

Project

report1

\_\_pycache\_\_

data

2D-class1\_A.dat

2D-class1\_B.dat

2D-class1\_C.dat

2D-class1\_D.dat

2D-class2\_A.dat

2D-class2\_B.dat

2D-class2\_C.dat

2D-class2\_D.dat

2D-test\_A.dat

2D-test\_B.dat

2D-test\_C.dat

2D-test\_D.dat

results

.DS\_Store

answer.pdf

main.py

report1.aux

report1.log

report1.pages

report1.pdf

report1.synctex.gz

report1.tex

report1ReadMe.pdf

resources.py

resourcesNewVersion.py

temp

temp.txt

sample2.c

main.py

Settings

report1.tex

resources...

1 import resourcesNewVersion as re

2 import numpy as nu

3 from collections import namedtuple

4 import matplotlib.pyplot as pl

5

6

7

8 # NOTE: - global constants

9

10 FILES = [

11 'data/2D-class1\_' + re.DATASET\_ID + '.dat',

12 'data/2D-class2\_' + re.DATASET\_ID + '.dat',

13 'data/2D-test\_' + re.DATASET\_ID + '.dat'

14 ]

15

16

17

18 # NOTE: - read data:

19

20 class1, class2, testData = re.ClassData(FILES[0]

21

22

23

24 # NOTE: - 1(a) 判別基準評価:

25

26 Results = namedtuple('Results', 'correct nearestNeighbor euclideanDistance weightDistance similarity')

27 results = Results(

28 correct = testData.correctClasses,

29 nearestNeighbor = testData.nearestNeighborMethod,

30 euclideanDistance = testData.euclideanDistanceMethod,

31 weightDistance = testData.weightDistanceMethod,

32 similarity = testData.similarityMethod(class1, class2)

33 )

34

35 Errata = namedtuple('Errata', 'nearestNeighbor euclideanDistance weightDistance similarity')

36 errata = Errata(

37 nearestNeighbor = re.errataOf(results.nearestNeighbor, testData.euclideanDistanceMethod, testData.weightDistanceMethod, testData.similarityMethod, class1, class2),

38 euclideanDistance = re.errataOf(results.euclideanDistance, testData.euclideanDistanceMethod, testData.weightDistanceMethod, testData.similarityMethod, class1, class2),

39 weightDistance = re.errataOf(results.weightDistance, testData.euclideanDistanceMethod, testData.weightDistanceMethod, testData.similarityMethod, class1, class2),

40 similarity = re.errataOf(results.similarity, testData.euclideanDistanceMethod, testData.weightDistanceMethod, testData.similarityMethod, class1, class2)

41 )

42

43 RecognitionRates = namedtuple('RecognitionRates', 'nearestNeighbor euclideanDistance weightDistance similarity')

44 recognitionRates = RecognitionRates(

45 nearestNeighbor = re.recognitionRateOf(errata.nearestNeighbor, testData.euclideanDistanceMethod, testData.weightDistanceMethod, testData.similarityMethod, class1, class2),

46 euclideanDistance = re.recognitionRateOf(errata.euclideanDistance, testData.euclideanDistanceMethod, testData.weightDistanceMethod, testData.similarityMethod, class1, class2),

47 weightDistance = re.recognitionRateOf(errata.weightDistance, testData.euclideanDistanceMethod, testData.weightDistanceMethod, testData.similarityMethod, class1, class2),

48 similarity = re.recognitionRateOf(errata.similarity, testData.euclideanDistanceMethod, testData.weightDistanceMethod, testData.similarityMethod, class1, class2)