

## 線形回帰

$\{(\mathbf{x}_i, y_i)\}_{i=1}^N$  をラベル付けされたデータとし、 $N$  をデータの数、 $\mathbf{x}_i$  を  $D$  次元特徴ベクトル、 $y_i$  を  $\mathbf{x}_i$  のラベルとする。 $\mathbf{w}$  を  $D$  次元ベクトル、 $b$  を実数とし、

$$f_{\mathbf{w},b} := \mathbf{w}\mathbf{x} + b$$

とおく。この式を用いて、未知の  $D$  次元特徴ベクトル  $\mathbf{x}$  に対して、ラベル  $y = f_{\mathbf{w},b}(\mathbf{x})$  を予測する。最適な  $\mathbf{w}, b$  は

$$\min_{\mathbf{w},b} \frac{1}{N} \sum_{i=1}^N (f_{\mathbf{w},b}(\mathbf{x}_i) - y_i)^2$$

で求められる。

例 0.1. (コードは線形回帰.ipynb) ボストン住宅価格のデータセットを用いて、線形回帰のモデルのプログラムを確認する。

```
from sklearn.datasets import load_boston
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

boston = load_boston() #データを取得

boston_df = pd.DataFrame(boston.data, columns = boston.feature_names)
boston_df['MEDV'] = boston.target

reg = LinearRegression() #f_{w,b}=wx+bを設定

X = boston_df[['RM']].values #特徴ベクトル
Y = boston_df['MEDV'].values #ラベル

reg.fit(X, Y) #w,bの最適解を求める
reg.coef_ # w の値
>array([9.10210898])
reg.intercept_ # b の値
>-34.67062077643857
```

## 参考文献

- [1] Andriy Burkov. (2019). The hundred-page machine learning book.
- [2] Marc Peter Deisenroth., A. Aldo Faisal., Cheng Soon Ong. (2020). Mathematics for machine learning. Cambridge University Press.
- [3] Aurélien Geron. (2019). Hands-on machine learning with Scikit-Learn, Keras & TensorFlow. 2nd Edition. Oreilly.
- [4] 小縣信也., 斎藤翔汰., 溝口聡., 若杉一幸. (2021). ディープラーニング E 資格エンジニア問題集 インプレス.
- [5] Sebastian Raschka., Vahid Mirjalili. (2019). Python machine learning. Third Edition. Packt.