

课后作业：支持向量机(Support Vector Machine)

作者：欧新宇 (Xinyu OU)

本文档所展示的测试结果，均运行于：Intel Core i7-7700K CPU 4.2GHz

【作业提交】

将分类结果保存到文本文档进行提交(写上每一题的题号和题目，然后再贴答案)，同时提交源代码。

1. 测试结果命名为: ex07-结果-你的学号-你的姓名.txt
2. 输出图片命名为: ex07-性能对比图-你的学号-你的姓名.png (.jpg)
3. 源代码命名为: ex07-01-你的学号-你的姓名.py, ex07-02-你的学号-你的姓名.py

结果文件，要求每小题标注题号，两题之间要求空一行

要求在“鸢尾花”数据集上完成以下任务，要求如下：

1. 要求训练集和测试集的分割比例为60% : 40%
2. 分别使用LinearSVC, linear核、RBF核、sigmoid核、多项式核的支持向量机完成准确率的测试，要求同时输出训练集和测试集的准确率评分结果。（ex07-01）
3. 使用RBF核的SVM模型在不同Gamma值下进行建模，并输出性能对比图，同时输出训练集和测试集。（ex07-02, ex07-性能对比图）
 - o Gamma取值范围: `array([0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0])`
 - o 尽量使用 `np.linspace(start, end, num)` 函数实现数列的生成

数据集载入方法

```
1 from sklearn import datasets
2 iris = datasets.load_iris()
3
```

• 输出模型准确率

```
1 # TODO: 1. 导入必须库 以及 定义必要的函数
2 # 导入机器学习数据集处理工具
3 from sklearn import datasets
4 from sklearn.model_selection import train_test_split
5 # 导入支持向量机SVM
6 from sklearn import svm
7
8 # TODO: 2. 创建/导入数据
9 iris = datasets.load_iris()
10
11 # TODO: 3. 数据预处理，包括训练集、测试集划分，数据正则化，数据清洗等
12 X = iris.data
13 y = iris.target
14 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4)
15
```

```

16
17 # TODO: 4. 构建模型, 并进行模型训练 (或称为拟合数据)
18 C = 1.0
19 models = (svm.LinearSVC(C=C, max_iter=5000),
20            svm.SVC(kernel='linear', C=C, gamma='auto'),
21            svm.SVC(kernel='rbf', gamma=0.7, C=C),
22            svm.SVC(kernel='sigmoid', C=C, gamma='auto'),
23            svm.SVC(kernel='poly', degree=3, C=C, gamma='auto'))
24 # TODO: 考虑输出变量models
25 models = (clf.fit(X_train, y_train) for clf in models)
26
27 titles = ('LinearSVC (linear kernel)',
28           'SVC with linear kernel',
29           'SVC with RBF kernel',
30           'SVC with sigmoid kernel',
31           'SVC with ploy(degree=3) kernel')
32
33
34 # TODO: 5. 输出模型准确率
35 for model, title in zip(models, titles):
36     print("{0}, 训练集准确率: {1:.3f}, 测试集准确率: {2:.3f}".format(
37         title, model.score(X_train, y_train), model.score(X_test, y_test)))
38

```

```

1 -----
2
3 ValueError                                Traceback (most recent call last)
4
5 <ipython-input-1-a3dfe96c4685> in <module>
6     7
7     8 # TODO: 2. 创建/导入数据
8 ----> 9 iris = datasets.load_iris()
9     10
10    11 # TODO: 3. 数据预处理, 包括训练集、测试集划分, 数据正则化, 数据清洗等

```

```

1 C:\ProgramData\Anaconda3\lib\site-packages\sklearn\datasets\_base.py in
load_iris(return_X_y)
2     382     """
3     383     module_path = dirname(__file__)
4 --> 384     data, target, target_names = load_data(module_path, 'iris.csv')
5     385     iris_csv_filename = join(module_path, 'data', 'iris.csv')
6     386

```

```

1 C:\ProgramData\Anaconda3\lib\site-packages\sklearn\datasets\_base.py in
load_data(module_path, data_file_name)
2     234     data_file = csv.reader(csv_file)
3     235     temp = next(data_file)
4 --> 236     n_samples = int(temp[0])
5     237     n_features = int(temp[1])
6     238     target_names = np.array(temp[2:])

```

```

1 ValueError: invalid literal for int() with base 10: '花萼长度'

```

- 输出不同Gamma下RBF的准确率

```

1  '''
2  @Author: Xin-Yu Ou (欧新宇)
3  @Description: 鸢尾花数据集上RBF-SVM的超参数Gamma对比
4  @LastEditorTime: 2020-01-26
5  '''
6  # TODO: 1. 导入必须库 以及 定义必要的函数
7  import numpy as np
8  import matplotlib.pyplot as plt
9  # 导入机器学习数据集处理工具
10 from sklearn import datasets
11 from sklearn.model_selection import train_test_split
12 # 导入支持向量机SVM
13 from sklearn import svm
14
15 # TODO: 2. 创建/导入数据
16 iris = datasets.load_iris()
17
18 # TODO: 3. 数据预处理, 包括训练集、测试集划分, 数据正则化, 数据清洗等
19 X = iris.data
20 y = iris.target
21 X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.6)
22
23
24 # TODO: 4. 构建模型, 并进行模型训练 (或称为拟合数据)
25 C = 1.0
26 # gamma = np.concatenate((np.linspace(0.1, 1, 10), np.linspace(2, 10, 9)),
27 # axis=0)
28 gamma = np.linspace(0.1, 1, 10)
29 num = gamma.shape[0]
30 scores = np.zeros([2, num]) # 第1-4列分别为: score_train_rbf, score_test_rbf
31
32 for i in range(num):
33     n = i + 1
34     model = svm.SVC(kernel='rbf', gamma=gamma[i], C=C)
35     model.fit(X_train, y_train)
36
37     scores[0, i] = model.score(X_train, y_train)
38     scores[1, i] = model.score(X_test, y_test)
39
40 plt.figure(dpi=100)
41 plt.plot(gamma, scores[0, :], label="Train")
42 plt.plot(gamma, scores[1, :], label="Test")
43 plt.xticks(gamma)
44
45 plt.legend(loc='best')
46 plt.savefig('results/Ch07Hw01SVM.png', dpi=150)
47 plt.show()

```

