

MachineLearningHomework1_KMeansPictureSegment

201250068 陈骏

环境配置

```
Python 3.7
Package:
  pandas
  skimage
  sklearn
```

项目代码

代码结构

```
D: .
|  PicRead.py
|  README.md
|  SVM.py
|
|_ .idea
|   | .gitignore
|   | misc.xml
|   | modules.xml
|   | SVM.iml
|   | workspace.xml
|   |
|   |_ inspectionProfiles
|       profiles_settings.xml
|       Project_Default.xml
|
|_ data
|   t10k-images-idx3-ubyte
|   t10k-labels-idx1-ubyte
|   train-images-idx3-ubyte
|   train-labels-idx1-ubyte
|
|_ __pycache__
    PicRead.cpython-37.pyc
```

数据集

采用Fashion MNIST Dataset作为训练数据集，该数据集是一个使用很广泛的用于图片分类和图像识别的数据集。与MNIST手写数据集相同，该数据集的图片为28*28的灰度图片，也拥有十种分类结果。

SVM实现

SVM是一个分离超平面，在二维向量的分类中，这种平面表现为多条直线将二维平面分为多个区域，在高维向量中，表现为超平面。在线性支持向量机中，一个样例会计算其到分界超平面的垂直距离，而SVM则是将所有训练样本的其离最近的超平面的距离最小化。在sklearn的LinearSVC中，由函数_fit_liblinear实现迭代过程。

代码内容

```
from PicRead import read_idx3_ubyte_pixel_file
from PicRead import read_idx3_ubyte_label_file
import pandas as pd
from skimage.feature import hog
from sklearn.metrics import confusion_matrix, precision_score, recall_score
from sklearn.svm import SVC, LinearSVC, NuSVC

train_data, (height, width) = read_idx3_ubyte_pixel_file("data/train-images-idx3-ubyte")
train_labels = read_idx3_ubyte_label_file("data/train-labels-idx1-ubyte")
test_data = read_idx3_ubyte_pixel_file("data/t10k-images-idx3-ubyte")[0]
test_labels = read_idx3_ubyte_label_file("data/t10k-labels-idx1-ubyte")

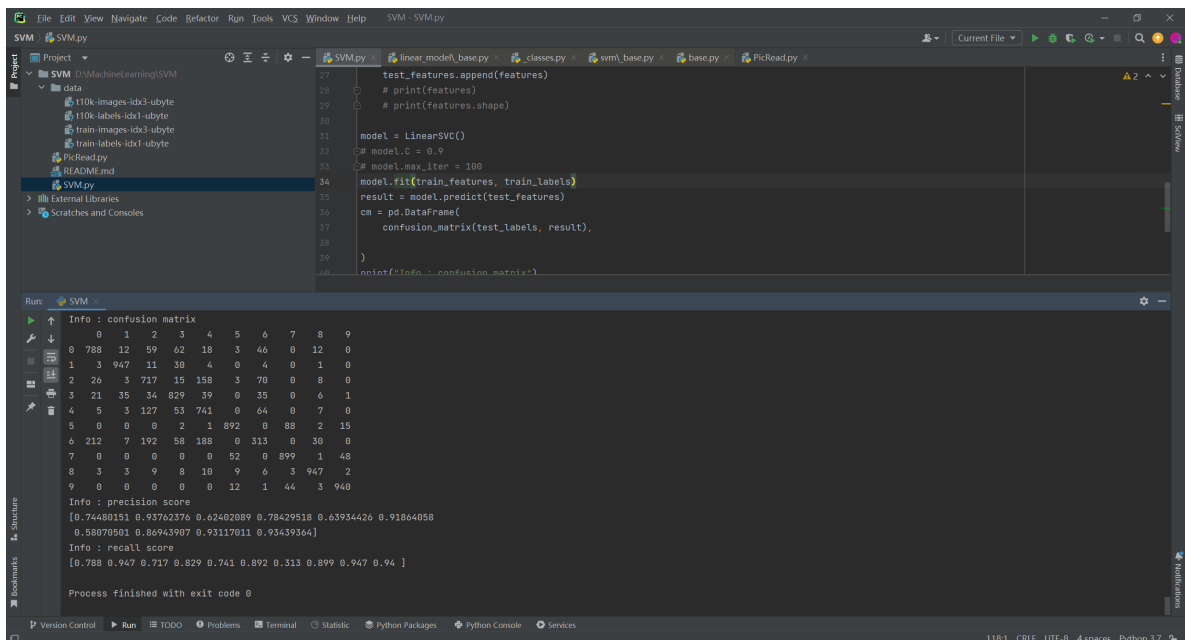
# print(train_data)
# print(train_labels)

train_features = []
test_features = []
for i in train_data:
    patches = hog(i, orientations=10)
    patches = patches.flatten()
    train_features.append(patches)
    # print(patches)
    # print(patches.shape)
for i in test_data:
    features = hog(i, orientations=10)
    features = features.flatten()
    test_features.append(features)
    # print(features)
    # print(features.shape)

model = LinearSVC()
# model.C = 0.9
# model.max_iter = 100
model.fit(train_features, train_labels)
result = model.predict(test_features)
cm = pd.DataFrame(
    confusion_matrix(test_labels, result),

)
print("Info : confusion matrix")
print(cm)
print("Info : precision score")
print(precision_score(test_labels, result, average=None))
print("Info : recall score")
print(recall_score(test_labels, result, average=None))
```

训练结果



The screenshot shows an IDE with a project named 'SVM'. The file explorer on the left shows a directory structure with files like 'data', '110k-images-idx3-ubyte', '110k-labels-idx1-ubyte', 'train-images-idx3-ubyte', 'train-labels-idx1-ubyte', 'PicRead.py', 'README.md', and 'SVM.py'. The main editor shows the code in 'SVM.py':27 test_features.append(features)
28 # print(features)
29 # print(features.shape)
30
31 model = LinearSVC()
32 # model.C = 0.9
33 # model.max_iter = 100
34 model.fit(train_features, train_labels)
35 result = model.predict(test_features)
36 cm = pd.DataFrame(
37 confusion_matrix(test_labels, result),
38)
39
40 print('Info : confusion matrix')The Run console at the bottom shows the output of the code execution:Info : confusion matrix
[[0 1 2 3 4 5 6 7 8 9]
 [0 788 12 59 62 18 3 46 0 12 0]
 [1 3 947 21 30 4 0 4 0 1 0]
 [2 26 3 717 15 158 3 70 0 8 0]
 [3 21 35 34 829 39 0 35 0 6 1]
 [4 5 3 127 53 741 0 64 0 7 0]
 [5 0 0 0 2 1 892 0 88 2 15]
 [6 212 7 192 58 188 0 313 0 30 0]
 [7 0 0 0 0 0 52 0 899 1 48]
 [8 3 3 9 8 10 9 6 3 947 2]
 [9 0 0 0 0 0 12 1 44 3 940]

Info : precision score
[0.74480151 0.93762376 0.62402089 0.78429518 0.63934426 0.91864058
 0.58078501 0.86943907 0.93117011 0.93439364]

Info : recall score
[0.788 0.947 0.717 0.829 0.741 0.892 0.313 0.899 0.947 0.94]

Process finished with exit code 0

优化尝试

特征提取

SVM模型是对向量进行分类，因此需要将图片特征提取为向量，可以采用卷积等方法提取。此处采用skimage.feature.hog进行方向梯度直方图提取作为向量。

算法优化

选取不同的模型与参数

Linear

线性分类，惩罚力度C，为0-1之间，C越大分类效果越好，但会导致过拟合。

Rbf

高斯模型分类，Gamma值越小，分类越连续，越大越分散。

Poly

多项式分类，维度值degree，参数效果类似于前两种。