情報数理科学 VII　　教師あり学習

松島　慎

2019 年 10 月 22 日

教師あり学習の問題設定

$(\mathbf{x}_i, y_i)$ の組がいくつか与えられている時 $(i = 1, 2, \ldots, n)$、未知のデータ $\mathbf{x}_{\text{未知}}$ に対する $y_{\text{未知}}$ を予測したい。どのように選べばよいか？

1　行列やベクトルの復習

データは行列やベクトルで表現される。まずは行列やベクトルの演算の基本を復習する。$\mathbf{x}_i \in \mathbb{R}^d$ というのはデータ $\mathbf{x}_i$ が d 次元実ベクトルで表されるということ。

$$\mathbf{x}_i = \begin{bmatrix} x_{i1} \\ \vdots \\ x_{id} \end{bmatrix}, \mathbf{x}_i^\top = \begin{bmatrix} x_{i1}, \ldots, x_{id} \end{bmatrix},$$

$X \in \mathbb{R}^{n \times d}$ というのはデータ X が $n \times d$ 行列で表されるということ。

$$X = \begin{bmatrix} \mathbf{x}_1^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix} = \underbrace{\begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1d} \\ \vdots & \ddots & & \\ x_{n1} & & & x_{nd} \end{bmatrix}}_{\text{計画行列（design matrix）}},$$

$$X^\top = \begin{bmatrix} \mathbf{x}_1, & \ldots, & \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} x_{11} & x_{21} & \ldots & x_{n1} \\ \vdots & \ddots & & \\ x_{1d} & & & x_{nd} \end{bmatrix}.$$

- 内積
  $\mathbf{x}$ の第 j 要素を x_j とする。$\mathbf{w}$ の第 j 要素を w_j とする。

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}, \mathbf{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$$

  $\mathbf{w}$ と $\mathbf{x}$ の内積を $\sum_{j=1}^d w_j x_j$ で定義する。この講義では $\mathbf{w}^\top \mathbf{x}$ と書くことが多い。
- 行列積

$$A = \begin{bmatrix} \mathbf{a}_1^\top \\ \vdots \\ \mathbf{a}_l^\top \end{bmatrix} = \begin{bmatrix} a_{11} & \ldots & a_{1m} \\ \vdots & \ddots & \\ a_{l1} & & a_{lm} \end{bmatrix},$$

$$B = \begin{bmatrix} \mathbf{b}_1, & \ldots, & \mathbf{b}_n \end{bmatrix} = \begin{bmatrix} b_{11} & \ldots & b_{1n} \\ \vdots & \ddots & \\ b_{m1} & & b_{mn} \end{bmatrix},$$

のとき

$$AB = \begin{bmatrix} \sum_i a_{1i} b_{j1} & \ldots & \sum_i a_{1i} b_{in} \\ \vdots & \ddots & \\ \sum_i a_{li} b_{i1} & & \sum_i a_{li} b_{in} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^\top \mathbf{b}_1 & \ldots & \mathbf{a}_1^\top \mathbf{b}_n \\ \vdots & \ddots & \\ \mathbf{a}_l^\top \mathbf{b}_1 & & \mathbf{a}_l^\top \mathbf{b}_n \end{bmatrix},$$

$$A = \begin{bmatrix} \mathbf{a}_1, & \ldots, & \mathbf{a}_m \end{bmatrix}, B = \begin{bmatrix} \mathbf{b}_1^\top \\ \vdots \\ \mathbf{b}_m^\top \end{bmatrix}$$

のとき

$$AB = \sum_i \mathbf{a}_i \mathbf{b}_i^\top$$

- 行列の正則性
  行列 $A \in \mathbb{R}^{n \times n}$ が正則であるとは $BA = AB = I$ となる行列 $B \in \mathbb{R}^{n \times n}$ が存在すること（A^{-1} とかく）
  行列 $A \in \mathbb{R}^{n \times n}$ が正則でない $\Leftrightarrow$ $\mathbf{0}$ でないベクトル $\mathbf{x}$ が存在し $A\mathbf{x} = \mathbf{0}$ を満たす。
- 線形方程式
  $A\mathbf{x} = \mathbf{b}$ の解は、A が正則な時 $\mathbf{x} = A^{-1}\mathbf{b}$ が唯一の解。
  A が非正則な場合、解が存在しないか、解に自由度がある。
  （例）$A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ のとき解は存在しない。同じ A で $\mathbf{b} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ の時、$\mathbf{x} = \begin{bmatrix} 1 \\ c \end{bmatrix}$ であれば（c は何でもよい）解となる。

- L_2 ノルム

$$\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^\top \mathbf{x}} = \sqrt{\sum_{j=1}^{d} x_j^2}$$

常に非負であり、$\mathbf{x} = \mathbf{0}$ の時に限り $\|\mathbf{x}\|_2 = 0$ である。
特に断らずに $\|\mathbf{x}\|$ と書いたら L_2 ノルムのこと

- L_1 ノルム

$$\|\mathbf{x}\|_1 = \sum_{j=1}^{d} |x_j|$$

常に非負であり、$\mathbf{x} = \mathbf{0}$ の時に限り $\|\mathbf{x}\|_1 = 0$ である。

- 勾配
$J : \mathbb{R}^d \to \mathbb{R}$ に対し勾配を以下で定義する。

$$\nabla J(\mathbf{w}) = \begin{bmatrix} \frac{\partial J(\mathbf{w})}{\partial w_1} \\ \vdots \\ \frac{\partial J(\mathbf{w})}{\partial w_d} \end{bmatrix}.$$

どの変数に関する微分かわかりにくいときは $\nabla_{\mathbf{w}} J(\mathbf{w})$ などと書く。
(例)
$J(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$ の時、$\nabla J(\mathbf{w}) = \mathbf{w}$ である。

- 勾配の性質
$f : \mathbb{R} \to \mathbb{R}$ に対し、

$$\nabla_{\mathbf{x}} f(\mathbf{a}^\top \mathbf{x}) = f'(\mathbf{a}^\top \mathbf{x})\mathbf{a}$$

. $f : \mathbb{R}^m \to \mathbb{R}$、$A \in \mathbb{R}^{m \times n}$ と $x \in \mathbb{R}^n$ に対し、

$$\nabla_{\mathbf{x}} f(A\mathbf{x}) = A^\top \nabla_{\mathbf{u}} f(\mathbf{u})\,|_{\mathbf{u}=A\mathbf{x}}$$

2 線形回帰の最小二乗法

最も簡単な例として線形回帰の最小二乗法を考える。

- 回帰問題
$\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$ が与えられたとき、未知の $\mathbf{x} \in \mathbb{R}^d$ から $\hat{y} \in \mathbb{R}$ を予測したい。

- 線形モデル
$\mathbb{R}^d$ から $\mathbf{w}$ を一つ選び未知の $\mathbf{x}$ に対し、$\hat{y}$ を以下の式で予測する

$$\hat{y} = \sum_{j=1}^{d} w_j x_j = \mathbf{w}^\top \mathbf{x}$$

線形モデルの学習とは得られたデータに「最もふさわしい」関数を以下の関数の集合から一つ選ぶこと。

$$F = \left\{ f : \mathbf{x} \mapsto \mathbf{w}^\top \mathbf{x} | \mathbf{w} \in \mathbb{R}^d \right\}$$

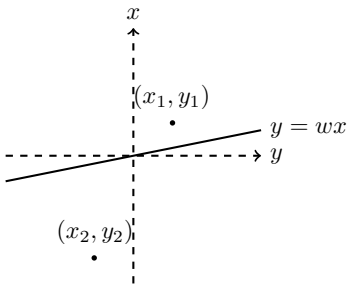

図 1　$J(\mathbf{w})$ のイメージ

線形モデルを用いることで「どのように予測すればよいか？」が「どのような $\mathbf{w}$ を選べばよいか？」へ帰着する。選ばれた関数を予測関数または予測器という。他の数理モデル（関数の集合）を考えることもできる（ バイアス項（切片）は？）が、しばらく線形モデルに注目する。

- 最小二乗法
以下の目的関数 $J(\mathbf{w})$ を最小化するような $\mathbf{w}$ を一つ選び、それを予測器とする。

$$\begin{aligned} J(\mathbf{w}) &= \sum_i \left(y_i - \sum_j w_j x_{ij} \right)^2 \\ &= \sum_i \left(y_i - \mathbf{w}^\top \mathbf{x}_i \right)^2 \\ &= \|\mathbf{y} - X\mathbf{w}\|_2^2 . \end{aligned}$$

ここで $\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n$ である。

- $X^\top X$ が正則である場合
勾配が $\mathbf{0}$ になる点が以下のように表される。

$$\nabla J(\mathbf{w}^*) = \mathbf{0} \to \overbrace{X^\top X}^{\text{グラム行列 (GramianMatrix)}} \mathbf{w}^* = X^\top \mathbf{y}$$
$$\to \mathbf{w}^* = \left(X^\top X \right)^{-1} X^\top \mathbf{y}$$

これが最小解であることは $J(\mathbf{w}^* + \mathbf{w}) = \|\mathbf{y} - X\mathbf{w}^*\|^2 + \|X\mathbf{w}\|^2$ であることからわかる。

$$
\begin{aligned}
J(\mathbf{w}) &= \sum_i \left(y_i - \sum_j w_j x_{ij} \right)^2 \\
\frac{\partial J(\mathbf{w})}{\partial w_{j'}} &= \frac{\partial}{\partial w_{j'}} \sum_i \left(y_i - \sum_j w_j x_{ij} \right)^2 \\
&= \sum_i -2 x_{ij'} \left(y_i - \sum_j w_j x_{ij} \right) \\
&= -2 \underbrace{\sum_i x_{ij'} y_i}_{X^\top \mathbf{y} \text{ の第 } j' \text{要素}} + 2 \underbrace{\sum_i \sum_j x_{ij'} x_{ij} w_j}_{X^\top X \mathbf{w} \text{の第 } j' \text{要素}} \\
&\Rightarrow \nabla J(\mathbf{w}) = -2X^\top \mathbf{y} + 2X^\top X \mathbf{w}
\end{aligned}
$$

$$
\begin{aligned}
J(\mathbf{w}) &= \sum_i (y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle)^2 \\
\nabla J(\mathbf{w}) &= \sum_i \left(2(y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle) \right) \mathbf{x}_i. \\
&(\nabla_{\mathbf{x}} F(\langle \mathbf{a}, \mathbf{x} \rangle) = F'(\langle \mathbf{a}, \mathbf{x} \rangle)\mathbf{a}) \\
&= 2 \sum_i \mathbf{x}_i y_i - 2 \left(\sum_i \mathbf{x}_i \mathbf{x}_i^\top \right) \mathbf{w}. \\
&= 2 \begin{bmatrix} \mathbf{x}_1 \dots \mathbf{x}_n \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} - 2 \begin{bmatrix} \mathbf{x}_1 \dots \mathbf{x}_n \end{bmatrix} \begin{bmatrix} \mathbf{x}_1^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix} \mathbf{w}.
\end{aligned}
$$

$$
\begin{aligned}
J(\mathbf{w}) &= \|\mathbf{y} - X\mathbf{w}\|_2^2. \\
\nabla J(\mathbf{w}) &= X^\top \left(2(\mathbf{y} - X\mathbf{w}) \right). \\
&\left(\nabla_{\mathbf{x}} F(A\mathbf{x}) = A^\top \nabla_{\mathbf{u}} F(\mathbf{u}) \mid_{\mathbf{u}=A\mathbf{x}}, \nabla_{\mathbf{x}} \|\mathbf{x}\|_2^2 = 2\mathbf{x} \right)
\end{aligned}
$$

- $X^\top X$ が正則でない場合
  最小解は複数存在する。すなわち、$\mathbf{w}^*$ が与えられたとき、$\mathbf{0}$ でない $\mathbf{w}'$ が存在して、任意の c に対して $J(\mathbf{w}^* + c\mathbf{w}') = \|\mathbf{y} - X\mathbf{w}^*\|^2 + \|cX\mathbf{w}'\|^2 = \|\mathbf{y} - X\mathbf{w}^*\|^2 = J(\mathbf{w}^*)$ となる。

3 過学習と正則化

学習の結果得られた関数は未知のデータを正しく予測できるだろうか？本講義では深入りしないが学習の標準的な評価は以下のような統計モデルの下で行う。すなわち、与えられたデータ $(\mathbf{x}_i, y_i)$ と新しく来るデータ $(\mathbf{x}_{\text{test}}, y_{\text{test}})$ は同じ分布から独立に生成されると考える。この時学習の本当の目的は汎化誤差

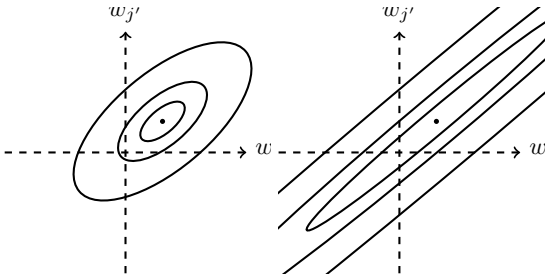

図 2　$J(\mathbf{w})$ のイメージ

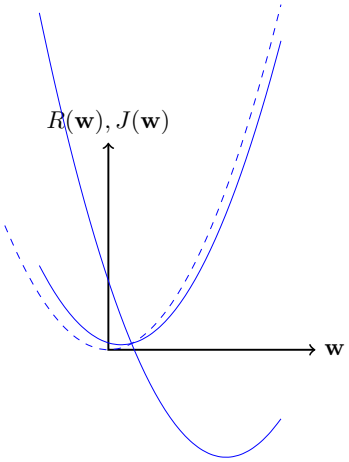

図 3　汎化誤差と経験誤差のイメージ

$(R(\mathbf{w}) = \mathbb{E}_{\mathbf{x}_{\text{test}}, y_{\text{test}}} \left(y_{\text{test}} - \mathbf{w}^\top \mathbf{x}_{\text{test}} \right)^2)$ を最小化する $\mathbf{w}$ を求めることであると考えることができる。

最小二乗法で最小化した目的関数は汎化誤差の近似であると考えられる。そのような文脈では $R(\mathbf{w}) = \frac{1}{n} \sum_i \left(y_i - \mathbf{w}^\top \mathbf{x}_i \right)^2$ のことを経験誤差という。データの数が限りなく増えていけば、経験誤差の近似はよくなっていく。汎化誤差の最小解を $\mathbf{w}^*$、経験誤差の最小解を $\hat{\mathbf{w}}$ と表すとすると、$\hat{\mathbf{w}}$ は $\mathbf{w}^*$ に近づいていく。データ数が一定である場合、関数空間が悪いと経験誤差と汎化誤差の近似が悪くて、経験誤差が低くても汎化誤差が高くなってしまうことがある。このような現象を過学習という。

過学習を防ぐために正則化を行うことが多い。経験誤差と汎化誤差の差を減らすために関数空間自体の大きさを小さくする。

$$
F_\tau = \left\{ f : \mathbf{x} \mapsto \mathbf{w}^\top \mathbf{x} | \, \|\mathbf{w}\|_2 \leq \tau \right\}
$$

とすると τ を小さくすることで過学習が抑えられる。適切に τ を定めて以下の最適化問題を解く。

- Ivanov 型正則化
  –

$$
J(\mathbf{w}) = \|\mathbf{y} - X\mathbf{w}\|_2^2
$$

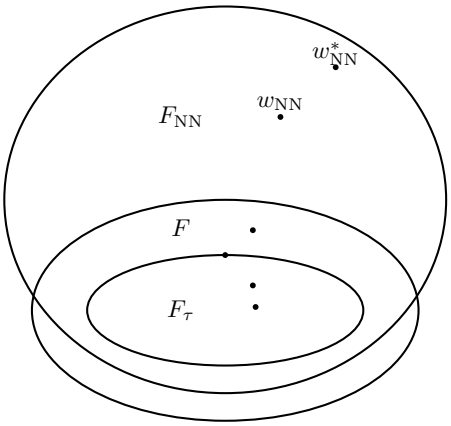

図 4　汎化誤差と経験誤差のイメージ

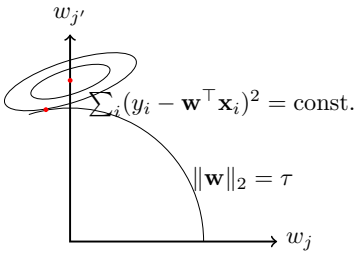

図 5　正則化のイメージ

を制約 $\|\mathbf{w}\|_2 \leq \tau$ のもとで最小化

これではやはり解が複数存在する場合がある。適切に λ を定めて以下の最適化問題を解く。

- Tikhonov 型正則化
  - リッジ線形回帰

$$J(\mathbf{w}) = \|\mathbf{y} - X\mathbf{w}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$
$$\Rightarrow \mathbf{w}^* = \left(X^\top X + \lambda I\right)^{-1} X\mathbf{y}$$

- Ivanov 型正則化と Tikhonov 型正則化にはある種の同値性がある

$$\mathbf{w}^*_\lambda = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{y} - X\mathbf{w}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$
$$\Rightarrow -2X^\top y + 2X^\top X\mathbf{w}^*_\lambda + \lambda \mathbf{w}^*_\lambda = \mathbf{0}$$

$$\mathbf{w}^*_\tau = \underset{\|\mathbf{w}\|_2^2 \leq \tau}{\operatorname{argmin}} \|\mathbf{y} - X\mathbf{w}\|_2^2$$
$$\Rightarrow \mathbf{w}^*_\tau = \underset{\mathbf{w}}{\operatorname{argmin}} \max_{\alpha \geq 0} \|\mathbf{y} - X\mathbf{w}\|_2^2 - \alpha(\tau - \|\mathbf{w}\|_2^2)$$
$$\Rightarrow \exists \alpha^* \geq 0, \mathbf{w}^*_\tau = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{y} - X\mathbf{w}\|_2^2 - \alpha^*(\tau - \|\mathbf{w}\|_2^2)$$
$$\Rightarrow -2X^\top y + 2X^\top X\mathbf{w}^*_\tau + 2\alpha^*\mathbf{w}^*_\tau = \mathbf{0}$$

まとめ

- 過学習
  経験誤差は低いのに汎化誤差が高くなってしまうこと。
- 未学習
  モデルが単純すぎて、与えられたデータをうまく説明できず汎化誤差が高くなること
- 正則化
  モデルの複雑さ（関数空間の大きさ）を連続的に小さくすることで過学習を防ぐ。また、解が不定になることを防ぐ。

4　交差検証

検証セットによる検証を行い学習の正当性を検証する。学習曲線を描いて過学習と未学習を観察し、適切なモデルを選ぶ。

- ホールドアウト法
  X^{train} と X^{valid} が全データ X の分割になるように、分割する。y も同様に分割する。訓練データ X^{train} と y^{train} の組を使って目的関数を設計、その最小解として

$$\hat{\mathbf{w}}_\lambda$$

  を得る。

$$\mathrm{MSE} = \left\|\mathbf{y}^{\mathrm{valid}} - X^{\mathrm{valid}}\hat{\mathbf{w}}_\lambda\right\|^2 / n$$

  が最小になるように λ を決める。
- K-分割法
  各 k に関して $X^{\mathrm{train},k}$ と $X^{\mathrm{valid},k}$ が全データ X の分割になるように、また $X^{\mathrm{valid},k}\,(k = 1, \ldots, K)$ が X の分割になるようにデータを分割する。各 k に関して訓練データ $X^{\mathrm{train},k}$ と $y^{\mathrm{train},k}$ の組（データ数を n_k とする）を使って目的関数を設計、その最小解として

$$\hat{\mathbf{w}}_{\lambda,k}$$

  を得る。これらを用いて

$$\bar{\mathrm{MSE}} = K^{-1}\sum_{k=1}^{K}\left\|\mathbf{y}^{\mathrm{valid},k} - X^{\mathrm{valid},k}\hat{\mathbf{w}}_{\lambda,k}\right\|^2 / n_k$$

  を計算し、これがが最小になるように λ を決める。
- 反復無作為抽出法
  無作為に検証データをサンプルし各 λ の性能を評価
- LOO 交差検証
  $=K = n$ で行う K-分割法。

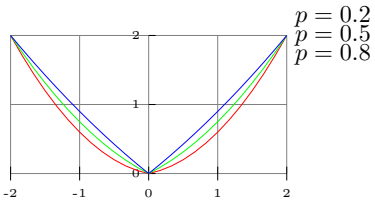

図 6　エラスティックネット正則化項

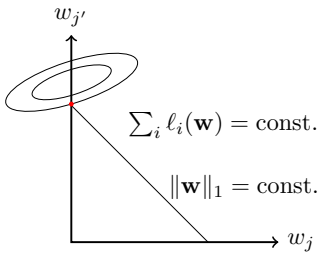

図 7　L_1 正則化のイメージ

5　正則化付き経験リスク最小化問題

正則化付き損失最小化問題 (Regularized Loss Minimization, RLM) とは以下の最小化問題

$$J(\mathbf{w}) = \sum_{i=1}^{n} \ell_i(\mathbf{w}) + \lambda r(\mathbf{w})$$

ここで r は正則化項。第一項は対して損失項という。損失項と正則化項を選ぶことで様々な問題がこれを用いて記述できる。

正則化項の例

- L_2 正則化項

$$r(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|_2^2$$

- L_1 正則化項

$$r(\mathbf{w}) = \|\mathbf{w}\|_1$$

- L_p 正則化項

$$r(\mathbf{w}) = \frac{1}{p} \|\mathbf{w}\|_p^p$$

- L_∞ 正則化項

$$r(\mathbf{w}) = \|\mathbf{w}\|_\infty$$

- L_0 正則化項

$$r(\mathbf{w}) = \|\mathbf{w}\|_0$$

- エラスティックネット正則化項

$$r(\mathbf{w}) = p \cdot \frac{1}{2} \|\mathbf{w}\|_2^2 + (1-p) \|\mathbf{w}\|_1$$

L1 ノルムによる正則化は解の疎性を誘導する

5.1　回帰問題

- 二乗損失

$$\ell_i(\mathbf{w}) = (\mathbf{w}^\top \mathbf{x}_i - y_i)^2$$

- 絶対損失

$$\ell_i(\mathbf{w}) = |\mathbf{w}^\top \mathbf{x}_i - y_i|$$

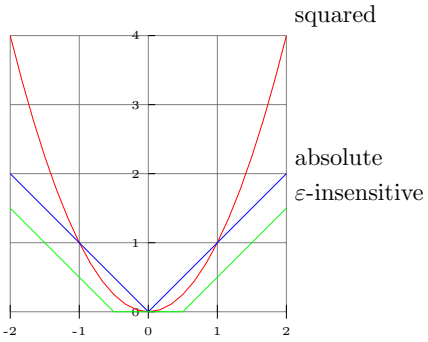

図 8　回帰問題の損失関数

- ε 許容損失 (ε-insensitive loss)

$$\ell_i(\mathbf{w}) = \max\left(-(\mathbf{w}^\top \mathbf{x}_i - y_i) - \varepsilon, 0, (\mathbf{w}^\top \mathbf{x}_i - y_i) - \varepsilon\right)$$

5.2　二値分類問題

$y_i \in \{+1, -1\}$ の場合、二値分類問題という。

- 識別関数

$$\hat{y}_i = \begin{cases} +1 & \mathbf{w}^\top \mathbf{x}_i > 0 \\ -1 & \mathbf{w}^\top \mathbf{x}_i \leq 0 \end{cases}$$

- ヒンジ損失関数

$$\begin{aligned} \ell_i(\mathbf{w}) &= \begin{cases} \max\left(0, 1 - \mathbf{w}^\top \mathbf{x}_i\right) & y_i = +1 \\ \max\left(0, 1 + \mathbf{w}^\top \mathbf{x}_i\right) & y_i = -1 \end{cases} \\ &= \max\left(0, 1 - y_i \mathbf{w}^\top \mathbf{x}_i\right) \end{aligned}$$

- ロジスティック損失関数

$$\ell_i(\mathbf{w}) = \log\left(1 + \exp\left(-y_i \mathbf{w}^\top \mathbf{x}_i\right)\right)$$

分類問題の例

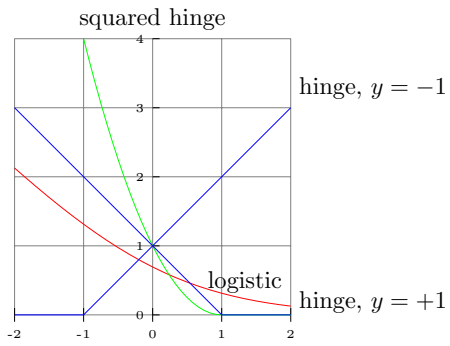

図 9　二値分類問題の損失関数

- サポートベクトルマシン（ヒンジ損失項＋ L2 正則化）

$$J(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i \mathbf{w}^\top \mathbf{x}_i\right) + \frac{\lambda}{2}\mathbf{w}^\top \mathbf{w}$$
$$= C\sum_{i=1}^{n} \max\left(0, 1 - y_i \mathbf{w}^\top \mathbf{x}_i\right) + \frac{1}{2}\mathbf{w}^\top \mathbf{w}$$

- L_1-ロジスティック回帰（ロジスティック損失項＋ L1 正則化）

$$J(\mathbf{w})$$
$$= \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i \mathbf{w}^\top \mathbf{x}_i\right)\right) + \lambda\sum_{j=1}^{d} |w_j|$$
$$= C\sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i \mathbf{w}^\top \mathbf{x}_i\right)\right) + \sum_{j=1}^{d} |w_j|$$

分類問題の指標

- 正解率/Accuracy

$$\frac{\#\left\{i | \hat{y}_i = y_i\right\}}{n}$$

- 精度/Precision

$$\frac{\#\left\{i | \hat{y}_i = +1, y_i = +1\right\}}{\#\left\{i | \hat{y}_i = +1\right\}}$$

- 再現率/Recall

$$\frac{\#\left\{i | \hat{y}_i = +1, y_i = +1\right\}}{\#\left\{i | y_i = +1\right\}}$$

- ROC 曲線/Receiver Operating Characteristic curve

  横軸：

$$\frac{\#\left\{i | \hat{y}_i = +1, y_i = +1\right\}}{\#\left\{i | y_i = +1\right\}}$$

  縦軸

$$\frac{\#\left\{i | \hat{y}_i = +1, y_i = -1\right\}}{\#\left\{i | y_i = -1\right\}}$$

- Precision-Recall curve

  横軸：は再現率。縦軸は精度

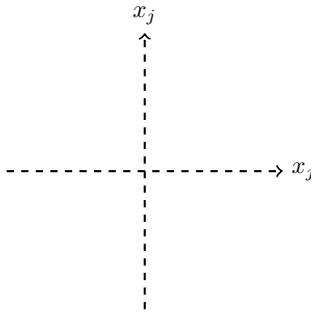

図 10　多クラス分類のイメージ

5.2.1　多クラス分類

$y \in [K] = \{1, 2, \ldots, K\}$ のとき K-クラス分類問題という。$\mathbf{w}_k$ をパラメータベクトルとする。

- 識別関数

$$f(\mathbf{x}) = \operatorname*{argmax}_{k \in [K]} \mathbf{w}_k^\top \mathbf{x}$$

- 多クラスヒンジ損失関数

$$\ell(\mathbf{x}, y) = \max\left(0, \max_{k \in [K] \setminus \{y\}} 1 - (\mathbf{w}_y^\top \mathbf{x} - \langle \mathbf{w}_k, \mathbf{x} \rangle)\right)$$
$$= \max_{k \in [K]} [\![y = k]\!] - (\mathbf{w}_y^\top \mathbf{x} - \mathbf{w}_k^\top \mathbf{x})$$

- 多クラスロジスティック損失関数

$$\ell(\mathbf{x}, y) = \log\left(\sum_{k=1}^{K} \exp\left(\mathbf{w}_k^\top \mathbf{x}\right)\right) - \langle \mathbf{w}_y, \mathbf{x} \rangle$$

6　カーネル法

6.1　特徴写像

導入として回帰問題における次のようなモデルを考える。どのように学習すれば（係数を選べば）よいか？

$$y = \alpha x^2 + \beta x + \gamma$$

特徴写像 $\boldsymbol{\phi}(x)$ を用いればこれは線形モデルで表される。

$$y = \alpha x^2 + \beta x + \gamma = \underbrace{\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}}_{\mathbf{w}}^\top \underbrace{\begin{pmatrix} x^2 \\ x \\ 1 \end{pmatrix}}_{\boldsymbol{\phi}(x)}$$

各データに関する損失関数を $\ell_i(\mathbf{w}^\top \boldsymbol{\phi}(x_i))$ と書くことにすると、回帰だけでなく、以下の問題を解けば非線形なモデルが学習できる。

$$J(\mathbf{w}) = \sum_i \ell_i\left(\mathbf{w}^\top \boldsymbol{\phi}(x_i)\right) + \lambda r(\mathbf{w})$$

多次元の入力データに関しても特徴写像 $\boldsymbol{\phi} : \mathbb{R}^d \to \mathbb{R}^D$ を考えることでより複雑な関数を学習することができる。

6.2 カーネルトリック

- 表現定理

$$J(\mathbf{w}) = \sum_i \ell_i \left(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i)\right) + \frac{\lambda}{2}\mathbf{w}^\top \mathbf{w}$$

の解 $\mathbf{w}^*$ はある $(\alpha_i^*)_{i\in[n]}$ により以下の様に表される

$$\mathbf{w}^* = \sum_{i=1}^n \alpha_i^* \boldsymbol{\phi}(\mathbf{x}_i)$$

ℓ_i が微分可能であるとき、これは以下から簡単にわかる。

$$\sum_i \ell_i'({\mathbf{w}^*}^\top \boldsymbol{\phi}(\mathbf{x}_i))\boldsymbol{\phi}(\mathbf{x}_i) + \lambda \mathbf{w}^* = 0$$
$$\Rightarrow \mathbf{w}^* = -\lambda^{-1} \sum_i \ell_i'({\mathbf{w}^*}^\top \boldsymbol{\phi}(\mathbf{x}_i))\boldsymbol{\phi}(\mathbf{x}_i)$$

（微分が 0 を考える）

- カーネル関数

表現定理により以下の最小化問題を解けば十分

$$\begin{aligned}
&\underset{\alpha_i}{\text{minimize}}\, J\left(\sum_i \alpha_i \boldsymbol{\phi}(\mathbf{x})\right) \\
\Leftrightarrow &\underset{\alpha_i}{\text{minimize}} \sum_i \ell_i \left(\left(\sum_{i'} \alpha_{i'} \boldsymbol{\phi}(\mathbf{x}_{i'})\right)^\top \boldsymbol{\phi}(\mathbf{x}_i)\right) \\
&\qquad + \frac{\lambda}{2}\left(\sum_i \alpha_i \boldsymbol{\phi}(\mathbf{x}_i)\right)^\top \left(\sum_i \alpha_i \boldsymbol{\phi}(\mathbf{x}_i)\right)
\end{aligned}$$

(i, i') 要素が

$$K_{ii'} = \boldsymbol{\phi}(\mathbf{x}_i)^\top \boldsymbol{\phi}(\mathbf{x}_{i'})$$

である行列 K を考える。すると目的関数は

$$\underset{\alpha_i}{\text{minimize}} \sum_i \ell_i \left(\mathbf{e}_i^\top K\boldsymbol{\alpha}\right) + \frac{\lambda}{2}\boldsymbol{\alpha}^\top K\boldsymbol{\alpha}$$

と書ける。

予測器は ${\mathbf{w}^*}^\top \boldsymbol{\phi}(\mathbf{x}) = \left(\sum_i \alpha_i^* \boldsymbol{\phi}(\mathbf{x}_i)\right)^\top \boldsymbol{\phi}(\mathbf{x})$ という形になる。したがって、カーネル関数 $k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$ があたえられれば、特徴写像がわからなくても学習、識別が可能。

$$K_{ii'} = k(\mathbf{x}_i, \mathbf{x}_{i'})$$
$$\left(\sum_i \alpha_i \boldsymbol{\phi}(\mathbf{x}_i)\right)^\top \boldsymbol{\phi}(\mathbf{x}) = \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x})$$

6.3 カーネル関数の例

- 線形カーネル

$$k(\mathbf{x}_i, \mathbf{x}_{i'}) = {\mathbf{x}_i}^\top \mathbf{x}_{i'} = \sum_j x_{ij} x_{i'j}$$

- 多項式カーネル

$$k(\mathbf{x}_i, \mathbf{x}_{i'}) = \left(\gamma {\mathbf{x}_i}^\top \mathbf{x}_{i'} + r\right)^d$$

- ガウスカーネル/RBF カーネル

$$k(\mathbf{x}_i, \mathbf{x}_{i'}) = \exp\left(-\gamma \left\|\mathbf{x}_i - \mathbf{x}_{i'}\right\|^2\right)$$

Mercer の定理よりカーネル関数の正当性がわかる。

任意の n,$(\mathbf{x}_i)_{i\in[n]}$ において

$$K_{ii'} = k(\mathbf{x}_i, \mathbf{x}_{i'})$$

となる行列 K が常に対称かつ半正定値となる時、

$$k(\mathbf{x}_i, \mathbf{x}_{i'}) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_{i'}) \rangle$$

となる適当な特徴写像 $\boldsymbol{\phi} : \mathbb{R}^d \to V$（と適当な関数空間）が存在する。

※ V は有限次元ベクトル空間とは限らない。$\langle \cdot, \cdot \rangle$ は有限次元ベクトル空間とは限らないベクトル空間の内積（$\langle \phi, \phi' \rangle = \sum_{j=1}^\infty \phi_j \phi_j'$ とかける）。