Fisher's Iris, ADALINE, and Python

緑川章一 *

分類問題を解くアルゴリズムに ADALINE(Adaptive Linear Neuron) と呼ばれるものがある。これは、M 種類のデータ $\{x_0, x_1, x_2, \cdots, x_{M-1}\}$ があるとき、その線形線形結合から、

$$y(x_0, x_1, \cdots, x_{M-1}) = \sum_{i=0}^{M-1} w_i x_i + \theta \tag{1}$$

を作り、連続値である $y(x_0, x_1, \cdots, X_{M-1})$ を離散化することにより目的の値 y を得るものである。関数 $y(x_0, x_1, \cdot, x_{M-1})$ の係数 w_i の値は、この $y(x_0, x_1, \cdot, x_{M-1})$ と、離散化された値 y との差の平方が最小となるように決める。すなわち、M 種類のデータの個数が n コある場合には、

$$Q(w_0, w_1, \cdots, w_{M-1}, \theta) = \sum_{j=0}^{n-1} \Big(y(x_0^j, x_1^j, \cdots, x_{M-1})^j - y^j \Big)^2 \tag{2}$$

が最小となるようにする。そして、この値を求める方法が最小二乗法 (least squares method) である。

この ADALINE のアルゴリズムを、Fisher が線形判別分析に用いたアヤメに適応しよう。アヤメのデータは、Python の scikit-learn からダウンロードすることができる。ここには、0 から 2 までの数字でラベリングされた 3 種類のアヤメ、

Setosa	セトーサ	0
Versicolour	ヴェルシカラー	1
Virginica	ヴァージニカ	2

のデータが載っている。測定部位は、

w_0	がく片の長さ	sepal length (cm)
w_1	がく片の幅	sepal width (cm)
w_2	花弁の長さ	petal length (cm)
w_3	花弁の幅	petal width (cm)

の 4 カ所である。

*Shoichi Midorikawa

実際のデータは、こんな感じ。

	0	1	2	3	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

アヤメの種類のラベルに用いた数字を教師データ (目的変数)y、x_0, x_1, x_2, x_3 を説明変数として、最小二乗法により $\boldsymbol{w} = (w_0, w_1, w_2, w_3)$ と θ を求める。得られた値は、それぞれ、

$$\boldsymbol{w} = (-0.11190585,\ -0.04007949,\ 0.22864503,\ 0.60925205), \tag{3}$$

$$\theta = 0.18649524720625021 \tag{4}$$

である。

これらの値を (1) 式に代入し、改めて 150 個のアヤメにたいして、$y(x_0, x_1, x_2, x_3)$ の値を計算する。この値の小数点以下を四捨五入してアヤメの種類の推定を行う。

作成したプログラムは、次ページの通りである。

プログラム

```
#  application of ADALINE to iris data

import pandas as pd
import numpy as np
from pandas import Series,DataFrame

from sklearn.datasets import load_iris
iris = load_iris()
#print(iris.DESCR)
#print(iris.feature_names)

iris_df = DataFrame(iris.data)
iris_df.to_csv("iris_df.csv")
iris_df['target'] = DataFrame(iris.target)
#print(iris_df.head())

from sklearn.linear_model import LinearRegression
#インスタンス
lr = LinearRegression()

#説明変数を縦 (1) の列と指定して削除
X = iris_df.drop("target", 1)
#Y に目的変数を入れる
Y = iris_df.target

lr.fit(X,Y)
w = lr.coef_
theta = lr.intercept_

j=0
for i in range(150):
    # i 行を取り出し、それをベクトル XV とする
    XV=X.loc[i]
    y=np.dot(XV,w)+theta
    print('{:>3}'.format(i), '{:<10.6f}'.format(y), Y[i], end="")
    if(round(y) != Y[i]):
        print("   incorrect")
        j+=1
    else:
        print("")

print("間違いの数：",j)
```

計算結果は、以下の通りである。

```
 0 -0.082549  0
 1 -0.040128  0
 2 -0.048628  0
 3 0.012300   0
 4 -0.075367  0
 5 0.058291   0
 6 0.038337   0
 7 -0.044486  0
 8 0.019832   0
 9 -0.082197  0
10 -0.101273  0
11 0.000759   0
12 -0.089863  0
13 -0.102504  0
14 -0.226652  0
15 -0.041049  0
16 -0.033167  0
17 -0.021624  0
18 -0.032198  0
19 -0.010783  0
20 -0.043520  0
21 0.054150   0
22 -0.122062  0
23 0.176836   0
24 0.069353   0
25 -0.005590  0
26 0.100229   0
27 -0.070875  0
28 -0.089732  0
29 0.019966   0
30 0.012783   0
31 0.032602   0
32 -0.155848  0
33 -0.155367  0
34 -0.021272  0
35 -0.105064  0
36 -0.150176  0
37 -0.125101  0
38 -0.007040  0
39 -0.055677  0
40 -0.033298  0
41 0.070750   0
42 -0.015056  0
43 0.218071   0
```

```
44 0.141600   0
45 0.031987   0
46 -0.048844  0
47 -0.014573  0
48 -0.090082  0
49 -0.063343  0
50 1.202484   1
51 1.284824   1
52 1.324337   1
53 1.185438   1
54 1.312530   1
55 1.257340   1
56 1.398661   1
57 0.905746   1
58 1.175481   1
59 1.241039   1
60 0.956317   1
61 1.280199   1
62 0.950717   1
63 1.315224   1
64 1.058742   1
65 1.171471   1
66 1.382365   1
67 0.975923   1
68 1.347285   1
69 1.021517   1
70 1.592146   1   incorrect
71 1.098255   1
72 1.415528   1
73 1.197381   1
74 1.129269   1
75 1.186669   1
76 1.263762   1
77 1.495441   1
78 1.341610   1
79 0.853935   1
80 1.013851   1
81 0.930061   1
82 1.052045   1
83 1.547738   1   incorrect
84 1.404746   1
85 1.382496   1
86 1.300989   1
87 1.187371   1
88 1.169056   1
```

89	1.177422	1	
90	1.203947	1	
91	1.288351	1	
92	1.078917	1	
93	0.898564	1	
94	1.203945	1	
95	1.119805	1	
96	1.184738	1	
97	1.151650	1	
98	0.871689	1	
99	1.165882	1	
100	2.244226	2	
101	1.752895	2	
102	1.900160	2	
103	1.742324	2	
104	2.005364	2	
105	2.004259	2	
106	1.602589	2	
107	1.790469	2	
108	1.759322	2	
109	2.154352	2	
110	1.715447	2	
111	1.731481	2	
112	1.842274	2	
113	1.810162	2	
114	2.053513	2	
115	1.955142	2	
116	1.693070	2	
117	2.044794	2	
118	2.199544	2	
119	1.483988	2	incorrect
120	1.990647	2	
121	1.786465	2	
122	1.963023	2	
123	1.590288	2	
124	1.887170	2	
125	1.721043	2	
126	1.574606	2	
127	1.600645	2	
128	1.917917	2	
129	1.561479	2	
130	1.798483	2	
131	1.831969	2	
132	1.978842	2	
133	1.449234	2	incorrect

```
134 1.533028   2
135 2.000596   2
136 2.087835   2
137 1.700253   2
138 1.588971   2
139 1.804211   2
140 2.055097   2
141 1.857468   2
142 1.752895   2
143 2.047566   2
144 2.130871   2
145 1.906721   2
146 1.682094   2
147 1.746327   2
148 1.992372   2
149 1.668756   2
間違いの数： 4
```

150 個のアヤメのデータをこの分類器にかけた場合、失敗したのはわずかに 4 個であったので、正解率は 97.3%である。

この状況を視覚的に理解するために、(3), (4) の値を用いて、

$$x = y(x_0,\ x_1,\ x_2,\ x_3) = w_0x_0 + w_1x_1 + w_2x_2 + w_3x_3 + \theta, \tag{5}$$

$$y = \begin{cases} 0 & \text{セトーサ } (Setosa) \\ 1 & \text{ヴェルシカラー}(Versicolour) \\ 2 & \text{ヴァージニカ } (Virginica) \end{cases} \tag{6}$$

とおいて 150 コのアヤメのデータをプロットすると、以下のようになる。