Practical exercices
Classification of images and time series

20 octobre 2020

The aim of this exercise is to manipulate classification (linear SVM, kernels) on toy dataset, images and time series, using SKLEARN [1] and TSLEARN [2]. All data and codes are available here : http://sites.

1 Classification on toy datasets

1.1 Useful function

Before starting, the function plot_boundary(clf,X,Y) enables to plot the result of a boundary obtained by the discriminative technique clf on sets X and Y. An illustration is visible in figure 1(b).

1.2 Linear SVM on linearly separable data

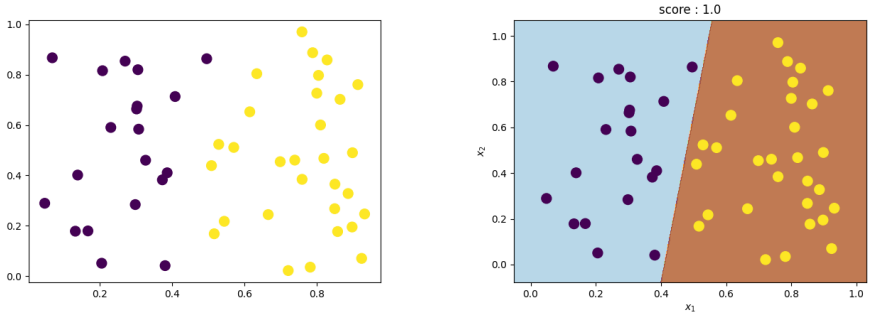
In this section, we generate linearly separable data and find a classifier. The first part generates the dataset (visible in figure 1 (a)) and initializes the classifier with the command clf = LinearSVC(C=1) (regularization constant $C = 1$). **To do : perform the training**. As one can see, training scores can be print out.

In the next, we add some noise in the data and we evaluate the effect of the regularization coefficient C. **To do : perform the training for various regularization coefficients**.

1.3 SVM on non linear data

Here we charge datasets issued from SKLEARN and extract the *moon* dataset visible in figure 2. Then we

1. https://scikit-learn.org/stable/
2. https://tslearn.readthedocs.io/en/stable/



(a) (b)

FIGURE 1 – Illustration of a linear set (a ; X in blue and Y in yellow) and boundary function (b) obtained by a classifier

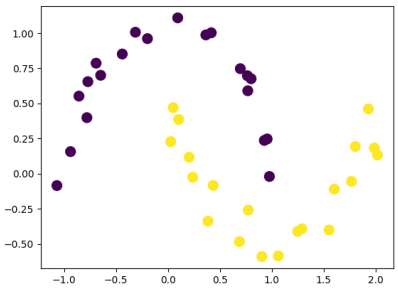


FIGURE 2 – Illustration of the non linear mon dataset

apply SVM with polynomial and gaussian kernel with various values of regularization coefficient C. Performances of test and train are also plotted out.

2 Classification on images

Here the goal is to classify the image in 3(a) from training samples in 3(b).

2.1 First classification

The first lines load the data (they are stored as pickle files, a python format to share data), visualise them, modify them in a correct shape and split the dataset in train and test files (some lines are to fill). Then the classification has to be performed (SVM with gaussian kernel) and they results are printed out. As you way see, they are not good.

2.2 Second classification

The incorrect results come from the unbalanced training dataset. In the following lines we plot the number of training samples in each class and one observes that classes are not balanced. In the next step we re-equilibrate the dataset and perform again the classification. What can we conclude ?

3 Manipulation of time series

Here we use tslearn library. A function plot_series is defined to plot sets of time series and their associated barycenter.

3.1 Barycenter computation

The first part charges some series and compute Euclidian and DTW-based barycenter. Noting to do here, just run the code and analyse results.

3.2 KNN on time series

Here the idea is to perform KNN based on DTW distance. We look for some queries, extract the closest time series and plot them. Noting to do here, just run the code and analyse results.

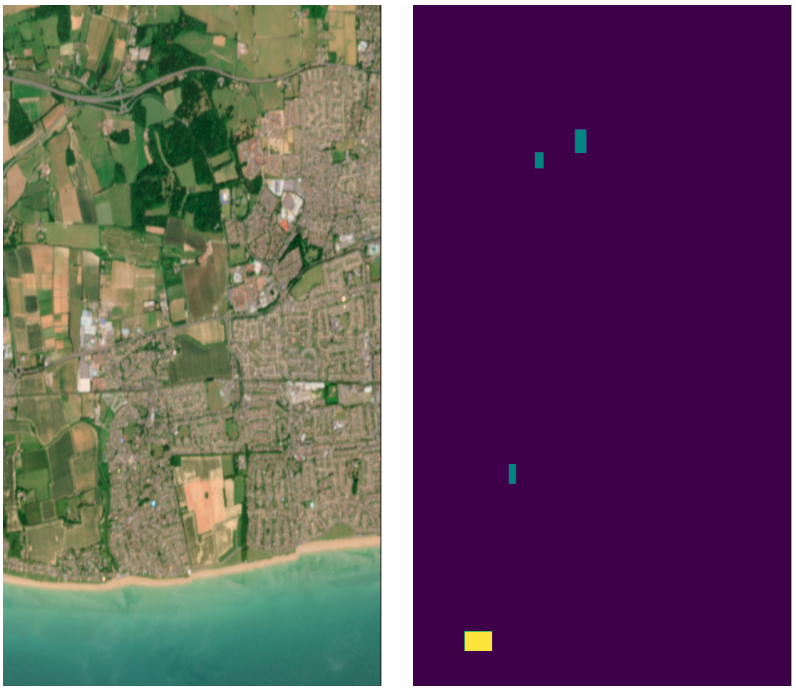


FIGURE 3 – Satellite image (a) and training samples (1 in blue : strong vegetation, 2 in yellow : water, 0 in purple : others)

3.3 SVM on time series

Here we use TimeSeriesSVC extracted from from tslearn.svm import TimeSeriesSVC command. The first lines charge and visualise some time series and we perform a classification using a GAK (Global Alignment Kernel) kernel, based on DTW. Noting to do here, just run the code and analyse results and plot support vectors.