

## 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: assignment1.py

```
1  # -*- coding: utf-8 -*-
2
3
4  import numpy as np
5  import matplotlib.pyplot as plt
6
7
8  def generate_data(sample_size=90, n_class=3):
9      x = (
10          np.random.normal(size=(sample_size//n_class, n_class))
11          + np.linspace(-3., 3., n_class)
12          ).flatten()
13      y = np.broadcast_to(
14          np.arange(n_class),
15          (sample_size // n_class, n_class)
16          ).flatten()
17      return x, y
18
19
20 def train(x, y, h, lamb, n_class):
21     n_sample = x.shape[0]
22     theta = np.zeros((n_sample, n_class))
23     K = np.exp(-(x - x[:, None])**2 / (2*h**2))
24     for label in range(n_class):
25         pi_y = (y == label).astype(int)
26         theta[:, label] = np.linalg.inv(K.T.dot(K) +
27             lamb*np.eye(n_sample)).dot(K).dot(pi_y)
28     return theta
29
30 def visualize(x, y, theta, h, num=100, path=None):
31     X = np.linspace(-5, 5, num=num)
32     K = np.exp(-(x - X[:, None])**2 / (2*h**2))
```

```

33     logit = K.dot(theta)
34     unnormalized_prob = logit - logit.min(axis=1, keepdims=True)
35     prob = unnormalized_prob / unnormalized_prob.sum(axis=1, keepdims=True)
36
37     plt.clf()
38     plt.xlim(-5, 5)
39     plt.ylim(-.3, 1.8)
40
41     plt.plot(X, prob[:, 0], c='blue')
42     plt.plot(X, prob[:, 1], c='red')
43     plt.plot(X, prob[:, 2], c='green')
44
45     plt.scatter(x[y == 0], -.1 * np.ones(len(x) // 3), c='blue', marker='o')
46     plt.scatter(x[y == 1], -.2 * np.ones(len(x) // 3), c='red', marker='x')
47     plt.scatter(x[y == 2], -.1 * np.ones(len(x) // 3), c='green', marker='v')
48
49     if path:
50         plt.savefig(path)
51     plt.show()
52
53
54 def main():
55     # settings
56     n_sample = 90
57     n_class = 3
58     h = 2
59     lamb = 1e-4
60     fig_path = '../figures/assignment1_result.png'
61     np.random.seed(0)
62
63     # load data
64     x, y = generate_data(n_sample, n_class)
65     #print(x)
66     #print(y)
67
68     # train
69     theta = train(x, y, h, lamb, n_class)
70
71     # result
72     print(f'#Sample: {n_sample}      #Class: {n_class}')
73     print(f'h = {h}      lambda = {lamb}')
74     print(f'theta: \n{theta}')
75     visualize(x, y, theta, h, path=fig_path)
76
77
78 if __name__ == '__main__':
79     main()

```