

Overview

For the task following dataset was chosen *pendigits* which is a Pen-Based Recognition of Handwritten Digits Data Set from LIBSVM.

The set is split into two files, learning set of 7,494 samples, and testing one of 3,498 samples.

There are 10 classes across the dataspace.

▼ Data preparation

Load training and testing set.

```
from sklearn import datasets

pendings = datasets.load_svmlight_file('/content/data_sets/pendigits.txt')
target = pendings[-1]
data = pendings[0]

pendings = datasets.load_svmlight_file('/content/data_sets/pendigits_test.txt')
target_test = pendings[-1]
data_test = pendings[0]
```

▼ Training and testing

Classification accuracy is at the level of 95%.

```
from sklearn import svm
clf = svm.SVC(gamma=0.001, C=100.)
clf.fit(data, target)
target_prediction = clf.predict(data_test)

diff = target_test - target_prediction

import numpy as np
nr_of_incorrect = np.count_nonzero(diff)
total_nr_of_samples = len(target_test)

accuracy = (total_nr_of_samples - nr_of_incorrect) / total_nr_of_samples
print('Accuracy: ', round(accuracy, 2), '%')

Accuracy:  0.95 %
```

▼ Training and performance for different kernel

Change of the default kernel rbf to linear.

New obtained accuracy is at the level of 95%.

```
clf.set_params(kernel='linear').fit(data, target)
target_prediction = clf.predict(data_test)

diff = target_test - target_prediction

nr_of_incorrect = np.count_nonzero(diff)
total_nr_of_samples = len(target_test)

accuracy = (total_nr_of_samples - nr_of_incorrect) / total_nr_of_samples
print('Accuracy: ', round(accuracy, 2), '%')

Accuracy:  0.95 %
```

▼ Change from multiclass coding to multilabel coding.

Using multiclass classifiers, accuracy was improved to the level of 98% for OneVsRestClassifier.

```
from sklearn.svm import SVC
from sklearn.multiclass import OneVsRestClassifier
from sklearn.preprocessing import LabelBinarizer

iris = datasets.load_iris()
classif = OneVsRestClassifier(estimator=SVC(random_state=0))
clf4 = classif.fit(data, target)

target_prediction = clf4.predict(data_test)

diff = target_test - target_prediction

nr_of_incorrect = np.count_nonzero(diff)
total_nr_of_samples = len(target_test)

accuracy = (total_nr_of_samples - nr_of_incorrect) / total_nr_of_samples
print('Accuracy for OneVsRestClassifier: ', round(accuracy, 2), '%')

Accuracy for OneVsRestClassifier:  0.98 %
```

Multilabel coding

For multilabel coding, for labels assigned for samles, correct ones were included 97% times.

```
y = LabelBinarizer().fit_transform(target)
clf5 = classif.fit(data, y)
target_prediction = clf5.predict(data_test)
```

```
nr_of_correct = 0

for i in range(len(target_test)):
    index = int(target_test[i])
    if(target_prediction[i][index] == 1):
        nr_of_correct += 1

accuracy = nr_of_correct / total_nr_of_samples
print('Accuracy for Multilabel coding: ', round(accuracy, 2), '%')

Accuracy for Multilabel coding:  0.97 %
```

Płatne usługi Colab - Tutaj możesz anulować umowy

✓ 1 s ukończono o 09:31

