University of St. Gallen

Machine Learning with Python

"Handwritten Digits Recognition
using scikit-learn"

Anja Spiess
Danni Zhai
Olga Lukashenko
Shengchang Qiu

Group Project:
Introduction to Programming/
Advanced Programming XCamp 2019
Prof. Mario Silic

24th May 2019

Contents

---

1. Introduction: task description

"Machine Learning is the science of getting computers to learn and act like humans do, and improve their learning over time in autonomous fashion, by feeding them data and information in the form of observations and real-world interactions."

The goal of this project is to use machine learning to recognize handwritten digits. For that purpose, we used the MNIST dataset of handwritten digits from LeCun, Cortes and Burges (http://yann.lecun.com/exdb/mnist/). This project is based on the following tutorial: https://www.datacamp.com/community/tutorials/machine-learning-python

In our project we proceeded with the **common structure of machine learning codes**:

1.Setting up, loading data sets (split in a later stage into training and test data sets)
2.Summarizing and inspecting the dataset
3.Pre-processing and visualizing the dataset (including normalization of data and the use of dimension reduction techniques)
4.Implementation of the adequate algorithm to construct the model
5.Making and visualizing predictions (Isomap and PCA)
6.Measuring the goodness of the applied classification method (confusion matrix and various quality metrics)

As explained in the tutorial, we followed to different methods (K-means Clustering and Support Vector Machines) to compare their performance in predicting the digits and get a sufficient accuracy of predictions (in this case, the SVM method had a much higher prediction accuracy as the K-means method).

2. Pre-requisites

To directly access the dataset of MNIST from Tensorflow, we need to firstly install the Tensorflow environment for Jupyter. To be specific, the following steps need to be firstly completed:

1. Install Anaconda
2. Install Tensorflow in Anaconda
3. Install package “matplotlib” and “sklearn” under Tensorflow environment
4. Run Jupyter Notebook under Tensorflow environment

Tensorflow offers direct access to MNIST dataset from its API. Additionally, we need to install the necessary “matplotlib” and “sklearn” packages into the newly setup Tensorflow environment. The “matplotlib” enables us to generate plots for our assessment of the dataset. The “sklearn” is the essential package in our machine learning project, all the algorithms applied in the project comes from this package.

3. Programming procedure

First Method: K-Means Clustering model

Step 1: Loading the input data from the MNIST database
Goal: Access the dataset of MNIST containing a training set of 60'000 examples, and a test set of 10,000 examples

Step 2: Exploring the characteristics of the input data
Goal: Gathering some insights into the data by performing an exploratory data analysis (EDA)

Step 3: Visualizing target images with the matplotlib library
Goal: Show the handwritten digits together with their correspnding labels

Step 4: Principal Component Analysis (PCA) = Dimensionality reduction technique
Goal: Reducing the dimensionality to grasp and visualize the data more easily

Step 5: Normalization
Goal: Pre-processing the data before being able to model it

Step 6: Inspect training data
Goal: Get a quick overview of the number of samples and features

Step 7: Clustering model with the k-means algorithm
Goal: Find the clusters (ideally 10) in the training set

Step 8: Visualizing cluster centres
Goal: Verify visually if the clustering process has been successful

Step 9: Predicting the labels
Goal: Verify if the model predict the right values

Step 10: Visualizing predictions with Isomap
Goal: Use the Isomap to visually verify the performance of the model

Step 11: Visualizing predictions with PCA
Goal: Compare the results with those obtained using the Isomap

Step 12: Confusion matrix
Goal: Analyze the degree of correctness of the model's predictions

Step 13: Measuring the goodness of fit of the cluster labels to the correct labels
Goal: Learn more about the quality of the clusters by applying different cluster quality metrics

<u>Second Method: Support Vector Machines classifier</u>

Step 1: Loading the input data from the MNIST database
Goal: Access the dataset of MNIST containing a training set of 60'000 examples, and a test set of 20'000 examples.

Step 2: Exploring the characteristics of the input data
Goal: Gathering some insights into the data by performing an exploratory data analysis (EDA)

Step 3: Visualizing target images with the matplotlib library
Goal: Show the handwritten digits together with their correspnding labels

Step 4: Principal Component Analysis (PCA) = Dimensionality reduction technique
Goal: Reducing the dimensionality to grasp and visualize the data more easily

Step 5: Optimization of classifier parameters using the GridSearchCV method
Goal: Decide on kernal and the value of C and gamma

Step 6: Classification with the Support Vector Machine
Goal: Perform classification by finding the hyperplane that differentiate the classes very well.

Step 7: Predicting the labels
Goal: Verify if the model predict the right values

Step 8: Visualizing classification
Goal: Verify visually if the classification has been successful

Step 9: Confusion matrix
Goal: Analyze the degree of correctness of the model's predictions

Step 10: Visualizing predictions with Isomap
Goal: Use the Isomap to visually verify the performance of the model

4. Code
See separate .ipynb files (K-means code and SVM code)