Code Guide

Ani Mkheidze

July 2023

This document presents a step-by-step explanation of the code produced for the paper, "Navigating Complexity: Evaluating the Effectiveness of Dimension Reduction and Clustering Approaches on Complex Datasets". There are 6 datasets that research has explored:: Jester, Swiss Roll, Broken Swiss Roll, Twinpeaks, Helix, and High-Dimensional dataset. For each of the datasets, there is a folder named for them that contains all the code. In the files, you will find two R files named data set name + results and dataset name + tuning. For some datasets, there are also additional files and their purpose will be described in the steps that will follow.

First we will discuss production of Datasets:

- **Jester:** In the Jester folder you will find three Excel files named: "jester-data-1", "jester-data-2", and "jester-data-3". They contain the data given at `https://eigentaste.berkeley.edu/dataset/`. Open the "Jester Results" R file and load the data by modifying the file paths given in the document to that of your local machine. Run code part that cleans the data from observations with missing values and produces a random sample using the seed given.

- **Swiss Roll, Broken Swiss Roll, Helix, and TwinPeaks:** For these data sets, the result file contains the function that generates the dataset in the first lines of code. Run that function. Then It is possible to run the second chunk of code that generates the data set object with predefined seed.

- **High-Dimensional:** The folder contains two Matlab files. *generate_data* and "combn". Open both of them and run *generate_data* function with the given instructions. Then export the results into an Excel file. If you do not wish to redo this process the Excel file is also given in the folder. Then in the results file modify the path to Excel file to the one corresponding to your local machine.

The procedure for all the data sets is similar after the production of data sets is completed and is as follows:

- Run NbClust functions to determine the optimal number of clusters and assign that value to the *clust_num* variable.
- Then navigate to the Tuning file and run it fully to determine the best hyperparameters for UMAP.
- Navigate back to the results file and input the optimal hyperparameters for UMAP. Now you can run the code fully and will get a matrix "*results_evaluation*" that holds results for internal evaluation indexes for all the DR techniques and cluster methods. You can also get 2-D visualizations by running the last chunk of code. You can get 3-D visualizations code for Swiss Roll, Broken Swiss Roll, Helix and TwinPeaks datasets.