



線形回帰

できる限り多くのデータ点の近くを通る直線を求めることで、その直線の式を使って新たなデータを大まかに予測することができる
このような手法を線形回帰という

ref: なっとく！機械学習 p40
ref: 線形代数の半歩先 p112、p121~122

線形回帰は、次のような手順で行われる

1. モデルとして直線や平面を仮定する
2. 誤差を測る指標（誤差関数）を設定する
3. 誤差が最小になるようにモデルのパラメータを調整する



モデルを表す式

線形回帰では、モデルとして一次式を使う

ref: 線形代数の半歩先 p122~123
ref: なっとく！機械学習 p42

$$f(\mathbf{x}) = w_0 + \sum_{d=1}^D w_d x_d$$

この式では、 D 個の特徴量 x_1, \dots, x_D を持つデータを考えている

モデルを表す式において、それぞれの特徴量にかける係数 w_1, \dots, w_D を重み (weight) と呼ぶ

また、モデルを表す式では、どの特徴量にも結びつかない定数 w_0 がある
この定数をバイアス (bias) と呼ぶ




誤差関数

どんな直線が適合するといえるのか、「データに当てはまる基準」も自分で設定することになる

誤差が少ない、すなわち「当てはまりがよい」ほど小さい値をとるような関数を**誤差関数** (**error function**) と呼ぶ

ref: なっとく！機械学習 p65～

ref: 線形代数の半歩先 p123

 **誤差関数** モデルの性能がどれくらいかを明らかにする指標であり、性能が悪いモデルに大きな値を割り当て、性能がよいモデルに小さな値を割り当てる関数

誤差関数は、**損失関数** (**loss function**) や**コスト関数** (**cost function**)、最適化問題としての側面に注目した場合は**目的関数**と呼ばれることもある

誤差関数を定義する一般的な方法としては、次の 2 つがある

- **絶対誤差** (**absolute error**) : 直線からデータ点までの垂直距離を合計したもの
- **二乗誤差** (**square error**) : 直線からデータ点までの垂直距離の二乗を合計したもの

線形回帰では、誤差関数を最小にするモデル（直線）を探すことになる

誤差関数として**二乗誤差**を用いた場合、その最小化問題は**最小二乗法**と呼ばれる



二乗誤差と最小二乗法

データの総数を N とすると、二乗誤差は、次のような数式で表される

ref: 線形代数の半歩先 p123～127

$$J(\boldsymbol{w}) = \sum_{n=1}^N (y_n - f(\boldsymbol{x}_n))^2$$

n 番目の実際の出力 y_n と、 n 番目の入力を使ったときのモデルの出力 $f(\boldsymbol{x}_n)$ との差を見ている

符号を正にするために二乗し、それをすべてのデータについて合計したものが二乗誤差である

この誤差関数 $J(\boldsymbol{w})$ を最小にするパラメータを探すことが目標となる

モデルの式を整理する

まずは、モデルの式を整理する

$$f(\boldsymbol{x}) = w_0 + \sum_{d=1}^D w_d x_d$$

右辺はベクトルの内積で書けそうだが、 w_0 が余分なので、 $\boldsymbol{x}_0 = 1$ と定義して、次のように書き換える

$$f(\boldsymbol{x}) = \sum_{d=0}^D w_d x_d$$

そして、次のようなベクトルを導入する

$$\boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_D \end{bmatrix}, \quad \boldsymbol{x}' = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix}, \quad \boldsymbol{x}'_n = \begin{bmatrix} 1 \\ x_{n,1} \\ x_{n,2} \\ \vdots \\ x_{n,D} \end{bmatrix}$$

すると、先ほどのモデルの式は、次のように \boldsymbol{w} と \boldsymbol{x}' の内積で表せる

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}'$$

N 個分のデータをまとめる

N 個分のデータをまとめた出力 \boldsymbol{y} と入力 \boldsymbol{X} を、それぞれ次のように書く

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \quad X = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,D} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,D} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N,1} & x_{N,2} & \cdots & x_{N,D} \end{bmatrix} = \begin{bmatrix} (\mathbf{x}'_1)^\top \\ (\mathbf{x}'_2)^\top \\ \vdots \\ (\mathbf{x}'_N)^\top \end{bmatrix}$$

X は $N \times (D + 1)$ 行列で、定数項の分だけ列が一つ増えている

この定数項の列を含まず、データだけを並べたものはデータ行列と呼ばれる

ただし、ここでは定数項の列を含めた X もデータ行列と呼ぶことにする

誤差関数をベクトルと行列で表す

ここまでの記号を使って、誤差関数 $J(\mathbf{w})$ を書き直す

まずは n 番目のデータにのみ注目すると、実際の値とモデルの差は、

$$\begin{aligned} y_n - f(\mathbf{x}_n) &= y_n - \mathbf{w}^\top \mathbf{x}'_n \\ &= y_n - (\mathbf{x}'_n)^\top \mathbf{w} \end{aligned} \quad \left. \vphantom{\begin{aligned} y_n - f(\mathbf{x}_n) &= y_n - \mathbf{w}^\top \mathbf{x}'_n \\ &= y_n - (\mathbf{x}'_n)^\top \mathbf{w} \end{aligned}} \right\} \text{内積の順番を変える}$$

ベクトルと行列を使うと、 N 個のデータに対しては次のように書ける

$$\mathbf{z} = \begin{bmatrix} y_1 - (\mathbf{x}'_1)^\top \mathbf{w} \\ y_2 - (\mathbf{x}'_2)^\top \mathbf{w} \\ \vdots \\ y_N - (\mathbf{x}'_N)^\top \mathbf{w} \end{bmatrix} = \mathbf{y} - X\mathbf{w}$$

この二乗をとった形は、 \mathbf{z} 自身との内積で書き表せる

$$J(\mathbf{w}) = \mathbf{z}^\top \mathbf{z} = (\mathbf{y} - X\mathbf{w})^\top (\mathbf{y} - X\mathbf{w})$$

ベクトルの微分で最小化問題を解く

誤差関数を最小にする \mathbf{w} を求めるためには、

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = 0$$

を解くことになる