NPGR035 Homework 3 (Data analysis)

Mgr. Matúš Goliaš, Doc. RNDr. Elena Šikudová PhD.

December 1, 2023

This document gives a brief description of the third homework and our requirements for an acceptable solution. This homework is more open-ended than the previous ones and the evaluation will be based on the completeness of the analysis. Additional details are provided on the practicals or directly in the source code.

Description

You are going to analyse data from the image segmentation domain. Since we have not discussed computer vision methods, you are given a set of features computed from images. The database of feature vectors was created in the Vision Group of University of Massachusetts. The full description of the data is in `segmentation_description.txt`.

Make sure to comment your code and explain what you are trying to do. You don't have to comment simple formulae, but you should note why you chose a particular method and not something else.

Task 1 Familiarise yourself with the template source file `hw3.py` and the data we are working with loaded from the file `segmentation_data.mat`. The data dictionary contains three important data arrays with the names: `feature_names`, `segmentation_features` and `segmentation_labels`.

Task 2 Analyse the features and identify those that are necessary. Lower the dimensionality of the data.

Task 3 Pick three (3) supervised classification methods. Use cross-validation to set the parameters of the models. Then train the classifiers on the full dataset. You can save the trained models to make testing faster.

Task 4 Use an unsupervised method to identify clusters in the data and find the optimal number of clusters.

Task 5 Write a report (in pdf format) where you include the following:

- Description of your approach in the previous tasks.
- Explanation and justification the selection of your models and methods.
- Report of the considered and selected parameters as well as the results acquired during the analysis.
- Commentary on the process (e.g. what went wrong, whether the classifiers behaved as expected, etc.).

Task 6 Complete the given code template and make sure that the evaluation function fulfils the following requirements. It performs the feature selection/transformation, uses(loads) your trained models and classifies the data. For each method it will report the confusion matrix and the macro-averaged precision and recall values. Plot the results for the three models into a precision-recall space.

Requirements

You are supposed to hand in two files `hw3.py` and the PDF file with your report, everything else is available on our side. The comments marked by `TODO` show the places that have to be implemented.

There is 100 points in total for this homework and, as usual, you have to get at least 50% to get your homework marked as completed. The points are awarded for completeness of individual tasks. It is necessary to complete all of the tasks but you are not expected to implement or describe everything perfectly, hence 50%.

Evaluation

Your solution is intended to be evaluated with the source file `evaluator.py` as in the previous homework. If you like using jupyter notebooks then you can convert the `hw3.py` source file into a notebook and submit it instead.

The tasks can be evaluated more or less individually. The evaluator has the following boolean arguments, which will be used to evaluate your solution:

- `--split_test` This boolean value is passed to the constructor of your implementation and it can be used as a condition to split your data into a training and testing set. This can help verify that `evaluate` method works correctly.
- `--feature` Runs the feature analysis (Task 2). This should be the investigation of the feature space in order to find an appropriate feature transformation. Note that feature transformation has to be applied on data both before training and testing.
- `--cluster` Runs the cluster analysis of the data (Task 4).
- `--train_search` This runs the `train_search` method, which should include the implementation of searching for the best parameters of your chosen models (Task 3).
- `--train_best` This runs the `train_best` method, which should train the selected models with the best parameters. This method can be useful if searching the model space takes too long. Otherwise, it can be ignored.
- `--evaluate` Runs the evaluation on testing data (Task 6). You do not have a separate test file so you can simulate having additional dataset by splitting your data based on `--split_test`.

You can use multiple of these arguments at once. We request that the following combinations will work:

- `--feature` Just the feature analysis.
- `--cluster` Cluster analysis on the selected/transformed data.
- `--train_search --evaluate` Runs your search of best model parameters on selected/transformed data and evaluates the best models on the test set. This is for an automatic search of parameters.
- `--train_best --evaluate` Trains your models with the best parameters on selected/transformed data and evaluates them on the test set. This is for manual search of parameters or a time-consuming automatic search.

The code and results have to, in some way, show your experiments and results described in the associated PDF. For instance, if you are writing about a set of parameters which you tried for each model and the feature selection criteria or drawing graphs with the results then all of these things should be present in the code.