CVIK: A Matlab-based cluster validity index toolbox for automatic data clustering

Adán José-García, Wilfrido Gómez-Flores

SoftwareX 22 (2023) 101359

Contents lists available at ScienceDirect

SoftwareX

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

Original software publication

CVIK: A MATLAB-based cluster validity index toolbox for automatic data clustering

Adán José-García [a], Wilfrido Gómez-Flores [b,*]

[a] Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
[b] Centro de Investigación y de Estudios Avanzados del IPN, Unidad Tamaulipas, 87130, Cd. Victoria, Tamaulipas, Mexico

ARTICLE INFO

Article history:
Received 14 December 2021
Received in revised form 1 February 2023
Accepted 8 March 2023

Dataset link: https://github.com/adanjoga/cvik-toolbox

Keywords:
Clustering
Cluster validity index
Automatic clustering

ABSTRACT

We present CVIK, a MATLAB-based toolbox for assisting the process of cluster analysis applications. This toolbox aims to implement 28 cluster validity indices (CVIs) for measuring clustering quality available to data scientists, researchers, and practitioners. CVIK facilitates implementing the entire pipeline of automatic clustering in two approaches: (i) evaluating candidate clustering solutions from classical algorithms, in which the number of clusters increases gradually, and (ii) assessing potential solutions in evolutionary clustering algorithms using single- and multi-objective optimization methods. This toolbox also implements distinct proximity measures to estimate data similarity, and the CVIs are capable of processing both feature data and relational data. The source code and examples can be found in this GitHub repository: https://github.com/adanjoga/cvik-toolbox.

Code metadata

Current code version	v1.0
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-21-00241
Legal Code License	GPLv3
Code versioning system used	git
Software code languages, tools, and services used	Matlab and C++
Compilation requirements, operating environments & dependencies	Linux, macOS, Microsoft Windows
If available Link to developer documentation/manual	https://github.com/adanjoga/cvik-toolbox
Support email for questions	adan.jose@cinvestav.mx, wgomez@cinvestav.mx

1. Motivation and significance

Cluster analysis is an unsupervised learning task for discovering underlying groups in unlabeled data using similarity measures. A distance metric usually determines the similarity between data points to form clusters with greater homogeneity than their counterparts [1]. A crucial free hyperparameter in a clustering algorithm is the number of groups to be discovered. For instance, *k*-means requires a *k*-value of groups [2], whereas hierarchical clustering needs a dendrogram cut-off level to form clusters [3]. In practice, a data analyst or rule of thumb usually defines the number of groups a priori; however, these heuristic approaches are unsuitable in real-world scenarios with challenging conditions, typically found in high-dimensional data and complex cluster shapes.

To cope with the uncertainty about the number of groups, automatic clustering methods involve a search procedure to determine a suitable clustering. A common approach is to use a clustering algorithm to generate several candidate partitions by varying the number of groups within a search range (e.g., the *k*-value in *k*-means). The quality of every clustering is then calculated using a cluster validity index (CVI). This way, the final clustering and its number of groups are obtained from the solution reaching the best CVI value [4].

* Corresponding author.
E-mail addresses: adan.jose@cinvestav.mx (Adán José-García), wgomez@cinvestav.mx (Wilfrido Gómez-Flores).

https://doi.org/10.1016/j.softx.2023.101359

Table 1
Toolkits proposed in the literature that implements CVIs.

Author	Year	Toolkit name	# CVIs	Language	Ref.
Brock et al.	2021	clValid	3	R	[9,10]
Walesiak and Dudek	2021	clusterSim	8	R	[11]
Robles-Berumen et al.	2020	LEAC	10	C++	[12]
Qaddoura et al.	2020	EvoCluster	5	Python	[13]
Z. Cebeci	2020	fcvalid	18	R	[14]
L. Nieweglowski	2020	clv	8	R	[15]
E. Dimitriadou	2020	cclust	14	R	[16]
C. Baker	2019	validclust	8	Python	[17]
B. Desgraupes	2018	clusterCrit	28	R	[18]
Charrad et al.	2015	NbClust	30	R	[19]
MATLAB 2013b	2013	evalclusters	4	MATLAB	[20]
Wang et al.	2009	CVAP	14	MATLAB	[21]
Balasko et al.	2005	FuzzClust	7	MATLAB	[22]

This search procedure uses a CVI as an external quality measure since it evaluates the candidate solutions outside the clustering algorithm [5]. Nonetheless, a CVI is naturally an objective function that has been incorporated as an internal quality measure within automatic clustering methods based on metaheuristics such as genetic algorithm (GA), differential evolution (DE), and particle swarm optimization (PSO) [6].

Since a CVI is a relevant component in any clustering algorithm, many CVIs have been proposed in the literature, measuring the relationship between cluster attributes such as cohesion, separation, connectedness, and symmetry [7]. However, notice that the great variety of existing CVIs leads to the problem of selecting one that performs adequately according to the underlying structure of the input data. In this regard, the "no free lunch" (NFL) theorem states that if any search algorithm performs exceptionally well on one set of objective functions, it must perform correspondingly poorly on all other objective functions [8]. Hence, from the viewpoint of cluster analysis, the NFL theorem implies considering a pool of CVIs to obtain the most suitable one that fits a specific clustering problem.

This fact has motivated the proposal of different cluster analysis toolkits in the literature that implement CVIs for measuring clustering quality in different applications. As a result, the research community has introduced and updated eight cluster analysis toolkits in the last three years, summarized in Table 1, suggesting the current interest in providing frameworks for developing clustering applications.

Notably, almost all the toolkits were developed in the R language, followed by MATLAB, Python, and C++. In the case of MATLAB, version 2013b introduced the function '`evalclusters`' with four CVIs, as a part of the Statistics and Machine Learning Toolbox [20]. In addition, the MATLAB-based toolboxes CVAP [21] and FuzzClust [22] include 14 and seven CVIs, respectively; both were released more than ten years ago and still need to be updated to incorporate newer CVIs and distance metrics recently proposed in the literature. Furthermore, it is noticeable that other toolkits developed in different programming languages are more complete than current MATLAB-based toolboxes.

Additionally, the current CVIs implemented in these toolkits only accept feature data for measuring cluster attributes. Such feature data can be represented by an $N \times D$ matrix, where N is the number of observations and D is the number of features. However, there are applications where the original feature space is unavailable, and only relational data is given in an $N \times N$ matrix that specifies pairwise similarities (or dissimilarities) between observations. In this case, users should modify CVIs to cope with relational data [23].

On the other hand, cluster analysis toolkits usually address automatic clustering in two ways: (1) by evaluating candidate clustering solutions from classical algorithms, like k-means, where the number of clusters variates iteratively, and (2) by evaluating potential solutions in evolutionary clustering algorithms using single- and multi-objective optimization approaches. Nevertheless, both automatic clustering methods have yet to be implemented in a single toolkit.

This paper introduces a MATLAB-based toolbox named CVIK that incorporates the most considerable quantity of CVIs implemented in this programming language to evaluate clustering quality. Furthermore, CVIK assists in the process of implementing automatic clustering applications through the following contributions:

- It implements 28 CVIs that can use six different proximity measures, including the Laplacian distance, point-symmetry distance, and maximum edge distance, which are not considered in other MATLAB-based toolkits.
- It increases the versatility of CVIs by accepting both feature and relational data.
- It includes automatic evolutionary clustering algorithms from the literature based on single- and multi-objective optimization, using CVIs as objective functions.
- It incorporates a framework of automatic clustering based on k-means, k-medoids, and hierarchical clustering, where CVIs evaluate candidate partitions as the number of groups gradually increases.

Hence, the CVIK toolbox may be helpful for the research community and practitioners interested in the practical applications of automatic cluster analysis.

2. Software description

The source code of the CVIK toolbox has been released under the GPLv3 license, and its technical documentation and examples are available in this GitHub repository: https://github.com/adanjoga/cvik-toolbox. The next subsections present how the CVIK toolbox software is integrated for the automatic clustering task. Besides, the CVIK toolbox's architecture and functionalities are described in detail.

2.1. Software integration

Fig. 1 illustrates the integration of the proposed CVIK toolbox into the automatic clustering pipeline for determining the most suitable clustering partition and its corresponding number of clusters. The task of automatic data clustering based on CVIs can be performed in two manners. First, following an automatic *a posteriori* clustering approach using traditional clustering algorithms, where the CVI is used as an external function that evaluates the quality of different candidate clustering solutions (see Fig. 1a). Second, an automatic clustering approach based on metaheuristic methods, where a CVI is integrated as a fitness function (see Fig. 1b). Thus, CVIK contains several elements that can be varied into the different components of both automatic clustering approaches presented in Fig. 1.

2.2. Software architecture

The CVIK toolbox is developed in MATLAB R2021a. In addition to the implemented CVIs, the toolbox includes other important elements for the automatic data clustering task, such as different proximity measures and external validity indices. Fig. 2 shows the organization of the CVIK toolbox, which is based on four main modules:

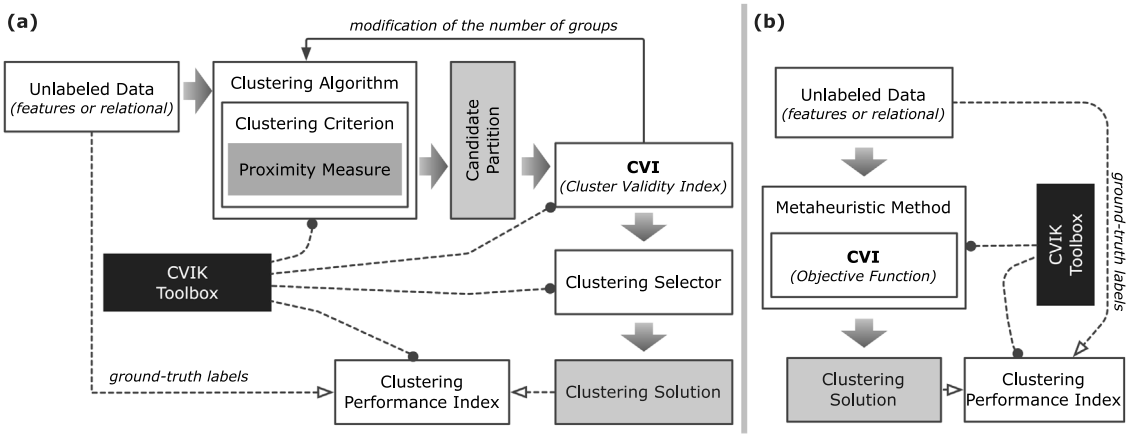

Fig. 1. Integration of the CVIK Toolbox into the clustering task pipeline to determine the number of clusters automatically. In (a), CVIK's indices are used as external validation functions for evaluating candidate partitions from traditional clustering algorithms. In (b), CVIK's indices are internal clustering criteria (objective functions) in automatic clustering methods based on metaheuristics.

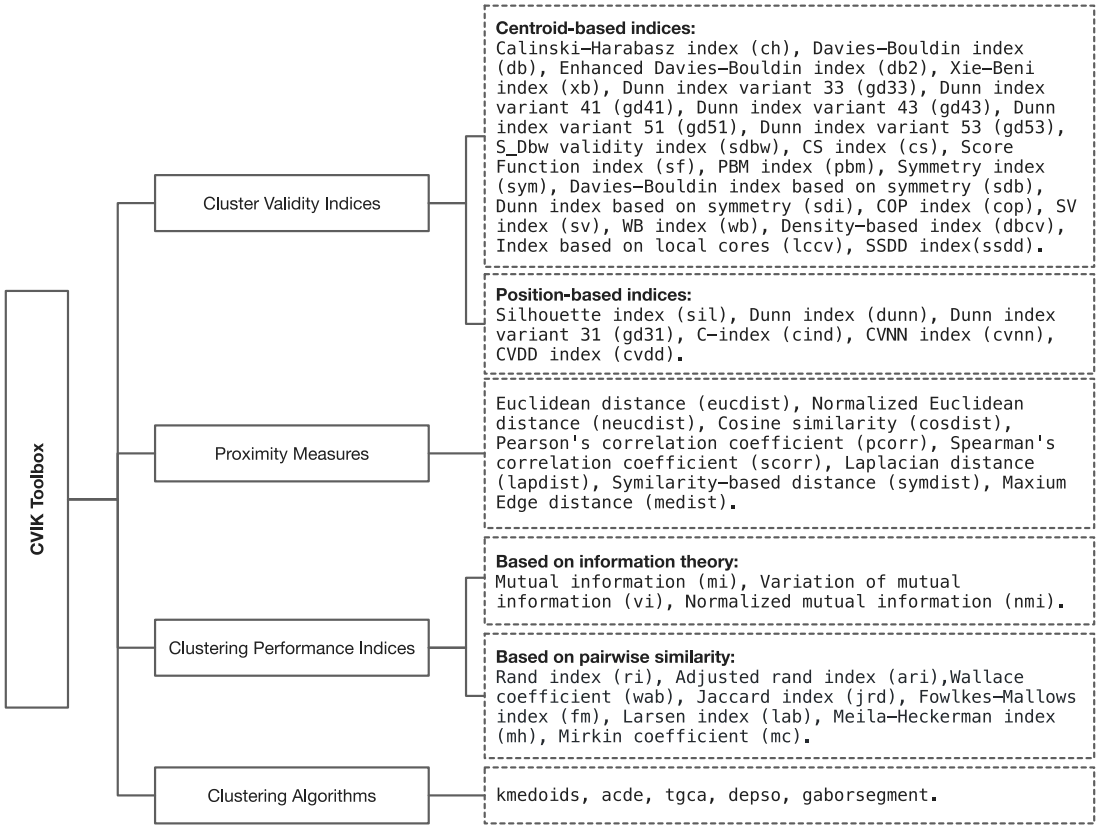

Fig. 2. Organization of the CVIK toolbox and its list of functions.

- **Cluster validity indices**. It contains 28 CVIs from the state-of-the-art. These indices are categorized according to how they operate on the input space: centroid-based indices, which calculate cluster prototypes, and position-based indices, which use the entire data points. Detailed information about the mathematical definition of all the CVIs can be found in the user's guide.
- **Proximity measures**. It contains ten proximity functions optimized by vectorization to make matrices and vectors operations efficiently. The CVIs module depends on these proximity functions to measure cluster quality.
- **Clustering performance indices**. It includes 11 external indices to validate the performance of the clustering algorithms when the actual cluster labels are known *a priori*. These indices are based on two approaches: information theory, in which mutual information measures the dependence between two partitions considered random variables, and pairwise similarity, in which the agreement between pairs of labels is measured.
- **Clustering algorithms**. It implements four clustering algorithms for the demonstration of the proposed toolbox. These methods are optimization-based algorithms for addressing automatic data clustering considering two approaches: gradually increasing the number of groups in the *k*-means

(or k-medoids) algorithm and performing single- and multi-objective optimization using population-based metaheuristics.

2.3. Software functionalities

In addition to the functions listed in Fig. 2, the CVIK toolbox provides three basic interfaces for usability purposes:

- `cviconfig`. This interface inputs the CVI name and returns the corresponding CVI function handle and a flag indicating if the index should be maximized or minimized. This information is crucial for selecting the best clustering solution for the chosen CVI.
- `evalcvi`. This interface takes the CVI name and the collection of clustering solutions and returns an evaluation object indicating the optimal number of clusters in terms of the provided index.
- `proxconfig`. This interface receives one of the available proximity measures and returns the corresponding function handle to compute the distance between observations.

3. Illustrative examples

The following subsections provide four illustrative examples using the CVIK toolbox to determine the number of clusters in different clustering scenarios automatically. The aim is to show the feasibility and versatility of the CVIK toolbox in different types of automatic clustering problems when using distinct unlabeled input datasets.

For didactic purposes, the first three examples use toy 2D datasets for easily visualizing the effects of automatic clustering by CVIK. We chose the well-known Iris dataset [24] in the first example about employing CVIs as external quality measures. The second example about using CVIs as objective functions utilizes synthetic datasets widely used in the clustering literature [7]. Likewise, the third experiment uses a synthetic dataset with non-linearly separable groups to show how a CVI based on the maximum edge distance can reveal clusters with arbitrary shapes. Finally, the fourth example is focused on unsupervised image segmentation based on clustering textures [25], considering a multidimensional feature space generated by decomposing the input image with a bank of Gabor filters. All these datasets are included in CVIK to run the examples presented herein.

3.1. CVIs as external quality measures

Usually, the CVIs are used as external quality measures to determine the best clustering solution and the optimal number of clusters from several partitions [5]. In this approach, a clustering algorithm is run over the input dataset by gradually increasing the number of groups such that a collection of M candidate partitions $\{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_M\}$ is generated. Then, a CVI is computed for all the resulting partitions, and the one obtaining the best CVI value (maximum or minimum) is selected as the final clustering [7]. Notice that the "external" characteristic is because the CVI is used outside of the clustering algorithm. In practice, the collection of clusterings can be generated by using the k-means or k-medoids algorithms, or a hierarchical clustering technique such as single, average, or complete linkage methods. Fig. 1a shows this automatic clustering approach when using a CVI as an external quality measure.

Fig. 3 illustrates the use of the Calinski–Harabasz (CH) index [26] to determine the best solution from a collection of clusterings generated by two well-known clustering algorithms on the Iris dataset. Fig. 3(a) presents the performance of the CH index over a clustering collection generated by k-means when varying the number of groups in the range $K \in [2, 10]$. It is noted that the CH index obtains its maximum value for $K = 3$, and the corresponding clustering solution has an Adjusted Rand index (ARI) [27] value of 0.72. Furthermore, Fig. 3(b) presents the CH index performance for the average-linkage hierarchical method. In this case, the algorithm generates a hierarchy of nested clusterings (called dendrogram); thus, a clustering collection is generated by varying the cut-off point in the hierarchy. Again, the CH index attains its maximum value for $K = 3$, and the clustering performance of the corresponding solution is ARI = 0.76.

3.2. CVIs as objective functions

Next, we present how to use the CVIs as objective functions in optimization-based clustering algorithms. Because CVIs are naturally objective functions that can be maximized or minimized, they have been integrated into evolutionary clustering approaches to address the automatic clustering problem so that the CVIs guide the search procedure towards promising clustering solutions iteratively [6]. There are considered two cases: single- and multi-objective optimization. Fig. 1b illustrates this automatic clustering approach when considering a CVI as an internal quality measure (i.e., objective function).

Fig. 4 illustrates some CVIs' behavior used as objective functions in two evolutionary clustering algorithms based on differential evolution (DE), which automatically vary the number of clusters during the search procedure. Fig. 4(a) presents the performance of the ACDE algorithm [28], which maximizes the CH index on a two-dimensional synthetic dataset. As observed, the algorithm reached convergence after 100 iterations, and the quality of the best partition solution with five clusters is ARI=0.95. Moreover, Fig. 4(b) presents the clustering solution obtained with the multi-objective algorithm MODEC [6] on a synthetic dataset with groups of different densities. First, the optimization procedure simultaneously minimizes the Xie–Beni (XB) [29] and PBM [30] indices. Then, from the Pareto front approximation (PFA), the normal boundary intersection method selects the best solution located at the knee of the curve [31]. Such as knee point has the largest distance from PFA to the line connecting the extreme points.

3.3. CVIs for arbitrary-shaped clusters

The proximity measure plays an essential role in a CVI as it quantifies the similarity (or dissimilarity) between data points that help discover the underlying cluster structure. For instance, the Euclidean distance is more suitable for spherically-shaped clusters, the maximum edge distance (MED) excels at identifying irregular-shaped components [32], and the Cosine distance is more suitable when orientation between patterns is more relevant than their magnitude [33]. In this regard, the following example shows how using an appropriate distance in a CVI leads to revealing arbitrary-shaped clusters. This automatic clustering approach when using a CVI as an external quality measure is illustrated in Fig. 1a.

Fig. 5 illustrates the performance of the Silhouette (SIL) index [34] using the MED distance on a synthetic dataset having two arbitrary-shaped clusters. First, from the original feature data, a dissimilarity matrix using the MED distance is created to obtain relational data. Next, a clustering collection is generated using the k-medoids algorithm by varying the number of groups iteratively. Finally, the best solution is selected using the SIL index based on the MED distance. It is noticeable that this approach successfully reveals the two non-linearly separable clusters in the

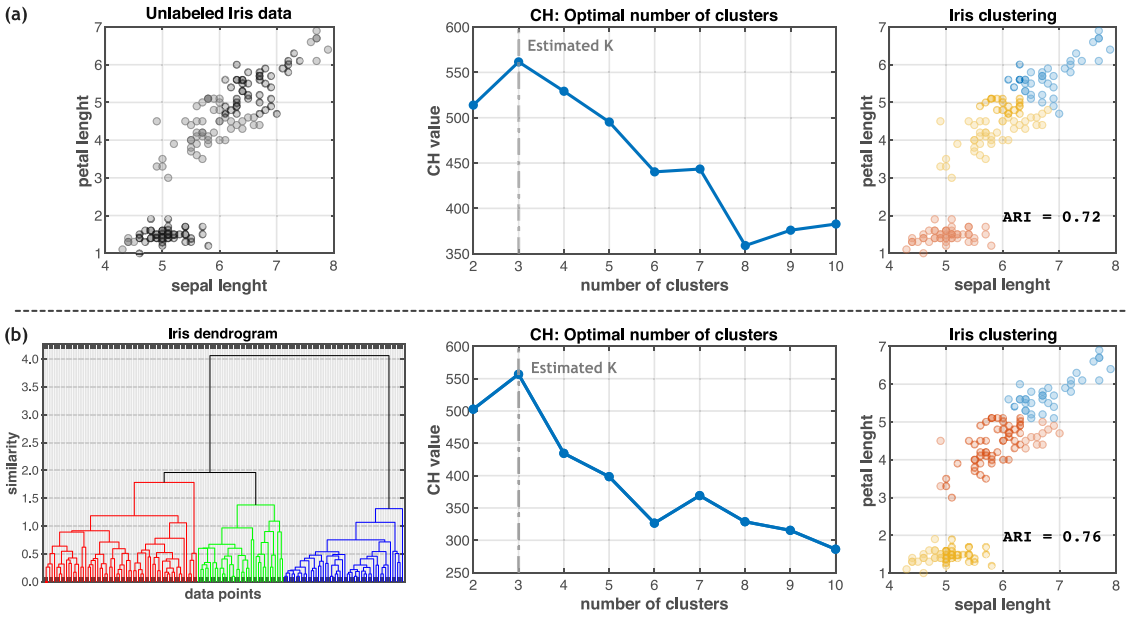

Fig. 3. Examples of the Calinski–Harabasz (CH) index used as an external quality measure for finding the optimal number of clusters in the Iris dataset using (a) the *k*-means algorithm and (b) the average-linkage hierarchical method. In (a), the unlabeled dataset, the CH value as a function of the number of clusters, $K = \{2, \ldots, 10\}$, and the best clustering solution are illustrated. Likewise, (b) shows the dendrogram, the CH value as a function of the number of clusters, and the best clustering solution.

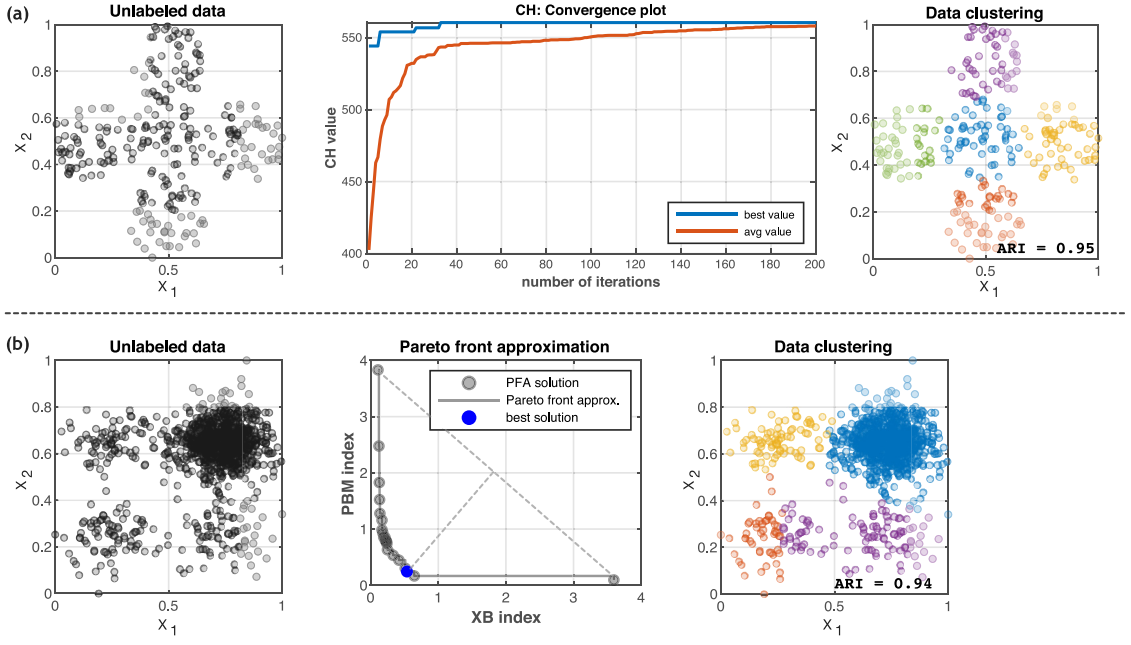

Fig. 4. Examples of some CVIs as objective functions in automatic data clustering problems when using two different optimization techniques (a) the single-objective ACDE algorithm and (b) the multi-objective MODEC approach. In (a), it is illustrated the unlabeled dataset, the convergence plot when maximizing the CH index, and the best clustering solution. In (b) is represented the unlabeled dataset, the Pareto front approximation obtained when minimizing the XB and PBM indices, and the best clustering solution obtained by the normal boundary intersection method.

input data. Hence, this example demonstrates the versatility of CVIK for clustering arbitrary-shaped clusters as the CVI operates on the dissimilarity matrix (relational data) instead of the original feature data.

3.4. Application to image segmentation

Image segmentation is the task of dividing an image into its constituent parts. Commonly, semantic segmentation methods are used for pixel-wise classification in known classes. However, semantic segmentation approaches require training datasets in which objects of different classes are manually labeled by humans, which can be labor-intensive and costly. Hence, if obtaining an annotated dataset for training is challenging, unsupervised segmentation is an alternative to extract the objects of interest by clustering similar local features such as color, intensities, and textures [35].

Here, we illustrate an example reproducing a well-known method proposed in the literature by Jain and Farrokhnia [25], in which the image segmentation task is addressed as a texture

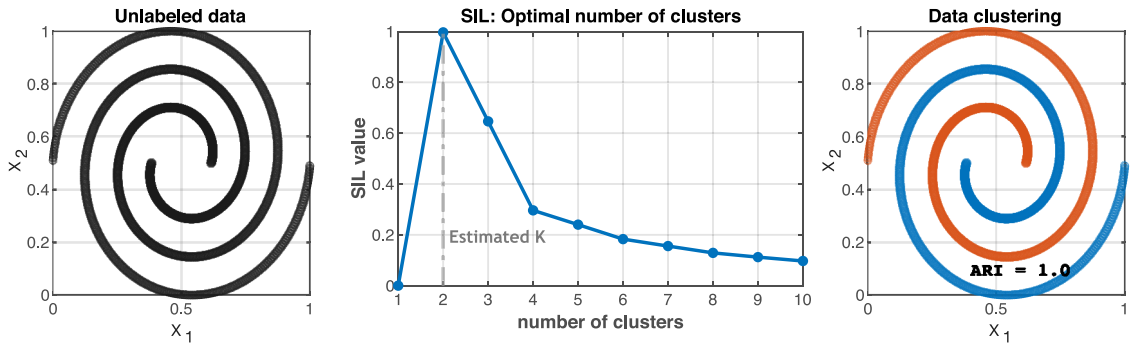

Fig. 5. Example of the SIL index based on MED distance for finding the natural partitioning and the number of clusters in data with arbitrary-shaped clusters. The collection of clustering solutions is generated using the k-medoids algorithm in the range $K \in [1, 10]$, and the best solution in terms of the SIL index for $K = 2$ is shown.

clustering problem. Texture can be defined as the local-spatial variation of pixel intensities at scales smaller than the scales of interest [36]. Such intensity variations produce texture patterns that visually differentiate objects in the image; hence, it is necessary to numerically depict them before applying a segmentation method.

The segmentation method decomposes the input image by a bank of Gabor filters to enhance local texture characteristics. Filtering is implemented in the frequency domain by varying the rotation and bandwidth of the filters to enhance textures with a specific orientation and scale. Therefore, each filter in the bank creates a feature map that describes a particular texture. Besides, adjacency information (i.e., x- and y-coordinates) is concatenated to feature maps to avoid losing spatial information. Next, this feature space is clustered by the k-means algorithm, where each pixel is assigned to a group according to texture similarity with its neighbors.

In this illustrative example, this segmentation method is embedded into the pipeline of automatic clustering by using the scheme shown in Fig. 1a to determine the suitable number of groups, i.e., disjoint texture regions in the image. The `Clustering Algorithm` module of CVIK includes the function called `gaborsegment` for texture segmentation.

Fig. 6 shows some examples of image segmentation based on Gabor filters and k-means algorithm, where the CH index evaluates potential partitions to determine the optimal number of disjoint regions.

4. Impact

As surveyed in Section 1, currently, there exist several toolkits that implement CVIs for clustering applications. However, CVIK is the first MATLAB-based toolbox that integrates 28 CVIs into algorithms for automatic data clustering by considering two approaches: 1) a linear search that gradually increases the number of clusters in algorithms like k-means until finding the best partition and 2) an evolutionary search based on differential evolution where the CVIs are used as fitness functions in single- and multi-objective optimization schemes. Automatic data clustering is useful in applications where the number of groups is unknown beforehand, for instance, image segmentation, where no standard reference segmentation is available, or data mining applications for multivariate data, where it is challenging to infer the proper number of clusters.

Moreover, CVIK implements CVIs that consider data density and connectivity in addition to the common intra-group cohesion and inter-group separation criteria. For instance, the DBCV [37], SSDD [38], and LCCV [39] indices are included.

CVIK also includes distinct proximity measures capable of handling feature and relational data. This property is useful for dealing with spherical-shaped and arbitrary-shaped clusters since the proximity between data points can be measured accordingly to the underlying data structure, opening the way for using CVIK in a wide range of data distributions.

5. Conclusions

This article introduced CVIK, a MATLAB-based cluster validity index toolbox for automatic data clustering applications. Many CVIs have been proposed in the literature to determine the appropriate number of clusters and the corresponding underlying partitioning. In this regard, CVIK includes 28 state-of-the-art CVIs for addressing this common clustering task, and the toolbox can serve as a guideline for selecting the most suitable CVI for different clustering applications. Furthermore, CVIK is easily extensible and configurable and allows for experimental studies in different data clustering scenarios, as described in Section 3. Besides, CVIK is supplemented by a user technical manual that includes a detailed overview of the software architecture and the implemented clustering techniques.

The proposed toolbox is limited to numerical input data. In this sense, if the automatic clustering problem consists of heterogeneous data (i.e., numerical, binary, symbolic attributes), these data must first be preprocessed to obtain a numerical dataset and then use the CVIK toolbox. Future work will consider the inclusion of CVIs capable of dealing intrinsically with heterogeneous datasets. In addition, it would be desirable to develop a graphical interface so that the end user can use this toolbox in a simpler way.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

https://github.com/adanjoga/cvik-toolbox

Acknowledgments

The authors are grateful to the Université de Lille and the Cinvestav-IPN for the economic support.

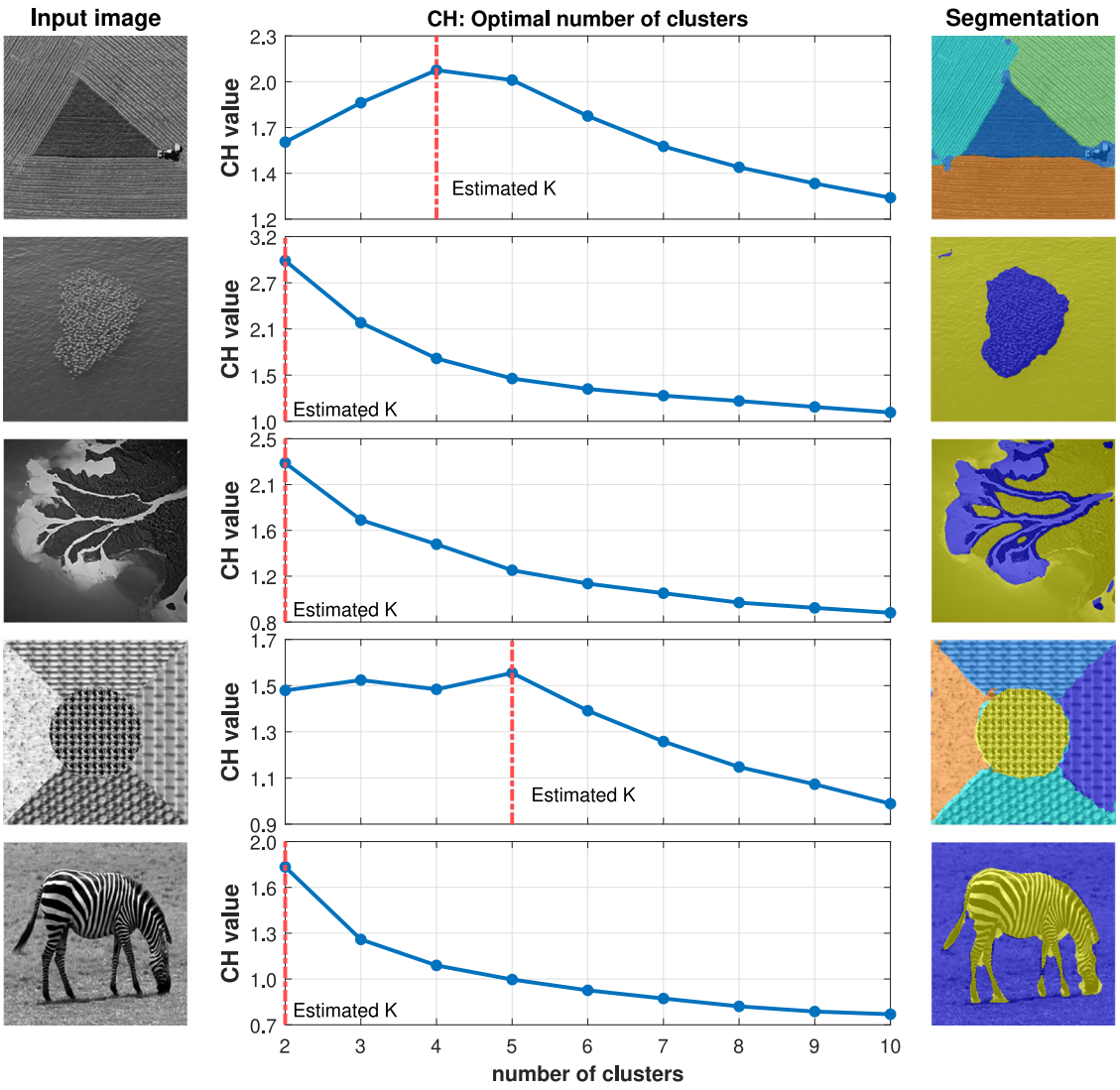

Fig. 6. Exemplification of the CVIK toolbox for image segmentation based on Gabor filters and *k*-means algorithm. Here the CH index is used as an external quality measure for finding the optimal number of segmented regions.

References

[1] Xu R, Wunsch DC. Clustering. Hoboken, New Jersey: John Wiley & Sons, Inc; 2008.

[2] Sinaga KP, Yang M-S. Unsupervised K-means clustering algorithm. IEEE Access 2020;8:80716–27. http://dx.doi.org/10.1109/ACCESS.2020.2988796.

[3] Karna A, Gibert K. Automatic identification of the number of clusters in hierarchical clustering. Neural Comput Appl 2022;34:119–34. http://dx.doi.org/10.1007/s00521-021-05873-3.

[4] Ezugwu AE, Shukla AK, Agbaje MB, Oyelade ON, José-García A, Agushaka JO. Automatic clustering algorithms: A systematic review and bibliometric analysis of relevant literature. Neural Comput Appl 2021;33:6247–306. http://dx.doi.org/10.1007/s00521-020-05395-4.

[5] Arbelaitz O, Gurrutxaga I, Muguerza J, Pérez JM, Perona I. An extensive comparative study of cluster validity indices. Pattern Recognit 2013;46(1):243–56. http://dx.doi.org/10.1016/j.patcog.2012.07.021.

[6] José-García A, Gómez-Flores W. Automatic clustering using nature-inspired metaheuristics: A survey. Appl Soft Comput 2016;41:192–213. http://dx.doi.org/10.1016/j.asoc.2015.12.001.

[7] José-García A, Gómez-Flores W. A survey of cluster validity indices for automatic data clustering using differential evolution. In: The genetic and evolutionary computation conference. Lille, France: ACM; 2021, p. 1–9. http://dx.doi.org/10.1145/3449639.3459341.

[8] Wolpert D, Macready W. No free lunch theorems for optimization. IEEE Trans Evol Comput 1997;1(1):67–82. http://dx.doi.org/10.1109/4235.585893.

[9] Brock G, Pihur V, Datta S, Datta S. clValid: An R package for cluster validation. J Stat Softw 2008;25(4):1–22. http://dx.doi.org/10.18637/jss.v025.i04.

[10] Brock G, Pihur V, Datta S, Datta S. clValid: Validation of clustering results. 2021, URL https://cran.r-project.org/package=clValid, R package version 0.7.

[11] Walesiak M, Dudek A. clusterSim: Searching for optimal clustering procedure for a data set. 2021, URL https://cran.r-project.org/package=clusterSim, R package version 0.49-2.

[12] Robles-Berumen H, Zafra A, Fardoun HM, Ventura S. LEAC: An efficient library for clustering with evolutionary algorithms. Knowl-Based Syst 2019;179:117–9. http://dx.doi.org/10.1016/j.knosys.2019.05.008.

[13] Qaddoura R, Faris H, Aljarah I, Castillo PA. EvoCluster: An open-source nature-inspired optimization clustering framework in Python. In: Castillo PA, Jiménez Laredo JL, Fernández de Vega F, editors. Applications of evolutionary computation. Cham: Springer International Publishing; 2020, p. 20–36. http://dx.doi.org/10.1007/s42979-021-00511-0.

[14] Cebeci Z. Fcvalid: An R package for internal validation of probabilistic and possibilistic clustering. Sakarya Univ J Comput Inf Sci 2020;3:11–27. http://dx.doi.org/10.35377/saucis.03.01.664560.

[15] Nieweglowski L. clv: Cluster validation techniques. 2020, URL https://CRAN.R-project.org/package=clv, R package version 0.3-2.2.

[16] Dimitriadou E. cclust: Convex clustering methods and clustering indexes. 2020, URL https://cran.r-project.org/package=cclust, R package version 0.6-22.

[17] Baker C. Validclust. 2019, URL https://validclust.readthedocs.io/en/latest, Phyton package 11be673b.

[18] Desgraupes B. Clustercrit: Clustering indices. 2018, URL https://cran.r-project.org/package=clusterCrit, R package version 1.2.8.

[19] Charrad M, Ghazzali N, Boiteau V, Niknafs A. NbClust: An R package for determining the relevant number of clusters in a data set. J Stat Softw 2014;61(6):1–36. http://dx.doi.org/10.18637/jss.v061.i06.
[20] Mathworks M. Evalclusters. 2013, Matlab function, URL https://www.mathworks.com/help/stats/evalclusters.html.
[21] Wang K, Wang B, Peng L. CVAP: Validation for cluster analyses. Data Sci J 2009;8:88–93. http://dx.doi.org/10.2481/dsj.007-020.
[22] Balasko B, Abonyi J, Feil B. Fuzzy clustering and data analysis toolbox. 2005, Matlab toolbox, URL https://www.abonyilab.com/fclusttoolbox.
[23] José-García A, Handl J, Gómez-Flores W, Garza-Fabre M. An evolutionary many-objective approach to multiview clustering using feature and relational data. Appl Soft Comput 2021;108. http://dx.doi.org/10.1016/j.asoc.2021.107425.
[24] Fisher RA. The use of multiple measurements in taxonomic problems. Ann Eugen 1936;7(2):179–88. http://dx.doi.org/10.1111/j.1469-1809.1936.tb02137.x.
[25] Jain AK, Farrokhnia F. Unsupervised texture segmentation using Gabor filters. Pattern Recognit 1991;24(12):1167–86. http://dx.doi.org/10.1016/0031-3203(91)90143-S.
[26] Calinski T, Harabasz J. A dendrite method for cluster analysis. Comm Statist Theory Methods 1974;3(1):1–27. http://dx.doi.org/10.1080/03610927408827101.
[27] Hubert L, Arabie P. Comparing partitions. J Classification 1985;2(1):193–218. http://dx.doi.org/10.1007/BF01908075.
[28] Das S, Abraham A, Konar A. Automatic clustering using an improved differential evolution algorithm. IEEE Trans Syst Man Cybern 2008;38(1):218–37. http://dx.doi.org/10.1109/TSMCA.2007.909595.
[29] Xie XL, Beni G. A validity measure for fuzzy clustering. IEEE Trans Pattern Anal Mach Intell 1991;13(8):841–7. http://dx.doi.org/10.1109/34.85677.
[30] Pakhira MK, Bandyopadhyay S, Maulik U. Validity index for crisp and fuzzy clusters. Pattern Recognit 2004;37(3):487–501. http://dx.doi.org/10.1016/j.patcog.2003.06.005.
[31] Das I. On characterizing the "knee" of the Pareto curve based on normal-boundary intersection. Struct Optim 1999;18:107–15. http://dx.doi.org/10.1007/BF01195985.
[32] Bayá AE, Granitto PM. How many clusters: A validation index for arbitrary-shaped clusters. IEEE/ACM Trans Comput Biol Bioinform 2013;10(2):401–14. http://dx.doi.org/10.1109/TCBB.2013.32.
[33] Mikolov T, Le QV, Sutskever I. Exploiting similarities among languages for machine translation. 2013, arXiv e-prints, arXiv:1309.4168.
[34] Rousseeuw PJ. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 1987;20:53–65. http://dx.doi.org/10.1016/0377-0427(87)90125-7.
[35] Aganj I, Harisinghani MG, Weissleder R, Fischl B. Unsupervised medical image segmentation based on the local center of mass. Sci Rep 2018;8(13012). http://dx.doi.org/10.1038/s41598-018-31333-5.
[36] Petrou M, Garcia-Sevilla P. Image processing: dealing with texture. Chichester, England: John Wiley and Sons Ltd; 2006.
[37] Moulavi D, Jaskowiak PA, Campello RJGB, Zimek A, Sander J. Density-based clustering validation. In: SIAM international conference on data mining. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2014, p. 839–47. http://dx.doi.org/10.1137/1.9781611973440.96.
[38] Liang S, Han D, Yang Y. Cluster validity index for irregular clustering results. Appl Soft Comput 2020;95:106583. http://dx.doi.org/10.1016/j.asoc.2020.106583.
[39] Cheng D, Zhu Q, Huang J, Wu Q, Yang L. A novel cluster validity index based on local cores. IEEE Trans Neural Netw Learn Syst 2019;30(4):985–99. http://dx.doi.org/10.1109/TNNLS.2018.2853710.