Handwritten symbol recognition using Scikit-learn

Peter-Tibor Zavaczki

march 7, 2018

Chapter 1

About Scikit-learn

1.1 Tool Purpose

Scikit-learn is a machine learning library for Python, which features various classification, regression and clustering algorithms, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

1.2 Installing scikit-learn

Prior to using scikit-learn, Python (>= 2.7 or >= 3.3) has to be installed along with the NumPy (>= 1.8.2) and SciPy (>= 0.13.3) libraries.

1.2.1 Installing steps

Some versions of Ubuntu come installed with Python 2.7.12 and Python 3.5.2, so for this installation we will consider that and install scikit-learn for Python 3.5.2. To ease installing packages for Python we will use pip. To install pip, we need the command *sudo apt install python3-pip*. Please note that we have a 3 after python to signal that we will install pip for Python 3.x. After the previous step we install NumPy and SciPy by using the command *sudo pip3 install numpy scipy*. This will download the libraries' latest version and automatically install them. As a final step, we use the command *sudo pip3 install scikit-learn* to install scikit-learn.

1.3 Studied example

The studied example is **Recognizing hand-written digits** by **Gael Varoquaux**, a handwritten digit classificator by machine learning. It can recognize the 0-9 handwritten digits and convert them to digital characters.

1.3.1 How to run the example(s)

To run the given example, you need to have matplotlib, installed with *sudo pip3 install matplotlib* and python3-tk, installed with *sudo apt-get install python3-tk*.

Then just use the command *python3 ./plot_digits_classification.py* from the folder of origin to run the example.

1.3.2 Algorithm

The given example relies on a few libraries which it imports and works with. These are matplotlib.pyplot, and from sklearn, the datasets, svm, metrics libraries. After the libraries have been loaded, the application loads the processed dataset using the datasets.load_digits() command.

The images and the targets of the digits dataset is zipped into touples and added to a lists, so that it can be worked with in the following nested for-each loop (first level iterating by index, second level iterating by (image, prediction) touple). Please note that the loop only takes the first 4 (image, target) touples! In the mentioned for loop, at each iteration a new subplot is activated, the axis' are turned off (we are displaying an image and labelling it to see what it is, we are not actually displaying an actual plot), an image is shown on the axis with the gray_r colormap set and the interpolation set to nearest (this is the best choice when a small image is enlarged), then a title is set for each sepparate subplot, which signals the character trained using that data sample.

In the following step the number of image samples is saved in the n_samples variable. In the data variable a reshaped version of the digits' images' array is stored in the (samples, feature) matrix format. A Support Vector Classifier is instantiated with a gamma of value 0.001 and then the first half of the dataset is used for training.

After training with the first half of the dataset, the second half's targets are stored in the 'expected' variable and the predictions from the second half of the dataset are stored in the 'predicted' variable.

After predicting the second half of the dataset, we print the SVC's prameters, which in this case is all default, except for gamma and the classification report, which consists of listing the possible cases and the precision, recall and f1-scores calculated for them, along the number of samples of that case in the predicted dataset. The second part of printing the prediction data is the confusion matrix, which represents the expected values on the rows and the predicted values on the columns.

In the next step, the second half of the dataset and the predictions are zipped into a list of touples, so that the first four predicted images can be displayed similarly to the first four training images before.

As a last step, the plot is shown so that we can see the results.