プログラム

実行環境と用いた言語・ライブラリを以下の表 1 に示す。

表 1: プログラムの実行環境

OS	:	Microsoft Windows 10 Pro (64bit)
CPU	:	Intel(R) Core(TM) i5-4300U
RAM	:	4.00 GB
使用言語	:	Python3.6
可視化	:	matplotlib ライブラリ

Listings 1: assignment2.py

```
# -*- coding: utf-8 -*-


import numpy as np
import scipy.linalg
import matplotlib.pyplot as plt


def generate_data_1(n=100):
    x = np.concatenate([
            np.random.randn(n, 1) * 2,
            np.random.randn(n, 1)
        ], axis=1)
    return x


def generate_data_2(n=100):
    x = np.concatenate([
            np.random.randn(n, 1) * 2,
            2 * np.round(np.random.rand(n, 1)) - 1 + np.random.randn(n, 1) / 3.
        ], axis=1)
    return x


def calc_norm(x, c, save_memory=False):
    if save_memory:
        n_x = x.shape[1]
        n_c = c.shape[0]
        d = x.shape[-1]
        norm = np.zeros((n_c, n_x))
        x = np.reshape(x, (n_x, d))
        c = np.reshape(c, (n_c, d))
```

```
        for i in range(len(x)):
            x_i = x[i]
            norm[i, :] = np.sum((x_i - c)**2, axis=-1)
    else:
        norm = np.sum((x - c) ** 2, axis=-1)
    return norm


def gauss_kernel(x, c, h, save_memory=False):
    norm = calc_norm(x, c, save_memory)
    ker = np.exp(- norm / (2*h**2))
    return ker


def calc_design_matrix(x, c, h, kernel):
    return kernel(x[None], c[:, None], h)


def train(x, n_components=1, h=1., kernel=gauss_kernel):
    K = calc_design_matrix(x, x, h, kernel)

    W = K
    D = np.diag(W.sum(axis=1))
    L = D - W

    X = x.T
    A = X.dot(L).dot(X.T)
    B = X.dot(D).dot(X.T)

    eigen_values, eigen_vectors = scipy.linalg.eig(A, B)

    # normalize
    for i in range(len(eigen_vectors)):
        eigen_vectors[i] = eigen_vectors[i]/np.linalg.norm(eigen_vectors[i])

    T = eigen_vectors[::-1][:n_components]
    return T


def visualize(x, T, h=1., grid_size=100, path=None):
    plt.xlim(-6., 6.)
    plt.ylim(-6., 6.)
    plt.plot(x[:, 0], x[:, 1], 'rx')
    plt.plot(np.array([-T[:, 0], T[:, 0]]) * 9, np.array([-T[:, 1], T[:,
    1]]) * 9)

    if path:
```

```
79          plt.savefig(path)
80      plt.show()
81
82
83  def main():
84      # settings
85      n = 100
86      n_components = 1
87      h = 1.0
88      fig_path = '../figures/assignment2_result_data1.png'
89      np.random.seed(0)
90
91
92      # generate data
93      x = generate_data_1(n)
94      #print(x.shape)
95
96
97      # train
98      T = train(x, h=h)
99
100
101     # result
102     print(f'#data: {n}  #Component: {n_components}  h = {h}')
103     #print('T = \n', T)
104
105     visualize(x, T, h=h, path=fig_path)
106
107
108 if __name__ == '__main__':
109     main()
```