

```

1  def calcCutoffDistance(classesCoun, instancesMa, ordinateCoun, categoryTarge,
2  categoryOpposit, vectorWeightsCur, argClasse):
3  '''получает количество классов, количество экземпляров в самом объемном классе,
4  количество ординат, номера целевой и противоположной категории и матрицу аргументов
5  по классам возвращает количество отсеченных и расстояние до стенки коридора'''
6  # Расчет скалярного произведения
7  iMatrix = 0
8  while iMatrix < classesCoun:
9      iRow = 0
10     while iRow < instancesMa:
11         if (argClasse[iMatrix][iRow][0] * argClasse[iMatrix][iRow][0] +
12             argClasse[iMatrix][iRow][1] *
13             argClasse[iMatrix][iRow][1]) == 0:
14             break
15         argClasse[iMatrix][iRow][ordinateCoun + 2] = np.dot(argClasse[iMatrix,
16             iRow, :ordinateCoun + 1], vectorWeightsCur)
17         iRow += 1
18         iMatrix += 1
19     # установка признака отсеченности целевой категории
20     # получение максимума скалярного произведения из НЕ целевых категорий
21     maxNoTarget = setFlagCuOffCategoryTarget(categoryTarge, classesCoun,
22         instancesMa, ordinateCoun, argClasse)
23     # установка признака отсеченности противоположной категории
24     # получение минимума скалярного произведения из НЕ противоположных категорий
25     minNoOpposite = setFlagCuOffCategoryOpposite(categoryOpposit, classesCoun,
26         instancesMa, ordinateCoun, argClasse)
27     # расчет количества отсеченных плюсом, минусом и их суммы
28     countCutOffPlus = calculationCutOffPlusNumber(categoryTarge, instancesMa,
29         ordinateCoun, argClasse)
30     countCutOffMinus = calculationCutOffMinusNumber(categoryOpposit, instancesMa,
31         ordinateCoun, argClasse)
32     countCutOffSum = countCutOffPlus + countCutOffMinus
33     # расчет минимального расстояния до одной из стенок коридора
34     distanceCutOffPlus = calcDistancePlus(maxNoTarget, categoryTarge, instancesMa,
35         ordinateCoun, argClasse)
36     distanceCutOffMinus = calcDistanceMinus(minNoOpposite, categoryOpposit,
37         instancesMa, ordinateCoun, argClasse)
38     if distanceCutOffPlus < distanceCutOffMinus:
39         distanceCutOffMi = distanceCutOffPlus
40     else:
41         distanceCutOffMi = distanceCutOffMinus
42     return countCutOffSum, distanceCutOffMi
43
44 def calcDescentDirection(classesCoun, instancesMa, ordinateCoun, categoryTarge,
45 categoryOpposit, vectorWeightsCur, argClasse):
46 '''получает количество классов, количество экземпляров в самом объемном классе,
47 количество ординат, номера целевой и противоположной категории и матрицу аргументов
48 по классам возвращает направление спуска +-1'''
49 deltaMultiplierCur = 1
50 weightsOl = vectorWeightsCurr[iDelta]
51 vectorWeightsCurr[iDelta] = weightsOl + deltaMultiplierCur
52 ww = calcCutoffDistance(classesCoun, instancesMa, ordinateCoun, categoryTarge,
53 categoryOpposit, vectorWeightsCur, argClasse)
54 countCutOffCurr = ww[0]
55 distanceCutOffCurr = ww[1]
56 if (countCutOffCurr < countCutOffPrev) or (distanceCutOffCurr >=
57     distanceCutOffPrev):
58     deltaMultiplierCur = -1
59 return deltaMultiplierCur
60
61 def calculationCutOffPlusNumber(categorySpecified, instanMax, ordCount, aClasses):
62 '''получает номер целевой категории, количество экземпляров в самом объемном классе,
63 количество ординат и матрицу аргументов по классам возвращает количество отсеченных
64 плюсом'''
65 countCutOffPlu = 0
66 iRow = 0
67 while iRow < instanMax:
68     if aClasses[categorySpecified][iRow][ordCount + 3] == 1:
69         countCutOffPlu += 1
70     iRow += 1
71 return countCutOffPlu

```

```

55 def calculationCutOffMinusNumber(categorySpecified, instanMax, ordCount, aClasses):
56     '''получает номер противоположной категории, количество экземпляров в самом объемном
классе, количество ординат и матрицу аргументов по классам возвращает количество
отсеченных минусом'''
57     countCutOffMinu = 0
58     iRow = 0
59     while iRow < instanMax:
60         if aClasses[categorySpecified][iRow][ordCount + 3] == 1:
61             countCutOffMinu += 1
62             iRow += 1
63     return countCutOffMinu
64
65 def setFlagCuOffCategoryTarget(categorySpecified, clCount, instanMax, ordCount,
aClasses):
66     '''получает номер целевой категории, количество классов, количество экземпляров в
самом объемном классе, количество ординат и матрицу аргументов по классам
устанавливает признак отсеченности целевой категории возвращает максимум скалярного
произведения из НЕ целевых категорий'''
67     maNoTarget = 0
68     iMatrix = 0
69     while iMatrix < clCount:
70         if iMatrix == categorySpecified:
71             break
72         iRow = 0
73         while iRow < instanMax:
74             if aClasses[iMatrix][iRow][ordCount+2] > maNoTarget:
75                 maNoTarget = aClasses[iMatrix][iRow][ordCount+2]
76                 iRow += 1
77             iMatrix += 1
78     iRow = 0
79     while iRow < instanMax:
80         if aClasses[categorySpecified][iRow][ordCount+2] > maNoTarget:
81             aClasses[categorySpecified][iRow][ordCount + 3] = 1
82         else:
83             aClasses[categorySpecified][iRow][ordCount + 3] = 0
84             iRow += 1
85     return maNoTarget
86
87 def setFlagCuOffCategoryOpposite(categorySpecified, clCount, instanMax, ordCount,
aClasses):
88     '''получает номер противоположной категории, количество классов, количество
экземпляров в самом объемном классе, количество ординат и матрицу аргументов по
классам устанавливает признак отсеченности противоположной категории возвращает
минимум скалярного произведения из НЕ противоположной категорий'''
89     miNoTarget = 0
90     iMatrix = 0
91     while iMatrix < clCount:
92         if iMatrix == categorySpecified:
93             break
94         iRow = 0
95         while iRow < instanMax:
96             if aClasses[iMatrix][iRow][ordCount+2] < miNoTarget:
97                 miNoTarget = aClasses[iMatrix][iRow][ordCount+2]
98                 iRow += 1
99             iMatrix += 1
100     iRow = 0
101     while iRow < instanMax:
102         if aClasses[categorySpecified][iRow][ordCount+2] < miNoTarget:
103             aClasses[categorySpecified][iRow][ordCount + 3] = 1
104         else:
105             aClasses[categorySpecified][iRow][ordCount + 3] = 0
106             iRow += 1
107     return miNoTarget
108
109
110
111
112
113
114
115
116

```

```

117 def calculationCutOffPlusNumber(categorySpecified, instanMax, ordCount, aClasses):
118     '''получает номер целевой категории, количество экземпляров в самом объемном классе,
    количество ординат и матрицу аргументов по классам возвращает количество отсеченных
    плюсом'''
119     countCutOffPlu = 0
120     iRow = 0
121     while iRow < instanMax:
122         if aClasses[categorySpecified][iRow][ordCount + 3] == 1:
123             countCutOffPlu += 1
124             iRow += 1
125     return countCutOffPlu
126
127 def calculationCutOffMinusNumber(categorySpecified, instanMax, ordCount, aClasses):
128     '''получает номер противоположной категории, количество экземпляров в самом объемном
    классе, количество ординат и матрицу аргументов по классам возвращает количество
    отсеченных минусом'''
129     countCutOffMinu = 0
130     iRow = 0
131     while iRow < instanMax:
132         if aClasses[categorySpecified][iRow][ordCount + 3] == 1:
133             countCutOffMinu += 1
134             iRow += 1
135     return countCutOffMinu
136
137 def calcDistanceMinus(miNoTarget, categorySpecified, instanMax, ordCount, aClasses):
138     '''получает минимум скалярного произведения, номер целевой категории, количество
    экземпляров в самом объемном классе, количество ординат и матрицу аргументов по
    классам возвращает минимальное расстояние до отрицательной стенки коридора'''
139     distanceCutOffPrev = -miNoTarget
140     distanceCutOffCurr = -miNoTarget
141     iRow = 0
142     while iRow < instanMax:
143         if aClasses[categorySpecified][iRow][ordCount + 3] == 0:
144             distanceCutOffCurr = aClasses[categorySpecified][iRow][ordCount + 2] -
145             miNoTarget
146             if distanceCutOffCurr < distanceCutOffPrev:
147                 distanceCutOffPrev = distanceCutOffCurr
148             iRow += 1
149     return distanceCutOffPrev
150
151 def calcDistancePlus(maNoTarget, categorySpecified, instanMax, ordCount, aClasses):
152     '''получает максимум скалярного произведения, номер целевой категории, количество
    экземпляров в самом объемном классе, количество ординат и матрицу аргументов по
    классам возвращает минимальное расстояние до положительной стенки коридора'''
153     distanceCutOffPrev = maNoTarget
154     distanceCutOffCurr = maNoTarget
155     iRow = 0
156     while iRow < instanMax:
157         if aClasses[categorySpecified][iRow][ordCount + 3] == 0:
158             distanceCutOffCurr = maNoTarget -
159             aClasses[categorySpecified][iRow][ordCount + 2]
160             if distanceCutOffCurr < distanceCutOffPrev:
161                 distanceCutOffPrev = distanceCutOffCurr
162             iRow += 1
163     return distanceCutOffPrev

```