

Exercício 02 - MC886

Victor Seixas Souza - ra118896

5 de outubro de 2016

- 1) Qual a accuracy média (na validação de fora).
- 2) Quais os valores de C e γ a serem usados no classificador final (fazer o 3-fold no conjunto todo).

1. Qual a acurácia média?

A acurácia média encontrada foi 0.920 ± 0.027 .

2. Quais os valores de C e γ a serem usados no classificador final?

O valor de C usado foi 2^5 e o de γ foi 2^{-5} .

	$C = 2^{-5}$	$C = 2^{-2}$	$C = 2^0$	$C = 2^2$	$C = 2^5$
$\gamma = 2^{-15}$	0.565 ± 0.001	0.565 ± 0.001	0.565 ± 0.001	0.565 ± 0.001	0.789 ± 0.029
$\gamma = 2^{-10}$	0.565 ± 0.001	0.621 ± 0.049	0.813 ± 0.033	0.850 ± 0.025	0.892 ± 0.008
$\gamma = 2^{-5}$	0.565 ± 0.001	0.636 ± 0.008	0.903 ± 0.007	0.913 ± 0.007	0.913 ± 0.007
$\gamma = 2^0$	0.565 ± 0.001	0.565 ± 0.001	0.569 ± 0.004	0.569 ± 0.004	0.569 ± 0.004
$\gamma = 2^5$	0.565 ± 0.001	0.565 ± 0.001	0.565 ± 0.001	0.565 ± 0.001	0.565 ± 0.001

Tabela 1: Acurácia variando parametros C e γ

```
victor @ randomgraph> python proj2.py
Best C = 32
Best gamma = 0.03125
Accuracy: 0.920 (+/- 0.027)
```

Listing 1: Sidas geradas pelo programa

```

1 from numpy import genfromtxt
2 from sklearn.cross_validation import StratifiedKFold, cross_val_score
3 from sklearn.svm import SVC
4
5 X = genfromtxt('data1.csv', delimiter=',', skip_header=1, usecols=range(166))
6 Y = genfromtxt('data1.csv', delimiter=',', skip_header=1, usecols=(166))
7
8 # Generate indexes for Cross Validation
9 skf3 = StratifiedKFold(Y, 3)
10 skf5 = StratifiedKFold(Y, 5)
11
12 # Searching parameters
13 C_range = [2**-5, 2**-2, 2**0, 2**2, 2**5]
14 gamma_range = [2**-15, 2**-10, 2**-5, 2**0, 2**5]
15 C_best = C_range[0]
16 gamma_best = gamma_range[0]
17 current_score = 0
18 # Grid Search
19 for C in C_range:
20     for gamma in gamma_range:
21         clf = SVC(kernel='rbf', C=C, gamma=gamma)
22         scores = cross_val_score(clf, X, Y, cv=skf3)
23         scores_mean = scores.mean()
24         #print("C", C, "gamma", gamma, "acc", scores.mean(), "+-", scores.std())
25         if scores_mean >= current_score:
26             current_score = scores_mean
27             C_best = C
28             gamma_best = gamma
29 print("Best C =", C_best)
30 print("Best gamma =", gamma_best)
31
32 # SVC with best parameters
33 clf = SVC(kernel='rbf', C=C_best, gamma=gamma_best)
34 scores = cross_val_score(clf, X, Y, cv=skf5)
35 print("Accuracy: %0.3f (+/- %0.3f)" % (scores.mean(), scores.std() * 2))

```