Density-Based Clustering for Adaptive Density Variation - Supplementary

Li Qian*, Claudia Plant†, Christian Böhm*
*Institute of Informatics, Ludwig Maximilian University of Munich, Munich, Germany
{li.qian, boehm}@dbs.ifi.lmu.de
†Faculty of Computer Science, ds:UniVie, University of Vienna, Vienna, Austria
claudia.plant@univie.ac.at

I. IMPLEMENTATION AND PARAMETER SETTINGS

The proposed algorithm DBADV is implemented in Python. Source code, synthetic data sets, and supplementary material are available at https://dmm.dbs.ifi.lmu.de/cms/downloads. We run all experiments on a machine equipped with six cores (Intel Core i7-8700 3.19GHz) and 32GB memory.

We conduct experiments on two synthetic data sets *Shape3* and *Shape5* generated by us, five common data sets with multivariate including Seeds, Dermatology, Image Segment, Page Blocks, and Crowdsourced Mapping (Crowdsourced) from the UCI Machine Learning Repository [1], and one face image data set warpPIE10P with high dimensionality from the Feature Selection Repository[1]. The statistics of these data sets are briefly described in Table I. We preprocess datasets using the MinMaxScaler such that each feature ranges in $[0, 1]$ before the experiments. Besides, we run all initialization dependent algorithms (e.g., k-means, Spectral clustering) 10 times and report the best results.

Table I
STATISTICS OF DATA SETS

Data set	# Samples	# Features	# Classes
Shape3	2250	2	3
Shape5	3150	2	5
Seeds	210	7	3
Dermatology	358	34	6
Image Segment	2310	19	7
Page Blocks	5473	10	5
Crowdsourced	10545	28	6
warpPIE10P	210	2420	10

To extensively evaluate the performance of the proposed DBADV, we take a range of competitive clustering algorithms as our baselines, including density-based methods DBSCAN[2] and its variants OPTICS[2], HDBSCAN[3], CRAD[4], DBSCAN-DScale[5], SpectACl[6], and other categories of clustering algorithms k-means[2], Spectral clustering[7], Self-tuning spectral clustering[7], Affinity Propagation[2], and Sync[8]. For each baseline, we search for the best achievable clustering result in a reasonable range of parameters, or the default value of parameters recommended by the authors, as shown in Table II. Notably, Sync is a parameter-free clustering algorithm.

Table II
PARAMETERS AND THEIR SEARCH RANGE FOR EACH CLUSTERING ALGORITHM. $nSample$ IS THE NUMBER OF SAMPLES; N DENOTES THE NUMBER OF CLUSTERS; $median$ PRESENTS THE MEDIAN OF THE SIMILARITY MATRIX.

Algorithm	Parameters with search range
DBADV	$MinPts \in \{1, 2, ..., 30\}$; $perp \in \{1, 2, ..., 30\}$; $prob = 0.977$
DBSCAN	$MinPts \in \{1, 2, ..., 30\}$; $eps \in \{0.01, 0.02, ..., 1\}$
OPTICS	$MinPts \in \{1, 2, ..., 30\}$; $\xi \in \{0.01, 0.02, ..., 0.99\}$
HDBSCAN	$minSample = minCluster \in \{2, 3, ..., 30\}$; $\alpha = 1$
CRAD	$StepSize = 1; Nbin \in (0.2 * nSample - 100, 0.2 * nSample + 100)$
DBSCAN-DScale	$MinPts \in \{1, 2, ..., 30\}$; $eps \in \{0.01, 0.02, ..., 1\}$; $\eta \in \{0.1, 0.2, ..., 0.5\}$
SpectACl	$eps \in \{0.01, 0.02, ..., 1\}$; $numCluster = N; d = 50$
k-means	$numCluster = N$
Spectral	$numCluster = N; \sigma = 0.04$
Self-tuning	$numCluster \in \{2, 3, ..., 100\}$; $K = 7$
Affinity Prop.	$lam \in \{0.5, 0.55, ..., 0.95\}$; $p = median$

REFERENCES

[1] D. Dua and C. Graff, "{UCI} Machine Learning Repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[1] https://jundongl.github.io/scikit-feature/datasets.html
[2] https://scikit-learn.org/stable/modules/clustering.html
[3] https://github.com/scikit-learn-contrib/hdbscan
[4] https://github.com/DataMining-ClusteringAnalysis/CRAD-Clustering/
[5] https://sourceforge.net/projects/distance-scaling/
[6] https://sfb876.tu-dortmund.de/spectacl/index.html
[7] http://www.vision.caltech.edu/lihi/Demos/SelfTuningClustering.html
[8] http://dm.uestc.edu.cn/publication/