Supplementary Material for: Chameleon 2: An Improved Graph-Based Clustering Algorithm

TOMAS BARTON, Czech Technical University in Prague, Institute of Molecular Genetics ASCR
TOMAS BRUNA and PAVEL KORDIK, Czech Technical University in Prague

1 COMPARING CLUSTERINGS

There are many methods used for evaluation of clusterings. This article uses Normalized Mutual Information (NMI) [2], which comes from information theory background. Another type of measure is based on counting pairs.

We first introduce a notation used for defining different measures. Let X be a finite set of instances with cardinality $|X| = n$. A clustering $\mathbb{C}$ is a set $\{C_1, C_2, \ldots, C_k\}$ of non-empty subsets of X, so that the union is equal to X as follows:

$$X = \bigcup_{i=1}^{k} C_i.$$

Let $\mathbb{C}' = \{C'_1, \ldots, C'_l\}$ denote a second clustering of X with a number of clusters equal to l. The *confusion matrix* (or contingency table) M of pairs $\mathbb{C}, \mathbb{C}'$ is a $k \times l$ matrix whose ij-th entry equals the number of elements in the intersection of clusters C_i and C'_j:

$$M_{ij} = |C_i \cap C'_j|, \ 1 \leq i \leq k, \ 1 \leq j \leq l.$$

See Table 1 for more visual representation.

Table 1. The Contingency Table

$\mathbb{C} \setminus \mathbb{C}'$	C'_1	C'_2	$\ldots$	C'_l	Sums
C_1	M_{11}	M_{12}	$\ldots$	M_{1l}	$\lvert C_1 \rvert$
C_2	M_{21}	M_{22}	$\ldots$	M_{2l}	$\lvert C_1 \rvert$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
C_k	M_{k1}	M_{k2}	$\ldots$	n_{kl}	$\lvert C_k \rvert$
Sums	$\lvert C'_1 \rvert$	$\lvert C'_2 \rvert$	$\ldots$	$\lvert C'_l \rvert$	$\sum_{ij} M_{ij} = n$

NMI Variants. Several NMI normalizations are possible based on the observation that [4]

$$I(\mathbb{C}, \mathbb{C}') \leq \min\{H(\mathbb{C}), H(\mathbb{C}')\}.$$

Strehl and Ghosh [4] use following definition of NMI, which is referred in the main article as NMI_{sqrt},

$$NMI_{sqrt}(\mathbb{C}, \mathbb{C}') = \frac{I(\mathbb{C}, \mathbb{C}')}{\sqrt{H(\mathbb{C})H(\mathbb{C}')}},$$

1556-4681/2019/01-ART10 $15.00
https://doi.org/10.1145/3299876

which prefers the normalization by the geometric mean. A slightly different normalization was proposed by Kvålseth [2] as follows:

$$NMI_{max}(\mathbb{C}, \mathbb{C}') = \frac{I(\mathbb{C}, \mathbb{C}')}{\max\{H(\mathbb{C}), H(\mathbb{C}')\}},$$

or average, as proposed by Kvålseth [2] and later by Fred and Jain [1] as follows:

$$NMI_{avg}(\mathbb{C}, \mathbb{C}') = \frac{2\, I(\mathbb{C}, \mathbb{C}')}{H(\mathbb{C}) + H(\mathbb{C}')}.$$

2 DATASET VISUALIZATIONS

All datasets used in our experiments contain distinguishable pattern. Datasets are highlighted according to golden standard (true class labels – provided either by original dataset author, or manually labeled by us) against which clustering results were evaluated. In case of datasets contaminated by noise, clusters are formed by areas with high data points density (Figures 1–2).

Datasets with their labels in ARFF format are available online at https://github.com/deric/clustering-benchmark.

3 UNSUPERVISED EVALUATION

Another approach for clustering evaluation is by using unsupervised criteria (indicies) that does not operate with knowledge of true class labels. The Silhouette index is frequently used in the literature, partially due to ease of interpretation. Unlike many other indicies Silhouette index is defined on a fixed interval.

Silhouette Index. Silhouette [3] combines criteria for cohesion and separation of clusters. It is computed individually for each object in dataset. Computing could be divided into following steps:

(1) For the object x, calculate its average distance to all other objects in its cluster. We call this value $a(x)$, given by

$$a(x) = \frac{1}{|C_k| - 1} \sum_{y \in C_k, y \neq x} d(x, y). \tag{1}$$

(2) For the object x and any cluster not containing this object, calculate object's average distance to all the objects in the given cluster. Find the minimum value and call it b, given by

$$b(x) = \min_{j, j \neq k} \left[\frac{1}{|C_j|} \sum_{y \in C_j} d(x, y) \right]. \tag{2}$$

(3) For the object x, the Silhouette coefficient is

$$s(x) = \frac{b(x) - a(x)}{\max\{b(x), a(x)\}}. \tag{3}$$

(4) Repeat previous steps for all objects in cluster to get an average cluster's value.
(5) Afterwards sum cluster's Silhouettes and divide by number of clusters.

$$f_S(\mathbb{C}) = \frac{1}{|\mathbb{C}|} \sum_{k=1}^{|\mathbb{C}|} \left(\frac{1}{|C_k|} \sum_{x \in C_k} s(x) \right). \tag{4}$$

The value of the Silhouette is in a range between −1 and 1. A negative value is undesirable because it means that the average distance within cluster is greater than minimal distance to other cluster.

The Silhouette coefficient is the default evaluation criterion in MATLAB. In order to obtain same numerical results as in MATLAB, one must use Euclidean distances without applying square root.

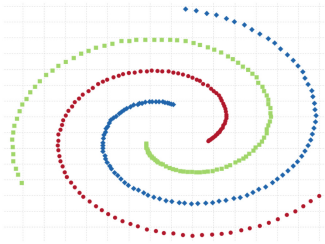
3.1 Results

Table 2 shows clustering results with best Silhouette values found with each algorithm on our benchmark suite. Silhouette index from definition favors centroid clusters. Unsurprisingly *k*-means and single-linkage hierarchical clustering are between top 3 algorithms. It is important to note

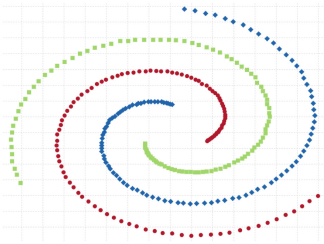
Table 2. Best Silhouette Values Obtained for Each Algorithm and Dataset Included in Our Benchmark

Dataset	Ch2*	HC-SL	*k*-means	DBSCAN	CURE	CL-G	AP	CW	Ch1
3-spiral	0.515	−0.087	**0.535**	0.072	0.512	0.522	0.428	0.496	0.260
aggregation	**0.770**	0.679	0.689	0.679	0.663	0.726	0.634	0.679	0.431
atom	0.638	0.242	0.488	**0.748**	0.133	0.116	−0.014	0.333	−0.098
chainlink	0.623	0.179	0.529	**0.639**	0.343	0.222	0.445	0.409	0.262
cluto-t4.8k	0.702	**0.895**	0.585	0.631	0.416	0.400	0.374	0.235	0.423
cluto-t5.8k	0.640	**0.936**	0.699	0.716	0.715	0.607	0.111	0.225	0.557
cluto-t7.10k	0.700	**0.898**	0.584	0.757	0.377	0.349	0.505	0.280	0.344
cluto-t8.8k	0.760	**0.902**	0.605	0.535	0.464	0.202	0.408	0.281	0.152
compound	0.512	**0.894**	0.641	0.839	0.643	0.557	0.697	0.463	−0.060
cure-t2-4k	0.572	**0.827**	0.679	0.543	0.388	0.535	0.506	0.285	0.230
D31	0.753	**0.782**	0.748	0.672	0.538	0.730	0.535	0.559	0.462
dense-disk-5k	0.533	0.763	0.480	**0.833**	0.555	0.570	0.161	0.385	0.385
diamond9	**0.753**	0.571	**0.753**	0.587	0.741	**0.753**	0.322	0.373	0.276
disk-in-disk	0.601	**0.676**	0.527	0.639	0.306	0.139	0.247	0.412	0.331
dpb	0.570	**0.888**	0.626	0.784	0.621	0.523	0.455	0.312	0.300
DS-850	0.758	0.683	**0.766**	0.612	0.697	0.754	0.311	0.366	0.536
flame	0.598	**0.709**	0.534	0.674	0.674	0.460	0.533	0.385	0.077
impossible	0.642	**0.873**	0.627	0.678	0.367	0.153	0.776	0.427	0.201
jain	0.638	0.434	0.670	0.600	**0.674**	0.553	0.483	0.490	0.390
long1	0.607	0.357	0.614	**0.620**	0.376	0.357	0.605	0.420	0.358
longsquare	0.683	0.755	0.732	**0.820**	0.781	0.678	0.810	0.452	0.539
lsun	**0.705**	0.629	0.683	0.640	0.629	0.629	0.609	0.436	0.036
pathbased	0.682	0.478	**0.692**	0.594	0.421	0.576	0.191	0.417	0.121
s-set1	0.867	0.862	**0.880**	0.766	0.647	0.833	0.424	0.438	0.596
sizes1	0.766	**0.827**	0.788	0.582	0.476	0.494	0.569	0.235	0.462
smile1	0.568	0.499	0.715	**0.749**	0.489	0.471	0.508	0.408	0.334
spiralsquare	0.718	**0.725**	0.709	0.650	0.694	0.455	0.479	0.467	0.417
target	0.739	0.739	**0.789**	0.383	0.739	0.176	0.307	0.395	0.203
triangle1	0.891	0.891	**0.899**	0.891	0.891	0.600	0.684	0.326	0.529
twodiamonds	**0.832**	0.543	**0.832**	0.831	0.818	**0.832**	0.823	0.424	−0.018
wingnut	0.630	0.630	0.646	**0.666**	0.448	0.630	0.547	0.422	0.664
zelnik4	0.666	**0.862**	0.763	0.696	0.664	0.347	0.715	0.151	−0.316
AVG	**0.676**	0.673	0.672	0.660	0.559	0.498	0.475	0.387	0.293
σ (SD)	0.097	0.242	0.108	0.151	0.173	0.206	0.202	0.105	0.223

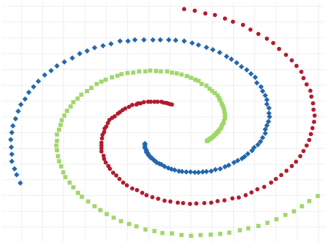
that clustering results selected by unsupervised index can be very far from expected labels. Thus, such evaluation only verifies that algorithm is capable of producing compact clusters. Nonetheless, singleton clusters are not properly penalized, which can explain high ranking of single-linkage algorithm.



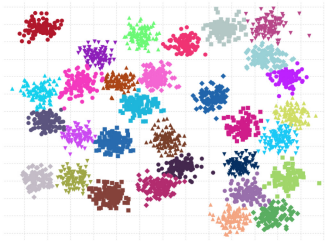
(a) *Chameleon 2* on *3-spiral*



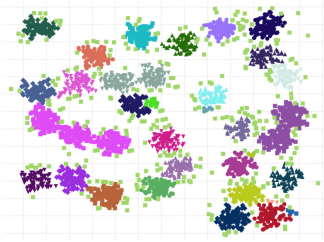
(b) *DBSCAN* on *3-spiral*



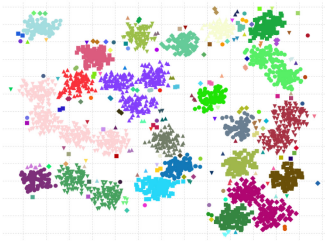
(c) *HC-SL* on *3-spiral*



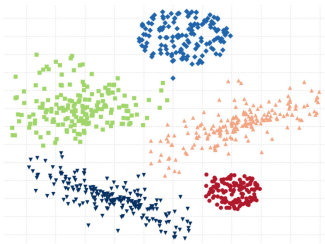
(d) *Chameleon 2* on *d31*



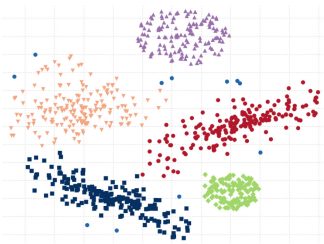
(e) *DBSCAN* on *d31*



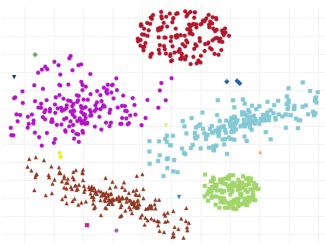
(f) *HC-SL* on *d31*



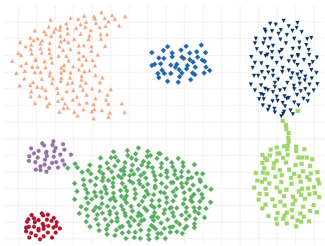
(g) *Chameleon 2* on *ds-850*



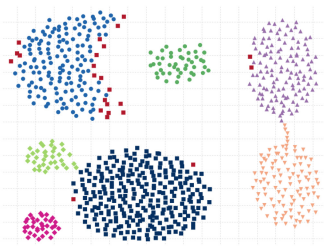
(h) *DBSCAN* on *ds-850*



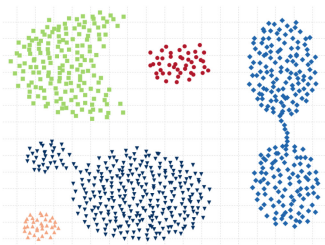
(i) *HC-SL* on *ds-850*



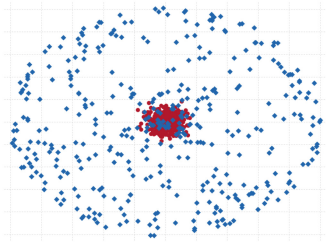
(j) *Chameleon 2* on *aggregation*



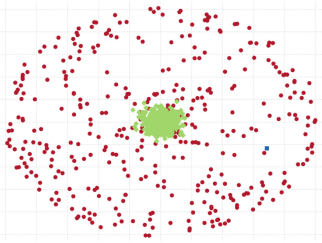
(k) *DBSCAN* on *aggregation*



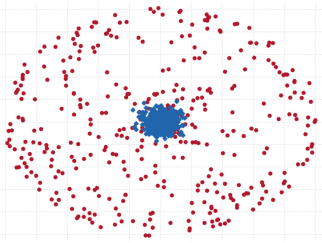
(l) *HC-SL* on *aggregation*



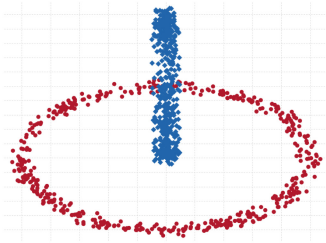
(a) *Chameleon 2* on *atom*



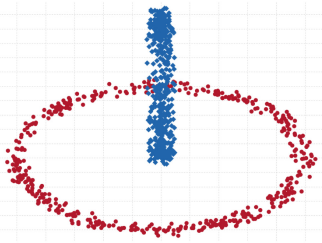
(b) *DBSCAN* on *atom*



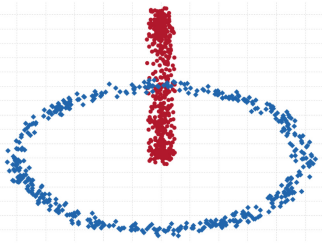
(c) *HC-SL* on *atom*



(d) *Chameleon 2* on *chainlink*

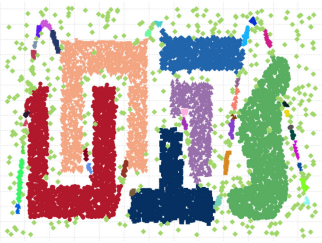


(e) *DBSCAN* on *chainlink*



(f) *HC-SL* on *chainlink*

(g) *Chameleon 2* on *cluto-t4-8k*



(h) *DBSCAN* on *cluto-t4-8k*

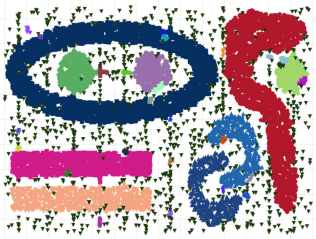
(i) *HC-SL* on *cluto-t4-8k*

(j) *Chameleon 2* on *cluto-t5-8k*

(k) *DBSCAN* on *cluto-t5-8k*

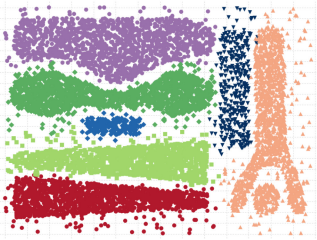
(l) *HC-SL* on *cluto-t5-8k*

(a) *Chameleon 2* on *cluto-t7-10k*

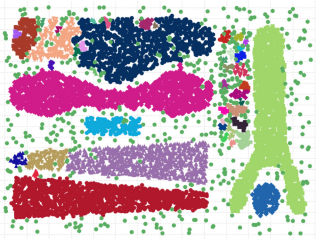


(b) *DBSCAN* on *cluto-t7-10k*

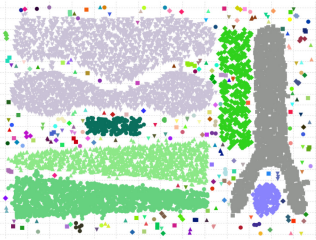
(c) *HC-SL* on *cluto-t7-10k*



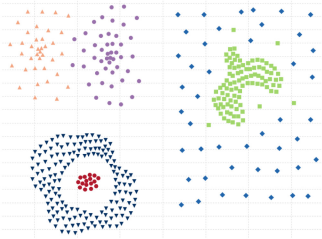
(d) *Chameleon 2* on *cluto-t8-8k*



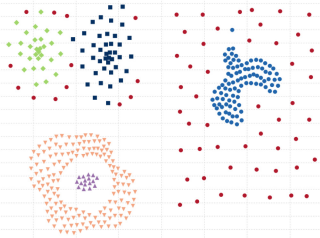
(e) *DBSCAN* on *cluto-t8-8k*



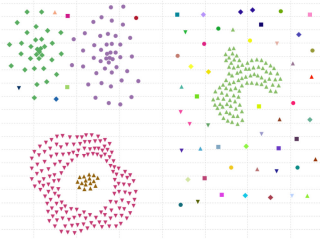
(f) *HC-SL* on *cluto-t8-8k*



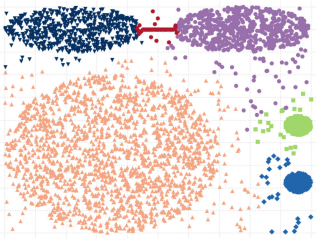
(g) *Chameleon 2* on *compound*



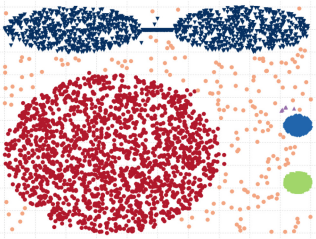
(h) *DBSCAN* on *compound*



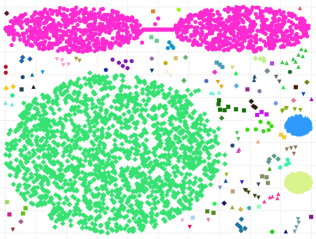
(i) *HC-SL* on *compound*



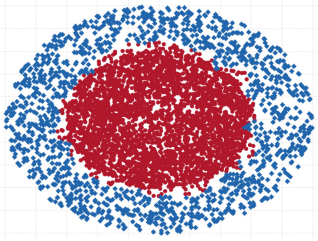
(j) *Chameleon 2* on *cure-t2-4k*



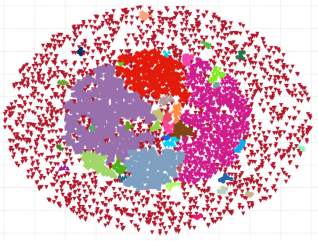
(k) *DBSCAN* on *cure-t2-4k*



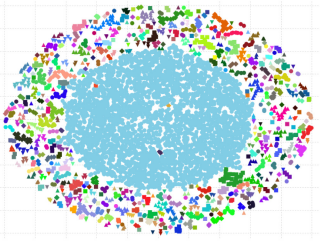
(l) *HC-SL* on *cure-t2-4k*



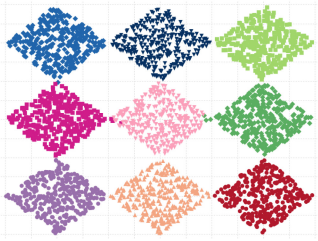
(a) *Chameleon 2* on *dense-disk-5k*



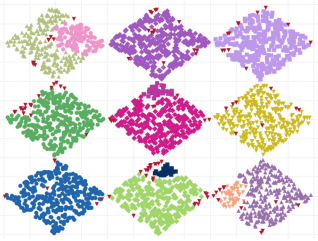
(b) *DBSCAN* on *dense-disk-5k*



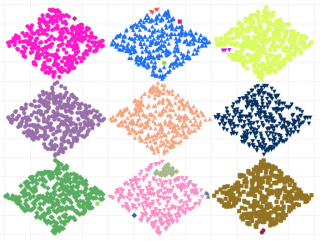
(c) *HC-SL* on *dense-disk-5k*



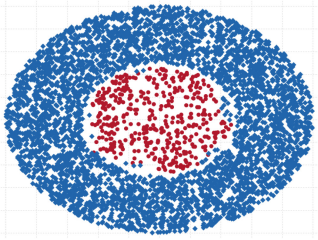
(d) *Chameleon 2* on *diamond9*



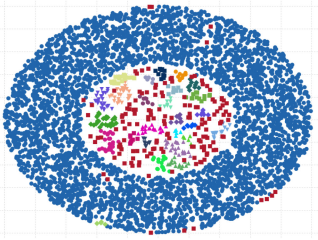
(e) *DBSCAN* on *diamond9*



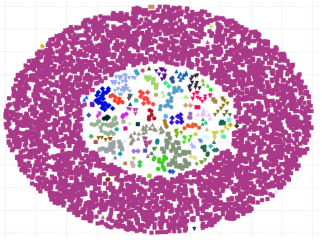
(f) *HC-SL* on *diamond9*



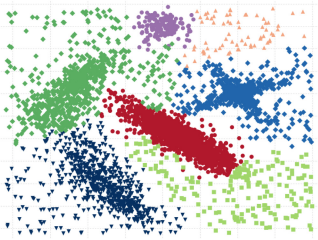
(g) *Chameleon 2* on *disk-in-disk*



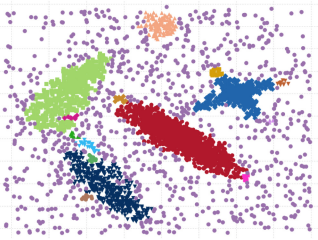
(h) *DBSCAN* on *disk-in-disk*



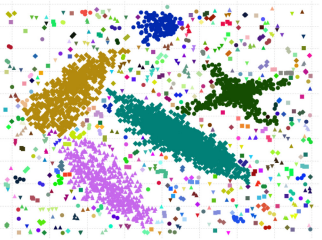
(i) *HC-SL* on *disk-in-disk*



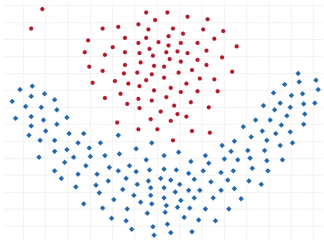
(j) *Chameleon 2* on *dpb*



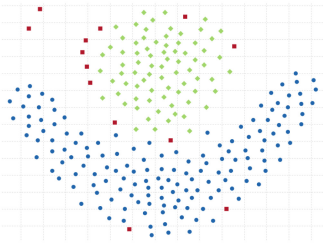
(k) *DBSCAN* on *dpb*



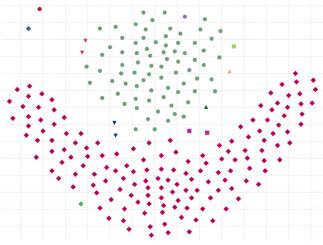
(l) *HC-SL* on *dpb*



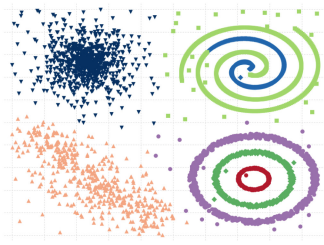
(a) *Chameleon 2* on *flame*



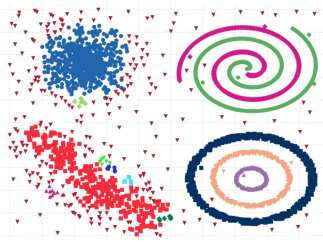
(b) *DBSCAN* on *flame*



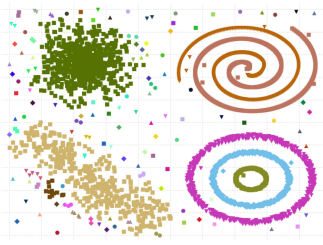
(c) *HC-SL* on *flame*



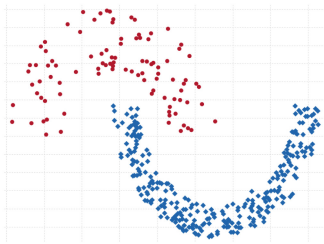
(d) *Chameleon 2* on *impossible*



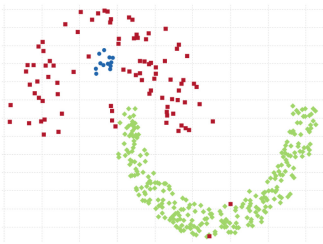
(e) *DBSCAN* on *impossible*



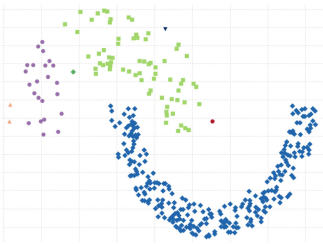
(f) *HC-SL* on *impossible*



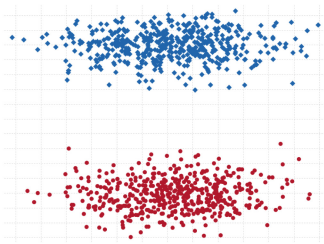
(g) *Chameleon 2* on *jain*



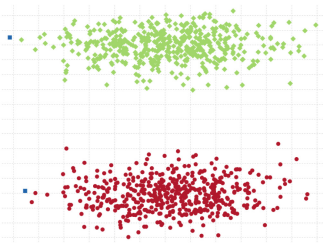
(h) *DBSCAN* on *jain*



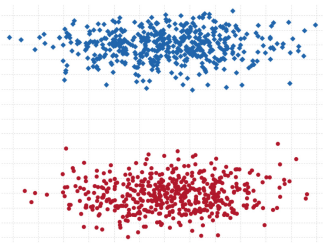
(i) *HC-SL* on *jain*



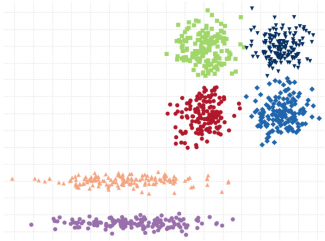
(j) *Chameleon 2* on *long1*



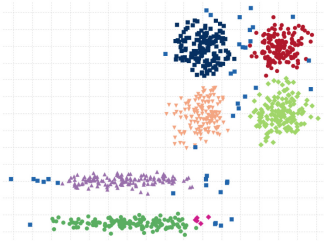
(k) *DBSCAN* on *long1*



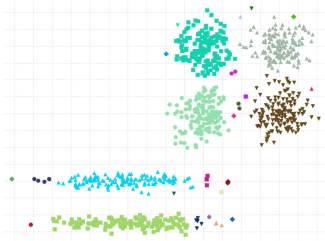
(l) *HC-SL* on *long1*



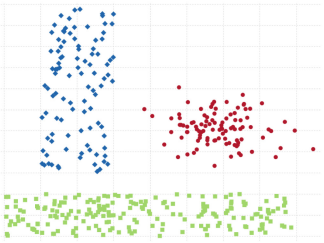
(a) *Chameleon 2* on *longsquare*



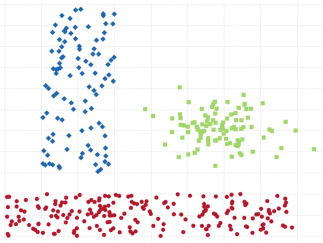
(b) *DBSCAN* on *longsquare*



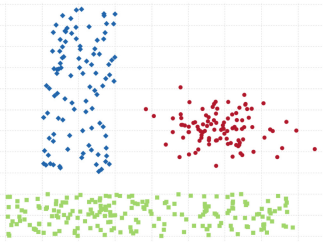
(c) *HC-SL* on *longsquare*



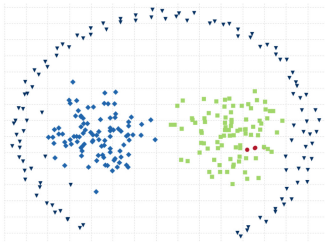
(d) *Chameleon 2* on *lsun*



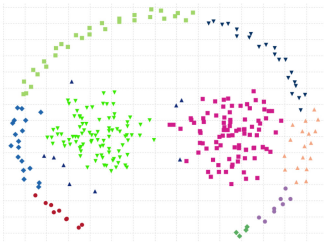
(e) *DBSCAN* on *lsun*



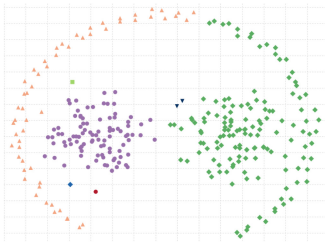
(f) *HC-SL* on *lsun*



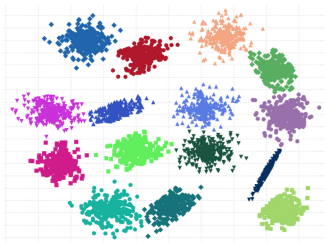
(g) *Chameleon 2* on *pathbased*



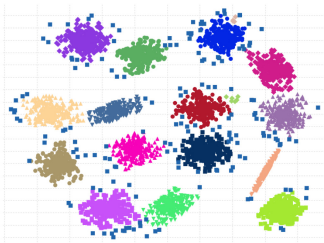
(h) *DBSCAN* on *pathbased*



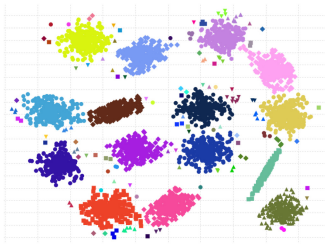
(i) *HC-SL* on *pathbased*



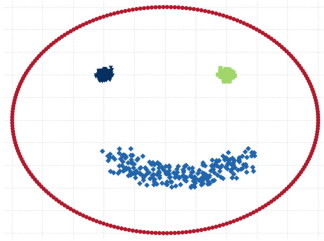
(j) *Chameleon 2* on *s-set1*



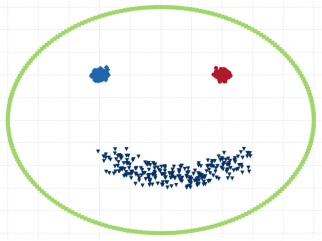
(k) *DBSCAN* on *s-set1*



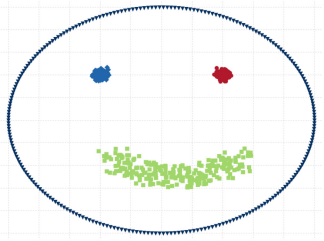
(l) *HC-SL* on *s-set1*



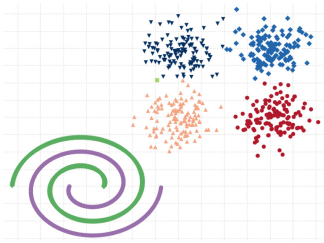
(a) *Chameleon 2* on *smile1*



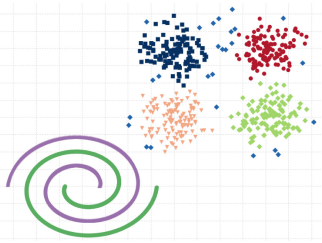
(b) *DBSCAN* on *smile1*



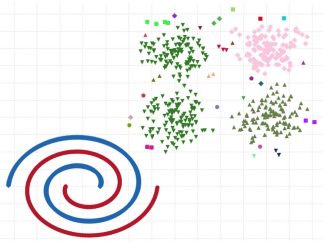
(c) *HC-SL* on *smile1*



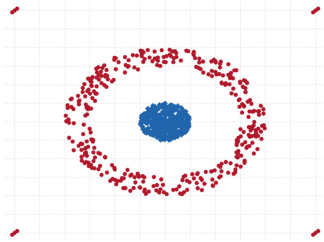
(d) *Chameleon 2* on *spiralsquare*



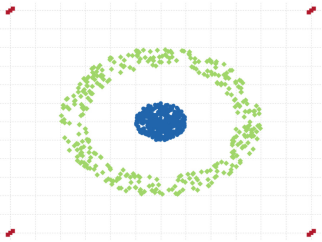
(e) *DBSCAN* on *spiralsquare*



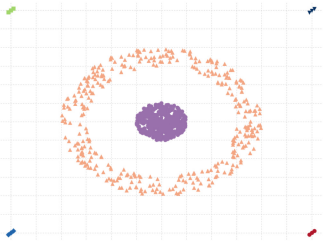
(f) *HC-SL* on *spiralsquare*



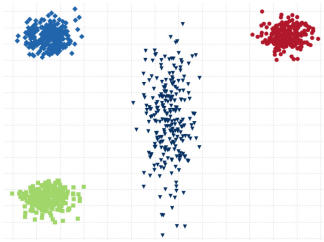
(g) *Chameleon 2* on *target*



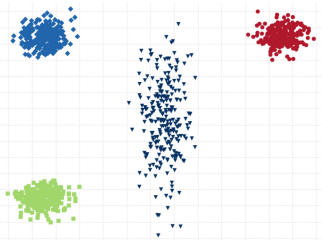
(h) *DBSCAN* on *target*



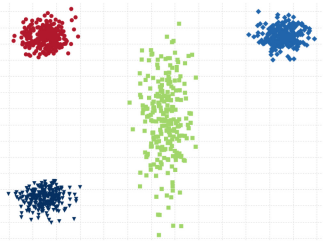
(i) *HC-SL* on *target*



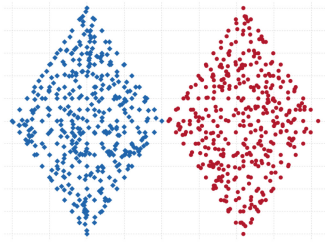
(j) *Chameleon 2* on *triangle1*



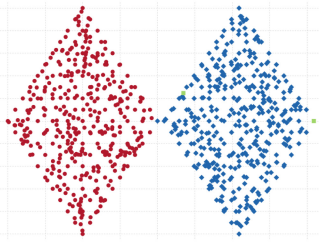
(k) *DBSCAN* on *triangle1*



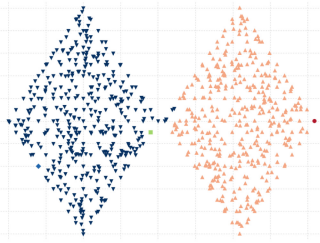
(l) *HC-SL* on *triangle1*



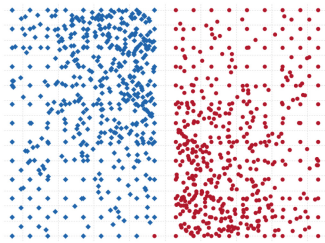
(a) *Chameleon 2* on *twodiamonds*



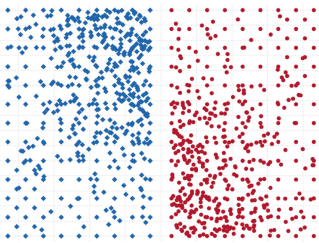
(b) *DBSCAN* on *twodiamonds*



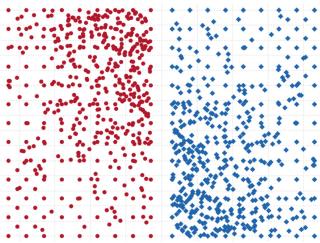
(c) *HC-SL* on *twodiamonds*



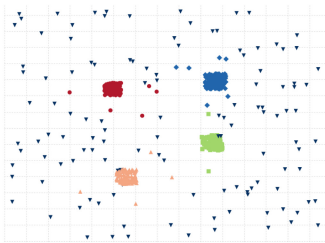
(d) *Chameleon 2* on *wingnut*



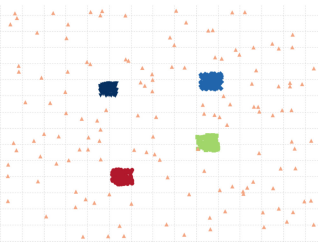
(e) *DBSCAN* on *wingnut*



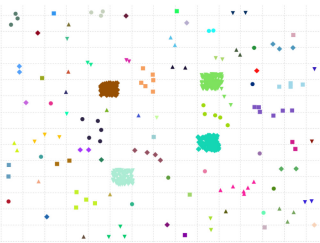
(f) *HC-SL* on *wingnut*



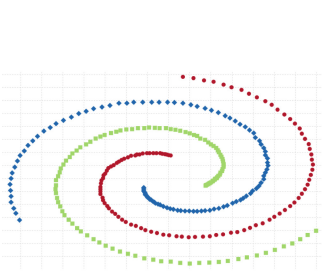
(g) *Chameleon 2* on *zelnik4*



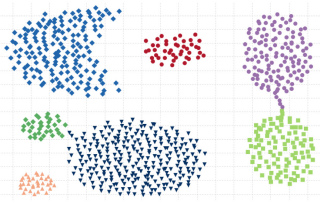
(h) *DBSCAN* on *zelnik4*



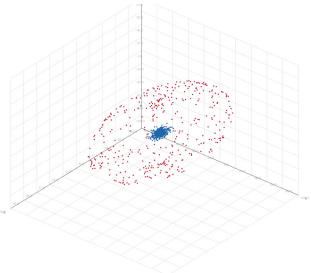
(i) *HC-SL* on *zelnik4*



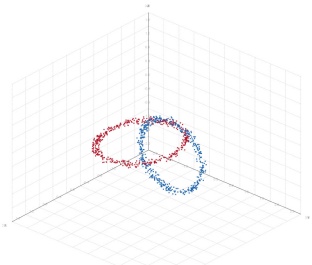
(a) Dataset *3-spiral*.



(b) Dataset *aggregation*.



(c) Dataset *atom*.

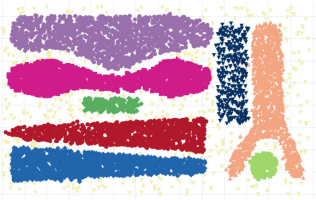


(d) Dataset *chainlink*.

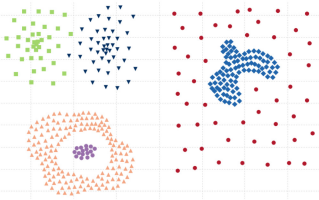
(e) Dataset *cluto-t4.8k*.

(f) Dataset *cluto-t5.8k*.

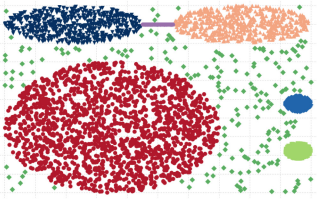
(g) Dataset *cluto-t7.10k*.



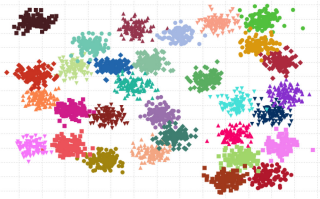
(h) Dataset *cluto-t8.8k*.



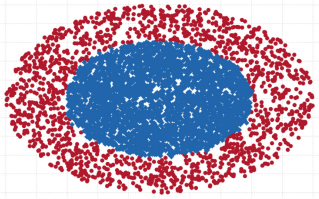
(i) Dataset *compound*.



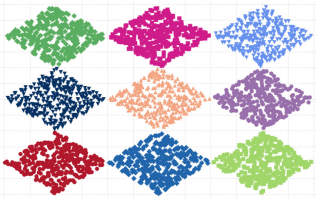
(j) Dataset *cure-t2-4k*.



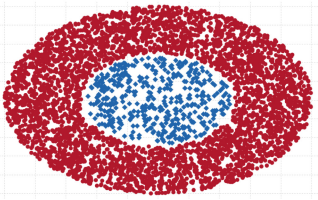
(k) Dataset *D31*.



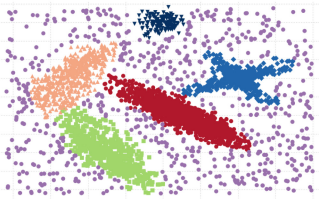
(l) Dataset *dense-disk-5k*.



(m) Dataset *diamond9*.

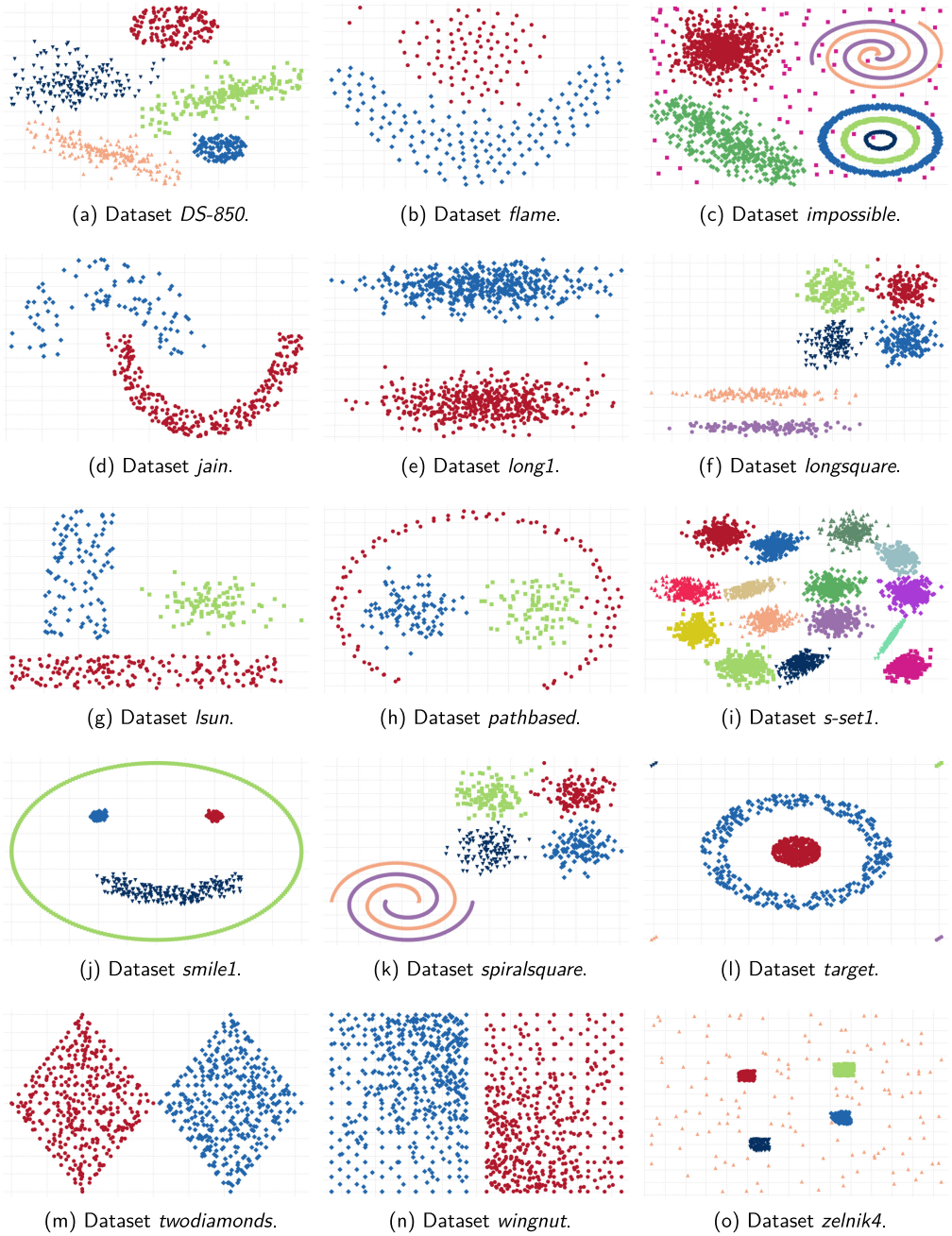


(n) Dataset *disk-in-disk*.



(o) Dataset *dpb*.

Fig. 1. Visualization of datasets used in experiments with ground truth assignments.



(a) Dataset *DS-850*. (b) Dataset *flame*. (c) Dataset *impossible*.

(d) Dataset *jain*. (e) Dataset *long1*. (f) Dataset *longsquare*.

(g) Dataset *lsun*. (h) Dataset *pathbased*. (i) Dataset *s-set1*.

(j) Dataset *smile1*. (k) Dataset *spiralsquare*. (l) Dataset *target*.

(m) Dataset *twodiamonds*. (n) Dataset *wingnut*. (o) Dataset *zelnik4*.

Fig. 2. Visualization of datasets used in experiments with ground truth assignments, second part.

REFERENCES

[1] A. Fred and A. Jain. 2003. Robust data clustering. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2. Citeseer.

[2] Tarald O. Kvålseth. 1987. Entropy and correlation: Some comments. *IEEE Trans. Syst. Man Cybern.* 17, 3 (1987), 517–519.

[3] Peter Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20, 1 (1987), 53–65. DOI: https://doi.org/10.1016/0377-0427(87)90125-7

[4] Alexander Strehl and Joydeep Ghosh. 2002. Cluster ensembles – A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* 3 (December 2002), 583–617.