先端データ解析論　第2回レポート

電子情報学専攻　48-176403 石毛真修

2017年4月25日

大問 1.

$y = \frac{sin(\pi x)}{\pi x} + 0.1x + 0.2 \times random$ を ガウスカーネルモデル $f_\theta(x) = \sum_{j=1}^{n} \theta_j K(x_i, x_j)$ で近似し，更にそのハイパーパラメータである λ とカーネル関数の分散 h を，交差確認法を使って決定する.

リスト 1: Regression Part

```
import matplotlib.pyplot as plt
import numpy as np


def org_model(x):
  return np.sin(np.pi * x) / (np.pi * x) + 0.1 * x


def get_samples(x_samples, f):
    return f(x_samples) + 0.2* np.random.randn(len(x_samples))


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


kerns = np.vectorize(kern)
def kern_matrix(x_samples, h=0.2):
    return np.array([kerns(xi, x_samples, h) for xi in x_samples])


def estimate_theta(samples_x, samples_y, lamb=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)                          c
    return np.matmul(Q, p)


def kern_model_gen(x_samples, y_samples, lamb=1, h=0.2):
    est_theta = estimate_theta(x_samples, y_samples, lamb, h)
    def _model(x):
        return np.dot(est_theta, kerns(x, x_samples, h))
    v_model = np.vectorize(_model)
    return v_model


  np.random.seed()
  x_min, x_max = -3, 3
  n = 50
  N = 1000
  x = np.linspace(x_min, x_max, n)
  X = np.linspace(x_min, x_max, N)

  y = org_model(x)
  # _y = get_samples(x, org_model)

  lamb = 0.1
  h = 0.7
  est_model = kern_model_gen(x, _y, lamb, h)
  Y = est_model(X)

  plt.scatter(x, _y)
  plt.plot(x, y, 'r-', X, Y, 'b-')
  plt.show()
```

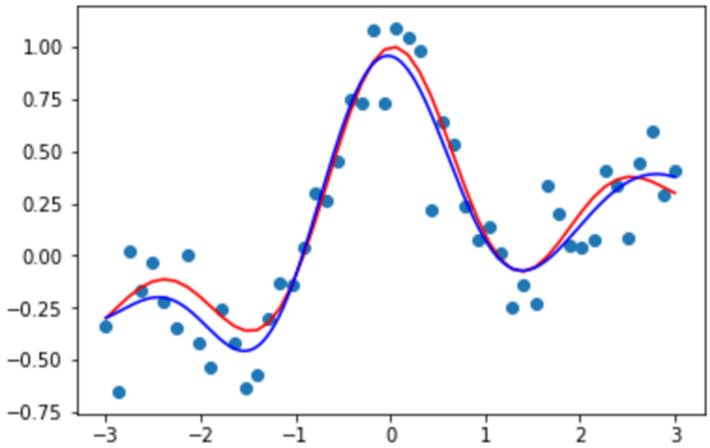

図 1: 曲線 $y = \frac{sin(\pi x)}{\pi x} + 0.1x$ (赤) と $\lambda = 0.1$, $h = 0.7$ のときの回帰曲線 (青).

リスト 2: Cross Validation and Grid Search

```
def split_train_test(x_samples, y_samples, k, i):
    batch_size = ceil(len(x_samples) / k)
    start_idx = i * batch_size
    end_idx = start_idx + batch_size
    if end_idx > len(x_samples):
        end_idx = len(x_samples)
    x_train = np.concatenate((
            x_samples[:start_idx], x_samples[end_idx:]))
    y_train = np.concatenate((
            y_samples[:start_idx], y_samples[end_idx:]))
    x_test = x_samples[start_idx:end_idx]
    y_test = y_samples[start_idx:end_idx]
    return x_train, x_test, y_train, y_test


def cross_validation(x_samples, y_samples, model_gen, k=10):
    ave_err = 0
    for i in range(k):
        x_train, x_test, y_train, y_test =\
                split_train_test(x_samples, y_samples, k, i)
        est_model = model_gen(x_train, y_train)
        diff = est_model(x_test) - y_test
        err = np.dot(diff, diff) / len(x_test)
        ave_err += err
    ave_err /= k
    return ave_err


def grid_search(model_gen, x_samples, y_samples, l_search_points,
                h_search_points):                                     c
    err_min = 10000
    likely_solution = [np.nan, np.nan]
    grid_err = np.zeros((len(h_search_points), len(l_search_points)))
    for i, l in enumerate(l_search_points):
        for j, h in enumerate(h_search_points):
            def _model_gen(x, y):
                return model_gen(x, y, l, h)
            err = cross_validation(x_samples, y_samples, _model_gen)
            grid_err[j, i] = err
            if err < err_min:
                err_min = err
                likely_solution = [l, h]
    return err_min, likely_solution, grid_err


l_search_points = np.logspace(-2, 3, 6)
h_search_points = np.linspace(0.001, 2, 50)

err_min, params, grid_err = grid_search(
  kern_model_gen, x, _y, l_search_points, h_search_points)

from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt

X, Y = np.meshgrid(l_search_points, h_search_points)
fig = plt.figure()
ax = Axes3D(fig)
ax.set_xlabel('Lambda')
ax.set_ylabel('h')
ax.set_zlabel('Square Error')

ax.plot_surface(X, Y, grid_err, rstride=1, cstride=1)
plt.show()
```

このコードを実行することにより，$\lambda = 0.1$，$h = 0.78$ のとき平均誤差が 0.0374637682683 となるような回帰曲線が得られた．

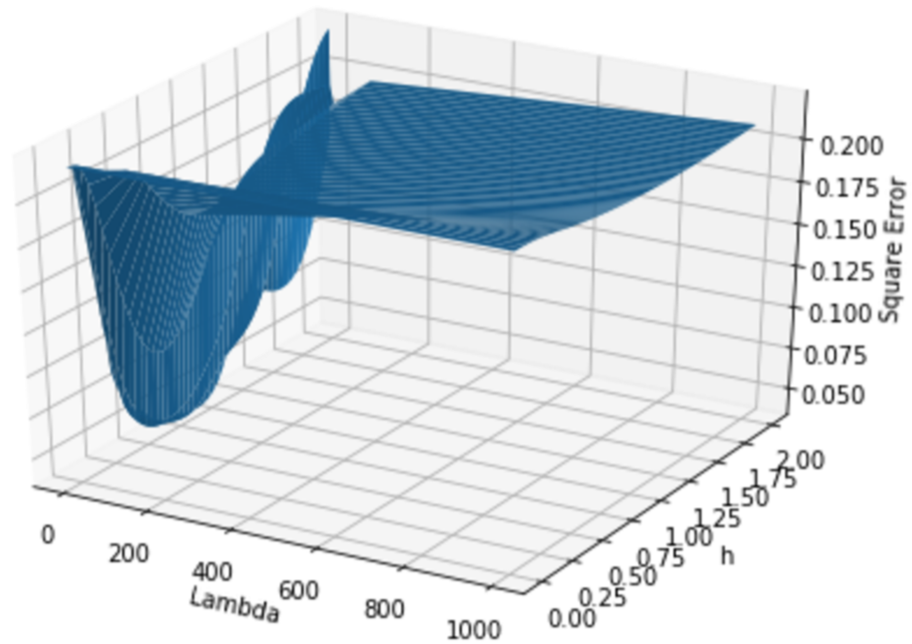

図 2: グリッドサーチの結果

大問 2.

線形モデル $f_\theta = \sum_{j=1}^{b} \theta_j \phi_j(\mathbf{x})$ を用いた l_2-正則化回帰に対するひとつ抜き交差確認による二乗誤差が，次のようになることを示す．

$$\frac{1}{n}\|\tilde{\mathbf{H}}^{-1}\mathbf{H}\mathbf{y}\|^2$$

導出

標本 $(\mathbf{x}_i, y_i)$ を除いて学習したパラメータ $\hat{\theta}_i$ は次のように表される．

$$\hat{\theta}_i = (\mathbf{\Phi}_i^{\mathrm{T}}\mathbf{\Phi}_i + \lambda\mathbf{I})^{-1}\mathbf{\Phi}_i^{\mathrm{T}}\mathbf{y}_i = \mathbf{U}_i^{-1}\mathbf{\Phi}_i^{\mathrm{T}}\mathbf{y}_i$$

$$\begin{aligned}\mathbf{U}_i &= \mathbf{\Phi}_i^{\mathrm{T}}\mathbf{\Phi}_i + \lambda\mathbf{I} \\ &= \mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi} - \phi_i\phi_i^{\mathrm{T}} + \lambda\mathbf{I} \\ &= (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi} + \lambda\mathbf{I}) - \phi_i\phi_i^{\mathrm{T}}\end{aligned}$$

であるので,

$$
\begin{aligned}
\mathbf{U}_i^{-1} &= \{(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi} + \lambda\mathbf{I}) - \phi_i\phi_i^{\mathrm{T}}\}^{-1} \\
&= (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi} + \lambda\mathbf{I})^{-1} + \frac{(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi} + \lambda\mathbf{I})^{-1}\phi_i\phi_i^{\mathrm{T}}(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi} + \lambda\mathbf{I})^{-1}}{1 - \phi_i^{\mathrm{T}}(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi} + \lambda\mathbf{I})^{-1}\phi_i} \\
&= \mathbf{U}^{-1} + \frac{\mathbf{U}^{-1}\phi_i\phi_i^{\mathrm{T}}\mathbf{U}^{-1}}{\gamma_i}
\end{aligned}
$$

ただし,

$$
\gamma_i = 1 - \phi_i^{\mathrm{T}}(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi} + \lambda\mathbf{I})^{-1}\phi_i
$$

以上から,

$$
\hat{\theta}_i = \left\{\mathbf{U}^{-1} + \frac{\mathbf{U}^{-1}\phi_i\phi_i^{\mathrm{T}}\mathbf{U}^{-1}}{\gamma_i}\right\}(\mathbf{\Phi}^{\mathrm{T}}\mathbf{y} - \phi_i y_i) = \left\{\mathbf{U}^{-1} + \frac{\mathbf{U}^{-1}\phi_i\phi_i^{\mathrm{T}}\mathbf{U}^{-1}}{\gamma_i}\right\}\left(\sum_{j\neq i}^{b} y_j\phi_j\right)
$$

一方で最終的な誤差 E は,

$$
E = \frac{1}{n}\sum_{i}^{n}\left(\phi_i^{\mathrm{T}}\hat{\theta}_i - y_i\right)^2
$$

$$
\begin{aligned}
\phi_i^{\mathrm{T}}\hat{\theta}_i - y_i &= \phi_i^{\mathrm{T}}\left\{\mathbf{U}^{-1} + \frac{\mathbf{U}^{-1}\phi_i\phi_i^{\mathrm{T}}\mathbf{U}^{-1}}{\gamma_i}\right\}\left(\sum_{j\neq i}^{b} y_j\phi_j\right) - y_i \\
&= \left\{\phi_i^{\mathrm{T}}\mathbf{U}^{-1} + \frac{\phi_i^{\mathrm{T}}\mathbf{U}^{-1}\phi_i\phi_i^{\mathrm{T}}\mathbf{U}^{-1}}{\gamma_i}\right\}\left(\sum_{j\neq i}^{b} y_j\phi_j\right) - y_i \\
&= \left(1 + \frac{1-\gamma_i}{\gamma_i}\right)\phi_i^{\mathrm{T}}\mathbf{U}^{-1}\left(\sum_{j\neq i}^{b} y_j\phi_j\right) - y_i \\
&= \frac{1}{\gamma_i}\phi_i^{\mathrm{T}}\mathbf{U}^{-1}\left(\sum_{j\neq i}^{b} y_j\phi_j\right) - y_i \\
&= -\left(y_i - \sum_{j\neq i}^{b} y_j\frac{\phi_i^{\mathrm{T}}\mathbf{U}^{-1}\phi_j}{\gamma_i}\right)
\end{aligned}
$$

よって,

$$
nE = \sum_{i}^{n}\left(y_i - \sum_{j\neq i}^{b} y_j\frac{\phi_i^{\mathrm{T}}\mathbf{U}^{-1}\phi_j}{\gamma_i}\right)^2
$$

$$\mathbf{H} = \mathbf{I} - \mathbf{\Phi}\mathbf{U}^{-1}\mathbf{\Phi}^{\mathrm{T}} \qquad \tilde{H}_{ii} = 1 - \phi_i^{\mathrm{T}}\mathbf{U}^{-1}\phi_i = \gamma_i$$

であるので,

$$\tilde{\mathbf{H}}^{-1}\mathbf{H}\mathbf{y} = \begin{bmatrix} 1 & -\frac{\phi_1^{\mathrm{T}}\mathbf{U}^{-1}\phi_2}{\gamma_1} & -\frac{\phi_1^{\mathrm{T}}\mathbf{U}^{-1}\phi_3}{\gamma_1} & ... & -\frac{\phi_1^{\mathrm{T}}\mathbf{U}^{-1}\phi_n}{\gamma_1} \\ -\frac{\phi_2^{\mathrm{T}}\mathbf{U}^{-1}\phi_1}{\gamma_2} & 1 & -\frac{\phi_2^{\mathrm{T}}\mathbf{U}^{-1}\phi_3}{\gamma_2} & ... & -\frac{\phi_2^{\mathrm{T}}\mathbf{U}^{-1}\phi_n}{\gamma_2} \\ & . & & & \\ & & . & & \\ & & & . & \\ -\frac{\phi_n^{\mathrm{T}}\mathbf{U}^{-1}\phi_1}{\gamma_2} & ... & ... & ... & 1 \end{bmatrix}\mathbf{y}$$

$$= \begin{bmatrix} y_1 - \sum_{j\neq 1}^{b} y_j \frac{\phi_1^{\mathrm{T}}\mathbf{U}^{-1}\phi_j}{\gamma_1} \\ y_2 - \sum_{j\neq 2}^{b} y_j \frac{\phi_2^{\mathrm{T}}\mathbf{U}^{-1}\phi_j}{\gamma_2} \\ . \\ . \\ . \\ y_n - \sum_{j\neq n}^{b} y_j \frac{\phi_n^{\mathrm{T}}\mathbf{U}^{-1}\phi_j}{\gamma_n} \end{bmatrix}$$

以上から,

$$\|\tilde{\mathbf{H}}^{-1}\mathbf{H}\mathbf{y}\|^2 = \sum_{i}^{n} \left(y_i - \sum_{j\neq i}^{b} y_j \frac{\phi_i^{\mathrm{T}}\mathbf{U}^{-1}\phi_j}{\gamma_i} \right)^2 = nE$$