先端データ解析論　第5回レポート

電子情報学専攻　48-176403 石毛真修

2017年5月16日

大問 1.

訓練標本入力の平均がゼロであり，2値の出力を $y \in \left\{+\frac{n}{n_+}, -\frac{n}{n_-}\right\}$ であるとき，線形モデル $f_\theta(\mathbf{x}) = \theta^{\mathrm{T}}\mathbf{x}$ を使った最小二乗分類器の決定境界の向きが,

$$\hat{\mathbf{\Sigma}}^{-1}(\hat{\mu}_+ - \hat{\mu}_-)$$

であることを示す.

証明

最小二乗分類なので,

$$\begin{aligned}\hat{\theta} &= argmin_\theta \frac{1}{2}\sum_{i=1}^{n}(f_\theta(\mathbf{x}_i) - y_i) \\ &= \left(X^{\mathrm{T}}X\right)^{-1} X^{\mathrm{T}}\mathbf{y}\end{aligned}$$

$$\begin{aligned}X^{\mathrm{T}}\mathbf{y} = \sum_{i=1}^{n}\mathbf{x}_i y_i &= \sum_{i:y=+}^{n}\mathbf{x}_i y_i + \sum_{i:y=-}^{n}\mathbf{x}_i y_i \\ &= \sum_{i:y=+}^{n}\mathbf{x}_i\frac{n}{n_+} + \sum_{i:y=-}^{n}\mathbf{x}_i\frac{-n}{n_-} = n(\hat{\mu}_+ - \hat{\mu}_-)\end{aligned}$$

一方,

$$\hat{\mathbf{\Sigma}} = \frac{1}{n}X^{\mathrm{T}}X$$

なので,

$$\begin{aligned}\hat{\theta} &= \left(X^{\mathrm{T}}X\right)^{-1} X^{\mathrm{T}}\mathbf{y} = \left(n\hat{\mathbf{\Sigma}}\right)^{-1} X^{\mathrm{T}}\mathbf{y} \\ &= \left(n\hat{\mathbf{\Sigma}}\right)^{-1} n\left(\hat{\mu}_+ - \hat{\mu}_-\right) \\ &= \hat{\mathbf{\Sigma}}^{-1}\left(\hat{\mu}_+ - \hat{\mu}_-\right)\end{aligned}$$

決定境界は，訓練標本入力の平均がゼロであるので，

$$\theta^{\mathrm{T}}\mathbf{x} = \left\{\hat{\mathbf{\Sigma}}^{-1}(\hat{\mu}_{+} - \hat{\mu}_{-})\right\}^{\mathrm{T}}\mathbf{x} = 0$$

となる．よって傾きは，$\hat{\mathbf{\Sigma}}^{-1}(\hat{\mu}_{+} - \hat{\mu}_{-})$

大問 2.

ガウスカーネル回帰により，手書き数字を分類する．今回は，一対多による多クラス分類を行った．まず，次のようなデータ直線 $y = x$ に正規分布から生成されるノイズを載せたデータを作成する．

```
import numpy as np
import pandas as pd
import sklearn.units import shuffle


# Read data
train_data = [
    pd.read_csv('digit_train0.csv', header=None),
    pd.read_csv('digit_train1.csv', header=None),
    pd.read_csv('digit_train2.csv', header=None),
    pd.read_csv('digit_train3.csv', header=None),
    pd.read_csv('digit_train4.csv', header=None),
    pd.read_csv('digit_train5.csv', header=None),
    pd.read_csv('digit_train6.csv', header=None),
    pd.read_csv('digit_train7.csv', header=None),
    pd.read_csv('digit_train8.csv', header=None),
    pd.read_csv('digit_train9.csv', header=None),
]

test_data = [
    pd.read_csv('digit_test0.csv', header=None),
    pd.read_csv('digit_test1.csv', header=None),
    pd.read_csv('digit_test2.csv', header=None),
    pd.read_csv('digit_test3.csv', header=None),
    pd.read_csv('digit_test4.csv', header=None),
    pd.read_csv('digit_test5.csv', header=None),
    pd.read_csv('digit_test6.csv', header=None),
    pd.read_csv('digit_test7.csv', header=None),
    pd.read_csv('digit_test8.csv', header=None),
    pd.read_csv('digit_test9.csv', header=None),
]


def kern(x, c, h=0.2):
    norm = np.linalg.norm(x - c)
    return np.exp(- norm**2 / (2 * (h**2)))


def kern_matrix(x_samples, h=0.2):                                    C
    def kerns(x, sample_X):
        return np.apply_along_axis(lambda xi: kern(x, xi), axis=1, arr=sample_X)
    return np.apply_along_axis(lambda x: kerns(x, x_samples), axis=1, arr=x_samples)


def estimate_theta(samples_x, samples_y, lamb=0.1, h=0.2):
    K = kern_matrix(samples_x, h)
    Q = np.linalg.inv(np.matmul(K, K) + lamb * np.eye(len(samples_x)))
    p = np.matmul(np.transpose(K), samples_y)
    return np.matmul(Q, p)


def kern_model_gen(x_samples, y_samples, lamb=0.1, h=0.2):
    est_theta = estimate_theta(x_samples, y_samples, lamb, h)
    def _model(x):
        return np.dot(est_theta, np.array([kern(x, xs, h) for xs in x_samples]))
    def v_model(X):
        return np.apply_along_axis(_model, axis=1, arr=X)
    return v_model


Judgers = [class_judger_generator(i) for i in range(10)]


def classifier(X):
    judges = np.array([j(X) for j in Judgers])
    return np.argmax(judges, axis=0)


def accuracy():
    y = classifier(test_X)
    test_y = np.array([[i] * 200 for i in range(10)]).flatten()
    correct_num = np.sum(np.equal(y, test_y))
    return correct_num / len(test_y)


print(accuracy())
```

このコードを実行した結果，96.3 % の正解率であった．