Statistical Signal Processing
A.A. 2017/2018

Computer Lab 1 – k-NN classifier

Exercise 1 – Synthetic dataset

In this exercise, you will employ a synthetic dataset (file `synthetic.mat`), containing labelled training data and test data for two classes. Each example is 2-dimensional.

Task: your task is to implement a k-NN classifier in Matlab, which calculates the probability that a given test example belongs to each class, and outputs a class label as the class with the highest probability. You will evaluate the classifier performance computing the average classification accuracy (i.e. the fraction of test examples that have been classified correctly).

In particular, you should perform the following:

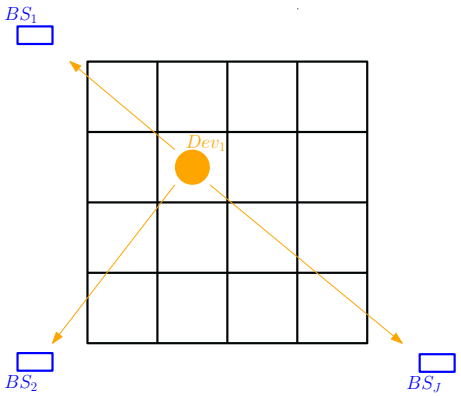
- Train a k-NN classifier for different values of k
- Compare accuracy on the training set and the test set
- Identifying overfitting and underfitting in the obtained results

Other indications:

- Students are not allowed to employ Matlab's `knnsearch( )` function (this also holds for exercise 2). They are supposed to employ basic functions. It is allowed to employ the `mink( )` function.

Exercise 2 - User localization from RSSI

Consider the following scenario, in which we wish to localize a user employing a non-GPS system (e.g., in indoor localization). The user holds a transmission device (e.g., a smartphone or other sensor with transmission capabilities). Localization is based on measurements of the Received Signal Strength Indicator (RSSI) from *D* sensors (base stations) placed in the area in which the localization service is provided (more detailed information can be found in [1]). The area is divided into N_C square cells, and localization amounts to identifying the cell in which the user is located.



In a **training** stage, the transmission device is placed in the center of each cell and broadcasts a data packet, and RSSI is measured by each sensor. This yields one measurement, corresponding to a vector of length *D*. The process is repeated *M* times for each cell, and for all N_C cells. The training stage provides a 3-dimensional array of size $N_C DM$.

In a **test** stage, the user is located in an unknown cell. The transmission device broadcasts a data packet, and each sensor measures the RSSI and communicates it to a fusion center. The fusion center treats the received RSSI values as a test vector of length *D*. It applies a k-NN classifier, comparing the test vector with all $M \cdot N_C$ training vectors available in the training set. For each test vector, the k-NN classifier outputs the probability that each cell contains the user.

Available data: you are provided with a .mat file (`localization.mat`) containing two variables, called `traindata` and `testdata`. These variables have the same size, and are 3-dimensional arrays of size *D*=7, *M*=5, and $N_C = 24$. The 24 cells have the following arrangement:

1	2	3	4
5	6	7	8
...	...	...	...
...	...	...	...
...	...	...	...
21	22	23	24

The training data can be seen as labelled data where each cell is a class, and you are given *M* data vectors for each cell. Regarding the test data, a test vector consists of a single measurement; so each measurement has to be used individually and you can perform up to *M* tests for each cell. The data correspond to real acquisition experiments performed outdoors nearby Politecnico di Torino, using an STM32L microcontroller with 915 MHz 802.15.4 transceiver (see picture below).

Task: your task is to implement a k-NN classifier in Matlab for the classification task described above, and evaluate its performance.

Performance evaluation: The performance is defined in terms of accuracy in the localization task, and it has to be averaged over all cells. Accuracy metrics are defined as follows:

- Average accuracy: the posterior probability associated to the cell that the user is actually located in. It is suggested to also consider the case that also the neighboring cells to the correct one are correct.
- Top-k accuracy: the user is located successfully if the correct cell is found among the k nearest neighbors.

Exercise 3 – From classification to regression

This exercise is very similar to Exercise 2. The key difference, though, is that the output of the k-NN classifier should not be the most likely cell the user is located in, but rather an estimate of the user spatial coordinates as a real-valued pair *(x,y)*. The variable `cell_coordinates` in file `localization.mat` contains horizontal and vertical coordinates of the center of each cell in the first and second column respectively. The location of the user is estimated, for the horizontal and vertical coordinates, as a weighted mean of the coordinates of the k nearest-neighbour cells; see the following formula for the horizontal coordinate, and the same is done for the vertical one:

$$\hat{x} = \frac{1}{k}\sum_{i=1}^{k} \xi_i$$

where the sum is over the k nearest-neighbour cells, and ξ_i is the horizontal coordinate of the center of cell *i*.

[1] Bay, A., Carrera, D., Fosson, S.M., Fragneto, P., Grella, M., Ravazzi, C., Magli, E., "Block-Sparsity-based Localization in Wireless Sensor Networks", *EURASIP Journal on Wireless Communications and Networking*, vol. 2015 n. 182, pp. 1-15, 2015