188.413 Self-Organizing Systems(VU 3.0)
2021 WS

HandsOn Assignments: SOM

This is the description of the "implementation" option for assignment 3, i.e. for groups wishing to implement SOM visualizations in Python rather than perform extensive experiments using the SOM. It focuses on implementing one of a set of visualizations, providing a short report on how it was implemented, how it is to be used, as well as a few examples of visualizations obtained.

A) Jupyter notebook with selected SOM visualizations

- You will receive a Jupyter notebook that contains the modules for SOM training and some basic visualizations (U-matrix, hit histogram, SDH), to which you should add your own code. The notebook also contains code to read SOM files trained using the Java SOM toolbox so that you can compare the visualizations you create with the ones provided by the toolbox.

B) Implementation

1) ***Select and register*** **the (group of) visualizations** you picked for implementation with your group number in the TUWEL Wiki. Each set of visualizations may be picked by max 2 groups (first come, first serve - do it early to be able to choose a visualization you also find interesting to work on – if there are more groups than expected wishing to choose the coding option over the analytics option we can add more groups)
2) **Implement the (set of) visualizations** according to the group you selected. You may, of course, refer to the source code of the Java SOM toolbox for inspiration on the coding, or consult other external sources)
3) Provide a documentation of the implementation and explanations, ideally integrated as text directly in the Jupyter Notebook (to be exported as PDF for the final report, or, alternatively, as a separate report)
4) Implementations should be provided via a public repository (e.g. Github) and specify an Open Source license, preferably Apache (or MIT).

Visualizations to choose from (groups a-h, some containing several sub-visualizations):

a) MetroMaps (as overlay on any provided coloring, e.g. the U-Matrix), including proper labeling of aggregated lines via a control-able parameter as well as labeling the high/low endpoints

b) Sky Metaphor (Stardinates): (as overlay on some provided coloring, e.g. a greyscale/black-white U-Matrix), interpolating the "true" position of a mapped instance between the three neighboring units.

c) 2 topographic error visualizations selected from
   i) Topographic error
   ii) Intrinsic distance
   iii) Topographic product
   iv) Topographic function
   v) SOM distortion

d) labelSOM, i.e. selecting those attributes per unit/cluster that show the lowest variance and the highest absolute values within a specific unit / cluster

e) “Chernoff faces”: using metaphor graphics to depict attribute values as overlays on units (or, as averages, for regions/clusters), allowing for some flexibility in the mapping between attributes and metaphor features.

f) Mnemonic SOM: Reading a black/white silhouette image to determine the shape of the SOM, computing a look-up table for shortest distance between two units for the training process, and mapping it into a regular SOM data structure to support the standard visualizations..

g) Aligned SOMs: Training a layer of n SOMs stacked on top of each other with differently weighted subsets of attributes, storing/exporting them as individual or linearly arranged SOMs.

h) Three simpler visualizations combined:

   i) Activity Histograms, allowing to iterate through the data set being projected onto the SOM instance by instance or selecting a specific instance.
   ii) Minimum spanning tree of data and units, indicating the root-to-leave trajectory by different line widths
   iii) Cluster connections with several selectable/iterable thresholds

C) Evaluation Report

1) Pick the Chainlink Data Set and the 10-Clusters dataset from http://www.ifs.tuwien.ac.at/dm/somtoolbox/datasets.html
2) Train a 40x20 (small) and a 100x60 (large) SOM. Make sure that the SOMs are properly trained, i.e. that the structures to be expected in the SOM become clearly visible. (Note: if you run into performance issues, try running the code natively in Python, which is sometimes substantially faster than running it in a Jupyter notebook)
3) Show the visualizations, providing examples with different parameter settings and comparisons that allow a validation of the correctness of the implementation. Specifically, test a few extreme values for the visualization.
4) Read a SOM pre-trained with the JAVA SOM Toolbox (import functions are provided in the notebook) and compare your visualization with the one produced by the Java SOMToolbox (using either the pre-trained SOMs provided with the toolbox, or any that your colleagues who do the analytics option of the exercise share with you).
5) Describe, preferably directly within the Jupyter notebook (with options to select one of the two input files specified above and an arbitrary input file from local storage) and provide (export) as separate PDF report:
   - The implementation developed, explaining key parts of the code
   - The way to operate the code, i.e. adapting parameters, and discussing their effect on the visualization
   - Present the various evaluations performed under item 3) above, demonstrating the correctness of the implementation, and the plausibility of the information to be gained.
   - Compare the visualization with the identical visualizations (reading the same trained SOM files) using the SOM Java Toolbox

D) Feedback

1. (**optional**) Provide feedback on the exercise in general, and specifically whether having this option of a more implementation-oriented assignment makes sense, etc. (this section is, obviously, optional and will not be considered for grading. You may also decide to provide that kind of feedback anonymously via the feedback mechanism in TISS – in any case we would appreciate learning about it to adjust the exercises for next year.)

HandsOn Assignments: SOM

HandsOn Assignments: SOM

Submission guidelines:

- **Upload ONE [zip/tgz/rar] file** to TUWEL that **contains all your files** (all scripts/programs you wrote and subsidiary information for repeating experiments) as well as the report document (PDF) export fro Jupyter Notebook or separate report) containing your names and student IDs as well as the link to the repository (e.g. Github)**.** You must follow this naming convention for the zip file and report document:
  - ***SOS2021_ExSOM-C_group<groupno>_<studID1>_<studID2>_<studID3>.[zip/tgz/rar]***
  - Example: A submission of group 99 with 3 students (ids: 019999, 029999, 039999) looks like this:
    SOS2021_ExSOM-C_group99_0019999_0029999_0039999.[zip/tgz/rar]
  - Apply the same naming convention to the report (but obviously with pdf extension)
- **Put your names and your studentIDs in the report** (as author info), and clearly specify the chosen (group of) **visualizations in the title**.