Operations Research Letters 55 (2024) 107137

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

CABRA: Clustering algorithm based on regular arrangement

Jorge C-Rella [a,b,*]

[a] Department of Mathematics, Research Group MODES, CITIC, University of A Coruña, A Coruña, Spain
[b] Department of Risks, ABANCA Financial Services, A Coruña, Spain

ARTICLE INFO

Article history:
Received 9 October 2023
Received in revised form 16 November 2023
Accepted 6 June 2024
Available online 12 June 2024

Keywords:
Clustering
Segmentation
Classification
Outlier detection

ABSTRACT

Clustering is an unsupervised learning technique for organizing complex datasets into coherent groups. A novel clustering algorithm is presented, with a simple grouping concept depending on only one hyperparameter, which makes it suitable for further extensions to any topology and space. It is compared to state-of-the-art algorithms, overall achieving a better performance independently on the structure and complexity of the data, making the proposed algorithm a valuable tool for real applications such as market segmentation, sentiment analysis and anomaly detection.

1. Introduction

Clustering is an unsupervised learning technique for organizing populations into coherent groups, extracting some inherent structure in the data in terms of meaningful subclasses. Resulting clusters should not only have good statistical properties (compact, well-separated, connected, and stable) [3], but also give relevant and useful results. Effective clustering algorithms can facilitate numerous applications, including gen clustering, customer grouping, pattern recognition, image analysis and data compression, to name a few.

There are two basic types of clustering algorithms [8]: partitioning and hierarchical. The former constructs a partition of a database into k clusters, being k a parameter usually not available beforehand. The partitioning algorithm typically starts with an initial partition and then uses an iterative strategy to optimize an objective function. Finally, each cluster is represented by its gravity center, e.g. the mean point of the group [10]. Some of the most popular partitioning clustering algorithms include k-means [10], k-medoids, and Gaussian mixture models [2]. On the other hand, hierarchical algorithms create a hierarchical decomposition of the data represented by a dendrogram defined by a distance metric, until each subset consists of only one object. The dendrogram can either be created by merging (agglomerative approach) or dividing clusters (divisive approach) at each step. In contrast to partitioning algorithms, hierarchical algorithms do not need the number of clusters, k, as an input. However, a termination condition (pruning) has to be defined indicating when the merge or division process should be terminated, which is usually not straightforward.

State-of-the-art clustering notions include small distances between cluster members [13], dense areas of the data space [6,1], centroid models [10], connectivity models [8,11] or particular statistical distributions [2]. Regarding the challenges to overcome, appropriate hyperparameters (such as the number of clusters, the underlying distribution of the data, a density threshold or the concept of closeness in terms of a metric or a number of observations) are often not known in advance. The estimation of these hyperparameters is not straightforward as it is metric-dependent and subject to some degree of subjectivity. Therefore it can be a daunting task determining the most appropriate algorithm, which depends on the specific application and the characteristics of the data.

In this paper it is considered the hard clustering approach, meaning that each point belongs exactly to one and only one cluster. The proposed *Clustering Algorithm Based on Regular Arrangement* (CABRA) extends density and connectivity-oriented previous approaches considering a new grouping concept. It starts assuming the null hypothesis that the data is generated by an uniform distribution on each dimension. Then, points that are at a distance smaller than a quantile of the distribution of the minimum distance are grouped together. This quantile is the only hyperparameter of the algorithm, which autonomously determine the number of clusters. Furthermore, as the algorithm relies on a null hypothesis and not on characteristics of the data or initial values, it is robust to noise and provides stable results. Consequently, the proposed algorithm has a simple parameter tuning based on a straightforward grouping logic, which makes it suitable for fur-

* Correspondence to: Department of Mathematics, Research Group MODES, CITIC, University of A Coruña, A Coruña, Spain.
E-mail address: jorge.crella@udc.es.

https://doi.org/10.1016/j.orl.2024.107137

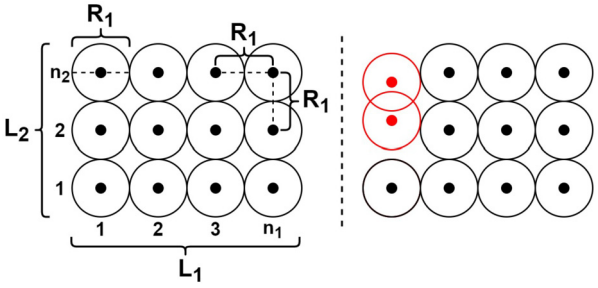

Fig. 1. Regular arrangement assumption.

ther generalizations and applications in a broad range of clustering tasks.

The paper is organized as follows. Section 2 introduces the proposed clustering algorithm, explaining its grouping logic, steps and some refinements. Section 3 introduces the state-of-the-art clustering algorithms considered to compare the CABRA performance and summarize the results obtained over various commonly used datasets. Section 4 gives conclusions and further extensions of the proposed algorithm.

2. Clustering based on regular arrangement

The algorithm starts from the null hypothesis that the data has an independent uniform distribution across all its dimensions. This can be understood as the stochastic version of all the points being as far apart as possible from each other in the data support, case where it would be no clustering structure except from the trivial groupings. The left graph in Fig. 1 represents this setting. Under the null hypothesis, the underlying distribution of the data is known, so it can be calculated the distribution of the minimum distance between any two points. Afterwards, a quantile of this distribution is taken as distance threshold for clustering. Two observations closer than this quantile are considered to break the null hypothesis and are assigned to the same group. The quantile, q, is a hyperparameter that needs to be tuned. Depending on the data, a high value of q can be too strict and the algorithm would estimate only one cluster, or vice versa. Thus, the algorithm has flexibility depending on the data with just one hyperparameter and a straightforward grouping criteria.

It is assumed that the uniform distribution hypothesis is extended across dimensions, i.e. there is a direct relation between the support of the data in a dimension j, L_j, and the number of different points in that dimension, n_j. As example, consider a two-dimensional data set of size $n = 12$ and support $L_1 \times L_2 = [0, 4] \times [0, 3]$. Then, it is assumed that there are $n_1 = 4$ different values in the first dimension and $n_2 = 3$ in the second one, like a 4×3 grid as represented in the left graph in Fig. 1. Note that $n = n_1 n_2$. Under this setting, it is easy to realize that the minimum distance between points is $R_1 = L_1/(n_1 - 1) = L_2/(n_2 - 1)$. Note that the minimum distance is the same independently of the considered dimension. Thus, the distribution of the minimum distance between two points in the multidimensional space is the same as for the distribution in one of the dimensions, and the threshold estimation can be addressed as a one-dimensional problem. Extending this setting to its stochastic version where the data is independently uniformly distributed on each dimension, the algorithm grouping threshold is obtained.

To construct the algorithm, it is needed to know the distribution of the minimum distance between two points of an uniform distributed sample. First, the number of different points per dimension, n_j, has to be estimated when there are more than two dimensions. This is achieved extending the logic explained above for Fig. 1. For a d-dimensional sample $\mathcal{X} = \{\mathbf{x}_i = (x_{i1}, \cdots, x_{id})\}_{i=1}^{n}$, let n_j be the number of different values in the j-th dimension and $[0, L_j]$ the support of the j-th dimension. It is assumed $\min_i x_{ij} = 0$ without loss of generality. Thus, the number of different points in the j-th dimension is:

$$n_j = \frac{L_j}{\sqrt[d]{\frac{\prod_{j=1}^{d} L_j}{n}}} \tag{1}$$

Note that $\prod_{j=1}^{d} n_j = n$. The second step consists in obtain the grouping threshold, R_q. By assumption, there is a direct relation between n_j and L_j. Consequently, the expected distribution of the minimum distance between two points of a d-dimensional sample is the distribution of the minimum distance in each of the dimensions (as explained for the example in Fig. 1). Thus, without loss of generality, the grouping threshold, R_q, is calculated over the first dimension ($j = 1$). Considering a different dimension, the same value for R_q will be obtained as previously mentioned. Consider the distribution of the minimum distance between two points from an univariate uniformly-distributed sample, already developed in [5]. Let $\mathcal{U} = \{u_i\}_{i=1}^{n}$, whit $u_i \sim U[0, 1]$ and $M = \min_{1 \le i \ne j \le n} |u_i - u_j|$. Then,

$$P(M > m) = (1 - (n - 1)m)^n \tag{2}$$

As we are considering an uniform distribution, the same results hold multiplying by L when $u_i \sim U[0, L]$. Considering equation (2), the q quantile of the distribution of $M = \min_{1 \le i \ne j \le n_1} |u_i - u_j|$ with $u_i \sim U[0, L_1]$ is given by:

$$R_q = \frac{1 - (1 - q)^{\frac{1}{n_1}}}{n_1 - 1} L_1 \tag{3}$$

where $q \in [0, 1]$. The quantile R_q is taken as distance grouping threshold. To break the null hypothesis, two points have to be at a shorter distance than R_q, case when they are considered to belong to the same group. Extending this reasoning, the CABRA algorithm is developed.

Note that the grouping threshold, R_q, depends on the data support. Thus, outliers increasing the range L_j in any of the dimensions can affect the estimated threshold and thus the algorithm output. Depending on the size of the outlier, this can result in an inconsistent or impractical clustering. For example, an outlier in the first dimension would increase the value of R_q, thus resulting in a more relaxed grouping threshold that could even end with just one single estimated cluster. To address this, it can be of practical interest to consider the points contained in some quantiles of the support e.g. 0.5% and 99.5% in order to address outliers. An initial exploratory analysis of the data can help in this stage. In addition, depending on the dimension of the data, can be of interest to omit outliers from a multidimensional perspective.

2.1. Algorithm

Intuitively, the idea of the algorithm is to consider a ball with radius R_q as defined in equation (3) centered in each point of the data sample and group together those points whose balls intersect. Then, the centroids of the formed groups are calculated and used as the as the data set for the next step. The radius R_q (3) is recalculated over the new centroids sample and the clusters are constructed analogously, with the same ball intersection logic. If two centroids are are clustered together in this step, all points represented by these centroids are considered to be in the same cluster. This process is iterated until no more clusters are obtained or some stopping criteria are met.

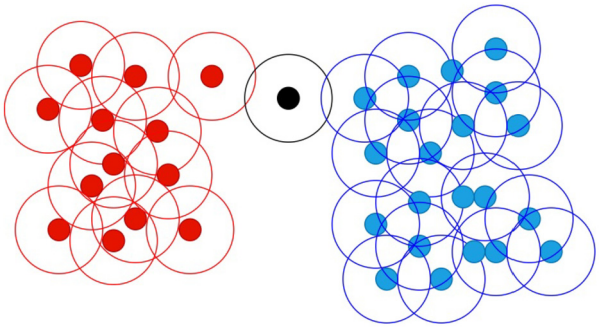

Fig. 2. Data set with superimposed balls of radius R_q where a single point (black dot) produces one cluster and the dense clustering criteria produces two separate clusters (red and blue dots) plus a noise point (black dot). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

A toy example with three steps is shown in Fig. 3. In the left, it is represented the two-dimensional data considered, with the assigned clustering in each step represented with different colors. In the right, it is represented the centroids at that iteration, with the corresponding ball with radius R_q superimposed calculated at that step. The centroids are calculated as the median point of all the points assigned to a cluster, as can be seen in the second and third step. Note that the radius R_q grows at each step as it is considered the same space but with less points. In the left, when a centroid is linked with another one, all the points defining that centroid are assigned to the same cluster (second step). In the last step no new cluster is found and the algorithm stops.

The clustering algorithm is described in detail in Algorithm CABRA. The stopping criterion considered is the reduction of the silhouette coefficient from one step to another as defined in [12], but any other criteria could be considered. Centroids are calculated as the mean of all points assigned to the same cluster, but any criteria can be used, such as median or mode.

With this grouping criterion, two different clusters may be connected by a single point that is connected to a point on each cluster. An example is shown in Fig. 2, where the effect of a single point (black dot) is shown. A refinement of the introduced algorithm is developed to avoid the effect of this single "link" point. For a point to be introduced into a cluster, two different points of the cluster must be at a smaller distance than R_q from the point, i.e. they must share at least one other connection. This eliminates "weak" connections just because two points are close. This is inspired by the DBSCAN algorithm [6], so "denser" clusters are obtained. The code in Algorithm CABRA shows the dense version of the algorithm. In the example in Fig. 2, the dense approach estimates two different clusters and one point that does not belong to any cluster, which seems more reasonable. In some applications it might be of interest to have this clustering criteria and in others to get a single cluster, for which both algorithms are considered useful depending on the problem. It can even be considered a stricter clustering criterion, requiring more than two connected points to consider them clustered together. However, since the main drawback is the presence of a single link point grouping different clusters, which is considered a weak and inconsistent grouping criterion, further restrictions are not considered. Thus, throughout the paper, the dense version of the algorithm is considered, which omits single link points as the black dot in Fig. 2, as this is the criterion most likely to be of interest in practice.

In order to speed up the iterative process, it is of practical utility to consider one value for q in the first step and another one for the remaining steps. This way, in the first step, there can be grouped several points reducing the sample size in the successive steps. It could even be explored a different value of q in each step, selecting its value by cross-validation for example. Nevertheless, with this simple approach of considering one value of q in the first step and another one in the successive steps, a reduction in the computational burden has been observed without compromising the results, even improving them in some scenarios thanks to the extra flexibility. Thus, further extensions although interesting are not explored in this paper for brevity sake.

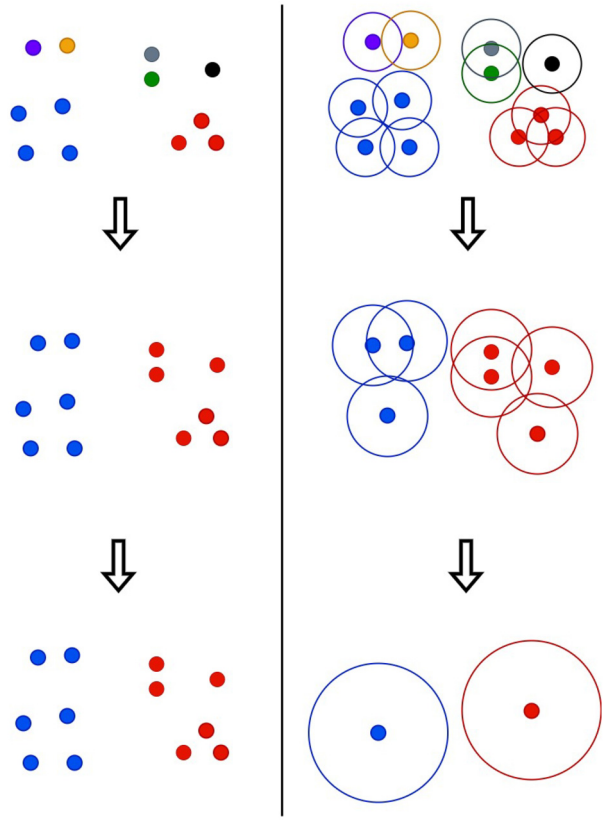

Fig. 3. Two-dimensional example of CABRA algorithm functioning. In the left it is represented with colors the assigned cluster to each point at each step. In the right, it is represented the considered centroids at each step with a ball of the corresponding radius R_q superimposed.

Algorithm CABRA: Clustering algorithm based on regular arrangement.

Data $\mathcal{X} = \{\mathbf{x}_i = (x_{i1}, \cdots, x_{id})\}_{i=1}^{n}$
Input q parameter
$\mathbf{g}^0 \leftarrow (1, 2, \cdots, n)$, the vector indicating the clusters in step 0. Each point is assigned to a different cluster in this first step
$t \leftarrow 0$
$Sil_{old} \leftarrow -2$
$Sil \leftarrow Silhouette(\mathbf{g}^0, \mathcal{X})$
while $Sil > Sil_{old}$ **do**
  $Sil_{old} \leftarrow Sil$
  $\mathcal{X}^t \leftarrow$ medioids of the clusters given by g^t
  $L_j = \max_i x_{i1}^t - \min_i x_{i1}^t,\ j = 1, \cdots, d$
  $n_1 = \frac{L_1}{\sqrt[d]{\frac{\prod_{j=1}^{d} L_1}{n}}}$
  $R_q = \frac{1-(1-q)^{\frac{1}{n_1}}}{n_1 - 1} L_1$
  Assign the same cluster in the vector of clusters in step t, $\mathbf{g}^t$, to the instances in $\mathcal{X}^t$ that are at a distance smaller than R_q and share another connection (a different point at distance smaller than R_q)
  $\mathbf{g} \leftarrow \mathbf{g}^t \circ \mathbf{g}^{t-1} \circ \cdots \circ \mathbf{g}^0$, i.e., the cluster assigned to each original point in $\mathcal{X}$ is the assigned to its medioid in the last iteration and so on
  $Sil \leftarrow Silhouette(\mathbf{g}, \mathcal{X})$
  $t \leftarrow t + 1$
end while
Output Clustering groups defined by g

3. Performance study

The performance of the proposed algorithm is compared with the most outstanding state-of-the-art clustering algorithms. Several widely used datasets are considered to evaluate the advantages and disadvantages of different algorithms depending on the structure and underlying distribution of the data. All datasets are two-dimensional in order to easily present the obtained clusters, but the proposed algorithm can be extended to any multi-dimensional dataset since it relies only on the distance between points. All samples (except *Skewed* and *Symetric*) consist of labeled data, so an evaluation of the clustering and classification performance is performed.

There are a number of metrics that can be used to assess clustering quality. However, there is no single, universally accepted way to evaluate clustering algorithms. Consequently, in this study there are considered different performance metrics, including external evaluation using the true labels and internal evaluation using cohesion and separation measures. Specifically, there are considered the Purity, Adjusted Rand index (ARI), Silhouette Coefficient (SC), Calinski-Harabasz index (CHI) and Dunn index (DI). Purity is the proportion of correctly matched classes and cluster labels. The ARI is a measure of the similarity between the estimated groups and the true labels, adjusted for the chance grouping. The SC measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The CHI is based on the principle of variance ratio, calculated between the within-cluster diffusion and the between cluster dispersion. The DI captures the intuition that dense clusters that are well-spaced from each other are a 'good' clustering. For all the metrics, the greater the value, the better the clustering.

The considered algorithms are: UPGMA [13], K-means (KM) [10], DBSCAN [6], OPTICS [1], Mean-shift (MS) [4,7], partitioning around medoids (PAM) [9], fuzzy clustering (FC) as proposed in [8] and a Gaussian mixture (GM) model [2]. The dense CABRA algorithm is run with different values of q for the first and successive steps and the one obtaining the greatest Purity (or SC when labels are not available) is selected. It is considered the usual euclidean distance, but the algorithm can be extended to any metric and space. The clusters obtained with each algorithm over the different data sets are represented in Figs. 4-15 and the results summarized in Table 1.

It can be seen the good performance of CABRA regardless of the structure and noise in the data. The worst Purity is obtained in the *Path* and *Flame* datasets (89% and 94% respectively), but it is attained a performance similar to OPTICS, the winning algorithm. In the density-separable datasets as *R15, Network* and *Basic4*, UPGMA, KM, FC, PAM and GM obtain the better results, but CABRA attains a similar performance as well. When the data shows some structure or is not well separated, these algorithms have a worse performance, something that does not occur with CABRA. DBSCAN and OPTICS achieve overall good results, but CABRA always attains a similar or higher Purity as well. Regarding the datasets without labels, even though CABRA does not obtain the best metrics, it visually obtains the best results in view of Fig. 11 and 12 after GM, which does not win in any metric either. Depending on the dataset, different algorithms obtain the best results depending on the structure, noise, and complexity of the data. However, CABRA obtains in all the datasets either the best results or comparable to the winning algorithm, something that no other algorithm achieves. Taking this into account, it is considered that CABRA is the algorithm with the best overall performance, consistently obtaining reasonable clusters regardless of the structure and complexity of the data. Adding that it can handle outliers, the stability that it offers by not depending on any starting point, and that it does not need to know the number of clusters in advance, it is concluded that CABRA is an algorithm that offers all the advantages of the state-of-the-art algorithms in a simpler and more intuitive way.

4. Conclusions

A new clustering algorithm is proposed extending density and connectivity-oriented previous approaches, taking its distinct advantages and condensing them with a straightforward grouping criteria. It only relies on one hyperparameter, the quantile q. Thus, it is not needed to estimate the number of clusters beforehand neither perform any pruning step, making easier the algorithm tuning. Outliers can be addressed considering some quantiles of the data support when constructing the distance grouping threshold R_q (3). In addition, the algorithm does not depend on starting points, for which stable results are obtained. Lastly, it only depends on a distance metric over a multidimensional feature space, for which the algorithm can be extended to any topology and space.

Several state-of-the-art algorithms along with CABRA were evaluated on a variety of datasets, with a wide range of structures, shapes and complexity. All algorithms have a good performance over certain settings, but have problems when considering other data structures. Nevertheless, CABRA has shown to consistently obtain coherent clusters as shown in Figs. 4-15 and to overall outperform previous approaches as illustrated in Table 1. In all the settings there are obtained results better than or similar to the best state-of-the-art algorithm, concluding with the satisfactory performance of the proposed algorithm independently on the setting and problem complexity. Adding the adaptability of the algorithm to any setting and space, the straightforward grouping logic and how easy it is to tune the model, CABRA is considered a promising algorithm for clustering problems.

Declaration of competing interest

None.

Data availability

Data will be made available on request.

Acknowledgements

This research has been financed by the Grant PID2020-113578-RB-I00, funded by MCIN/AEI/10.2039/501100011033/. It has also been supported by the Xunta de Galicia (Grupos de Referencia Competitiva ED431C-2020/14) and by CITIC that is supported by Xunta de Galicia, convenio de colaboración entre la Consellería de Cultura, Educación, Formación Profesional e Universidades y las universidades gallegas para el refuerzo de los centros de investigación del Sistema Universitario de Galicia (CIGUS). The first author was financed by the Axencia Galega de Innovación Industrial PhD Grant 14-IN606D-2021-2607768.

References

[1] M. Ankerst, M. Breunig, P. Kröger, J. Sander, OPTICS: ordering points to identify the clustering structure, SIGMOD Rec. 28 (1999) 49–60, https://doi.org/10.1145/304182.304187.
[2] J.D. Banfield, A.E. Raftery, Model-based Gaussian and non-Gaussian clustering, Biometrics 49 (1993) 803–821, http://www.jstor.org/stable/2532201.
[3] G. Brock, V. Pihur, S. Datta, S. Data, clvalid, an r package for cluster validation, J. Stat. Softw. 25 (2008), https://doi.org/10.18637/jss.v025.i04.

Table 1
Results for the different datasets and algorithms introduced.

Dataset	Algorithm	Purity	ARI	SC	CHI	DI	Dataset	Algorithm	Purity	ARI	SC	CHI	DI
Aggregation	UPGMA	99.49	99.13	47.36	11.43	4.67	Multi	UPGMA	50.64	27.04	38.86	8.17	1.68
	KM	75.76	71.14	46.20	12.79	3.40		KM	42.73	21.05	38.30	10.69	0.97
	DBSCAN	82.49	80.58	28.69	5.33	4.79		DBSCAN	60.18	43.63	37.18	3.77	4.23
	OPTICS	95.30	94.53	37.25	7.01	0.87		OPTICS	80.73	78.77	26.13	4.10	1.23
	PAM	81.47	68.14	46.60	12.71	3.25		PAM	42.27	23.89	41.82	11.22	1.20
	FC	68.65	57.56	32.38	9.75	1.25		FC	38.82	27.02	28.82	8.19	0.68
	GM	81.47	79.09	45.23	11.76	3.45		GM	52.82	34.93	29.27	8.59	0.49
	CABRA	98.86	97.68	47.62	11.61	3.58		CABRA	95.27	94.95	-6.77	0.95	2.52
Flame	UPGMA	80.00	35.70	35.89	1.40	4.38	Basic4	UPGMA	99.93	99.79	45.37	37.93	5.23
	KM	82.92	43.12	37.40	1.52	3.40		KM	99.40	98.14	45.34	38.02	0.90
	DBSCAN	65.42	2.60	21.73	0.10	4.95		DBSCAN	99.73	99.62	41.58	25.23	4.53
	OPTICS	95.42	90.21	25.67	0.48	3.84		OPTICS	95.03	92.27	32.84	16.92	0.53
	PAM	85.00	48.80	35.99	1.36	3.80		PAM	99.23	97.61	45.32	37.97	1.23
	FC	84.58	47.63	35.97	1.36	3.86		FC	98.27	94.66	45.06	37.87	0.23
	GM	74.58	23.62	35.01	1.37	4.47		GM	99.70	99.06	45.39	37.99	0.90
	CABRA	94.17	81.76	17.02	0.37	4.07		CABRA	99.70	99.54	-13.61	7.64	2.97
Path	UPGMA	75.33	47.17	50.89	3.02	5.10	Boxes2	UPGMA	69.03	40.71	35.13	20.67	1.39
	KM	76.00	47.97	50.92	3.04	4.57		KM	44.87	24.89	38.70	24.21	0.49
	DBSCAN	36.67	0.00	-100.00	0.00			DBSCAN	82.03	68.46	35.81	10.53	0.75
	OPTICS	89.33	71.86	10.38	0.22	1.23		OPTICS	93.43	88.54	-16.36	3.71	0.18
	PAM	75.67	47.57	50.93	3.04	3.23		PAM	54.37	39.90	39.85	25.42	0.59
	FC	72.33	42.42	47.69	2.70	1.96		FC	61.10	25.04	38.51	24.69	0.46
	GM	70.67	43.04	47.15	2.38	2.09		GM	79.37	46.33	28.05	14.99	0.04
	CABRA	88.67	72.21	-10.28	0.12	2.63		CABRA	99.37	98.85	-30.10	2.19	0.99
R15	UPGMA	99.50	98.93	75.22	48.60	18.65	Network	UPGMA	99.62	99.06	69.90	85.75	6.44
	KM	80.50	81.39	60.92	24.32	2.09		KM	99.54	98.95	70.05	86.54	4.42
	DBSCAN	89.83	88.37	67.19	7.97	0.29		DBSCAN	98.44	98.42	64.59	52.16	1.35
	OPTICS	95.83	93.99	71.20	9.78	0.39		OPTICS	95.18	93.99	66.39	46.75	0.92
	PAM	99.67	99.28	75.27	48.71	19.41		PAM	99.54	98.95	70.05	86.54	4.42
	FC	99.67	99.28	75.27	48.71	19.41		FC	99.54	98.95	70.00	86.41	3.55
	GM	99.67	99.28	75.27	48.71	19.41		GM	99.51	98.76	69.84	84.81	3.68
	CABRA	92.83	91.22	27.86	20.90	6.17		CABRA	99.05	98.73	24.54	16.26	1.38
Skewed	UPGMA			40.57	13.47	2.55	Spirals	UPGMA	100.00	100.00	52.60	29.01	8.39
	KM			44.20	19.01	0.82		KM	90.98	75.95	52.58	31.31	0.76
	DBSCAN			-3.32	0.87	6.73		DBSCAN	100.00	100.00	52.60	29.01	8.39
	OPTICS			27.71	5.57	0.78		OPTICS	93.69	90.49	-28.07	6.34	0.24
	PAM			44.44	18.59	1.23		PAM	94.76	85.10	52.64	30.70	0.26
	FC			42.87	17.61	0.21		FC	94.80	85.21	52.68	30.76	0.33
	GM			39.17	14.10	3.64		GM	71.31	31.17	41.21	18.78	0.17
	CABRA			6.61	7.02	2.41		CABRA	100.00	100.00	52.60	29.01	8.39
Asymmetric	UPGMA			59.75	17.59	3.62	Triangle	UPGMA	63.25	41.62	38.12	3.86	4.26
	KM			64.54	34.19	0.48		KM	53.38	25.74	45.28	5.78	2.29
	DBSCAN			2.27	0.82	0.20		DBSCAN	65.18	50.87	22.29	1.08	1.53
	OPTICS			50.38	9.76	0.64		OPTICS	94.00	90.66	29.76	1.83	1.03
	PAM			64.55	34.18	1.34		PAM	72.53	40.04	43.92	5.11	1.74
	FC			64.50	34.09	1.34		FC	68.47	35.24	44.22	5.31	1.18
	GM			62.34	30.82	3.40		GM	99.03	97.07	44.35	4.45	3.10
	CABRA			37.91	17.48	2.38		CABRA	95.55	93.00	-36.75	0.49	1.58

[4] Y. Cheng, Mean shift, mode seeking, and clustering, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1995) 790–799, https://doi.org/10.1109/34.400568.

[5] M. Earnest, Average minimum distance between n points generate i.i.d. with uniform dist, Math. Stack Exch. (2016), https://math.stackexchange.com/q/2001026.

[6] M. Ester, K. Hans-Peter, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), 1996, pp. 226–231.

[7] K. Fukunaga, L. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition, IEEE Trans. Inf. Theory 21 (1975) 32–40, https://doi.org/10.1109/TIT.1975.1055330.

[8] L. Kaufman, P. Rousseeuw, Finding Groups in Data: an Introduction to Cluster Analysis, Wiley, New York, 1990.

[9] L. Kaufman, P.J. Rousseeuw, Partitioning Around Medoids (Program PAM), John Wiley & Sons, Ltd., 1990.

[10] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Statistics, vol. 1, University of California Press, Berkeley, 1967, pp. 281–297, http://projecteuclid.org/euclid.bsmsp/1200512992.

[11] F. Nielsen, Hierarchical Clustering, Springer International Publishing, Cham, 2016, pp. 195–211.

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

[13] R.R. Sokal, C.D. Michener, A statistical method for evaluating systematic relationships, Univ. Kans. Sci. Bull. 38 (1958) 1409–1438, https://api.semanticscholar.org/CorpusID:61950873.

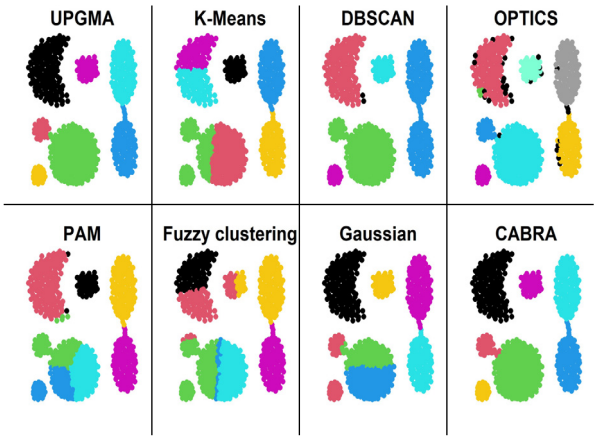

Fig. 4. Aggregation data set results.

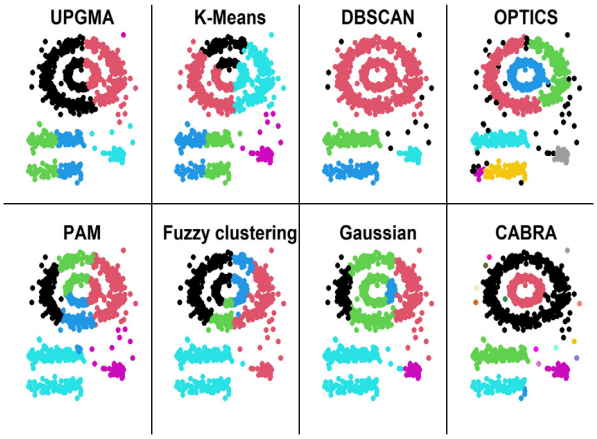

Fig. 7. Multi data set results.

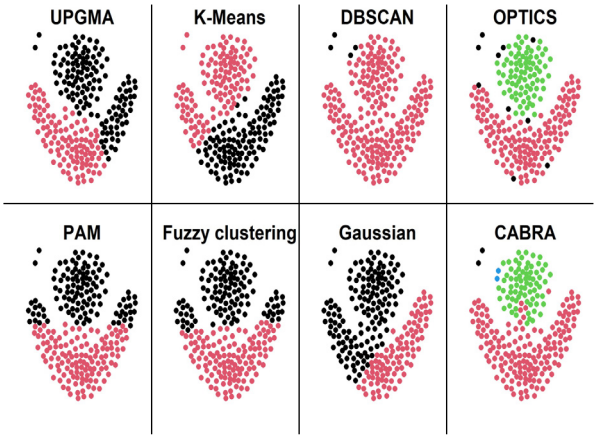

Fig. 5. Flame data set results.

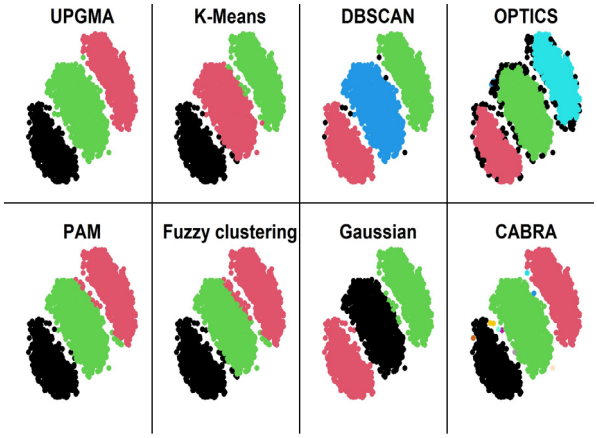

Fig. 8. Basic4 data set results.

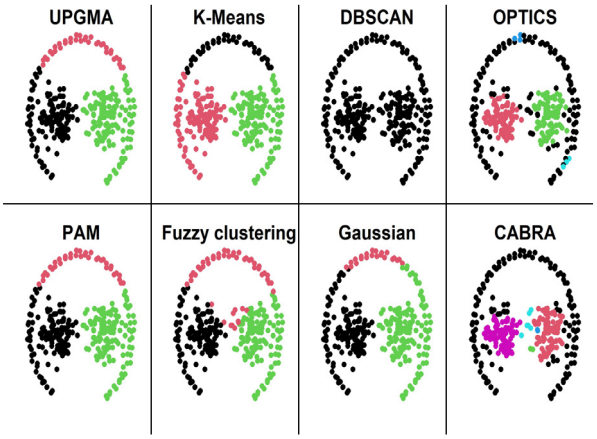

Fig. 6. Path data set results.

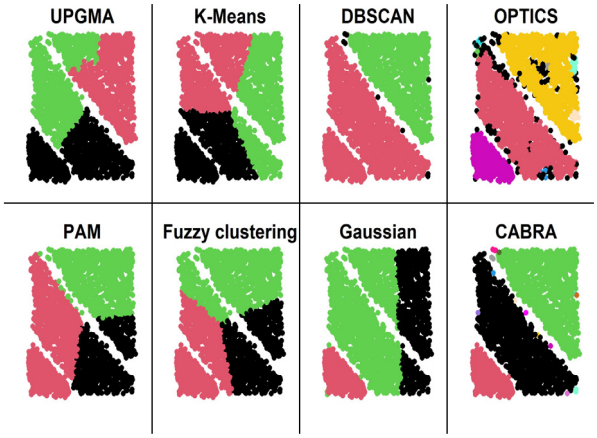

Fig. 9. Boxes2 data set results.

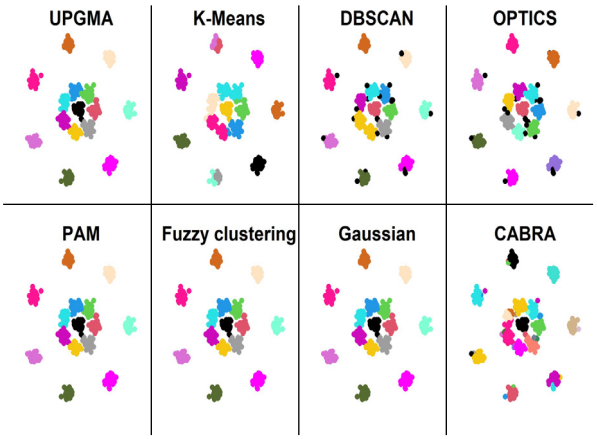

Fig. 10. R15 data set results.

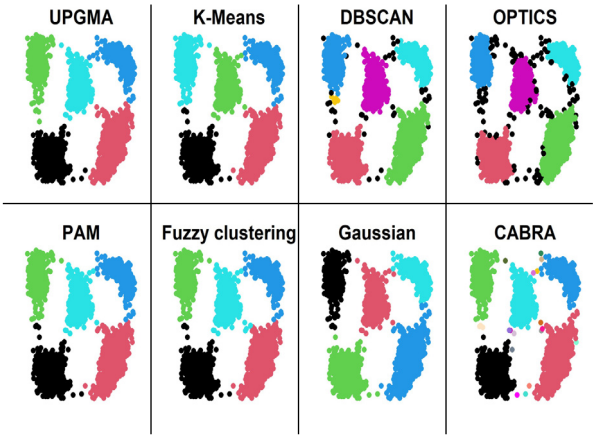

Fig. 13. Network data set results.

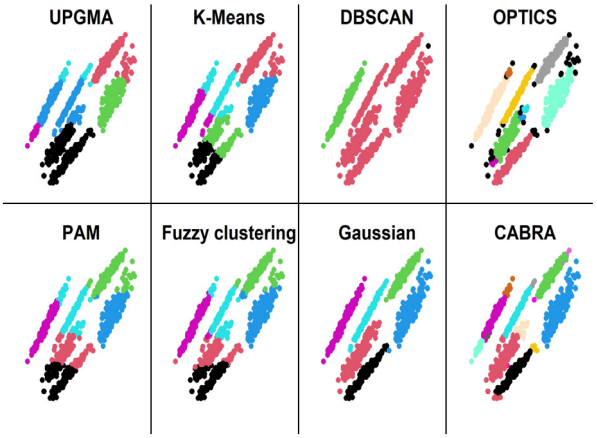

Fig. 11. Skewed data set results.

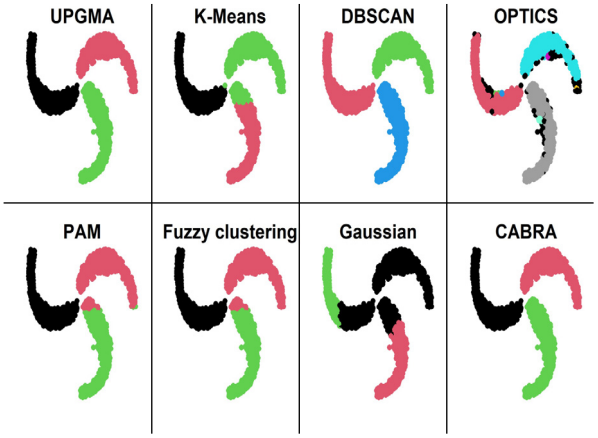

Fig. 14. Spirals data set results.

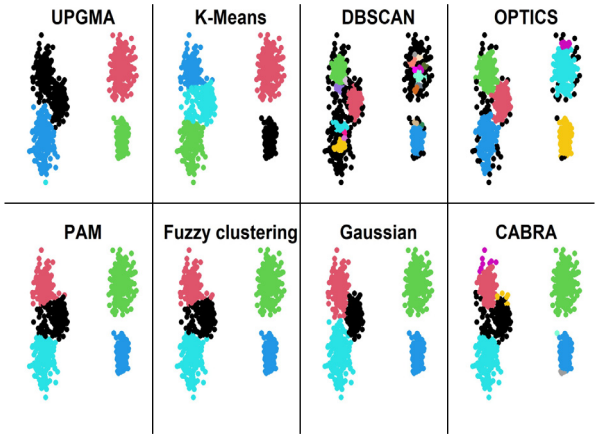

Fig. 12. Asymmetric data set results.

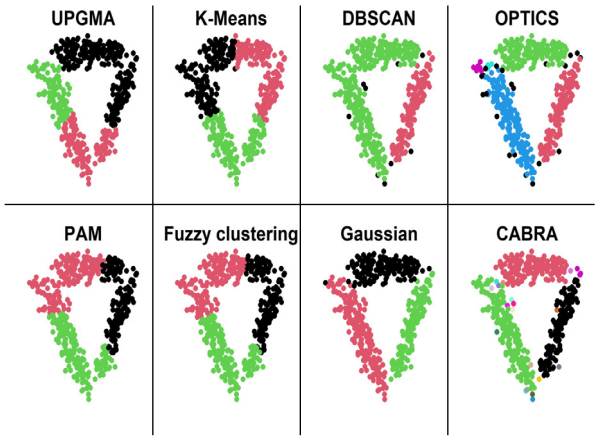

Fig. 15. Triangle data set results.