Data Mining: Preparation for Practical Assignments

Due on Thu & Fri, June 06-07 2019, 10:15am-13:15 & 14:15am-17:15

Task 1

The k-means algorithm ≈ 30 min.

In the *data* folder for our todays' tutorial you will find an implementation of the *k-means* cluster algorithm. Use the `simple_KMEANS.ipynb` for an implementation in `scikit learn` and discuss the underlying principles of the clustering procedure with your group partner. The following questions may help you to recall and understand the algorithm:

- What is the initialization procedure?
- How does the algorithm find clusters?
- When does the algorithm stop (termination criterion)?

Task 2

What is the best k? ≈ 30 min.

A major drawback of the *k-means* algorithm is that you have to determine the number of k clusters beforehand. However, it is possible to find the best k e.g. calculating the *sum of squared distances* or the *silhoutte coefficient*. Use the `bestK.ipynb` file to loop over a range of k from 2 to 15 (to assume just one cluster is not sensible). What is the best number of clusters for the given data?

Optional: Create other data distributions with e.g. `make_blobs` to test for other best k values

Task 3

Clustering comparisons ≈ 45 min.

In the lecture you learnt also about other clustering techniques and we will now integrate some other algorithms for a little comparitive study on datasets with different properties. In `clusteringComparisons.ipynb` we prepared some datasets for you. First, run the *k-means* algorithm. What are the results on the datasets? Second, run *agglomerative clustering* on the datasets trying 3 different linkage types {*single, average, complete*} and the *DBSCAN* algorithm. What are your results? Explain the differences between the applied methods.

Optional: In the last tutorial we gave some sources to get data from; feel free to use another dataset and compare the performance across the different clustering methods.

Task 4

Self Organizing Maps ≈ 45 min.

Another popular unsupervised learning algorithm is the *Self-Organizing Map* (SOM) or *Kohonen Map*. Similar to the previous task, get the corresponding Python code to run in a terminal. **Please note**: the data to feed the SOM is in the directory `SOM-Data`, the corresponding file in `SOM-Python`; the GNU directory is necessary to run the SOM simulation in the terminal, please **ignore**.

Discuss the algorithm with your group partner. Answering the following questions may help you to understand the algorithm:

- What is the initialization procedure?
- How does the algorithm determine the *best matching unit* (BMU). Give also an intuitive idea of this concept.
- What is the role of the neighborhood function?
- What is the role of the learning rate?
- When does the algorithm stop (termination criterion)?

@home Task

For the next tutorial, prepare the following topics

- Fuzzy Logic, Modelling a fuzzy system