Practical work 06 – 30th of October 2018 Classification with Support Vector Machines (SVM)

Summary for the organisation :

- Submit the solutions of the practical work before Monday 12h00 next week in Moodle.
- Preferred modality : **one**[1] pdf generated from your iPython notebook(s) and an archive with your notebook(s).
- Alternative modality : one pdf report with annotated code and outputs.
- The file name must contain the number of the practical work, followed by the names of the team members by alphabetical order, for example `02_dupont_muller_smith.zip`.
- Put also the name of the team members in the body of the notebook (or report).
- Only one submission per team.

Exercice 1 Digit classification system using different SVM classifiers

The objective of this exercise is to build a classification system able to classify the images of handwritten digits (0–9) coming from the MNIST database and using SVM with different types of kernels (linear, polynomial, RBF,...). For that purpose, you will use the SVM library available in *Scikit-learn* (`http://scikit-learn.org`).

a. Getting the training and test sample sets from the MNIST database

a) Load MNIST digit dataset - see previous PW.

b) Visualize (plot) the images of some digits of the MNIST database. You should get something similar to Figure 1.

c) Build the final training and test sets, which should be balanced, i.e. have the same number of samples per class (i.e. digit), for example 200 per class for training, and 100 for test.

1. And only one so use a tool to merge your pdfs if more than one.

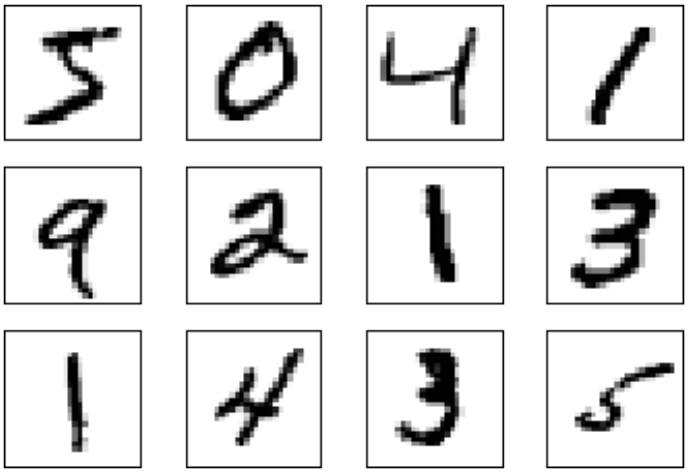

FIGURE 1 – Some digits of the MNIST database

b. Classification of digits based on raw pixel values using SVM and different kernels

Create, train and test several SVM classifiers with different kernels (linear, polynomial, RBF,...). For the training, perform a cross-validation using 10 folds, and test different with several C and kernel parameter values (e.g. for γ for RBF kernel) in order to get the best classifier. After the test, display the classification performances and confusion matrix of each SVM classifier (see class *metrics*) The following references will help you :

a) http://scikit-learn.org/stable/modules/svm.html
b) http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
c) http://scikit-learn.org/stable/modules/grid_search.html
d) http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
e) http://scikit-learn.org/stable/modules/model_evaluation.html#confusion-matrix and ...#classification-report
f) http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

c. (Optional) Impact of preprocessing and feature extraction

Analyse the impact of the classification performances using the following preprocessing and feature extraction steps :

a) Preprocessing step : convert images to binary (i.e. black and white) representations ;
b) Feature extraction steps :
   — Horizontal and vertical projections (i.e. compute the sum of grey pixel values along the the X and Y-axis, see Figure 2) ;

— Local binary patterns (refer to : https://fr.wikipedia.org/wiki/Motif_binaire_local) ;
— Any other usable image features that you may have found.

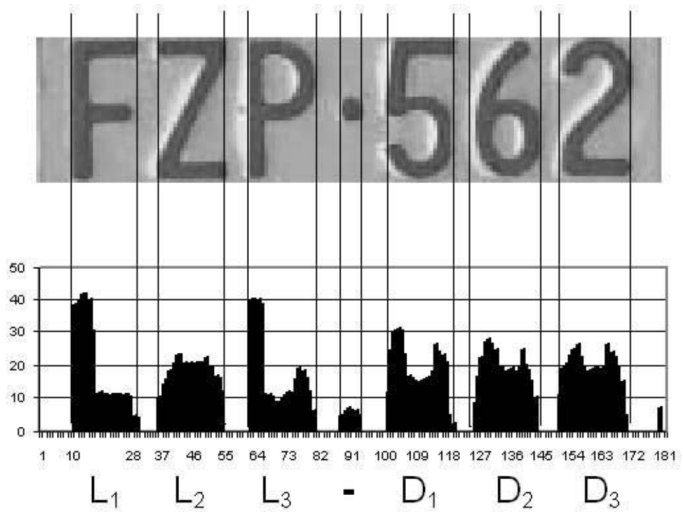

FIGURE 2 – Vertical projection of an image

d. Analysis of the results

Analyse the results obtained with the best SVM classifier.

a) Which kernel and parameters were used ?
b) Which digit classes are the best/worse recognized against which ? Why ?
c) What is the impact of the sizes of the training and test sets on the classification performance ?