

Structural k -means user guide v1

June 24, 2022

Đoàn Quang Văn

Center for Computational Sciences

University of Tsukuba

E-mail: doan.van.gb@u.tsukuba.ac.jp

Copyright notices

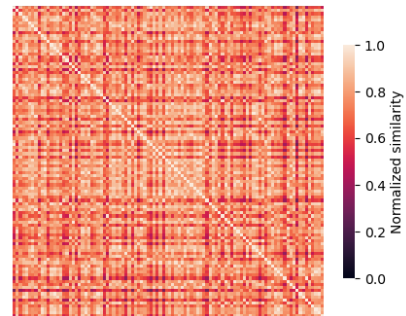
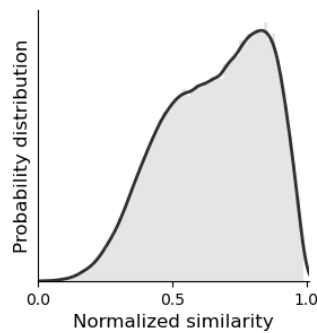
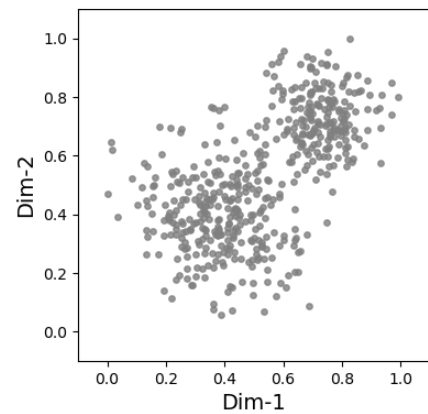
Structural k -means (or S k -means) algorithm was developed by a team led by Dr. Quang-Van DOAN at the Center for Computational Sciences (CCS), the University of Tsukuba. S k -means can be used by any person or entity for any purpose without any fee or charge. We request that any user include this notice on any partial or full copies of S k -means. S k -means is provided on an "AS IS" basis and any warranties, either express or implied, including but not limited to implied warranties of non-infringement, originality, merchantability, and fitness for a particular purpose, are disclaimed. In no event we shall be liable for any damages, whatsoever, whether direct, indirect, consequential, or special, that arise out of or in connection with the access, use or performance of S k -means, including infringement actions.

Full program of S k -means is repositied in the Github: <https://github.com/doan-van/S-k-means>. Scripts were written on Python language. Scripts includes k -means main programs; wrap-up code (for three test runs); clustering outcome analysis: a) similarity-distributions, b) silhouette analysis results, c) clustering uncertainty degree evaluations. Details of the scripts are described as follows:

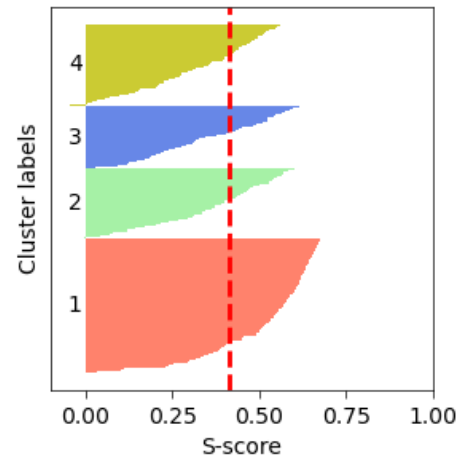
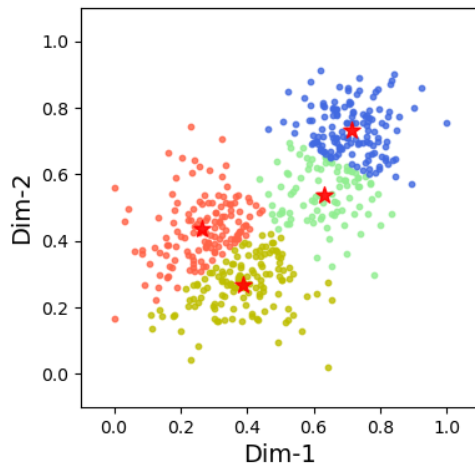
Main program

Scripts	Descriptions
kmean.py	<ul style="list-style-type: none">main program containing functions of the k-means algorithm $k\text{-means}(\mathbf{X}, k, \text{sim}, \text{ini})$, where \mathbf{X} is a set of input vectors, k is number of clusters. sim is classification scheme (one can select "ssim", "str", "ed", "md", which mean structural similarity, Pearson correlation coefficient, Euclidean distance, or Manhattan distance). ini is initialization scheme. One can select "rand" for randomized initialization, or "pp" for k-means ++ initialization schemeNo furthermore libraries (other than default ones in anaconda framework) is needed to run the program.This script is self-standing. One can run it directly from terminal <code>\$ python kmean.py</code> to generate some demonstration result.

- In demonstration runs, k -means is applied for randomly distributed 2-dimensional data. One can select preferable setting, by modifying some lines (for k , sim , etc.)
- The clustering result is saved in [demo/run_?/ cluster.csv](#), and plotting are also automatically generated and saved in the same directory. For example, the scatter plot of the input data (right plot). Also diagnosing results such as the Similarity-Distribution, Similarity-Matrix of the input vectors is also automatically generated (below)



- Clustering results are plotted automatically together with the silhouette analysis. See figures below for data which are grouped into 4 clusters and Silhouette score plotted besides.

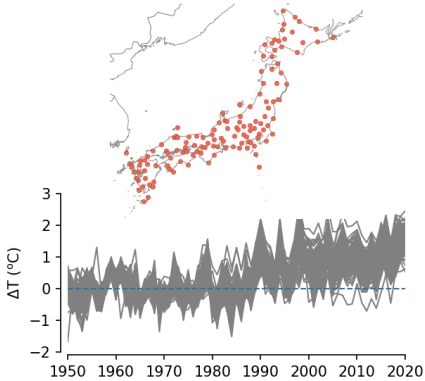
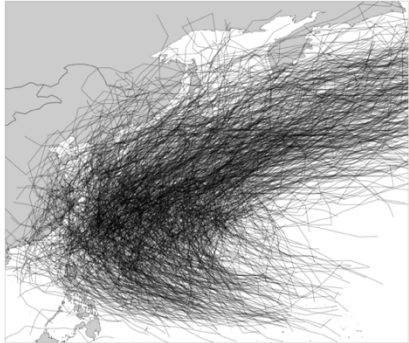


	<ul style="list-style-type: none"> In the case of multiple runs, one can assess the uncertainty/consensus of clustering results using Clustering Uncertainty Evaluation framework proposed together with S k-means. This analysis is also automatically conducted and the plot is generated (right figure) 	
main_kmean.py	<ul style="list-style-type: none"> script to wrap up kmean.py for three test problems. With several number of loops commands, i.e., 3 (tests, 'TC_II', 'AM_t_1950_y', 'SLP_DJF') x 4 (classification schemes, 'ssim', 'ed', 'md', 'str') x 11 (number of cluster k) x 10 (initializations) = 1320 runs are conducted. Output data are generated and save into directory specified in the script, for each run the Silhouette analysis is also automatically conducted. The Silhouette score, and figure are automatically saved in the output directory (explained latter in Output section) Users have to specify the link to input and output directories before running the script. 	

Input data

Three demonstration tests include clustering weather pattern (WP), time series of long-term historical climate change (CC) data, tropical cyclone (TC) best tracks. Input files for these three problems, named SLP_DJP.nc, AM_t_1950_y.nc, TC_II.nc are located in directory **input_data/** or can be download from [Google Drive link](#).

Data file	Descriptions
SLP_DJF.nc	<p>Data used for Weather Pattern (WP) clustering. To group winter weather pattern in Japan (see figure below). Use mean sea level pressure (SLP), which are obtained from ERA-Interim reanalysis. The data have horizontal resolution of 0.75° on a regular grid but are re-gridded to an equal-area scalable earth-type grid at a spatial resolution of 200×200 km, thus grid size of 35×35. Winter months, i.e., December, January, February (DJF), for ten years 2005-2014 over the region of $20 - 50^\circ\text{N}$ and $115 - 165^\circ\text{E}$ are used. The total number of samples is 902.</p>

AM_t_1950_y.nc	<p>Climate Change (CC) clustering. Input data are temperature-increase time series for 70-year (1951 – 2020) from weather stations run by Japan Meteorological Agency. There are 134 weather sites data from which is used as sample. Annual means of each time series are calculated. Climate change component are determined by subtracting the average of the first 30 years (1951 – 1980) from each value series.</p> 
TC_II.nc	<p>Tropical Cyclone (TC) tracking clustering. Best TC tracks from 1951 to 2020 are retrieved from the Japan Regional Specialized Meteorological Center (RSMC) (https://www.jma.go.jp/jma/jma-eng/jma-center/rsmc-hp-pub-eg/besttrack.html). In this study, only TC that passing the Japan region, defined as the region of 25 – 45°N and 126 – 150°E, are used for analysis. Hence, the total number of TC to feed the k-means is 863. TC tracks are reconstructed so that they have the equal length of 20 segments by the method.</p> 

Output data

Output data of three demonstration runs can be downloaded from [Google Drive link](#).

Structure of output directory is as follow:

`output/AA/BB/CC/rand`

where AA indicates the test name, BB indicates the initialization, CC indicates the number of clusters (k)

Also, the results pairwise similarities of input vectors is also contained in

`output/AA/DD_sim_btw.nc`

where DD indicates the similarity indices

Plotting

Scripts for plotting are saved in directory `plot/`. All figures in the manuscript can be replotted by using these scripts. Scripts are written in python language. Full version of figures can be downloaded from [Google Drive link](#):

Scripts	Description
plot_fig02_S-Ds.py	For plotting fig 02 in the manuscript, i.e., the S-Distributions of input vectors.
plot_fig03_WP.py	For plotting fig 03 in the manuscript, i.e., the results for the WP test.
plot_fig04_cc.py	For plotting fig 04 in the manuscript, i.e., the results for the CC test.
plot_fig05_tc.py	For plotting fig 05 in the manuscript, i.e., the results for the TC test.
plot_fig0607_gen_res.py	For plotting fig 06 and 07 in the manuscript. Collective results of Silhouette scores and time consumed for running algorithms.
plot_chord.py plot_fig08-10_CUD.py	For plotting fig 08, 09, 10 in the manuscript. For plotting fig 08, the function for chord diagram is called from plot_chord.py
write_docx.py	This script is to automatically generating the Supplementary document in both the docx and pdf formats.