数理情報工学特論第一
【機械学習とデータマイニング】
2章：回帰（３）

かしま　ひさし
鹿島 久嗣
（数理６研）

kashima@mist.i.～

東京大学
THE UNIVERSITY OF TOKYO

教師つき学習の王道である「回帰」について学びます

- 回帰問題の定義
- 線形回帰問題の定式化
- 線形回帰問題の初等的解法
- リッジ回帰：L_2正則化による過学習の回避
- 交差確認によるハイパーパラメータの決定
  - Leave-one-out交差確認
- 回帰問題の確率モデル的解釈

- 回帰の応用
- カーネル回帰
- L_1正則化

回帰の重要な応用

線形回帰の自明でない使い方を2つ紹介します

- 時系列予測
- 分類

時系列予測

- 線形回帰の1つの使い方として、時系列予測に用いることができる
- 時系列とは、時刻 $t = 1,2,\ldots$ に関連づけられた、実数値の列 $x_1, x_2, \ldots$ ($x_t \in \mathcal{R}$)
- 時系列予測の目的は、ある時刻tにおける時系列の値 $x_t \in \mathcal{R}$ をそれ以前の値 $x_1, x_2, \ldots, x_{t-1}$ から予測すること
- 時系列予測のための代表的モデルの1つが、**自己回帰モデル**もしくは**AR(Auto Regressive)モデル**と呼ばれるモデル
- D次の自己回帰モデル：

$$x_t = w_1 x_{t-1} + w_2 x_{t-2} + \cdots + w_D x_{t-D}$$

  – ある時刻tにおける値は、過去D時点分の値から決まる

  – モデルのパラメータは $(w_1, w_2, \ldots, w_D)$

自己回帰モデルの学習は、線形回帰としてみることができます

- D次の自己回帰モデルは、

$$x_t = w_1 x_{t-1} + w_2 x_{t-2} + \cdots + w_D x_{t-D}$$

- これは、特徴ベクトルを $(x_{t\text{-}1}, x_{t\text{-}2}, \ldots, x_{t\text{-}D})$ 、パラメータを$(w_1, w_2, \ldots, w_D)$と考えれば、まさに線形回帰のモデル
- 時刻1から時刻Tまでの時系列の値 $x_1, x_2, \ldots, x_T$ が与えられたときに時刻T+1の値を予測したいものとする
- 時刻Tまでの時系列 $x_1, x_2, \ldots, x_T$ から長さD+1の窓をずらしながらT-D 個の訓練データを作ることができる。
- これらから、線形回帰の方法を用いてパラメータ **w**を推定し、それをもとに x_T を予測する。
- 時系列予測については、より詳細なモデルや特化した解法がある

回帰で分類問題を解く

- 回帰を用いて分類問題（例えば、2値分類 $y \in \{+1,-1\}$ の場合）を解く
- 便宜的に各訓練データの出力を $y^{(i)} \in \{+1,-1\}$として回帰を適用する
  - 予測時には、$f(x; \mathbf{w})$ が0以上の値であるなら出力「+1」、そうでないならば「-1」として予測する
- 出力は+1か-1のどちらかなので、回帰の仮定（出力にガウス的ノイズが入る）は成立せず、このような適用は厳密には少しおかしい
- 分類問題をより適切にモデル化する方法は複雑になるため、分類問題を手軽に扱えるという意味で、このやり方は妥協に値する。
- なお、実際は訓練データの出力を $y^{(i)} \in \{+1,-1\}$とするのではなく、$y^{(i)} \in \{+1/|N_+|\}, -1/|N_-|\}$としたほうがよい
  - 出力が+1の訓練データの数を$|N_+|$、出力が-1の訓練データの数を$|N_-|$
  - フィッシャー判別に対応

カーネル回帰

データ数よりも次元数が大きい場合の回帰：データ数に依存した計算量にすることができます

- リッジ回帰の解は、以下で与えられた

$$\mathbf{w}^* = \left(\mathbf{\Phi}^\top \mathbf{\Phi} + \lambda \mathbf{I}\right)^{-1} \mathbf{\Phi}^\top \mathbf{y}$$

- これは次元数×次元数（$D \times D$）の行列の逆行列計算（もしくは連立方程式の解）が必要
  - $\mathrm{O}(D^3)$
- もしも（訓練データ数 N と比較して）次元数Dが大きい場合にはどうしたらよいだろうか？
- 実は、計算量が訓練データ数に依存するように問題を変換することができる
  - $\mathrm{O}(N^3)$

ここで便利なのが、ウッドベリーの公式です：逆行列計算のサイズを小さくできることがあります

- ウッドベリー(Woodbury)の公式：

$$\left(\mathbf{A}+\mathbf{U}\mathbf{C}\mathbf{V}^\top\right)^{-1}=\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{U}\left(\mathbf{C}^{-1}+\mathbf{V}^\top\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}^\top\mathbf{A}^{-1}$$

  - シャーマン・モリソンの公式の一般化になっている
    - C ≡ Iとして、UとVをベクトルとするとシャーマン-モリソンに一致する

$$\left(\mathbf{A}+\mathbf{u}\mathbf{v}^\top\right)^{-1}=\mathbf{A}^{-1}-\frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^\top\mathbf{A}^{-1}}{1+\mathbf{v}^\top\mathbf{A}^{-1}\mathbf{u}}$$

- 重要なポイントは：
  - LOOCVのときと同じく$\mathbf{A}^{-1}$が出てくるところ
  - **U**と**V**が縦長の行列の場合、右辺のほうが逆行列のサイズが小さいところ

ウッドベリーの公式を用いると、リッジ回帰の計算量を $O(D^3) \to O(N^3)$（次元の3乗からデータ数の3乗）にできます

- ウッドベリー

$$\left(\mathbf{A}+\mathbf{U}\mathbf{C}\mathbf{V}^\top\right)^{-1}=\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{U}\left(\mathbf{C}^{-1}+\mathbf{V}^\top\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}^\top\mathbf{A}^{-1}$$

  で$\mathbf{A} \equiv \lambda\,\mathbf{I}$、$\mathbf{U} \equiv \mathbf{V} \equiv \boldsymbol{\Phi}^\top$、$\mathbf{C} \equiv \mathbf{I}$とおいてみると、リッジ回帰の解における逆行列の部分を以下のように書き直せる

$$\left(\boldsymbol{\Phi}^\top\boldsymbol{\Phi}+\lambda\mathbf{I}\right)^{-1}=\frac{1}{\lambda}\mathbf{I}+\frac{1}{\lambda^2}\boldsymbol{\Phi}^\top\left(\mathbf{I}+\frac{1}{\lambda}\boldsymbol{\Phi}\boldsymbol{\Phi}^\top\right)^{-1}\boldsymbol{\Phi}$$

  - 左辺では$D \times D$の逆行列計算が、右辺では$N \times N$の逆行列計算
    - 両辺で逆行列の中の $\boldsymbol{\Phi}$と$\boldsymbol{\Phi}^\top$の順序が逆になっていることに注意
  - つまり、$D > N$ のときには、右辺のほうが計算量的に有利

- 右辺に出てくる行列 $\boldsymbol{\Phi}\boldsymbol{\Phi}^\top$は**グラム行列**と呼ばれ、その各要素は訓練データの特徴ベクトル間の内積になっている

- これはまさに**カーネル法**における**カーネル関数**であり、我々は「ウッドベリーの公式を用いて**カーネルリッジ回帰**を導いた」

L_1正則化

L_1正則化は、疎な解を与えるため、L_2正則化と並び重要な正則化のひとつです

- 1-ノルムを用いた正則化はL_1正則化と呼ばれる
  - L_2正則化と並び、よく利用される
- L_1正則化を用いた時の線形回帰の目的関数：

$$L(\mathbf{w}) = \| \mathbf{\Phi w} - \mathbf{y} \|_2^2 + \lambda \| \mathbf{w} \|_1$$

- L_1正則化は「疎な解」を与える
  - 目的関数を最小化する $\mathbf{w}^*$ において、多くの次元が丁度0になる
  - 特徴ベクトルϕの次元が非常に高く、その一方で、実際に予測に有用な特徴が少ない場合に特に有効
  - 学習済みのモデルを用いて予測を行う際、予測の計算量は $\mathbf{w}$ の 0 でない次元数に依存する

L_1正則化を用いた場合には（L_1正則化の場合と違い）最適化問題の解が閉じた形では求まりません

- L_1正則化を用いた回帰における最適化問題は、リッジ回帰のときと異なり、解が閉じた形で求まらないため、その解法は複雑になる
  - L_2正則化を用いた回帰（リッジ回帰）の解

$$\mathbf{w}^* = \left(\mathbf{\Phi}^\top \mathbf{\Phi} + \lambda \mathbf{I}\right)^{-1} \mathbf{\Phi}^\top \mathbf{y}$$

- 以降、2つの比較的簡単な方法を見ていくことにする
  - 方法1: 次元ごとの逐次解法
  - 方法2: L_2正則化への帰着

方法1: 次元ごとの逐次解法
1つのパラメータに注目した最適化を繰り返します

- L_1正則化項 $\| \mathbf{w} \|_1$ は：
  - 原点で微分できない
  - 原点が最適解となる場合が多い

  という問題があり、扱いづらい
- パラメータ $\mathbf{w}$ =($w_1, w_2, \ldots, w_D$)の、ある特定の次元 w_d に注目すると目的関数
  $L(\mathbf{w}) = \| \Phi\mathbf{w} - \mathbf{y} \|_2^2 + \lambda \| \mathbf{w} \|_1$ の最小化は簡単に行える
- そこで、
  1. 次元dを適当に選ぶ
  2. パラメータw_dについての最適化を行う

  を、収束するまで繰り返すアルゴリズムを考える

ある次元のパラメータ（w_d）に注目した目的関数を考えます

- i番目のデータに対するモデルの予測は、
  パラメータの d 次元目 w_d を特別扱いすれば：
  $$\mathbf{w}^\top \boldsymbol{\phi}(x^{(i)}) = w_d \phi_d(x^{(i)}) + \sum_{j \neq d} w_j \phi_j(x^{(i)})$$
- 目的関数を w_d についての関数であると思えば：
  $$L(w_d) = \sum_{i=1}^{N} \left(w_d \phi_d(x^{(i)}) - \left(y^{(i)} - \sum_{j \neq d} w_j \phi_j(x^{(i)}) \right) \right)^2 + \lambda |w_d| + \sum_{j \neq d} |w_j|$$
  - 残りのw_j ($j \neq d$)は定数であると思う
- 最後の項は w_d に関係ないので無視すると、最小化問題は：
  $$w_d^* = \operatorname*{argmin}_{w_d} \sum_{i=1}^{N} \left(w_d \phi_d(x^{(i)}) - \left(y^{(i)} - \sum_{j \neq d} w_j \phi_j(x^{(i)}) \right) \right)^2 + \lambda |w_d|$$

ひとまずは、正則化項を無視して解いてみます

- 以下の1変数最小化問題を解きたい

$$w_d^* = \underset{w_d}{\text{argmin}} \sum_{i=1}^{N} \left(w_d \phi_d(x^{(i)}) - \left(y^{(i)} - \sum_{j \neq d} w_j \phi_j(x^{(i)}) \right) \right)^2 + \lambda |w_d|$$

  - 厄介なのが最後の項 $\lambda\,|w_d|$
  - この項は$w_d = 0$ において、微分が定義されないために、単純に目的関数の微分を取って 0 とおいて…という方法が使えない

- ひとまず、最後の項を無視して：

$$\tilde{w}_d = \underset{w_d}{\text{argmin}} \sum_{i=1}^{N} \left(w_d \phi_d(x^{(i)}) - \left(y^{(i)} - \sum_{j \neq d} w_j \phi_j(x^{(i)}) \right) \right)^2$$

を解いてみると、この解は簡単に求まり：

$$\tilde{w}_d = \frac{\sum_{i=1}^{N} \left(y^{(i)} - \sum_{j \neq d} w_j \phi_j(x^{(i)}) \right) \phi_d(x^{(i)})}{\sum_{i=1}^{N} \phi_d(x^{(i)})^2}$$

正則化ナシの解を使って、目的関数を書き換えてみます

- これを用いて損失関数の部分を書き換えてみると、

$$w_d^* = \underset{w_d}{\mathrm{argmin}} \left(\sum_{i=1}^{N} \phi_d(x^{(i)})^2 \right) (w_d - \tilde{w}_d)^2 + \lambda |w_d|$$

  – 定数項は最小化に関係ないので無視した

- ここで、今後の表記を簡単にするために

$$\gamma \equiv \frac{\lambda}{2 \sum_{i=1}^{N} \phi_d(x^{(i)})^2}$$

  とおく

- γ を使って目的関数を書きなおすと：

$$w_d^* = \underset{w_d}{\mathrm{argmin}} \ \tilde{L}(w_d)$$

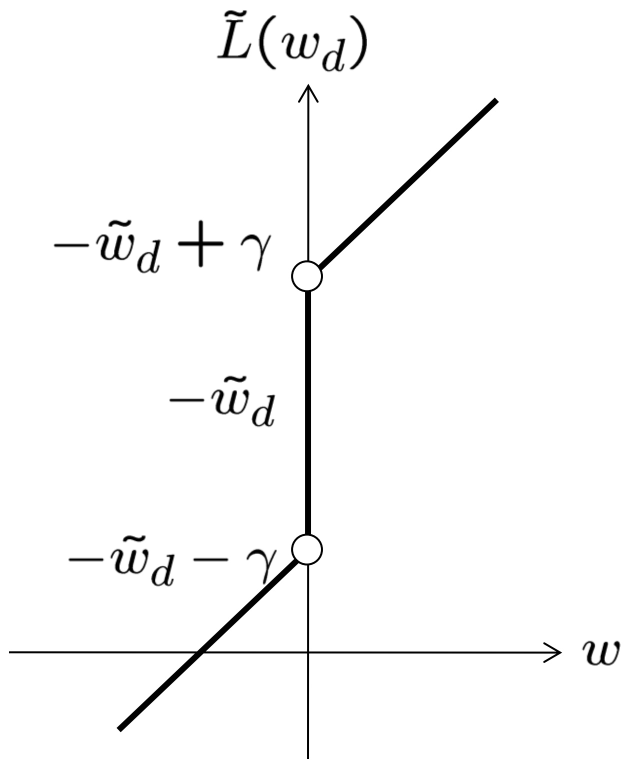
$$\tilde{L}(w_d) \equiv \frac{1}{2} (w_d - \tilde{w}_d)^2 + \gamma |w_d|$$

目的関数の微分を考えてみます（場合分け）

- $\tilde{L}(w_d) \equiv \frac{1}{2}(w_d - \tilde{w}_d)^2 + \gamma|w_d|$ の微分を計算してみると

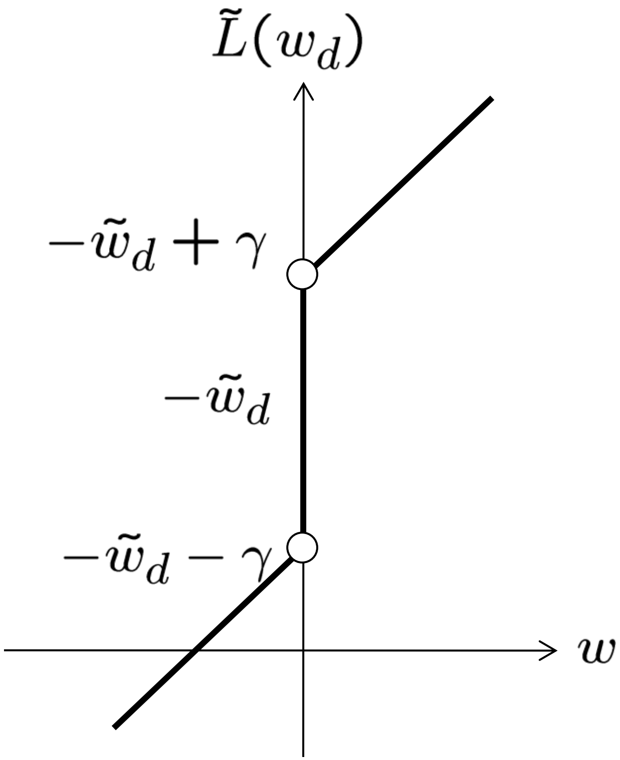
$$\frac{dL(w_d)}{dw_d} = \begin{cases} w_d - \tilde{w}_d + \gamma & (\text{if } w_d > 0) \\ w_d - \tilde{w}_d - \gamma & (\text{if } w_d < 0) \\ \text{undefined} & (\text{if } w_d = 0) \end{cases}$$

  - $w_d = 0$ のとき：
    - $|w_d|$が微分できないため定義されない
  - $w_d > 0$ のとき：
    - 傾き1で切片が$-\tilde{w}_d + \gamma$の一次関数
  - $w_d < 0$ のとき:
    - 傾き1で切片が$-\tilde{w}_d - \gamma$の一次関数
- これが0になるw_dを探す



求まった解を見てみると、確かに「疎」な傾向が見えてきます

- グラフが 0 と交わる w_d を探す
- 場合1：$-\tilde{w}_d + \gamma < 0$ すなわち $\tilde{w}_d > \gamma$ のときには解は $w_d^* = \tilde{w}_d - \gamma$
- 場合2：$-\tilde{w}_d - \gamma < 0$ すなわち $\tilde{w}_d < -\gamma$ のときには解は $w_d^* = \tilde{w}_d + \gamma$
- 場合3：$-\gamma \leq \tilde{w}_d \leq \gamma$ のとき
  - $w_d^* > 0$ とすると $w_d^* = \tilde{w}_d - \gamma \leq 0$ となり矛盾
  - $w_d^* < 0$ とすると $w_d^* = \tilde{w}_d + \gamma \geq 0$ となり矛盾
  - 解は必ず存在するので、従って $w_d^* = 0$ でないと困る
- 場合3を見ると、正則化ナシの解 $\tilde{w}_d$ が 0に近いところでは、L_1正則化の解が 0 になる（→ 疎になる）



方法2: L_2正則化（リッジ回帰）への帰着 簡単に解けるリッジ回帰の繰り返し解法に変換します

- L_1正則化項 $\| \mathbf{w} \|_1 \equiv \sum_{d=1}^{D} |w_d|$ を直接扱うのは若干煩雑
- 一方、L_2正則化を用いた回帰（リッジ回帰）は、逆行列の計算によって求まるのでシンプル

$$\mathbf{w}^* = \left(\mathbf{\Phi}^\top \mathbf{\Phi} + \lambda \mathbf{I}\right)^{-1} \mathbf{\Phi}^\top \mathbf{y}$$

- そこで、L_1正則化による回帰（ラッソ回帰）を、リッジ回帰の繰り返しによって解く方法を考える
- ポイントは、L_1正則化項の上界を、L_2正則化項を用いて作るところ
  - その代償として、パラメータが 2倍に増える

L_1正則化項の上界を、相加相乗平均によって作ります

- d 次元目についての正則化項 $|w_d|$ の上界を以下のように求める。

$$|w_d| = \sqrt{w_d^2} = \sqrt{\beta_d \frac{w_d^2}{\beta_d}} \leq \frac{1}{2}\left(\beta_d + \frac{w_d^2}{\beta_d}\right)$$

- 新たなパラメータ $\beta_d \geq 0$ が導入されていることに注意
- 最後の不等号は相加相乗平均、すなわち $a, b \geq 0$について

$$\sqrt{ab} \leq \frac{a+b}{2}$$

できた上界はL_2ノルムで書かれていることが分かります（パラメータの数は増えてしまいます）

- L_1正則化項 の上界は：

$$\| \mathbf{w} \|_1 \leq \sum_{d=1}^{D} \beta_d + \sum_{d=1}^{D} \frac{w_d^2}{\beta_d}$$

- 特筆すべきは、上界においては各パラメータの2乗が現れるため、微分不可能な点が無くなっている点
- 新たなパラメータ $\boldsymbol{\beta} \equiv (\beta_1, \beta_2, \ldots, \beta_D)$ を導入したことにより、$\mathbf{w}$と併せて、モデルパラメータの数は D から $2D$ に増加
- 従って、左辺の $\|\mathbf{w}\|_1$を $\mathbf{w}$ について最小化するかわりに、右辺を $\mathbf{w}$と $\boldsymbol{\beta}$ の両方について最小化することになる

増えたパラメータをまとめて最適化する必要があります

- L_1正則化を用いた時の線形回帰の目的関数は

$$L(\mathbf{w}) \equiv \| \mathbf{\Phi w} - \mathbf{y} \|_2^2 + \lambda \| \mathbf{w} \|_1$$

- $L(\mathbf{w})$ の上界 $\tilde{L}(\mathbf{w})$ を作ると：

$$L(\mathbf{w}) \leq \| \mathbf{\Phi w} - \mathbf{y} \|_2^2 + \lambda \left(\sum_{d=1}^{D} \beta_d + \sum_{d=1}^{D} \frac{w_d^2}{\beta_d} \right) \equiv \tilde{L}(\mathbf{w}, \boldsymbol{\beta})$$

- 最適なパラメータ$(\mathbf{w}^*, \boldsymbol{\beta}^*)$は、

$$(\mathbf{w}^*, \boldsymbol{\beta}^*) = \underset{\mathbf{w}, \boldsymbol{\beta}}{\operatorname{argmin}} \ \tilde{L}(\mathbf{w}, \boldsymbol{\beta})$$

によって求まることになる

アルゴリズム： $\mathbf{w}$ と β の推定を交互に行います

- アルゴリズムとしては、以下の2ステップを繰り返す
  - A) （βを現在の値で固定しておいて）$\mathbf{w}$ についての最小化
  - B) （$\mathbf{w}$を現在の値で固定しておいて）β についての最小化

$$(\mathbf{w}^*, \beta^*) = \underset{\mathbf{w},\beta}{\operatorname{argmin}} \; \tilde{L}(\mathbf{w}, \beta)$$

$$\tilde{L}(\mathbf{w}, \beta) \equiv \| \, \boldsymbol{\Phi}\mathbf{w} - \mathbf{y} \, \|_2^2 + \lambda \left(\sum_{d=1}^{D} \beta_d + \sum_{d=1}^{D} \frac{w_d^2}{\beta_d} \right)$$

ステップ A：β についての最小化は簡単です

- $\boldsymbol{w}$を固定したうえで、$\boldsymbol{\beta}$ についての最小化を行う
- 相加相乗平均

$$\sqrt{ab} \leq \frac{a+b}{2}$$

  の等号が成り立つのは、$a=b$のときである
- したがって、$\beta_d = w_d^2/\beta_d$ のとき、すなわち $\beta_d = |w_d|$ のときに上界

$$|w_d| \leq \frac{1}{2}\left(\beta_d + \frac{w_d^2}{\beta_d}\right)$$

  は最小化される

ステップB：
wについての最小化はリッジ回帰によって行えます

- 目的関数は $\tilde{L}(\mathbf{w},\boldsymbol{\beta}) \equiv \| \boldsymbol{\Phi}\mathbf{w} - \mathbf{y} \|_2^2 + \lambda \left(\sum_{d=1}^{D} \beta_d + \sum_{d=1}^{D} \frac{w_d^2}{\beta_d} \right)$

  $\tilde{w}_d \equiv \frac{w_d}{\sqrt{\beta_d}}$ と置く

$$\tilde{L}(\mathbf{w},\boldsymbol{\beta}) = \| \boldsymbol{\Phi}\mathrm{diag}\left(\sqrt{\boldsymbol{\beta}}\right)\tilde{\mathbf{w}} - \mathbf{y} \|_2^2 + \lambda \left(\sum_{d=1}^{D} \beta_d + \sum_{d=1}^{D} \tilde{w}_d^2 \right)$$

  - diag(・)は、引数のベクトルを対角成分に持つような行列
  - $\tilde{\mathbf{w}} \equiv (\tilde{w}_1, \tilde{w}_2, \ldots, \tilde{w}_D)$
- $\boldsymbol{\Phi}\mathrm{diag}(\sqrt{\boldsymbol{\beta}})$ を新たなデザイン行列と見ることによって、$\tilde{\mathbf{w}}$ についてのリッジ回帰として捉えることができる
- $\tilde{\mathbf{w}}$ について最適化を行い、$\mathbf{w}$ に戻せば、

$$\mathbf{w} = \mathrm{diag}\left(\sqrt{\boldsymbol{\beta}}\right)\left(\mathrm{diag}\left(\sqrt{\boldsymbol{\beta}}\right)\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\mathrm{diag}\left(\sqrt{\boldsymbol{\beta}}\right) + \lambda\mathbf{I}\right)^{-1}\boldsymbol{\Phi}^\top\mathbf{y}$$