

ロジスティック回帰モデル

$\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ 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。ただし、 y_i の値は $0, 1$ のいずれかであるとする。 \mathbf{w} を D 次元ベクトル、 b を実数とし、

$$f_{\mathbf{w}, b}(\mathbf{x}) := \frac{1}{1 + e^{\mathbf{w}\mathbf{x} + b}}$$

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 $y = f_{\mathbf{w}, b}(\mathbf{x})$ が 1 である確率を予測する。最適な \mathbf{w}, b は

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

で求められる。

例 0.1. (コードはロジスティック回帰モデル.ipynb) iris データセットを用いてロジスティック回帰モデルを実装する。

#データを取得

```
from sklearn import datasets
import numpy as np
iris = datasets.load_iris()
X = iris["data"][:, 3:] # petal width
y = (iris["target"] == 2).astype(np.int) # 1 if Iris virginica, else 0
```

```
from sklearn.linear_model import LogisticRegression
log_reg = LogisticRegression(solver="lbfgs") #ロジスティック回帰モデルを選択
log_reg.fit(X, y) #最適解を求める
```

```
log_reg.coef_ #w の最適解
>array([[4.3330846]])
```

```
log_reg.intercept_ #b の最適解
>array([-7.1947083])
```

```
log_reg.predict([[1.7], [1.5]]) #値を予測
>array([1, 0])
```

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.