Signal Classification with the k-NN Algorithm

Laboratory 4, DEDP

Objective

Implement and use the k-NN algorithm for classification of various signals.

Theoretical aspects

The k-NN algorithm

Suppose we have a set of **training signals** whose classes are known beforehand. For example:

- 100 signals of class A (e.g. images of cats)
- 100 signals of class B (e.g. images of dogs)
- ...

We have a new signal X. We need to decide to which class it belongs (A, B, etc).

The k-NN algorithm:

1. Compute the distances from X to all the signals in the training set
2. Choose the **closest** k **neighbors**, take the class of the majority of them (e.g. majority voting).

Datasets organization

Usually, we have at our disposal a large class of signals whose classes are known. The data is randomly split into:

- a **training set**: this data is used for the majority voting

- a **test set**: used only for **evaluation** of the algorithm performance. This data should never be used for training (the algorithm should never have seen this data before the testing).
- (optional) a **cross-validation set**: a subset of the training set, used to determine which values of k work best

The datasets are obtained by randomly splitting all the signals available at the beginning. They sizes of the datasets should be around:

- 60% of all data for the training set
- 20% of all data for the cross-validation set
- 20% of all data for the in the testing set

Data for this laboratory

In this laboratory we will use ECG signal data from the MIT-BIH Arrhythmia database.

The excerpt provided for this lab contains ECG signals from 4 classes, with 120 signals for class.

The ECG signals provided here are preprocessed: - all signals are segmented in segments corresponding to one heart beat - the signals are resized to fixed length 256 - the signals are resized so that the peak R wave is located at the center of the signal - the continuous component of all signals has been removed - the signals have been normalized to norm equal to 1

The signals are split into two files:

- training set: `ECG_train.mat`, 4 classes x 100 signals each
- test set: `ECG_test.mat`, 4 classes x 20 signals each

Exercises

1. Load the data files `'ECG_train.mat` and `'ECG_test.mat`. Explore the dataset:

    - display 5 signals from each class contained in the training set. Try to figure sou some visual differences.
    - display the first signal from the test dataset. Try to determine visually to what class it belongs to.

2. Implement a function `[class] = myKNN(signal, k, trainset)` for performing k-NN classification of a signal:

    - the function takes as input an unclassified signal `signal`, the parameter value `k`, and the training set matrix `trainset`
    - the function computes the Euclidean distance between `signal` and each vector from the training set

- the output `class` is defined by the majority of the k nearest neighbours of the signal

3. Call the function `myKNN` for each signal from the testing set and compare the classification results against the ground truth (`test_labels`). Use different values for k: $k = 1$, then $k = 5$, then $k = 15$. In each case, print the *confusion matrix*: A_{ij} = percentage of each signal of class i which are classified by our algorithm as being in class j.
4. Repeat the test in 4., this time adding a variable amount of gaussian noise to the test signals. How does the performance change?
5. Repeat the test in 4., this time adding a DC component to the test signals. How does the performance change?

Final questions

1. How does the confusion matrix look like in the ideal case? (perfect classification)