```
39      weightDistance = re.errataOf(results.weightDistance, testData.correctClasses),
40      similarity = re.errataOf(results.similarity, testData.correctClasses)
41  )
42
43  RecognitionRates = namedtuple('RecognitionRates', 'nearestNeighbor euclideanDistance we
44  recognitionRates = RecognitionRates(
45      nearestNeighbor = re.recognitionRateOf(errata.nearestNeighbor),
46      euclideanDistance = re.recognitionRateOf(errata.euclideanDistance),
47      weightDistance = re.recognitionRateOf(errata.weightDistance),
48      similarity = re.recognitionRateOf(errata.similarity)
49  )
50
51
52
53  # NOTE: - 1(b) 散布図と境界線
54
55  RecognitionLines = namedtuple('RecognitionLines', 'nearestNeighbor euclideanDistance we
56  recognitionLines = RecognitionLines(
57      nearestNeighbor = re.recogLineOfNe(class1, class2),
58      euclideanDistance = re.recogLineOfEu(class1, class2),
59      weightDistance = re.recogLineOfWe(class1, class2, nu.array([1.0, 20.0]).reshape(2,1
60      similarity = re.recogLineOfSi(class1, class2)
61  )
62
63
64  for plotTitle, result in zip(  results._asdict().keys(), results ):
65      re.plotResultOf(result, class1, class2, testData, plotTitle.capitalize(),
66          recognitionLines._asdict().get('{}'.format(plotTitle), False))
67
68  # re.plotResultOf(results.weightDistance, class1, class2, testData, 'Weight(1,20)', rec
69
70
71
72
73
74
75
76
77
```