

# 機械学習 識別モデル(練習問題回答例)

管理工学科  
篠沢佳久

# 回答例

# SGD(デルタールール)

必要なパッケージのインポート

```
import numpy as np
```

```
from sklearn.linear_model import SGDClassifier
```

```
from sklearn.preprocessing import StandardScaler
```

```
from sklearn.metrics import classification_report, accuracy_score,  
confusion_matrix
```

# クラス数, データ数, 次元数

```
CLASS = 3      # クラス数
```

```
DATA = 100     # データ数
```

```
D = 4          # 特徴の次元数
```

## 学習データの読み込み

```
train_file = "train-1.csv"
```

```
train_data = np.loadtxt(train_file,delimiter=",", usecols=(0,1,2,3))
```

```
train_label = np.loadtxt(train_file,delimiter=",", usecols=(4)).astype(np.int32)
```

## テストデータの読み込み

```
test_file = "test-1.csv"
```

```
test_data = np.loadtxt(test_file,delimiter=",", usecols=(0,1,2,3))
```

```
test_label = np.loadtxt(test_file,delimiter=",", usecols=(4)).astype(np.int32)
```

# 学習データの平均値, 標準偏差の計算

```
sc = StandardScaler()
```

```
sc.fit( train_data )
```

データの読み込みは練習問題①  
を参考にして下さい

```
print( "¥n [ 平均値 ]" )
```

```
print( sc.mean_ )
```

```
print( "¥n [ 分散 ]" )
```

```
print( sc.var_ )
```

```
print( "¥n [ 標準偏差 ]" )
```

```
print( sc.scale_ )
```

標準化はしなくてもかまいません

## # 標準化

```
train_data_std = sc.transform(train_data)
test_data_std = sc.transform(test_data)
```

## # 線形識別関数 (SGD)

```
model = SGDClassifier(eta0=0.1, max_iter=1000, learning_rate='optimal' ,
loss='hinge' , early_stopping=True, validation_fraction=0.1,
n_iter_no_change=5, penalty='l2')
```

## # 学習

```
model.fit(train_data_std, train_label)
```

## # 予測

```
predict = model.predict(test_data_std)
df = model.decision_function(test_data_std)
```

```
print( "¥n [ 重みベクトル ]" )  
print( model.coef_ )
```

```
print( "¥n [ 切片 ]" )  
print( model.intercept_ )
```

# 予測値, 正解ラベル

```
print( "¥n 予測値, 正解ラベル" )  
for i in range(len(test_data_std)):  
    for j in range(CLASS):  
        print( "{0:12.3f}".format( df[i][j] ) , end=" " )  
    print( "({0}) : {1}".format( predict[i], test_label[i] ) )
```

```
print( "¥n [ 予測結果 ]" )  
print( classification_report(test_label, predict) )
```

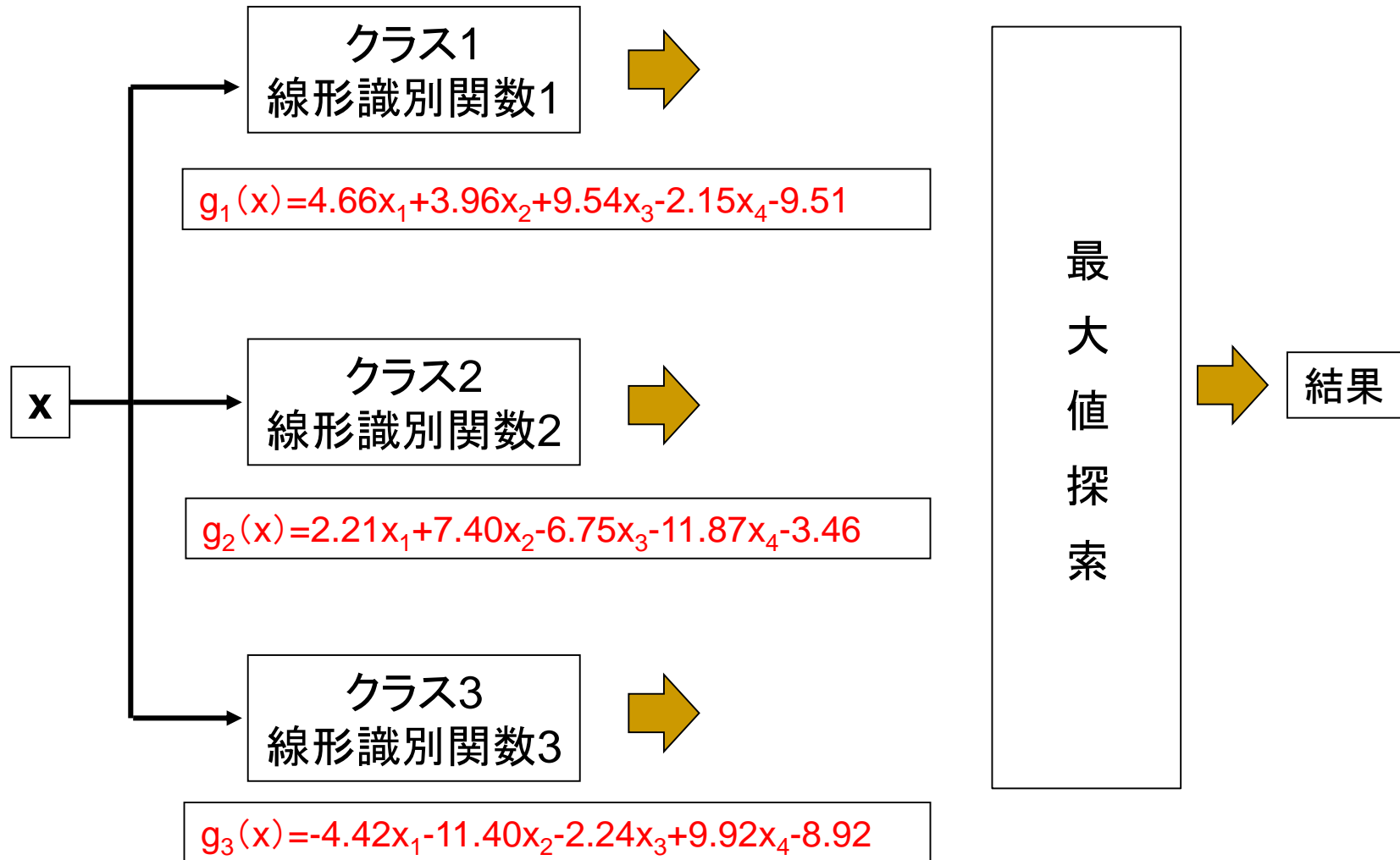
```
print( "¥n [ 正解率 ]" )  
print( accuracy_score(test_label, predict) )
```

```
print( "¥n [ 混同行列 ]" )  
print( confusion_matrix(test_label, predict) )
```

# 実行結果①



# 学習後の線形識別関数



# 実行結果②

```
コマンド プロンプト
-19.713      -12.394      23.417 (2) : 2
-7.541       -24.924      12.934 (2) : 2
-14.768      -18.691      20.002 (2) : 2
-19.086       22.021       2.883 (1) : 2
-3.974       -15.572       2.438 (2) : 2
-4.680       -5.955      -1.903 (2) : 2

[ 予測結果 ]
      precision    recall  f1-score   support

     0       0.68      0.67      0.67       100
     1       0.77      0.82      0.79       100
     2       0.68      0.64      0.66       100

 accuracy          0.71          300
 macro avg         0.71      0.71      0.71          300
weighted avg         0.71      0.71      0.71          300

[ 正解率 ]
0.71

[ 混同行列 ]
[[67 10 23]
 [11 82  7]
 [21 15 64]]
```