $f(m)$ の概形を示す.

$f(m)$ の概形は絶対値関数を表現したものとなる. 本章では $f(m)$ に Legendre 変換等を行い, 二次形式を求める.

絶対値関数は下に凸な関数なので Legendre 変換では, 式 (3.7) を利用する.

式 (4.1) で定義された $f(m)$ を次のように書き換える.

$$f(m) = \begin{cases} -m & (m < 0) \\ m & (m \geq 0) \end{cases} \quad (4.2)$$

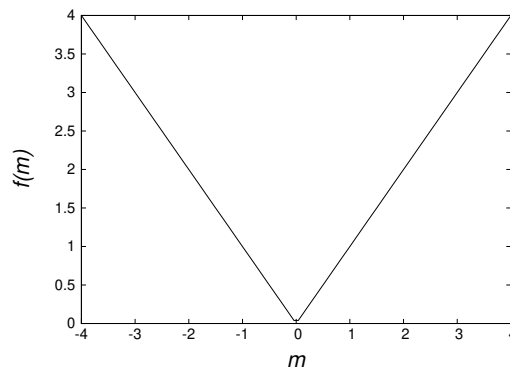


図 4.1: 絶対値関数の概形

次のように, 各区間に対して Legendre 変換を行う.

(a) $m < 0$ の場合 :

この範囲における傾きは常に -1 であるので,

$$\begin{aligned} -\min_m \{-m - mt\} &= -\min_m \{-m(1+t)\} \\ &= 0 \end{aligned}$$

共役関数は $f^*(t) = 0$ となり, t の取りうる値は $t = -1$ のみである.

(b) $m = 0$ の場合 :

この点における左微分は $f'_-(m) = -1$, 右微分は $f'_+(m) = 1$ であることから, 取りうる傾きは -1 から 1 までとなる. また, 点 $(0, 0)$ を通る直線となることから, 切片は常に 0 である. すなわち, 共役関数は点 $(-1, 0)$ と $(1, 0)$ を結ぶ直線, $f^*(t) = 0$ である. t の取りうる範囲は $t \in (-1, 1)$ である.

(c) $m > 0$ の場合 :

この範囲における傾きは常に 1 であるので,

$$\begin{aligned} -\min_m \{m - mt\} &= -\min_m \{m(1-t)\} \\ &= 0 \end{aligned}$$

共役関数は $f^*(t) = 0$ となり, t の取りうる値は $t = 1$ のみである.

以上より, $f^*(t) = 0$ ($-1 \leq t \leq 1$) である. これを再度 Legendre 変換すれば $f(m)$ が求まる. そのため, $f(m)$ の 2 次形式は次のようになる.

$$F(m) = -\min_t \{-mt\} \quad \text{subject to} \quad -1 \leq t \leq 1 \quad (4.3)$$

なお, $f(m)$ の 2 次形式であることを強調するために, 新たに $F(m)$ とした.

以上のようにして $f(m)$ の 2 次形式を表現できたが, $F(m)$ の \min 関数にマイナスの符号がついているため, $F(m)$ と他のコスト関数 $C(m)$ を組み合わせた最適化問題を二体イジングモデルを用いて解くことが困難になる. すなわち

$$\min_m \{C(m) + F(m)\} = \min_m \left\{ C(m) - \min_t \{-mt\} \right\} \neq \min_{m,t} \{C(m) - (-mt)\}$$

となり, m と t の最小化問題の形にすることができない. この問題に対処するために Wolfe の双対定理を導入する.

4.2 Wolfe の双対定理

Wolfe の双対定理は, 次のような不等式制約付き最小化問題 (主問題) を, ある種の最大化問題 (双対問題) に書き換えるものである.

$$\begin{cases} \text{minimize}_x & f_w(x) & (x \in \mathbb{R}^n) \\ \text{subject to} & h_i(x) \leq 0 & (i = 1, 2, \dots, l) \end{cases}$$

この最適化問題に対するラグランジュ関数は次のように表される.

$$L(x, z) = f_w(x) + z^T h(x)$$

このラグランジュ関数とラグランジュ乗数 $z \in \mathbb{R}^l$ に対して, 双対問題は次のように表される.

$$\begin{cases} \text{maximize}_{x,z} & L(x, z) & ((x, z) \in \mathbb{R}^n \times \mathbb{R}^l) \\ \text{subject to} & \nabla L(x, z) = 0 & (z \geq 0) \end{cases}$$

4.3 Wolfe の双対定理を利用した双対問題への変換

ここで, 式 (4.3) に対して Wolfe の双対定理を適用する. まず $F(m)$ を最適化問題の形式で書くと次のようになる.

$$\begin{cases} \text{minimize}_t & -mt \\ \text{subject to} & h_1(t) = -(t+1) \leq 0, h_2(t) = t-1 \leq 0 \end{cases} \quad (4.4)$$

Wolfe の双対定理を用いて最適化問題 (4.4) の双対問題を求めると次のようになる

$$\begin{cases} \text{maximize}_{t,z} & -mt - z_1(t+1) + z_2(t-1) \\ \text{subject to} & -m - z_1 + z_2 = 0 & (z_1 \geq 0, z_2 \geq 0) \end{cases} \quad (4.5)$$

なお, 最適化問題 (4.5) を二体イジングモデルを用いて解くためには, 目的関数と制約条件を1つの式にまとめる必要がある. そのため最適化問題の式に, 制約条件を表す次のようなペナルティ項を加える.

$$-M(-m - z_1 + z_2)^2 \quad (4.6)$$

ここで M は非常に大きい数, z_1, z_2 は共に0以上とする. 制約条件 $-1 \leq t \leq 1$, $z_1 \geq 0$, $z_2 \geq 0$ については, 二進数展開時に対応する値しか取らないように展開することで対応できる.

以上の議論より、最適化問題 (4.5) を二体イジング形式で表現すると次のようになる。

$$L(m, t, z_1, z_2) = -mt - z_1(t + 1) + z_2(t - 1) - M(-m - z_1 + z_2)^2 \quad (4.7)$$

$L(m, t, z_1, z_2)$ を用いることにより、次のような関係になる。

$$-\min_t \{-mt\} \xrightarrow{\text{Wolfe}} -\max_{t, z_1, z_2} \{L(m, t, z_1, z_2)\} = \min_{t, z_1, z_2} \{-L(m, t, z_1, z_2)\}$$

したがって次の関係が成り立ち、他のコスト関数と組み合わせることができない問題が解決される。

$$\min_m \{C(m) + \min_{t, z_1, z_2} \{-L(m, t, z_1, z_2)\}\} = \min_{m, t, z_1, z_2} \{C(m) - L(m, t, z_1, z_2)\}$$

4.4 シミュレーテッドアニーリングを用いた数値実験

シミュレーテッドアニーリングを用いて最適化問題 $-L(m, t, z_1, z_2)$ の計算を行う。各変数の設定は次のようにした。

- m は -4 から 0.01 ずつ増加させ、 4 になるまで各一回ずつ行う
- イタレーション回数は $100,000$ 回
- $t_{(0)}, z_{1(0)}, z_{2(0)}$ の初期値を 0 , $\sigma_{(0)}$ の初期値を 1 , 温度 $T_{(0)}$ の初期値を 50 , α の値を 0.995
- n 回目での $\sigma_{(n)}, T_{(n)}$ の値は次のようにする
 - $\sigma_{(n)} = \alpha * \sigma_{(n-1)}$
 - $T_{(n)} = \alpha * T_{(n-1)}$
- n 回目での t, z_1, z_2 の更新式は $\epsilon \sim \mathcal{N}(0, \sigma_{(n)})$ を用いて次のようにする
 - $t_{(n)} = t_{(n-1)} + \epsilon$ ($t < -1$ のとき $t = -1$, $t > 1$ のとき $t = 1$ にする)
 - $z_{1(n)} = z_{1(n-1)} + \epsilon - \frac{1}{4}\sigma_{(n)}$ ($z_1 < 0$ のとき $z_1 = 0$ にする)
 - $z_{2(n)} = z_{2(n-1)} + \epsilon - \frac{1}{4}\sigma_{(n)}$ ($z_2 < 0$ のとき $z_2 = 0$ にする)

このときの $-L(m, t, z_1, z_2), t, z_1, z_2$ 値は、図 4.2, 4.3, 4.4, 4.5 のようになる。

次に、図 4.2, 4.3 の結果より、コストの値は収束しているにも関わらず、 t の値は収束せずランダムに散らばっていることからコスト関数の式から t を取り除ける可能性がある。そこで、次の節では t を取り除けるかの検証を行う。

4.5 変数 t の消去

前節の数値実験の結果から変数 t は最適な値に落ち着くと言うよりは適当な値を取っているだけであり、今回の定式化では不要であると予想できる。そこで、変数 t を消去できるかを考える。

コスト関数は制約条件を用いて次のように式変形できる。

$$\begin{aligned} -L(m, t, z_1, z_2) &= mt + z_1(t + 1) - z_2(t - 1) && + M(-m - z_1 + z_2)^2 \\ &= mt + z_1(t + 1) - (m + z_1)(t - 1) + M(-m - z_1 + z_2)^2 \\ &= z_1 + (m + z_1) && + M(-m - z_1 + z_2)^2 \\ &= z_1 + z_2 && + M(-m - z_1 + z_2)^2 \quad (4.8) \end{aligned}$$

式 (4.8) をコスト関数とした最適化問題 $-L(m, z_1, z_2)$ を 4.4 節と同じ実験条件でシミュレーテッドアニーリングで実行した時の $-L(m, z_1, z_2), z_1, z_2$ の値は図 4.6, 4.7, 4.8 のようになる。

この結果から、 t を取り除いても結果に影響がないように思われる。

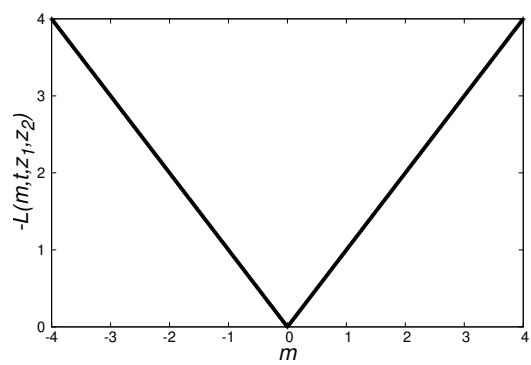


図 4.2: コストの概形

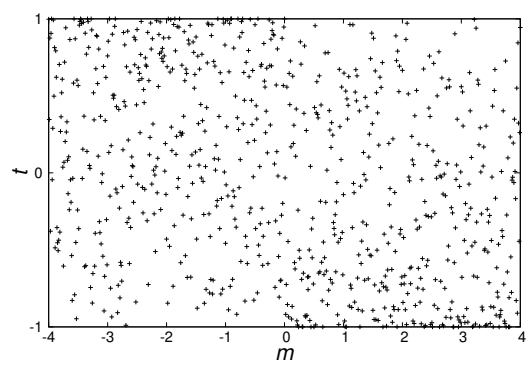


図 4.3: t の概形

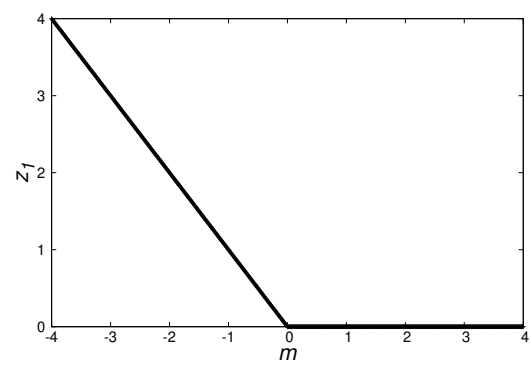


図 4.4: z_1 の概形

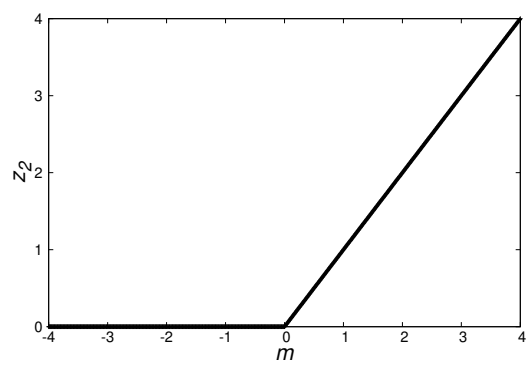


図 4.5: z_2 の概形

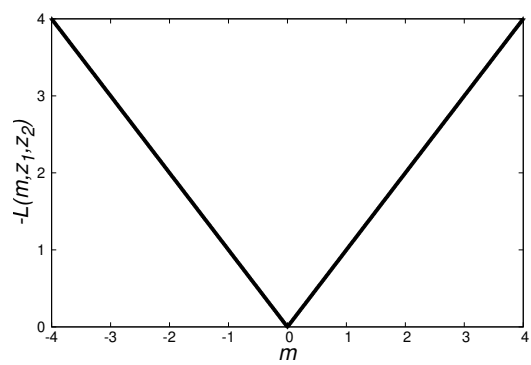


図 4.6: t を取り除いたコストの概形

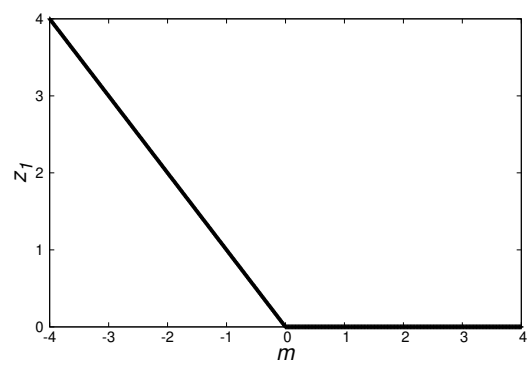


図 4.7: t を取り除いた z_1 の概形

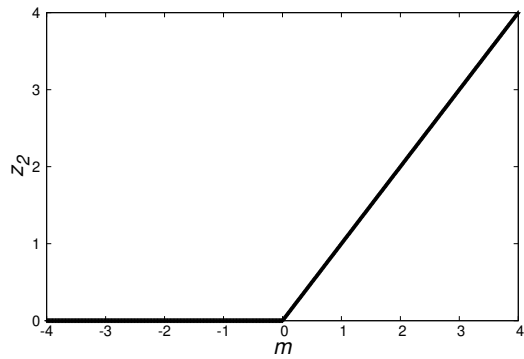


図 4.8: t を取り除いた z_2 の概形

第5章 シミュレーテッドアニーリング を利用したスパース性の検証

Lasso の正則化項を式 (4.7) に置き換え, シミュレーテッドアニーリングを用いて最適化問題を解く. Lasso のコスト関数は次のように表される.

$$\begin{aligned}\min_{\beta} S_{\lambda}(\beta) &= \min_{\beta} \{\|y - X\beta\|_2^2 + \lambda\|\beta\|_1\} \\ &= \min_{\beta} \{\|y - X\beta\|_2^2 + \lambda \sum_{i=1}^p \{z_{1i} + z_{2i} + M(-\beta_i - z_{1i} + z_{2i})^2\}\}\end{aligned}$$

ここで, 観測ベクトル: $y \in \mathbb{R}^n$, 計画行列: $X \in \mathbb{R}^{n \times p}$, 回帰係数ベクトル: $\beta \in \mathbb{R}^p$, 回帰係数ベクトルに対応するベクトル: $z_1, z_2 \in \mathbb{R}^p$ とする.

5.1 人工データ

行列の各行が正規分布 $\mathcal{N}(0, 1)$ に従う計画行列を $X \in \mathbb{R}^{n \times p}$ と置く. 次に, 解を $\beta \in \mathbb{R}^p$, 正規分布 $\mathcal{N}(0, 0.1)$ に従うノイズベクトルを $\epsilon \in \mathbb{R}^n$ と設定し, 観測ベクトル $y \in \mathbb{R}^n$ を次のようにとる.

$$y = X\beta + \epsilon \quad (5.1)$$

ただし, 解 β_j には $|\beta_j| \leq 10$ の値が入るように設定した.

5.2 数値実験

5.1 節の方法で作成したデータセット (データ数: $n = 300$, パラメータ数: $p = 5$) を利用して, Lasso をシミュレーテッドアニーリングで解く. この時の各変数の設定は次のようにした.

- イタレーション回数：5,000,000 回
- 正則化パラメータ： $\lambda = 1, 2n, 4n$
- $\beta_{(0)}, z_{1(0)}, z_{2(0)}$ の初期値を 0, $\sigma_{(0)}$ の初期値を 1, 温度 $T_{(0)}$ の初期値を 50
- (イタレーション回数) 回目での σ の値を $until = 10^{-2}$ とする
- α は次の式で求める

$$\begin{aligned}
 (\sigma \text{の初期値}) \cdot \alpha^{(\text{イタレーション回数})} &= until \\
 \alpha &= \left(\frac{until}{\sigma \text{の初期値}} \right)^{\frac{1}{\text{イタレーション回数}}} \quad (5.2)
 \end{aligned}$$

- n 回目での $\sigma_{(n)}, T_{(n)}$ の値は次のようにする
 - $\sigma_{(n)} = \alpha * \sigma_{(n-1)}$
 - $T_{(n)} = \alpha * T_{(n-1)}$
- $\beta, z_{1(n)}, z_{2(n)}$ の更新式は $\epsilon \sim \mathcal{N}(0, \sigma_{(n)})$ を用いて次のようにする
 - $\beta_{(n)} = \beta_{(n-1)} + \epsilon$
 - $z_{1(n)} = z_{1(n-1)} + \epsilon - \frac{1}{4}\sigma_{(n)} \quad (z_1 < 0 \text{ のとき } z_1 = 0 \text{ にする})$
 - $z_{2(n)} = z_{2(n-1)} + \epsilon - \frac{1}{4}\sigma_{(n)} \quad (z_2 < 0 \text{ のとき } z_2 = 0 \text{ にする})$

ただし、更新後の値の小数第三位を四捨五入することで離散化する。

また、Lasso(座標降下法) ではパラメータの初期値を 1 とし、更新する順番をランダムにして実験を行なった。

実験の結果より、正則化パラメータの値が大きくなると、0 と推定されるものが多くなりスパースになっていることがわかる。また、シミュレーテッドアニーリング (SA) は座標降下法 (CD) に近い推定結果を得られた。

CD と SA での推定結果の比較

	解 (β)	$\lambda = 1$		$\lambda = 2n$		$\lambda = 4n$	
		CD	SA	CD	SA	CD	SA
β_1	0.32797	0.33	0.33	0.00	0.00	0.00	0.00
β_2	1.05080	1.04	1.04	0.10	0.10	0.00	0.00
β_3	1.41335	1.35	1.36	0.26	0.26	0.00	0.00
β_4	9.05368	8.87	8.91	7.84	7.87	6.82	6.85
β_5	-9.43051	-9.07	-9.10	-8.16	-8.19	-7.16	-7.19

第6章 おわりに

6.1 まとめ

絶対値関数の QUBO 形式での導出を行い, 導出された式が正しいかをシミュレーテッドアニーリングを用いて実験を行なった. 実験の結果, 導出された式が正しいことがわかり, そしてパラメータを一つ取り除くことが可能だとわかった.

Lasso をシミュレーテッドアニーリングで解き, その時を結果を座標降下法で求めた時の結果と比較した. 結果は, シミュレーテッドアニーリングでも正則化パラメータを大きくすれば座標降下法での推定結果と同様にスパースな解を得られることがわかった. また, このことから導出した QUBO 形式が正しく動作しているようだとと言える.

6.2 今後の課題

今回, スパース性の検証にはデータ間に相関性のない人工データを利用した. しかし, Lasso はデータ間に相関性がある場合, 片方しか選択されずうまく推定を行えない. そこで, その欠点を改善した Elastic net も利用してデータ間に相関性のある人工データを用いて推定を行い, シミュレーテッドアニーリングとの比較を行う予定である. また, 実験ではコスト関数を二進展開せずに離散化して計算を行なったが, 二進展開をした場合での検証を行う必要がある.

参考文献

- [1] D-Wave Systems, D-Wave,
<https://www.dwavesys.com/home>
- [2] 富士通, Digital Annealer,
<http://www.fujitsu.com/jp/digitalannealer/>
- [3] 日立, CMOS アニールリング,
http://www.hitachi.co.jp/rd/portal/glossary/en_c/cmosaniiringu.html
- [4] V.S. Denchev, N. Ding, S.V.N. Vishwanathan, and H. Neven, “Robust classification with adiabatic quantum optimization,” in Proc. ICML’12, pp. 1003-1010 (2012)
- [5] G. Sato, M. Konoshima, Takuya Ohwa, Hirotaka Tamura, and Jun Ohkubo, “QUBO formulation for ReLU-type functions,” Preprint, arXiv:1811.03829 (2018).
- [6] R. Tibshirani, “Regression shrinkage and selection via the lasso,” J. Roy. Stat. Soc. B 58, pp. 267-288 (1996).
- [7] 富岡 亮太, スパース性に基づく機械学習 (講談社, 2015).
- [8] 川野 秀一, 松井 秀俊, 廣瀬 慧, スパース推定法による統計モデリング (共立出版株式会社, 2018).
- [9] A. Lucas, “Ising formulations of many NP problems,” Front. Phys. 2, Article 5 (2014).