知能システム レポート 1

三浦夢生

2020 年 11 月 1 日

1　目的

現在，機械学習において用いられるデータ整形のテクニックやアルゴリズム，データに適したアルゴリズムやパラメータの見つけ方を学ぶ.

2　用語

今回用いたアルゴリズムやテクニックいくつかの解説を簡単に行う.

2.1　Support Vector Classification

Support Vector Classification(以下 SVC という) とは，対象のデータセットのうち，予測に必要となる一部のデータ (これをサポートベクトルという) を決めて分類するアルゴリズムのことである．分類のための境界とサポートベクトルとの距離が最大となるような境界線を見つけることが目的となる.

2.2　クロスバリデーション

クロスバリデーションとは，学習したモデルに対し，いくつかのデータパターンを用意してアルゴリズムの妥当性やモデルの評価を行う手法である．今回は K 分割クロスバリデーションを用いた．これは与えられたデータセットを K 個に分割し，K-1 個をトレーニングデータ，1 個をテストデータにして評価することを K 回行って妥当性や信憑性を評価する手法である.

2.3　グリッドサーチ

各種アルゴリズムを用いる際にオプションとしてパラメータが設定できる．たとえば SVC の場合であれば正規化パラメータやカーネルタイプ等である．これら選択したパラメータの全パターンについて評価し最適なパラメータを見つけることをグリッドサーチという.

3　方法

Google Colaboratory 上で白ワインの品質データを用いて機械学習を行う．データは `https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/` 以下の winequality-white.csv を用いた.

また，学習に用いる手法を以下の様にいくつか組み合わせ，各手法の効果を比較する.

1. SVC による学習
2. SVC+ ラベルの付け直し
3. SVC+ ラベルの付け直し + クロスバリデーション
4. SVC+ ラベルの付け直し + クロスバリデーション + グリッドサーチ

4　実行結果

今回作成したソースコードをそれぞれ付録に示す．また，それらの実行結果を以下に示す．

4.1　SVC のみ

SVC のみの学習結果を以下に示す．

```
              precision recall f1-score support

           3 0.00 0.00 0.00 2
           4 0.00 0.00 0.00 33
           5 0.25 0.00 0.01 295
           6 0.45 1.00 0.62 439
           7 0.00 0.00 0.00 170
           8 0.00 0.00 0.00 39
           9 0.00 0.00 0.00 2

    accuracy 0.45 980
   macro avg 0.10 0.14 0.09 980
weighted avg 0.28 0.45 0.28 980

0.44693877551020406
```

4.2　SVC+ ラベルの付け直し

SVC とデータセットにおけるラベルの付け直しを行った学習結果を以下に示す．

```
              precision recall f1-score support

           0 0.00 0.00 0.00 34
           1 0.93 1.00 0.97 916
           2 0.00 0.00 0.00 30

    accuracy 0.93 980
   macro avg 0.31 0.33 0.32 980
weighted avg 0.87 0.93 0.90 980

0.9346938775510204
```

4.3　SVC+ ラベルの付け直し + クロスバリデーション

上記の方法にクロスバリデーションを付け加えた場合の学習結果を以下に示す．

```
[0.91428571 0.93061224 0.93979592 0.92849847 0.91624106]
```

4.4 SVC+ ラベルの付け直し + クロスバリデーション + グリッドサーチ

上記の方法にグリッドサーチを付け加えた場合の学習結果を以下に示す．ただし，グリッドサーチは正則化パラメータが 1,10,100 の場合かつカーネルが linear,rbf(gamma=0.001,0.0001),sigmoid(gamma=0.001,0.0001) の場合で行った．

```
0.9214285714285714
```

5 考察

SVC 単体の学習よりも，整形後のデータを用いた学習のほうが倍以上の精度をもっている．データ整形が機械学習に対して一定の効果を及ぼすことがわかる．

また，クロスバリデーションを施した際の平均的な正解率が 9 割を超えているため，SVC+ データ整形によって学習したモデルはそれなりに信頼ができるモデルであると言える．

グリッドサーチを行った際の正解率は他の手法で行った際と大きな差はない．むしろ多少ではあるが精度は落ちている．与えたパラメータの範囲にあるものよりもデフォルトのパラメータのほうが適していたとも言える．グリッドサーチを行うと他の場合よりも時間がかかったため，何でもかんでも付け加えれば良いということでもないようだ．

6 付録

今回作成したソースコードを以下に示す．

6.1 SVC のみ

```
import pandas as pd
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
import warnings

df = pd.read_csv("winequality-white.csv", sep=";", encoding="utf-8")

y = df["quality"]
x = df.drop("quality", axis=1)

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)

model = SVC()
model.fit(x_train, y_train)
warnings.filterwarnings("ignore")
```

```
18
19 y_pred = model.predict(x_test)
20
21 print(classification_report(y_test, y_pred))
22 print(accuracy_score(y_test, y_pred))
```

6.2 SVC+ ラベルの付け直し

```
1  import pandas as pd
2  from sklearn.svm import SVC
3  from sklearn.model_selection import train_test_split
4  from sklearn.metrics import accuracy_score
5  from sklearn.metrics import classification_report
6  import warnings
7
8  df = pd.read_csv("winequality-white.csv", sep=";", encoding="utf-8")
9
10 y = df["quality"]
11 x = df.drop("quality", axis=1)
12
13 new_list = []
14 for v in list(y):
15   if v <= 4:
16     new_list += [0]
17   elif v <= 7:
18     new_list += [1]
19   else:
20     new_list += [2]
21 y = new_list
22
23 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
24
25 model = SVC()
26 model.fit(x_train, y_train)
27 warnings.filterwarnings("ignore")
28
29 y_pred = model.predict(x_test)
30
31 print(classification_report(y_test, y_pred))
32 print(accuracy_score(y_test, y_pred))
```

6.3 SVC+ ラベルの付け直し + クロスバリデーション

```
1  import pandas as pd
```

```
from sklearn.svm import SVC
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
import warnings

df = pd.read_csv("winequality-white.csv", sep=";", encoding="utf-8")

y = df["quality"]
x = df.drop("quality", axis=1)

new_list = []
for v in list(y):
  if v <= 4:
    new_list += [0]
  elif v <= 7:
    new_list += [1]
  else:
    new_list += [2]
y = new_list

warnings.filterwarnings("ignore")
kfold_cv = KFold(n_splits=5, shuffle=True)

model = SVC()
scores = cross_val_score(model, x, y, cv=kfold_cv)

print(scores)
```

6.4 SVC+ ラベルの付け直し + クロスバリデーション + グリッドサーチ

```
import pandas as pd
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import GridSearchCV
import warnings

df = pd.read_csv("winequality-white.csv", sep=";", encoding="utf-8")

y = df["quality"]
x = df.drop("quality", axis=1)

new_list = []
for v in list(y):
```

```
  if v <= 4:
    new_list += [0]
  elif v <= 7:
    new_list += [1]
  else:
    new_list += [2]
y = new_list

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
parameters = [
            {"C": [1, 10, 100], "kernel": ["linear"]},
            {"C": [1, 10, 100], "kernel": ["rbf"], "gamma": [0.001, 0.0001]},
            {"C": [1, 10, 100], "kernel": ["sigmoid"], "gamma": [0.001, 0.0001]}
]

warnings.filterwarnings("ignore")
kfold_cv = KFold(n_splits=5, shuffle=True)
model = GridSearchCV(SVC(), parameters, cv=kfold_cv)
model.fit(x_train, y_train)

y_pred = model.predict(x_test)

print(accuracy_score(y_test, y_pred))
```