Practical work 07 – 28th of October 2025 Classification with Support Vector Machines (SVM)

Summary for the organisation :

- The practical works (PW) are optional. If you submit a PW, we provide **feedbacks** and you get potential **bonuses** on the final grade as explained in the first week. Submissions should be done before the date specified in Moodle.
- **Rule 1**. Submit 1 archive (***.zip**) with your Python notebooks. Do not include datasets unless specific instructions, but do include all necessary files to reproduce your experiments.
- **Rule 2**. The archive file name must contain the number of the practical work, followed by the family 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 (in first cell). Only one submission per team.
- **Rule 3**. We don't give bonuses for submissions that do not compile (missing files are a common source of errors...). So, make sure that your whole notebooks give the expected solutions by clearing all cells and running them all before submitting.

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`).

As always, if something does not work correctly, describe what you tried.

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.

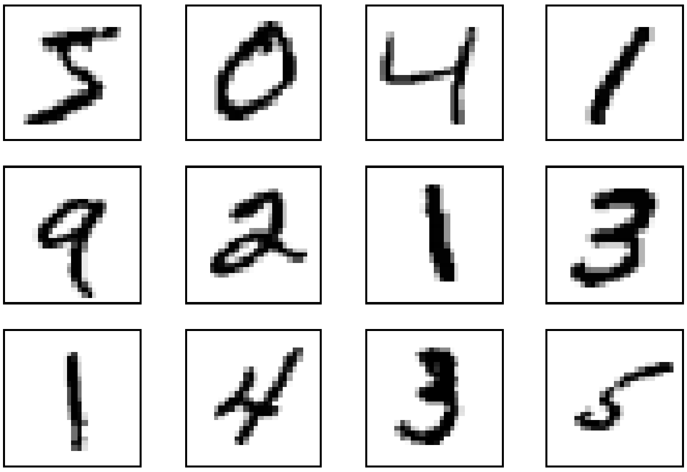

FIGURE 1 – Some digits of the MNIST database

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.

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 :

- `http://scikit-learn.org/stable/modules/svm.html`
- `http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`
- `http://scikit-learn.org/stable/modules/grid_search.html`
- `http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html`
- `http://scikit-learn.org/stable/modules/model_evaluation.html#confusion-matrix` and `...#classification-report`
- `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.

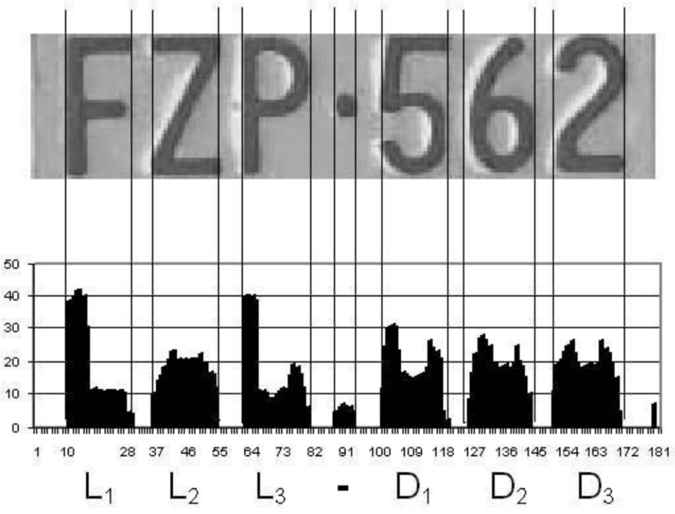

FIGURE 2 – Vertical projection of an image

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 ?

Exercice 2 Review questions

a) What are the two fundamental ideas a SVM are built on ? Summarize them with your own words.

b) With the hinge loss, training points can fall into three cases. Re-explain these cases with your own words.

c) What are the two implementations of SVMs available in SciKit Learn ? Which one would you take if you have a system that needs to incorporate incremental learning ?

d) A SVM can classify between 2 classes. Cite and explain in your own words the 2 strategies we have to build a multi-class (with K classes) system with SVM ?

e) Are the strategies of point d) equal in terms of cpu ? (elaborate your answer considering training and testing times)

f) Describe a machine learning task for which SVM would be a better choice than any of the algorithms previously studied. Explain why.