Data Mining: Preparation for Practical Assignments

Due on Thu & Fri, May 23- 24 2019, 10:15am-13:15 & 14:15am-17:15

Task 1

Supervised Learning with Decision Trees and Neural Networks ≈ 150 min. Today you will work 'hands on' on the popular `scikit-learn` framework, which offers a huge collection of data mining techniques and easy access to learning models and datasets. In the accompanying Jupyter notebooks, usage of this framework is denoted by `import sklearn`. We will use real datasets both directly importing from the scikit library or from the `openml` database[1].

General hints

Before you start:

- Datasets are matrices, where the number of rows corresponds to the number of observations, and the columns denote both your attributes and the label
- It is convenient to use X for the attributes and y for the class labels
- Start with simple `wine` example first to get familiar with the functions you can call to train and test your models, including setting your initial hyperparameters, and the evaluation metrics you can use to compare different experimental settings
- Check if you imported all necessary packages in case a function call is not working (header of notebook files)
- Use the `shape` command in case you want to check on the dimensionality of your data and that you put the correct data into X and y
- To split your data use `train_test_split` and include the desired split for your test (e.g. `0.3` means 30% of your data is used as test set). The function `cross_val_scores` does the splitting into training and test set already Include the number of folds for training your model
- We provide you with working examples for the first dataset. Note, that for MNIST, k-fold crossvalidation and gridsearch we only provide you code snippets, which you can integrate or adapt
- Use the functionalities of `matplotlib`, which you may remember from the 2nd tutorial, whenever you want to plot something

What are you expected to do?

Take special care of experimental reproduction and keep track of the following:

- Sampling of training and test data, crossvalidation
- Track of hyperparameter, random seeds
- Performance metrics

Please note that the confusion metric format in `scikit` is this:

[1] `https://www.openml.org/home`

	pred: no	pred:yes
class: no	TN	FP
class: yes	FN	TP

- Make **reasonable decisions** on any architectural design (e.g. number of nodes in an NN) and on value ranges for hyperparameters (e.g. depth of tree, learning rate in NN). It is fine to play around but in the end it is more rewarding to come up with conclusive experiments and their performance evaluation
- Make sure you can **explain and interpret** the evaluation metrics
- Take into account that computing power is limited, you may trap into convergence issues. Do not bother, instead take track at which point your classifier fails to converge and which result you got before termination. Try then to lower e.g. number of iterations, number of nodes, etc.

The 'Wine' dataset

An easy access to learning models is to first employ them on a clearly arranged and simple classification task as provided with the `wine` dataset. For the description of the dataset check `https://archive.ics.uci.edu/ml/datasets/wine`, which shows also papers using this dataset. Check out the files `simple_dt_wine.ipynb` and `simple_mlp_wine.ipynb` to get familiar with the `scikit` package. Read the comments carefully, perform several experiments and provide a performance evaluation sheet. The minimum expectation is that you:

- for decision tree: change the information criterion
- for MLP: change the learning rate, set the solver to 'sgd' (optionally: compare 'adam' and 'sgd')
- both: vary training and test set sizes
- both: integrate crossvalidation (see code snippets)
- both: integrate gridsearch (see code snippets)
- both: provide evaluation metrics for each experimental run (Please note how the confusion matrix is returned, e.g. the true positives are in the right corner)

Optional: Change other parameters that you find critical for performance

The 'MNIST' dataset

Another popular image benchmark is the MNIST database, containing 28×28 images of ciphers $0-9$. Please use the `MNIST.ipynb` as a start to perform cipher classification (for details check here `https://www.openml.org/d/554`. Perform several experimental runs varying parameter and provide an evaluation of your classifications. What is your highest accuracy?

Optional: Dataset of your choice

Both `scikit` and `openml` provide different datasets you can easily load and try out with your code, for example, 'phoneme' recognition' or 'robot navigation' at `openml`. From the two tasks before you know how to load the data. Choose a classifier and report on the performance.

Optional: Classifier of your choice

You may realized by now that the library has many classification tools. If you are familiar with another technique, e.g. SVM, feel free to try it out. However, if you choose something else, you are also expected to present and explain :).

@home Task

Now that you got some insights into supervised learning, we will proceed with another learning strategy: unsupervised learning. Therefore, refresh yourself with following topics:

- Self-Organized Maps (SOM)
- k-means clustering, DBScan, hierarchical clustering