人工知能第3回の講義です。第3回は線形回帰について学びます。

【復習】機械学習

- **データから関数を学習**
  - 関数は、入力(x)と出力(y)の関係を表す

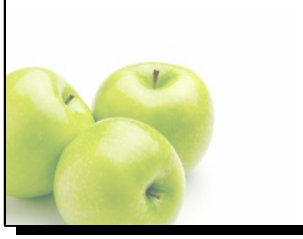
$$x \xrightarrow[f]{} y$$

  - 大量の入出力ペア(x, y)の集まり(データ)から関数fを予測

大量のデータ

(x_1, y_1)
(x_2, y_2)
(x_3, y_3)
$\vdots$
(x_n, y_n)

学習 → f



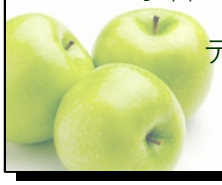
2

機械学習は、データから関数を学習することが本質です。関数は入力(x)と出力(y)の関係を表します。大量の入出力ペア(x, y)の集まり(データ)から入力と出力の関係をとらえた関数を自動的に獲得することが「学習」ということになります。

ただ単にデータがたくさんあれば良いわけではなく、このxのときはこんなyが正解、といったたくさんの入力と出力の例があることで、それらから、入力と出力の関係性を関数として学習する、というわけです。従って、学習には入力と出力のペアが揃った大量のデータが必要です。このように入力と出力のペアが揃ったデータから学習することを「教師つき学習」もしくは「教師あり学習」と言います。

【復習】機械学習

- **データから学習**
  - 入力 $\boldsymbol{x} = (x_1, x_2, \cdots, x_m)$ ←m次元ベクトル
    （画像の場合は、赤、青、緑の画像に対応する3つの行列(テンソル)）
  - データDは入力$\boldsymbol{x}$と出力yのペアの集合
  - 関数fはパラメータ(重み変数)の集合 $\boldsymbol{w} = (w_1, w_2, \cdots w_m)$から成る
    例: $f(\boldsymbol{x}) = w_1x_1 + w_2x_2 + \cdots + w_mx_m$
    - fは$f_w(\boldsymbol{x})$と書くとわかりやすい
  - 学習 = データDに対する誤差(損失)を最小にするパラメータ$\boldsymbol{w}$を求める
    データ全体の誤差(損失) $L(\boldsymbol{w}) = \sum_{(x,y)\in D} \big(y - f_w(\boldsymbol{x})\big)^2$



3

より一般的には、入力$\boldsymbol{x}$はベクトルで与えられることが多いです。1次元、2次元ではなく、多次元のベクトルを与えます。一般にm次元ベクトルとして与えます。画像の場合は、赤、緑、青の画像に対応する3つの行列(テンソル)として与えます。

データDを入力$\boldsymbol{x}$と出力yのペアの集合とします。

関数fはパラメータ(重み変数)の集合($\boldsymbol{w} = (w_1, w_2, \cdots w_m)$)からできていて、$\boldsymbol{w}$と$\boldsymbol{x}$から$y$の値を予測することになります。例えば、$f(\boldsymbol{x}) = w_1x_1 + w_2x_2 + \cdots + w_mx_m$といった形になります。$f$は$f_w(\boldsymbol{x})$と書くとわかりやすいですね。

このように与えられたデータDと関数fに対し、データDに対する誤差を最小にするパラメータ$\boldsymbol{w}$を求めることを、「学習」と言います。データ全体の誤差は、例えば、$L(\boldsymbol{w}) = \sum_{(x,y)\in D}\big(y - f_w(\boldsymbol{x})\big)^2$という式で計算できます。データの各ペア$(\boldsymbol{x}, y)$に対し、本当の正解の$y$から関数で予測される$f(\boldsymbol{x})$の値を引くことで誤差が得られ、データ全体においてその二乗和を計算すると、それがデータ全体の誤差となります。この誤差のことを機械学習の分野では一般に「損失」と呼びます。データ全体に対する損失$L(\boldsymbol{w})$には$\boldsymbol{x}$とyと$\boldsymbol{w}$の3種類の変数が含まれますが、$\boldsymbol{x}$とyにはデータから具体的な値が代入されますので、このデータ全体に対する損失は純粋にパラメータ$\boldsymbol{w}$だけの関数となります。これを最小化することでパラメータが求まります。

【復習】回帰問題と分類問題と構造予測

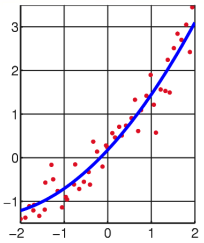
関数: $y = f(\boldsymbol{x})$

- **回帰問題 (regression)**
  - $y \in R$ (実数)

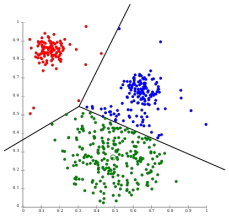
  例: 年齢予測、降水確率の予測、気温の予測
- **分類問題 (classification)**
  - $y \in \{C_1, C_2, \cdots, C_K\}$ (ラベル集合)

  例: 文書分類(政治、経済、スポーツ等)
- **構造予測 (structured prediction)**
  - $y \in G$ (グラフ集合)

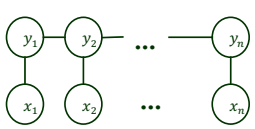
回帰

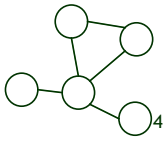


分類



構造予測





4

機械学習は関数$y = f(\boldsymbol{x})$を学習することでした。$\boldsymbol{x}$は様々な入力が想定されていますが、yの値は大きく分けると3種類のタイプに分けられます。そのタイプによって、機械学習の問題は、回帰問題、分類問題、構造予測の3つに分けられます。

回帰問題はyが実数値になる場合です。例えば、年齢予測、降水確率の予測、気温の予測などがあげられます。右上の図のように$\boldsymbol{x}$に対するyの値の曲線(関数)を学習することが目的となります。

分類問題はyがラベルになる場合です。「犬」「猫」ラベルのように、ラベル集合から正解を一つ選ぶ問題になります。ラベル集合は一般にこのように$C_1, C_2, \cdots, C_K$として与えられて、この中から正解を一つ選ぶ、ということになります。例えば、文書分類問題があげられ、「政治」「経済」「スポーツ」などのラベルからその文書のジャンルを選択することになります。スライド右の真ん中にあるように、$\boldsymbol{x}$の空間をいくつかの領域に分けることが目的となります。

最後は構造予測と呼ばれる問題で、これはyの値がグラフになる場合です。グラフというのはこのスライドの図のように丸と直線で結ばれた化合物の模型のような形をした構造のことを指します。例えば、DNAの系列解析や機械翻訳などもこの構造予測の一種となります。図のように与えられた$\boldsymbol{x}$に対応するグラフを生成することが目的となります。

損失関数

- **損失関数** $L(y, \hat{y})$
  - 誤差を数値化する関数。yは正解、$\hat{y}$は予測値

- **データ全体の損失** $L(\boldsymbol{w})$

$$L(\boldsymbol{w}) = \sum_{(\boldsymbol{x},y)\in D} L\big(y, f(\boldsymbol{x})\big)$$

学習：データ全体の損失が最小になる各パラメータの値を探すこと

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}}\, L(\boldsymbol{w})$$

- **回帰問題の損失関数(二乗損失)**

$$L(y, \hat{y}) = (y - \hat{y})^2$$

- **分類問題の損失関数(負対数尤度、交差エントロピー損失)**

$$L(y, \hat{y}) = -\log p_{\boldsymbol{w}}(y|\boldsymbol{x}) = \sum_{k=1}^{K} -y_k \log \hat{y}_k$$

5

続いて、学習の仕組みについて勉強していきます。学習はデータ全体の損失$L(\boldsymbol{w})$を最小化すること、と説明しました。データ全体の損失は、各データに対する損失を合計することで得られます。各データに対する損失を計算する関数$L(y, \hat{y})$のことを「損失関数」と呼びます(ただし、yは正解、$\hat{y}$は予測値)。問題の性質によっていろんな損失関数が用いられていて、例えば、回帰問題には先ほどまで説明した二乗損失が用いられ、分類問題では負対数尤度もしくは交差エントロピー損失と呼ばれる損失関数が用いられます。二乗損失は先ほどの説明と同じになりますが、正解の出力yと関数が予測する$f(\boldsymbol{x})$の値の差の二乗となります。データDの入出力ペア$(\boldsymbol{x}, y)$に対して、これらの合計が最小になるように$\boldsymbol{w}$を調整すると考えれば良いです。

負対数尤度は分類問題に対して用いられる損失関数です。統計学における最尤推定を行うための損失関数となっています。データ全体の各データの尤度(確率)を負の対数にして最小化しています。対数をつけても求めるべきパラメータは同じになります。対数にするのは微分を簡単にするだけではなく、データ全体の尤度(確率)を計算機で表現するためでもあります。確率自体非常に小さな値であり、データ全体の確率は各データの確率の積となるため、計算機で表せられない非常に小さな値となってしまいます。対数をつけることで計算機で表現可能な値となります。例えば、$\log_{10} 10^{-60} = -60$となります。つまり、分類問題は対数尤度の最大化(最尤推定)になっていることに注意しましょう。

教師データに対する損失が最小になるように各パラメータの値を探すことが学習、ということになります。

今日の学習内容

- **回帰問題の解法を学びます**
- **線形回帰モデル（入出力間の関係として線形関数を使った回帰モデル）を学びます**
  - 線形回帰：最小二乗法、リッジ回帰、正規方程式

6

今回は回帰問題の解法について学びます。

具体的な例として線形回帰モデルを学びます。線形回帰モデルというのは、入出力間の関係として線形関数を用いた回帰モデルのことです。回帰モデルは回帰問題を解くためのモデルです。

線形回帰モデルを解くための最小二乗法、過学習を防ぐためのリッジ回帰、線形回帰モデルのための最小二乗法の解となる正規方程式について学びます。

回帰モデル: 学習

- **回帰モデルのための教師データの例（学習時）**

乗客数	離陸重量	燃料重量比	巡行燃費	航続距離
120	50.0	0.250	0.65	3850
108	60.6	0.309	0.6	5820
108	65.1	0.312	0.6	6610
496	375.0	0.414	0.55	11800

学習 → 回帰モデル

入力（特徴量ベクトル）

目標出力（実数値）

7

まず、回帰問題のデータをみてみましょう。この表が回帰問題のデータとなっています。

飛行機の航続距離に関するデータとなっていて、乗客数、離陸時の重量、燃料重量比、巡航燃費がわかっているときの実際の航続距離を表したデータとなっています。これらのデータのうち左4列の乗客数、離陸時の重量、燃料重量比、巡航燃費が回帰モデルへの入力データとなります。このとき、航続距離がいくらになるのか出力する回帰モデルを学習したいとします。

学習時は、これらの4つ特徴が入力として与えられ、航続距離が出力として与えられて学習を行います。この場合は4つのデータから成り立ちますが、これらのデータから回帰モデルを学習します。

回帰モデルの特徴として、出力が実数値となることに注意してください。(出力がラベルの場合は識別モデルと呼ばれます)

回帰モデル：推論

- **予測の例（推論時）**

乗客数	離陸重量	燃料重量比	巡行燃費	航続距離
48	77.6	0.412	0.6	不明

入力
（特徴量ベクトル）

学習済み
回帰
モデル

出力
（実数値）

予測する

8

学習が終わった後、学習済みの回帰モデルを用いて予測を行います。未知のデータに対して、出力を予測することは「推論(inference)」と呼ばれます。

この例の場合、乗客数、離陸重量、燃料重量比、巡航燃費の4つの入力情報から成る未知のデータが与えられたとき、これらの入力を回帰モデルに与えて、出力値を得ます。この出力値が航続距離の予測値となります。

線形回帰モデル

- **入力ベクトルxから出力yを得る関数がxの線形関数（wとxの内積）**

$$y = w_0 + w_1x_1 + w_2x_2 + \cdots + w_Kx_K = \sum_{i=0}^{K} w_ix_i = \langle \boldsymbol{w}, \boldsymbol{x} \rangle = \boldsymbol{w}^T\boldsymbol{x}$$

ただし、$\boldsymbol{x} = \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_K \end{pmatrix}, \boldsymbol{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_K \end{pmatrix}$

Tは転置

$(w_0, w_1, \cdots, w_K)$のこと

※$\boldsymbol{x}$が1次元の時：$y = w_1x_1 + w_0$

9

線形回帰モデルは具体的にはどんな形の関数でしょうか。

線形式が関数となっているモデルは線形モデルと呼ばれます。回帰問題の線形モデルは線形回帰モデルと呼ばれ、分類問題の線形モデルは線形識別モデルと呼ばれます。

線形式なので、入力を$x_1, \cdots, x_K$としたとき、線形回帰モデルは$y = w_1x_1 + w_2x_2 + \cdots + w_Kx_K$となります。$w_0, w_1, \cdots, w_K$は学習すべき重み変数(パラメータ)と呼ばれています。各w_iは各入力x_iに対する係数となっています。入力と重み変数をそれぞれ一つのベクトルとして表現すると、$\boldsymbol{x} = \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_K \end{pmatrix}$、$\boldsymbol{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_K \end{pmatrix}$となります。機械学習の世界では一般的にベクトルは縦一列の行列として表現されることに注意してください。また、線形式の切片w_0を対応する入力として、$\boldsymbol{x}$の最初の要素を1としている点にも注意してください。切片w_0のことは機械学習の世界ではバイアス項と呼ばれています。このような表現を与えたとき、線形式は$\boldsymbol{w}$と$\boldsymbol{x}$の内積$\langle \boldsymbol{w}, \boldsymbol{x} \rangle$になります。各ベクトルを縦一列の行列として表現すると、この内積の計算は $\boldsymbol{w}^T\boldsymbol{x}$ となります。線形式を簡潔に表現することができるため、$\boldsymbol{w}^T\boldsymbol{x}$と表現することが多いです。右肩の$T$は転置を表します。

$\boldsymbol{x}$が1次元の時は$y = w_1x_1 + w_0$となります。

線形回帰モデルの学習

- **学習: 損失の最小化**

  $\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} L(\boldsymbol{w})$　　$L(\boldsymbol{w})$：データ全体の損失

- **線形回帰モデルの損失関数**
  - 最小二乗法

$$L(\boldsymbol{w}) = \sum_{(\boldsymbol{x},y)\in D} \underbrace{(y}_{\text{正解出力}} - \underbrace{\boldsymbol{w}^T\boldsymbol{x})^2}_{\text{予測出力}}$$

  - リッジ回帰

$$L(\boldsymbol{w}) = \sum_{(\boldsymbol{x},y)\in D} (y - \boldsymbol{w}^T\boldsymbol{x})^2 + \underbrace{\frac{\lambda}{2}\|\boldsymbol{w}\|^2}_{\text{正則化項}}$$

$$\|\boldsymbol{w}\|^2 = \left(\sqrt{w_0^2 + w_1^2 + \cdots + w_K^2}\right)^2 = w_0^2 + w_1^2 + \cdots + w_K^2$$

10

続いて、線形回帰モデルを教師データから学習することを考えます。

学習は、データ全体の誤差を最小化する重みベクトルを求めることで実現されます。正解出力と予測出力の差分のことを誤差または損失と呼びます。つまり、学習はデータ全体の損失の最小化として実現されます。

データ全体に対し、正解と予測の差の二乗和を最小化することで重み変数を求める手法は、最小二乗法と呼ばれます。線形回帰モデルを最小二乗法で求める際の損失は$L(\boldsymbol{w}) = \sum_{(\boldsymbol{x},y)\in D}(y - \boldsymbol{w}^T\boldsymbol{x})^2$として表現されます。つまり、データ$D$の各要素$(\boldsymbol{x}, y)$に対して、正解出力である$y$から、線形回帰モデルでの予測値$\boldsymbol{w}^T\boldsymbol{x}$を引くと、これがデータ$(\boldsymbol{x}, y)$に対する損失となります。この損失をデータ全体で足し合わせると、$\sum_{(\boldsymbol{x},y)\in D}(y - \boldsymbol{w}^T\boldsymbol{x})^2$となります。この式の中で$\boldsymbol{x}$と$y$は具体的な数値が入っているため、定数と考えてかまいません。従って、このデータ全体の損失を表す関数は残った変数である重み変数に対する関数となります。$L(\boldsymbol{w})$と表現されるように、重みベクトル$\boldsymbol{w}$に対する関数となっていることに注意してください。

線形回帰モデルの学習方法として、その他にリッジ回帰と呼ばれる学習方法があります。リッジ回帰の損失は、$L(\boldsymbol{w}) = \sum_{(\boldsymbol{x},y)\in D}(y - \boldsymbol{w}^T\boldsymbol{x})^2 + \frac{\lambda}{2}\|\boldsymbol{w}\|^2$となります。最小二乗法の損失に、$\frac{\lambda}{2}\|\boldsymbol{w}\|^2$という項がついています。$\|\boldsymbol{w}\|^2 = \left(\sqrt{w_0^2 + w_1^2 + \cdots + w_K^2}\right)^2 = w_0^2 + w_1^2 + \cdots + w_K^2$となるため、各重み変数の二乗和となっています。これは過学習を抑えるための項で、正則化項と呼ばれたり、罰則項とも呼ばれています。過学習が起きるときは極端に重み変数の値が大きな値になることが知られているので、重み変数の値が大きくなりすぎたときには罰則として損失に加わるような仕組みになっています。λは、データに対するモデルの損失$\sum_{(\boldsymbol{x},y)\in D}(y - \boldsymbol{w}^T\boldsymbol{x})^2$と正則化項$\frac{\lambda}{2}\|\boldsymbol{w}\|^2$のバランスを調整するパラメータで、人手で調整する必要があることから、ハイパーパラメータと呼ばれています。

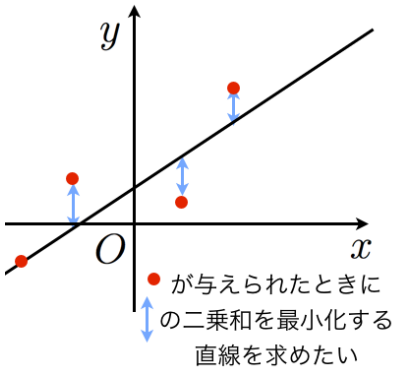
最小二乗法

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \sum_{(\boldsymbol{x},y)\in D} (y - \boldsymbol{w}^T\boldsymbol{x})^2$$

教師データDにおける入出力（$\boldsymbol{x}$とy）の関係を
表すもっともらしい関数を求める

各教師データとの差の二乗和
を最小にする関数

例： $\boldsymbol{x}$が1次元（モデルが直線）の時



● が与えられたときに
↕ の二乗和を最小化する
直線を求めたい

11

最小二乗法は、従って、教師データDにおける入出力(x, y)の関係を表すもっともらしい関数を求めることになります。具体的には、モデルが表す関数の出力と各データの正解の差(図の水色矢印)が小さくなるように関数を調整することになります。

回帰モデルの学習例

- **教師データ$D = \{(1,2), (2,3), (3,5)\}$から最小二乗法により学習される直線の式$f(x) = ax + b$を求めなさい。**

$$L(a,b) = \sum_{(x,y)\in D} \big(y - f(x)\big)^2$$

12

具体的に損失関数を最小化することで重み変数を求めてみましょう。損失関数に対する各重み変数の偏微分を全て0とすることで誤差関数を最小にする重み変数を求めます。

教師データ$D = \{(1,2), (2,3), (3,5)\}$から最小二乗法により学習される直線の式$f(x) = ax + b$を求めましょう。(約5分)

回帰モデルの学習

- **教師データ$D = \{(1,2),(2,3),(3,5)\}$から最小二乗法により学習される直線の式$f(x) = ax + b$を求めなさい。**

$$L(a,b) = \sum_{(x,y)\in D} \big(y - f(x)\big)^2$$

$$L(a,b) = (2 - a - b)^2 + (3 - 2a - b)^2 + (5 - 3a - b)^2$$

$$\frac{\partial L(a,b)}{\partial a} = -2(2 - a - b) - 4(3 - 2a - b) - 6(5 - 3a - b) =$$
$$= -46 + 28a + 12b = 0$$

$$\frac{\partial L(a,b)}{\partial b} = -2(2 - a - b) - 2(3 - 2a - b) - 2(5 - 3a - b) =$$
$$= -20 + 12a + 6b = 0$$

解くと、$a = \frac{3}{2}, b = \frac{1}{3}$。従って、$f(x) = \frac{3}{2}x + \frac{1}{3}$。

13

損失関数にデータの値を代入すると、$L(a,b) = (2 - a - b)^2 + (3 - 2a - b)^2 + (5 - 3a - b)^2$となります。

これをaとbでそれぞれ偏微分して0とする連立方程式をたてます。これを解くと、$a = \frac{3}{2}, b = \frac{1}{3}$となるため、求めるべき関数は$f(x) = \frac{3}{2}x + \frac{1}{3}$となります。

$\hat{\boldsymbol{w}}$を求めると…

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \sum_{(\boldsymbol{x},y)\in D} (y - \boldsymbol{w}^T\boldsymbol{x})^2 = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{y} - X\boldsymbol{w}\|^2$$

ただし、$\boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}$, $\boldsymbol{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_K \end{pmatrix}$, $X = \begin{pmatrix} \boldsymbol{x_1}^T \\ \vdots \\ \boldsymbol{x_N}^T \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1K} \\ \vdots & & \ddots & \vdots \\ 1 & x_{N1} & \cdots & x_{NK} \end{pmatrix}$

$$\underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{y} - X\boldsymbol{w}\|^2 = \underset{\boldsymbol{w}}{\operatorname{argmin}} (\boldsymbol{y} - X\boldsymbol{w})^T(\boldsymbol{y} - X\boldsymbol{w})$$

$\frac{\partial}{\partial \boldsymbol{w}}(\boldsymbol{y} - X\boldsymbol{w})^T(\boldsymbol{y} - X\boldsymbol{w}) = 0$を解けばよい

解くと… $(X^TX)\boldsymbol{w} = X^T\boldsymbol{y}$ 正規方程式

$\hat{\boldsymbol{w}} = (X^TX)^{-1}X^T\boldsymbol{y}$

14

線形回帰モデルには実は解析解が存在しています。誤差を各データごとに二乗して足し合わせていますが、これを行列形式で表現し、ベクトルで微分すると正規方程式と呼ばれる式が得られます。この正規方程式を解くことで解析解が得られます。

まず、重みベクトルは先程と同じ様に$\boldsymbol{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_K \end{pmatrix}$とします。出力は全データの出力を縦に並べたベクトル$\boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}$で表現します。入力は各データを横ベクトル$\boldsymbol{x} = \begin{pmatrix} 1 & x_1 & \cdots & x_K \end{pmatrix}$で表現し、全データの入力ベクトルxを縦方向に並べることで行列にします。つまり、$X = \begin{pmatrix} \boldsymbol{x_1}^T \\ \vdots \\ \boldsymbol{x_N}^T \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1K} \\ \vdots & & \ddots & \vdots \\ 1 & x_{N1} & \cdots & x_{NK} \end{pmatrix}$となります。この行列はデータ行列と呼ばれます。

そうすると、損失関数の最小化は、$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \sum_{(\boldsymbol{x},y)\in D} (y - \boldsymbol{w}^T\boldsymbol{x})^2 = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{y} - X\boldsymbol{w}\|^2$となります。$\underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{y} - X\boldsymbol{w}\|^2 = \underset{\boldsymbol{w}}{\operatorname{argmin}} (\boldsymbol{y} - X\boldsymbol{w})^T(\boldsymbol{y} - X\boldsymbol{w})$なので、$\frac{\partial}{\partial \boldsymbol{w}}(\boldsymbol{y} - X\boldsymbol{w})^T(\boldsymbol{y} - X\boldsymbol{w}) = 0$を解けば良いということになります。

この式を微分するのは難しいのですが、解くと、$(X^TX)\boldsymbol{w} = X^T\boldsymbol{y}$となります。この式は正規方程式と呼ばれています。この式に逆行列をかけることで、重みベクトルの解析解$\hat{\boldsymbol{w}} = (X^TX)^{-1}X^T\boldsymbol{y}$が得られます。

この正規方程式は難しい形をしていますが、元々、$X\boldsymbol{w} = \boldsymbol{y}$となる$\boldsymbol{w}$を見つけたかったわけなので、$X\boldsymbol{w} = \boldsymbol{y}$の左から$X^T$をかけて$(X^TX)\boldsymbol{w} = X^T\boldsymbol{y}$としていると考えるとわかりやすいです。$X\boldsymbol{w} = \boldsymbol{y}$となる$\boldsymbol{w}$が直接見つかればいいのですが、$X$は正方行列ではないので、逆行列が存在しません。そのため、X^Tを左からかけることによって、無理矢理正方行列にしている、と考えるとわかりやすいです。つまり、$(X^TX)^{-1}X^T$はX^{-1}のような役割を果たしていると言えます。この擬似的な逆行列はムーア-ペンローズの擬似逆行列と呼ばれています。

【参考】正規方程式の導出

$$\frac{\partial}{\partial \boldsymbol{w}}(\boldsymbol{y}-X\boldsymbol{w})^T(\boldsymbol{y}-X\boldsymbol{w})=0$$

$$\frac{\partial}{\partial \boldsymbol{w}}(\boldsymbol{y}^T\boldsymbol{y}-\boldsymbol{w}^TX^T\boldsymbol{y}-\boldsymbol{y}^TX\boldsymbol{w}+\boldsymbol{w}^TX^TX\boldsymbol{w})=0$$

左辺の各項目の$\boldsymbol{w}$による偏微分

転置の公式：$(AB)^T=B^TA^T$

内積の微分の公式：$\frac{\partial}{\partial \boldsymbol{w}}\boldsymbol{w}^T\boldsymbol{a}=\frac{\partial}{\partial \boldsymbol{w}}\boldsymbol{a}^T\boldsymbol{w}=\boldsymbol{a}$（$\boldsymbol{a}$は定数）

2次形式の微分の公式: $\frac{\partial}{\partial \boldsymbol{w}}\boldsymbol{w}^TA\boldsymbol{w}=(A+A^T)\boldsymbol{w}$

$$\frac{\partial}{\partial \boldsymbol{w}}\boldsymbol{y}^T\boldsymbol{y}=0,\quad \frac{\partial}{\partial \boldsymbol{w}}\boldsymbol{w}^TX^T\boldsymbol{y}=X^T\boldsymbol{y},\quad \frac{\partial}{\partial \boldsymbol{w}}\boldsymbol{y}^TX\boldsymbol{w}\ =[\boldsymbol{y}^TX]^T=X^T\boldsymbol{y}$$

$$\frac{\partial}{\partial \boldsymbol{w}}\boldsymbol{w}^TX^TX\boldsymbol{w}=(X^TX+(X^TX)^T)\boldsymbol{w}=2X^TX\boldsymbol{w}$$

$-2X^T\boldsymbol{y}+2X^TX\boldsymbol{w}=0$まとめると、$(X^TX)\boldsymbol{w}=X^T\boldsymbol{y}$

15

参考までに正規方程式の導出を紹介しておきます。行列微分の公式を知らないと解けないため、一般の大学生が習う知識で解くことは難しいです。

行列の公式として次のものを使います。

転置の公式：$(AB)^T=B^TA^T$

内積の微分の公式：$\frac{\partial}{\partial \boldsymbol{w}}\boldsymbol{w}^T\boldsymbol{a}=\frac{\partial}{\partial \boldsymbol{w}}\boldsymbol{a}^T\boldsymbol{w}=\boldsymbol{a}$（$\boldsymbol{a}$は定数）

2次形式の微分の公式: $\frac{\partial}{\partial \boldsymbol{w}}\boldsymbol{w}^TA\boldsymbol{w}=(A+A^T)\boldsymbol{w}$

求めたい式は$\frac{\partial}{\partial \boldsymbol{w}}(\boldsymbol{y}-X\boldsymbol{w})^T(\boldsymbol{y}-X\boldsymbol{w})=0$ですが、この式を展開して、$\frac{\partial}{\partial \boldsymbol{w}}(\boldsymbol{y}^T\boldsymbol{y}-\boldsymbol{w}^TX^T\boldsymbol{y}-\boldsymbol{y}^TX\boldsymbol{w}+\boldsymbol{w}^TX^TX\boldsymbol{w})=0$とします。

この式の各項に対して、ベクトル$\boldsymbol{w}$で微分をします。ちなみに、$\frac{\partial f}{\partial \boldsymbol{w}}=\begin{pmatrix}\frac{\partial f}{\partial w_0}\\ \frac{\partial f}{\partial w_1}\\ \vdots\\ \frac{\partial f}{\partial w_K}\end{pmatrix}$と定義されています。

各項を公式を使って微分すると、$\frac{\partial}{\partial \boldsymbol{w}}\boldsymbol{y}^T\boldsymbol{y}=0,\quad \frac{\partial}{\partial \boldsymbol{w}}\boldsymbol{w}^TX^T\boldsymbol{y}=X^T\boldsymbol{y},\quad \frac{\partial}{\partial \boldsymbol{w}}\boldsymbol{y}^TX\boldsymbol{w}\ =[\boldsymbol{y}^TX]^T=X^T\boldsymbol{y}$、$\frac{\partial}{\partial \boldsymbol{w}}\boldsymbol{w}^TX^TX\boldsymbol{w}=(X^TX+(X^TX)^T)\boldsymbol{w}=2X^TX\boldsymbol{w}$となり、まとめると、$-2X^T\boldsymbol{y}+2X^TX\boldsymbol{w}=0$となります。整理すると、$(X^TX)\boldsymbol{w}=X^T\boldsymbol{y}$になります。

$\boldsymbol{x}$が1次元（直線）の場合

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, \quad \boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}, \quad X = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix}$$

$$X^T X = \begin{bmatrix} N & \sum_{n=1}^{N} x_n \\ \sum_{n=1}^{N} x_n & \sum_{n=1}^{N} {x_n}^2 \end{bmatrix}, \quad (X^T X)^{-1} = \frac{1}{N\sum_{n=1}^{N} {x_n}^2 - (\sum_{n=1}^{N} x_n)^2} \begin{bmatrix} \sum_{n=1}^{N} {x_n}^2 & -\sum_{n=1}^{N} x_n \\ -\sum_{n=1}^{N} x_n & N \end{bmatrix},$$

$$X^T \boldsymbol{y} = \begin{bmatrix} \sum_{n=1}^{N} y_n \\ \sum_{n=1}^{N} x_n y_n \end{bmatrix}.$$

なので、 $\hat{w}_0 = \frac{1}{N}\left(\sum_{n=1}^{N} y_n - \hat{w}_1 \sum_{n=1}^{N} x_n\right)$,

$$\hat{w}_1 = \frac{N\sum_{n=1}^{N} x_n y_n - \sum_{n=1}^{N} x_n \cdot \sum_{n=1}^{N} y_n}{N\sum_{n=1}^{N} {x_n}^2 - (\sum_{n=1}^{N} x_n)^2}.$$

16

xが1次元(直線)の場合は、$X^T X$が2x2の行列になるため、その逆行列を直接計算することができます。従って、解析解は

$$\hat{w}_0 = \frac{1}{N}\left(\sum_{n=1}^{N} y_n - \hat{w}_1 \sum_{n=1}^{N} x_n\right)$$

$$\hat{w}_1 = \frac{N\sum_{n=1}^{N} x_n y_n - \sum_{n=1}^{N} x_n \cdot \sum_{n=1}^{N} y_n}{N\sum_{n=1}^{N} {x_n}^2 - (\sum_{n=1}^{N} x_n)^2}$$

となります。

演習

- **教師データ= {(1,2),(2,3),(3,5)} から最小二乗法により学習される直線の式を求めなさい。ただし、正規方程式の解を用いて答えなさい。**

17

前に求めた最小二乗法の解を正規方程式の解を用いて解いてみましょう。
教師データ$D = \{(1,2), (2,3), (3,5)\}$から最小二乗法により学習される直線の式$f(x) = ax + b$を求めましょう。(約5分)

演習（答え）

- **教師データ={(1,2),(2,3),(3,5)} から最小二乗法により学習される直線の式を求めなさい。**

$$N = 3,\quad \sum_{n=1}^{N} x_n = 6,\quad \sum_{n=1}^{N} y_n = 10,\quad \sum_{n=1}^{N} x_n y_n = 23,\quad \sum_{n=1}^{N} {x_n}^2 = 14$$

$$\hat{w}_1 = \frac{N\sum_{n=1}^{N} x_n y_n - \sum_{n=1}^{N} x_n \cdot \sum_{n=1}^{N} y_n}{N\sum_{n=1}^{N} {x_n}^2 - \left(\sum_{n=1}^{N} x_n\right)^2} = \frac{3\times 23 - 6\times 10}{3\times 14 - 36} = \frac{9}{6} = \frac{3}{2}$$

$$\hat{w}_0 = \frac{1}{N}\left(\sum_{n=1}^{N} y_n - \hat{w}_1 \sum_{n=1}^{N} x_n\right) = \frac{1}{3}\left(10 - \frac{3}{2}\times 6\right) = \frac{1}{3}$$

答え　$\frac{3}{2}x + \frac{1}{3}$

18

解は先程の解と同じになりますが、このようにして解析解を得ることができることがわかります。

リッジ回帰

- **最小二乗法に2次の正則化項を導入することで過学習を緩和**

※過学習：訓練データに過剰に適合してしまうこと

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \left\{ \sum_{(x,y)\in D} (y - \boldsymbol{w}^T \boldsymbol{x})^2 + \frac{\lambda}{2} \|\boldsymbol{w}\|^2 \right\}$$

正則化項

$$\|\boldsymbol{w}\|^2 = \sqrt{w_0^2 + w_1^2 + \cdots + w_K^2}$$

解くと

$$(X^T X + \lambda I)\boldsymbol{w} = X^T \boldsymbol{y}$$

$$\hat{\boldsymbol{w}} = (X^T X + \lambda I)^{-1} X^T \boldsymbol{y}$$

※λは正則化定数（予め人手で定める）

※Iは単位行列

19

リッジ回帰に対する解析解も求めることができます。同様にして解くと、正規方程式は

$$(X^T X + \lambda I)\boldsymbol{w} = X^T \boldsymbol{y}$$

となります。これに対し、逆行列を求めることで、

$$\hat{\boldsymbol{w}} = (X^T X + \lambda I)^{-1} X^T \boldsymbol{y}$$

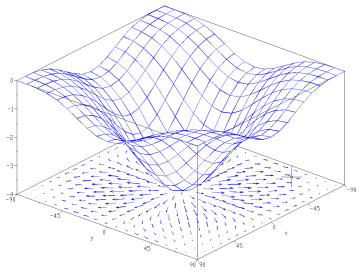
となります。ただし、Iは単位行列です。

最適化に関して

- **逆行列が簡単に求まる場合：正規方程式を利用**
- **逆行列演算が難しい場合：数値計算法を利用**
  - ニュートン法、準ニュートン法
  - 最急降下法
  - オンライン学習（確率的勾配法(SGD), AdaGrad, Adamなど）

  - 勾配を用いて計算する手法が多い

勾配　$\nabla L = \left(\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, \cdots, \frac{\partial L}{\partial w_m}\right)$



20

逆行列が簡単に求まる場合は、正規方程式を用いて解くときれいな解を高速に求めることができます。しかし、一般には逆行列を求めることが難しいため、数値計算法を利用して解を求めることになります。

数値計算法としては、ニュートン法、準ニュートン法、最急降下法が有名です。また、これらの学習方法はバッチ学習と呼ばれるデータ全体に対して最適化する手法が用いられていますが、深層学習では、一部のデータに対して最適化を繰り返すオンライン学習が一般的によく用いられています。オンライン学習では、確率的勾配降下法(SGD)、AdaGrad、Adamなどがあります。これらの手法はいずれも勾配に基づいて損失関数の最小化を行います。

まとめ

- **線形回帰モデル**
  - 最小二乗法
  - リッジ回帰
  - 正規方程式
- **最適化**
  - 正規方程式による最適解の導出
  - 逆行列演算が難しい場合は勾配に基づく数値計算法

21

今回のまとめです。
今回は線形回帰モデルについて勉強しました。最小二乗法とリッジ回帰について学んで、その解析解として正規方程式を学びました。最適化には正規方程式を解くことで最適解を導出する方法がありますが、逆行列の計算が必要です。逆行列演算が難しい場合は勾配に基づく数値計算法でその解を求めます。