Barbara Hammer, Philip Kenneweg, Bianca Schröder Universität Bielefeld

Maschinelles Lernen im Web (WiSe 24/25)
3. Sheet

Start: Monday, 11.11.2024.
End: The worksheets should be solved using Python, in groups of 2-3 people and will be presented in the Tutorials.
Discussion: Monday or Thursday, 25.11.2024 in the Tutorials.

Information

The worksheets will be made available in the Lernraum "392180 Maschinelles Lernen im Web (V)”. Worksheets will usually be released every two weeks on Monday, and discussed during the tutorials two weeks later. In order to successfully finish the course, 50% of the available points have to be obtained by presenting the results in the tutorials.
The week in between the release and discussion of the sheet will be used to discuss the theoretical exercises, the implementation of various algorithms as presented in the lecture, as well as go deeper into the relevant material.

You can use all sources from the internet, but you must add a reference. Please solve and prepare a short presentation of the methods and results. Each group needs to prepare solutions for all of the tasks and needs to present them in the tutorial.

Exercise 1: Visualizations

(*5 Points*)

T-SNE and UMAP are the most used non-linear dimensionality reduction methods. Both have crucial hyperparameter options (perplexity for T-SNE and n-neighbors for UMAP). Compare the two methods on a complex dataset of your choice (more than 10 dimensions, nonlinear). What are the differences between the two methods? (1 Pts.) What are the advantages and disadvantages of each method? (1 Pts.) Try out different hyperparameters (1 Pts.), visualize the results and find hyperparameter options that suggest faulty correlations (correlations that are not there but T-SNE or UMAP are suggesting them) in your visualizations. (2 Pts.) If you can not find any faulty correlations, try to modify your data set or create a custom data set to create some.

Exercise 2: Clustering

(*5 Points*)

Generate two artificial two dimensional data sets with at least 10 different intrinsic clusters (for example checkerboard structure)(1 Pts.). Transform the data into a format such that kmeans++, affinity propagation, and spectral clustering, respectively, can deal with the data and describe, which transformation you use and why (1 Pts.)(Note that some methods do this implicitly in scikit-learn functions. Please check this!). Produce clustering results for each of the algorithms using the intrinsic number of clusters and inspect the result visually (1 Pts.). Also provide a numeric evaluation (1 Pts.). Which method works best and why? (1 Pts.)