最小二乗法

山下 智樹

2025 年 9 月 9 日

目次

1 線形基底関数モデル

ここではパターン認識と機械学習 (PRML)[1] の線形基底関数モデルを用いた最小二乗法に関してまとめる．入力変数 (ベクトル) を $\mathbf{x}$ として，M 個の基底関数 (basis function) $\boldsymbol{\phi}(\mathbf{x}) = (\phi_0(\mathbf{x})\ \phi_1(\mathbf{x})\ \cdots\ \phi_{M-1}(\mathbf{x}))^T$ を用いた，線形基底関数モデル

$$y(\mathbf{x}, \mathbf{w}) = \sum_{i=0}^{M-1} w_i \phi_i(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) \tag{1}$$

を考える．ここで，$\phi_0(\mathbf{x}) = 1$ であり，w_0 はバイアスパラメータ (bias parameter) と呼ばれる定数項である．非線形な基底関数を用いると，$y(\mathbf{x}, \mathbf{w})$ は入力ベクトル $\mathbf{x}$ に関して非線形になるが，パラメータ $\mathbf{w}$ に関しては線形なので，線形モデル (linear model) と呼ばれる．データが N 個あるとすると，i 番目の入力変数 $\mathbf{x}_i$ に対するモデルの値 $y_i(\mathbf{x}_i, \mathbf{w})$ は，内積は変数の順番を入れ替えても変わらないので，

$$y_i(\mathbf{x}_i, \mathbf{w}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) \tag{2}$$

$$= \boldsymbol{\phi}(\mathbf{x}_i)^T \mathbf{w}. \tag{3}$$

1 番目から N 番目までの y をまとめてベクトル表記で $\mathbf{y} = (y_1\ y_2\ \cdots\ y_N)^T$ と書けば，データの個数分縦に並べて書くことで，

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix} \begin{pmatrix} w_0 \\ \vdots \\ w_{M-1} \end{pmatrix}. \tag{4}$$

最後の $\mathbf{w}$ に関しては，縦に N 個ではなく M 個並んだベクトルであるので，わざと高さをずらして書いた．中央の $N \times M$ 行列は計画行列 (design matrix) と呼ばれ，Φ で表せば，

$$\mathbf{y} = \Phi \mathbf{w} \tag{5}$$

のようにすっきり書ける．

ここでやりたいことは，N 個のデータに線形モデルが合うようにパラメータ $\mathbf{w}$ を決めることである．目的となる N 個のデータ，つまり目的変数をまとめてベクトル表記で $\mathbf{t} = (t_1\ t_2\ \cdots\ t_N)^T$ と書くと目的変数 $\mathbf{t}$ とモデルの値 $\mathbf{y}$ の二乗和誤差 (sum of squared errors) は

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}\left\{\mathbf{w}^T\Phi(\mathbf{x}_i) - t_i\right\}^2 \tag{6}$$

と書ける．係数の 1/2 は最大最小には関係ないので，計算が便利になるようにつけられている．この二乗和誤差を最小にするように $\mathbf{w}$ を決定する．ちなみに，加法性のガウスノイズを考えた尤度関数を最大化する手法 (最尤推定) でも，ここで扱う計算と全く同じやり方でパラメータ $\mathbf{w}$ が同様に決まるがここでは触れない．以下では，同じ計算をしているのだが，二つの解き方で二乗和誤差 $E(\mathbf{w})$ の最小化を考える．

2 正規方程式の導出

機械学習の表記などにあまり慣れてなくてもわかるような導出方法をここでまとめる．慣れている人は次のセクションの方が導出がシンプルでわかりやすいかもしれない．

二乗和誤差 $E(\mathbf{w})$ は次の様に書ける．

$$E(\mathbf{w}) = \frac{1}{2}||\Phi\mathbf{w} - \mathbf{t}||^2. \tag{7}$$

$(\Phi\mathbf{w} - \mathbf{t})$ は N 次元のベクトルであり，例えば第 i 成分は

$$\mathbf{w}^T\Phi(\mathbf{x}_i) - t_i \tag{8}$$

である．式 (7) は係数の 1/2 を別にすれば，このベクトルの長さ (L^2 ノルム) の 2 乗であるから，これは各成分の 2 乗和を計算すればよく，式 (6) と等価である．ベクトルの長さの 2 乗はそのベクトル同士の内積であるので，もう少し式 (7) を変形して

$$\frac{1}{2}||\Phi\mathbf{w} - \mathbf{t}||^2 = \frac{1}{2}(\Phi\mathbf{w} - \mathbf{t})^T(\Phi\mathbf{w} - \mathbf{t}) \tag{9}$$

$$= \frac{1}{2}(\mathbf{w}^T\Phi^T - \mathbf{t}^T)(\Phi\mathbf{w} - \mathbf{t}) \tag{10}$$

$$= \frac{1}{2}(\mathbf{w}^T\Phi^T\Phi\mathbf{w} - \mathbf{w}^T\Phi^T\mathbf{t} - \mathbf{t}^T\Phi\mathbf{w} + \mathbf{t}^T\mathbf{t}) \tag{11}$$

ここで転置行列の性質 $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$ と $(\mathbf{A} + \mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T$ を用いた．式 (11) の第 2 項と第 3 項はスカラーであり，転置を取っても変わらないので

$$\mathbf{t}^T\Phi\mathbf{w} = (\mathbf{t}^T\Phi\mathbf{w})^T \tag{12}$$

$$= (\Phi\mathbf{w})^T\mathbf{t} \tag{13}$$

$$= \mathbf{w}^T\Phi^T\mathbf{t} \tag{14}$$

となり，式 (11) の第 2 項と第 3 項は等しくなる．よって式 (11) は

$$\frac{1}{2}||\Phi\mathbf{w} - \mathbf{t}||^2 = \frac{1}{2}(\mathbf{w}^T\Phi^T\Phi\mathbf{w} - 2\mathbf{w}^T\Phi^T\mathbf{t} + \mathbf{t}^T\mathbf{t}) \tag{15}$$

となり，これを $\mathbf{w}$ で微分，つまり成分ごとに微分して勾配を $\mathbf{0}$ とおく．ここで，第 1 項に着目すると，中央の $\Phi^T\Phi$ は $(\Phi^T\Phi)^T = \Phi^T\Phi$ より，$(M \times M)$ 対称行列である．これを $S = \Phi^T\Phi$ とすると，$\mathbf{w}^TS\mathbf{w}$ は 2 次形式である．対称行列の 2 次形式の微分は，

$$\frac{\partial(\mathbf{w}^TS\mathbf{w})}{\partial\mathbf{w}} = 2S\mathbf{w} \tag{16}$$

となる.[*1]式 (15) 第 2 項の $\Phi^T\mathbf{t}$ の部分は $\mathbf{w}$ が入っていない普通の M 次元ベクトルであるので，$\mathbf{z} = \Phi^T\mathbf{t}$ とすると，$\frac{\partial(\mathbf{w}^T\mathbf{z})}{\partial\mathbf{w}} = \mathbf{z}$ の形なので単純に微分できて

$$-2\frac{\partial(\mathbf{w}^T\Phi^T\mathbf{t})}{\partial\mathbf{w}} = -2\Phi^T\mathbf{t}. \tag{17}$$

式 (15) 第 3 項は微分すると $\mathbf{0}$ になるので，結局のところ式 (15) を微分して $\mathbf{0}$ とおくと

$$S\mathbf{w} - \Phi^T\mathbf{t} = 0 \tag{18}$$

$$\Phi^T\Phi\mathbf{w} = \Phi^T\mathbf{t} \tag{19}$$

$$\mathbf{w} = \left(\Phi^T\Phi\right)^{-1}\Phi^T\mathbf{t} \tag{20}$$

が得られる．この式は最小二乗問題の正規方程式 (normal equation) として知られる．行列

$$\Phi^\dagger \equiv \left(\Phi^T\Phi\right)^{-1}\Phi^T \tag{21}$$

は行列 Φ のムーア-ペンローズの擬似逆行列と呼ばれる．行列が正方ではないとき，逆行列は定義されないが，逆行列の概念を非正方行列へ拡張した擬似逆行列であれば定義できる．

実際数値計算を行う際は，式 (20) のように逆行列を直接計算するのではなく，式 (19) をコレスキー分解などを使って解いた方が計算量が少なくて済む．式 (19) は $\mathbf{Ax} = \mathbf{b}$ という一般的な連立方程式の形をしているので，ライブラリを使えば高速に解ける．しかしながら，$\Phi^T\Phi$ が非正則に近いときは，数値計算上不安定になることがあるので，実際にこの問題を解く際は QR 分解が使われている．また，$\Phi^T\Phi$ が常に正則 (逆行列が存在する) であるとは限らないので，正則でない時は特異値分解 (SVD: singular value decomposition) や正則化項を加えることによって解決させる．SVD を使って最小二乗解を求めるライブラリがあるので（Python の numpy だと numpy.linalg.lstsq)，結局のところこれを使うと数値計算上安定に $\mathbf{w}$ を求められる．

3 正規方程式の導出 2

PRML[1] の導出は少しわかりづらいところがあるので，わかりやく導出をまとめておく.[*2]前のセクションの導出より，スッキリ美しく導出できる．

今考えている二乗和誤差 $E(\mathbf{w})$ は以下の式であった.

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}\left\{\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i) - t_i\right\}^2. \tag{22}$$

データの個数分二乗の項が出てくるが，書き下せば結局ただの 2 次式なので，最小化は勾配をとって $\mathbf{0}$ とすれば良い．$\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i)$ の部分はただの内積なのでベクトルの順序を $\boldsymbol{\phi}(\mathbf{x}_i)^T\mathbf{w}$ のように入れ替えておく．勾配の第 j 成分は，いつもの 2 次関数の微分と同じようにできて，$\frac{\partial}{\partial w_j}\left\{\boldsymbol{\phi}(\mathbf{x}_i)^T\mathbf{w}\right\} = \phi_j(\mathbf{x}_i)$ を用いれば,

$$\frac{\partial E}{\partial w_j} = \frac{1}{2}\sum_{i=1}^{N}2\phi_j(\mathbf{x}_i)\left\{\boldsymbol{\phi}(\mathbf{x}_i)^T\mathbf{w} - t_i\right\}. \tag{23}$$

式（4）を見るとわかるように，$\phi_j(\mathbf{x}_i)$ は計画行列 Φ の i 行 j 列成分，Φ_{ij} である．また，$\boldsymbol{\phi}(\mathbf{x}_i)^T\mathbf{w}$ は i 番目のデータに対する予測モデルの値であるから，これを y_i と書けば,

$$\frac{\partial E}{\partial w_j} = \sum_{i=1}^{N}\Phi_{ij}\left(y_i - t_i\right) \tag{24}$$

[*1] 別ノート「機械学習のための線形代数学」を参照

[*2] PRML[1] の (3.14) 式では，縦ベクトルではなく横ベクトルを用いて，
$\mathbf{0} = \sum_{n=1}^{N} t_n\boldsymbol{\phi}(\mathbf{x}_n)^T - \mathbf{w}^T\left(\sum_{n=1}^{N}\boldsymbol{\phi}(\mathbf{x}_n)\boldsymbol{\phi}(\mathbf{x}_n)^T\right)$ と表現しているが，このあたりの計算が少しわかりづらい気がする．

と書ける．さらに，$\mathbf{v} = \mathbf{y} - \mathbf{t}$ とおけば，$\mathbf{v}$ の i 成分は $v_i = y_i - t_i$ であるから，

$$\frac{\partial E}{\partial w_j} = \sum_{i=1}^{N} \Phi_{ij} v_i \tag{25}$$

のように整理できる．右辺の $\sum_{i=1}^{N} \Phi_{ij} v_i$ は，行列 Φ^T とベクトル $\mathbf{v}$ の積の j 成分であるから，

$$\frac{\partial E}{\partial w_j} = \sum_{i=1}^{N} \Phi_{ij} v_i = \left(\Phi^T \mathbf{v}\right)_j \tag{26}$$

となる．これで左辺も右辺も単純に j 成分だけを見ているので，ベクトル表記に戻せば，

$$\frac{\partial E}{\partial \mathbf{w}} = \Phi^T \mathbf{v} \tag{27}$$

$$= \Phi^T (\mathbf{y} - \mathbf{t}) \tag{28}$$

$$= \Phi^T \Phi \mathbf{w} - \Phi^T \mathbf{t} \tag{29}$$

となる．ここで，$\mathbf{y} = \Phi \mathbf{w}$ を用いた．これをゼロとおけば，

$$\Phi^T \Phi \mathbf{w} = \Phi^T \mathbf{t} \tag{30}$$

$$\mathbf{w} = \left(\Phi^T \Phi\right)^{-1} \Phi^T \mathbf{t}. \tag{31}$$

となって，正規方程式が得られた．

参考文献

[1] C. M. ビショップ，パターン認識と機械学習（丸善出版，2012）．