**The reason why cluster validity indices are wrong**

Eugen-Richard Ardelean1,\*, Mircea Susca2, Raluca Laura Portase1

1Department of Computer Science, Technical University of Cluj-Napoca, Cluj-Napoca, Romania

2Department of Automation, Technical University of Cluj-Napoca, Cluj-Napoca, Romania

**\* Corresponding authors:** [ardeleaneugenrichard@gmail.com](mailto:ardeleaneugenrichard@gmail.com)

**ORCID Author IDs:**

Eugen-Richard Ardelean: 0000-0002-0098-4228

Mircea Susca: 0000-0001-7409-4124

Raluca Laura Portase: 0000-0002-8985-4728

**Abstract:** Cluster validity indices, such as Silhouette, Calinski-Harabasz, and Davies-Bouldin, are consistently used to evaluate the performance of newly developed clustering algorithms and to estimate the number of clusters. In this work, we assess the actual validity of these indices.

**Keywords:** cluster validity index (cvi); external clustering validation index; internal clustering validation index; clustering performance metric; clustering quality; cluster analysis.

# Introduction

## Validation of clustering

Clustering is the unsupervised machine learning task of grouping data points based on similarity (1) with applications spanning diverse domains, such as data mining (1), pattern recognition, bioinformatics, market analysis (1) and many others. As an inherently unsupervised technique, clustering groups sets of data points without requiring prior knowledge of class labels (1–4) with the target of maximizing intra-cluster similarity (5) and minimizing inter-cluster similarity (6). Due to the unsupervised nature of clustering, a significant challenge emerges, specifically determining the quality of the grouping obtained (7) as there is no knowledge of the valid relationships between data points (1). Clustering has been shown to be an NP-hard problem (8–15), as a dataset having *N* data points and *K* clusters results in *KN*possible clustering options (8–15). Moreover, the performance of any clustering algorithm can be affected by several factors: dataset size, cluster overlap, cluster imbalance, dimensionality, and number of clusters (16,17).

The most common option to determine the performance of a clustering algorithm is through Cluster Validity Indices (CVIs) (2,5,11,13,16–21). The evaluation of the results of clustering has received considerable attention along the years and it is still a relevant question today (11,13,16,17). CVIs attempt to create mathematical models to estimate certain desired cluster attributes, such as cohesion (also called homogeneity, compactness, tightness, connectedness) and separation (also referred to as heterogeneity) (3,21,22) with the purpose of measuring how well a certain partitioning of the data reflects the structure of the data (2,13,18–20,22). Most CVIs were designed to address certain data characteristics, and as such they are data dependent (22).

As a result, selecting a specific internal validity index may critically affect the quality of the resultant clusters. Selecting and applying the most suitable CVs for a specific clustering algorithm, particularly when working with datasets of varying characteristics, can significantly enhance the quality of the clustering results produced by that algorithm.

One common use-case of CVIs is to estimate the number of clusters (3,22,23). Clustering algorithms can be reapplied with different parametrization (such as the *k* of K-Means (24)) with the purpose of obtaining different numbers of clusters to evaluate which of these renders the highest score of a chosen CVI (3,13). In such a way, the ‘optimal’ number of clusters may be determined without prior knowledge of the correct number of clusters.

Several studies measure the capability of CVIs to estimate the number of clusters (3,13). However, it was demonstrated that only a small subset of CVIs show invariance to the number of clusters (3). <AND?>

Another common use-case is the evaluation of performance (22) for newly-developed clustering algorithms in comparison to other clustering algorithms (13) based on one or more CVIs. <TBD>

CVIs can be categorized as external or internal (5,20). External indices measure how accurately the clustering labels match the ground truth (a predefined set of labels). In contrast, internal indices measure cluster attributes such as compactness, separation, structure and shape based on the clustering labels without requiring a ground truth (3,25). Internal indices measure cluster attributes based on the concepts (17) of intra-cluster (compactness) and inter-cluster (separability) distances, thus inadvertently evaluating the morphology of the clusters (especially when using the Euclidian distance) resulting in a bias towards the traditional dense well-separated clusters. Thus, there may be cases in which the correct clustering may receive lower performance values than a wrong clustering which results in better values for the cluster attributes estimated by the CVI.

All CVIs have inherent limitations (17,26) and should be used in conjunction with other indices, noting that domain knowledge, dataset characteristics, the type of clustering algorithm applied and distance metrics can all influence the values obtained. It has been shown that generally CVIs tend to fail their purpose in scenarios including non-globular clusters (17,18,20). There is no golden standard regarding CVIs which outperforms all others in all scenarios (23,26–28). In this work, we attempt to convince the reader that most CVIs do not even assign the highest evaluation score to the best clustering (i.e., the ground truth / the true labels / the reference).

In this work, the performance of X 27? CVIs was evaluated and compared using both handcrafted labels and labels obtained from commonly used clustering algorithms on X datasets with a diverse range of characteristics with the purpose of identifying the most appropriate CVI based on the data characteristics and discovering when and why these indices fail. The contributions of this work that distinguish it from prior studies are:

1. Comprehensive comparative analysis: a systematic evaluation of X 27? CVIs including both traditional and recently developed and both internal and external.
2. Diverse array of scenarios and datasets: the comparative analysis employs both synthetic and real datasets with diverse characteristics, such as cluster overlap, cluster imbalance, noise and high dimensionality.
3. Unique evaluation scenarios:
   1. Handcrafted labels showcasing simple scenarios of failure
   2. Analysis of breaking points
   3. Analysis of performance for validity and for estimating the number of clusters
   4. Computational efficiency? Stability across runs? Internal & external?

# Materials and Methods

## Cluster validity indices/metrics

We specify the terms / notations used in the formulas /equations here: the dataset containing *N* data points is divided into clusters , each cluster has points and the a cluster center / centroid , and denotes the Euclidean distance between points and , while in this case represents the centroid of the closest distinct cluster.

|  |  |
| --- | --- |
| **Term / Notation** | **Meaning** |
| N | The number of points in the dataset |
| D | The number of dimensions/features in the dataset |
|  | A point in the dataset |
| K | The number of clusters in the dataset |
|  | The cluster *k* |
|  | The number of points in cluster *k* |
|  | The center/centroid of cluster *k* |

General concepts used in CVI - equations:

Mean/center of data

Cluster mean / center / centroid

Sum-of-Squared Errors = Sum-of-Squares Within / within-cluster sum of squares

Internal cohesiveness/compactness is measured using the squared error function, the sum of squares within (SSW), or the sum of squares error (SSE). Thus, it determines the degree of closeness between the data points. It simply refers to the total of the squared distances between each point in the cluster and its centroid. The partition is indicated as good if the value of its variance is low. Therefore, the value of the compactness measure is minimized (29)

Separation: The external separation is quantified using the sum of squares between (SSB). Where it assumes the distances between separate clusters, typically, it is measured as the sum of the squared distance between the global average point and its centroid. A high value determines separation with high quality. Therefore, the value of the separation measure is maximized (30)

Although, the K-means clustering algorithm does not perform perfectly when the clusters are well-separated (31)

Sum-of-Squares Between / between-cluster sum of squares

--------------------------------------------------------------------

Davies-Bouldin index (DB) (32–34) is computed as the average similarity of clusters. The similarity is computed using the distance between clusters and their sizes. DBS has an inverse performance interval to the other metrics presented in this work. It has only a lower bound at 0, and lower values represent a higher performance. The following formulas describe the computation of this metric:

Here, *R* represents the similarity between clusters *i* and *j*, *si* is the mean of all distances between the points of cluster *i* and its centroid, *di,j* is the distance between clusters *i* and *j* given by their centroids, and *max(Ri,j)* is the maximum similarity of clusters *i* and *j*.

DB is sensitive to the choice of distance metric and to overlapping clusters (17).

Calinski-Harabasz index (CH) (35,36), or Variance Ratio Criterion, is computed as the ratio between the intra-cluster to inter-cluster dispersion. The dispersion is based on the sum of squared distances. For this metric, a higher value indicates a better result and it has no upper bound. The following formula describes the computation of this metric:

Wk = SSE

Bk = SSB

the cluster’s closeness or compactness is measured based on the distance between the cluster’s centroid and the data points within the cluster while the cluster’s separation from other clusters is measured using the distance from the cluster’s centroid to the global centroid

where (𝑆𝑤) is the intra-cluster scatter matrix, (𝑆𝐵 ) is the inter-cluster scatter matrix

Here, *tr(X)* is the trace of the dispersion matrix (either between *Bk* or within *Wk*), *n* is the dataset size and *k* is the number of clusters.

CH is sensitive to irregular cluster shapes and sizes and to outliers (17).

Silhouette index (S) (35,37) is computed as the ratio between the mean distance between a point and the rest of the points of that cluster and the mean distance between the point and all the points of the nearest cluster. SS has an interval of [-1, 1] where 1 represents well-separated dense clusters, 0 overlapping clusters, and -1 an incorrect clustering. Thus, SS evaluates as correct (and outputs higher scores for) the traditional structure of clusters. The following formula describes the computation of this metric:

Here, *b* is the mean of all distances between a point in cluster *i* and all points of the closest cluster *j*, and *a* is the mean of all distances between a point in cluster *i* and all other points in the same cluster.

S index can be sensitive to noise and outliers (17).

**CVI library** (38)

[**https://cluster-validity-indices.readthedocs.io/en/latest/\_autosummary/cvi.modules.html**](https://cluster-validity-indices.readthedocs.io/en/latest/_autosummary/cvi.modules.html)

Xie-Beni index (XB) (39) evaluates the compactness and separation of a clustering solution by comparing total within-cluster compactness to the closest inter-cluster separation. It was originally developed for fuzzy clustering but can be adapted for hard partitions by setting the memberships to 1 for a point’s assigned cluster and 0 otherwise. The index favors clusterings that produce small within-cluster squared errors while keeping cluster centres well separated. Lower XB values indicate better clustering solutions (more compact clusters and larger separation). The hard-partition XB is computed as:

The numerator: total fuzzy within-cluster squared error (weighted by membership degrees ).

XB does not consider the shape or density of clusters, it is sensitive to the number of clusters and its reliance on the Euclidian distance makes it susceptible to the curse of dimensionality (17).

the cluster cohesion is measured using the global mean squared distance of objects from the centroid of their cluster while the inter-cluster separation is measured using the minimum squared distance between pairs of clusters

Centroid-based Silhouette index (cSIL) (37,40) is the cluster-level analogue of SS that uses centroids and mean squared distances instead of pairwise sample distances. For each cluster it compares the cluster’s average squared within-cluster dispersion to the average squared distance from that cluster’s points to the nearest other centroid, producing one silhouette score per cluster which are then averaged. cSIL ranges in [−1,1]: values near 1 indicate well-separated, compact clusters (under the centroid/squared-distance model), values near 0 indicate overlapping or ambiguous clusters, and values near −1 indicate poor clustering (clusters that are closer to other cluster centroids than to their own centroid)

Generalized Dunn’s index 43 (GD43) (41,42) measures cluster quality as the ratio of the closest centre-to-centre separation to the worst (largest) cluster dispersion, where dispersion uses the half-power intra-cluster distances. GD43 is computed as the ratio between the minimum inter-cluster separation (here measured by centroid distance) and the maximum intra-cluster dispersion (a D3-style spread). GD43 is non-negative (no fixed upper bound) where larger values indicate better separation and compactness (values near 0 mean poor separation or a very dispersed cluster). Thus, GD43 rewards partitions whose closest cluster pair is still far relative to the worst cluster spread.

where *N* is the number of data points, *d(i,j)* is the Euclidean distance between data points and , and *max{d(i,k), k!=j}* is the maximum distance between data point *i* and any other data point in a different cluster.

Generalized Dunn’s index 53 (GD53) (41,42)is similar in spirit but replaces the centre-to-centre separation with a size-weighted average half-power distance between two clusters. As with GD43, higher GD53 values indicate better clustering. GD53 can be more robust than GD43 for non-spherical or differently sized clusters because it uses per-cluster half-power averages in the numerator rather than raw centre distances:

Partition Separation (PS) (43) comprises a measure of separation between prototypes / centroids. Larger values of PS indicate better clustering solutions (maximization). PS is computed from prototype (centroid) separations adjusted by cluster size/balance (a prototype-distance based score, with fuzzy and hard variants). PS has no fixed interval (scale depends on data) and higher PS means prototypes are farther apart and cluster sizes more balanced. Thus, PS evaluates prototype separation (not internal compactness) and outputs higher scores for well-spaced, balanced prototype partitions.

Renyi’s representative Cross Information Potential (rCIP) (44) represents each cluster by a Gaussian centered at the cluster centroid with a (regularized) sample covariance , then measures the pairwise *overlap* of those Gaussians. rCIP aggregates the overlaps between all distinct cluster pairs: small overlap means well-separated clusters. rCIP is non-negative (0 = perfect separation) and lower values indicate better clustering. The following formulas describe its computation:

WB-index (WB) (45) measures how good a clustering result is by balancing within-cluster compactness and between-cluster separation. Lower values generally indicate better clustering quality. WBI is computed as a scaled ratio of total within-cluster sum-of-squares to between-cluster dispersion (within / between SS combination, often multiplied or divided by K). WB is positive and smaller values indicate better partitions (it typically exhibits a minimum at the appropriate number of clusters). Thus, WB rewards compact, well-separated clusters (and is commonly used to select K by finding the minimum index).

WB considers average distances and not the shape or distribution of clusters, yet WB found as one of the most robust CVIs in a recent study (17).

**Permetrics library** (46)

DBCV (47) is particularly designed for evaluating density-based clustering algorithms such as DBSCAN (48) The DBCV metric ranges from -1 to 1, with values close to 1 indicating a well-separated cohesive clusters (in terms of density), a value of 0 indicates an ambiguous structure, while near -1 values indicate poor cluster structures. The DBCV metric is calculated using the following formula:

All-points-core-distance (APCD) of an object where KNN(x, i) is the distance between x and its i-th nearest neighbor in cluster k.

The mutual reacheability distance (MRD) between two object xi and xj

Minimum spanning trees (MST) are built for each cluster using MRDs. The density sparseness of a cluster (DSC), while density sparseness between clusters (DSBC) Ci and Cj can be defined as:

Dunn index (D) (42,49) aims to quantify the compactness and separation between clusters in a clustering solution by considering both the distance between points within the same cluster (intra-cluster distance) and the distance between points in different clusters (inter-cluster distance). Specifically. D is the ratio between the minimum distance between any two clusters and the maximum distance found within any cluster. A higher D value indicates better clustering quality, indicating that the clusters are well separated from each other while being compact internally; conversely, a lower D value may indicate that the clusters are too spread out or not well separated. The D index can be sensitive to the data scale and to outliers (17). The following formulas allow for the computation of this index:

Where the dataset is divided into clusters , and denotes the Euclidean distance between points and .

Duda Hart index (DH) (50) is defined as the ratio between the average pairwise distance within clusters and the average pairwise distance between clusters. A lower value of the DHI indicates better clustering, indicating that the clusters are more compact and well-separated. The DHI assumes Euclidian distances, but it can be implemented to use other suitable distance metrics based on the specific problem and data characteristics. The following formulas allow for the computation of DHI:

Where the dataset is divided into clusters , each cluster has points, and denotes the Euclidean distance between points and .

Sum of Squared Error index (SSE) (51) measures the sum of squared distances between each data point and its corresponding centroid or cluster center quantifying the compactness of the clusters. SSEI assigns each data point to its nearest centroid and calculates the squared Euclidean distance between the data point and its assigned centroid, the final value is obtained as the sum of the squared distances for all data points. Higher SSE values indicate higher dispersion or greater variance within the clusters, while lower SSE values indicate more compact and well-separated clusters. The following formulas allow for the computation of SSE:

R-Squared index (RS) is based on the idea of comparing the variance of the data before and after clustering. The RS measures the proportion of the total variance in the data that is explained by the clustering solution. The R-Squared index ranges from -inf to 1, with higher values indicating better clustering solutions. A negative value indicates that the clustering solution is worse than random, while a value of 0 indicates that the clustering solution explains no variance beyond chance. The following formulas allow for the computation of RS:

Where the dataset is divided into clusters , each cluster has points, and denotes the Euclidean distance between points and .

RS assumes a normal distribution (17).

Beale index (B) (52,53), also known as the “variance ratio criterion” or the “F-ratio”, measures the quality of a clustering solution by computing the ratio of the within-cluster sum of squares to the between-cluster sum of squares. The within-cluster sum of squares measures the variability of within each cluster, while the between-cluster sum of squares measures the variability between the clusters. A good clustering solution should have low within-cluster variation and high between-cluster variation, which results in a high B value. The B ranges from 0 to infinity, with higher values indicating better clustering solutions. However, B has a tendency to favor solutions with more clusters. The B can be calculated using the following formula:

Where the dataset containing *N* data points is divided into clusters , each cluster has points, and denotes the Euclidean distance between points and .

Ball Hall index (BH) (52,54) computes the average distance between each data point and its cluster centroid and then averages this across all clusters. It measures the compactness and separation of clusters in a clustering result. A lower BH value indicates better clustering, as it signifies that the data points are closer to their own cluster centroid than to the centroids of other clusters, indicating a clear separation between clusters. The following formula allows for the computation of BH:

Where the dataset containing *N* data points is divided into clusters , each cluster has points, and denotes the Euclidean distance between points and .

Hartigan index (H) (55) also known as the Hartigan’s criterion, is a measure used for evaluating the quality of clustering solutions. It is specifically designed for assessing the goodness of fit of a clustering algorithm, particularly the k-means algorithm. Lower values of the Hartigan index indicate better clustering solutions with lower within-cluster variance and higher separation between clusters. The Hartigan index can be calculated using the following formula:

Where the dataset containing *N* data points is divided into clusters , each cluster has points and the a cluster center / centroid , and denotes the Euclidean distance between points and , while in this case represents the centroid of the closest distinct cluster.

**PyCVI library** (56)

Score Function (SF) (57) is computed from two aggregated distances: a between-cluster term that measures how far cluster centroids lie from the global centroid, and a within-cluster term that measures average point-to-centroid scatter inside clusters. SF maps their difference through a double-exponential squashing function so that larger between-class separation and smaller within-class spread produce higher scores. SF has an interval of , where values close to 1 indicate well-separated, compact clusters, values near 0 indicate poor or overlapping clusters. The following formula describes the computation of this metric:

estimates cluster centroids ‘distances from the global centroids to evaluate the dispersion of clusters from each other. It also evaluates the clusters’ degree of closeness by measuring the distance between the data objects and their respective cluster centroids.

SD (58) is computed as a combination of two complementary terms: the average scattering of clusters, which measures intra-cluster compactness, and the total separation, which quantifies the distances between cluster centroids. Specifically, SD evaluates the trade-off between how compact the clusters are and how well separated they are from each other. Lower SD values indicate compact and well-separated clusters, while higher values denote overlapping or dispersed structures. Thus, SD assesses clustering quality by rewarding high separation and low internal variance. The following formulas describe the computation of this metric:

SDbw (59) evaluates clustering quality based on both cluster compactness and inter-cluster density separation. It combines two complementary components: *scattering*, which measures how compact clusters are relative to the overall dataset, and *density between-within*, which measures how much density exists between clusters compared to within them. Lower SDbw values indicate compact and well-separated clusters, while higher values correspond to overlapping or poorly separated structures. The following formulas describe the computation of this metric

the middle point of the line segment defined by the clusters’ centers

SDbw is sensitive to overlapping clusters and outliers (17).

Xie-Beni\* index (XB\*) (27) is a normalized variant of the original Xie-Beni index designed to reduce sensitivity to data scale and the number of clusters. Like XB, it assesses clustering quality by balancing cluster compactness against inter-cluster separation, but XB\* utilizes the maximum per-cluster average dispersion (it penalizes solutions that have one bad (loose/noisy) cluster even if others are tight - "bottleneck" style measure: the clustering is only as good as its worst cluster). It can be applied to both fuzzy and hard clusterings by defining membership degrees appropriately (1 for the assigned cluster, 0 otherwise in the hard case). The index rewards solutions with tightly grouped data points and well-separated cluster centers. Lower values correspond to higher quality clusterings. The hard-partition XB\* is computed as

Table 1 – Descriptive table of internal CVIs.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Complexity** | **Description** | **Range [worst, best]** | | **REF** | **CREF** |
| DB (↓) |  | Average similarity ratio of each cluster with its most similar cluster, where similarity is defined as the sum of within‐cluster scatter relative to between‐cluster separation. | | (+inf, 0] | (32–34) | (60) |
| CH (↑) |  | Ratio of between‐cluster dispersion to within‐cluster dispersion, normalized by the number of clusters and total points. | | [0, +inf) | (35,36) | (60) |
| S (↑) |  | Average across all data points for the normalized difference between its mean intra‐cluster distance and lowest mean inter‐cluster distance | | [-1, +1] | (35,37) | (60) |
| BH (↓) |  | Mean within-cluster variance (average squared distance of points to their cluster centroids), normalized by the number of clusters. | | (+inf, 0] | (52,54) | (46) |
| XBI (↓) |  | Total (fuzzy) intra-cluster scatter divided by the squared minimum inter-centroid distance. | | (+inf, 0] | (39) | (38) |
| XBI\*(↓) |  | Variant of Xie-Beni that modifies the separation term (e.g., average or normalized inter-centroid distance) to reduce bias. | | (+inf, 0] | (27) | (56) |
| DI (↑) |  | Minimum pairwise inter-cluster distance divided by the maximum intra-cluster diameter | | [0, +inf) | (42,49) | (46) |
| SSE (↓) |  | Sum of squared distances of all points to their assigned cluster centroids (total within-cluster sum of squares). | | (+inf, 0] | (51) | (46) |
| DH (↓) |  | Ratio between the average pairwise distance within clusters and the average pairwise distance between clusters | | (+inf, 0] | (50) | (46) |
| B (↓) |  | Ratio of the within-cluster sum of squares to the between-cluster sum of squares. | | (+inf, 0] | (52,53) | (46) |
| RS (↑) |  | Fraction of total variance explained by the clustering | | (-inf, 1] | - | (46) |
| DBCV (↑) |  | Density-based cluster validity measure comparing intra-cluster density to inter-cluster density | | [-1, +1] | (47) | (46) |
| H (↓) |  | Hartigan criterion measuring change in explained variance when increasing cluster count | | (+inf, 0] | (55) | (46) |
| cSIL (↑) |  | Centroid-based silhouette: per-point silhouette computed using centroid distances instead of full pairwise distances, averaged over points | | [-1, +1] | (37,40) | (38) |
| GD43 (↑) |  | Generalized Dunn variant (4/3) using specific choices for inter-cluster distance and intra-cluster diameter. | | [0, +inf) | (41,42) | (38) |
| GD53 (↑) |  | Generalized Dunn variant (5/3) with different distance/diameter definitions | | [0, +inf) | (41,42) | (38) |
| PS (↑) |  | Aggregate measure of separation between cluster prototypes relative to within-cluster spread | | [0, +inf) | (43) | (38) |
| rCIP (↓) |  | Cross-potential between cluster representative distributions | | (+inf, 0] | (44) | (38) |
| WB (↓) |  | Combined within-cluster scatter measure (sum-of-squares based) | | (+inf, 0] | (45) | (38) |
| SF (↑) |  | Composite validity score of between-cluster term (measures how far cluster centroids lie from the global centroid), and a within-cluster term (measures average point-to-centroid scatter inside clusters) | | [0, +1] | (57) | (56) |
| SD (↓) |  | Combined scattering (average within-cluster spread) and separation term | | (+inf, 0] | (58) | (56) |
| SDbw (↓) |  | Sum of average within-cluster scatter and density-based between-cluster term; | | (+inf, 0] | (59) | (56) |
| C (↓) |  |  | | [+1, 0] | (61) | **-** |
| I (↑) |  |  | | [0, +inf) | (62) | **-** |
| CDbw (↑) |  |  | | [0, +inf) | (63) | **-** |
| VIASCKDE (↑) |  |  | | [−1,+1] | (64) | **-** |
| CS (↓) |  |  | | (+inf, 0] | (65) | **-** |
| COP (↓) |  |  | | (+inf, 0] | (66) | **-** |

COP (66) measures the ratio between intra-cluster compactness (average distance from points to their cluster centroid) and a “farthest-neighbour” inter-cluster distance (for each cluster, the minimum over outside points of the maximum distance from that outside point to members of the cluster). COP takes values in ; values close to 0 indicate tight, well-separated clusters while large values indicate loose or poorly separated clusters. Thus, COP evaluates partitions that have small within-cluster scatter and relatively large farthest-neighbour separation.

C index (61)

can only be used for evaluating the validity of clustering results obtained using the k-means algorithm due to its assumption that the clusters are spherical, equally sized, and have the same density. It is sensitive to the scale of the data, presence of noise and overlapping clusters.

I index (62)

assumes that the clusters are convex and isotropic, which may not always be true in real-world datasets. It requires a large number of trials to find the optimal set of clustering parameters, which can be timeconsuming. It is not effective for datasets with skewed or imbalanced distributions, as it tends to produce similar scores for both wellclustered and poorlyclustered datasets.

CDbw (63) combines an internal-compactness measure with a separation measure that is penalized by density between clusters. Compactness: fraction of each cluster’s points lying within a neighborhood radius of its centroid (averaged). Separation: typical nearest-centroid distance, reduced when there is measurable data density in the midpoint region between centroids. The final score multiplies separation by compactness (squared in your centroid simplification), so high CDbw favors clusterings that are both tight internally and well separated with low inter-cluster density. Higher values indicate better cluster structure.

VIASCKDE index (64)

is an index that is not affected by the cluster shape, and thus, it can make a realistic evaluation of clustering performance regardless of the clusters’ shape. Unlike the existing cluster validation indices, our index calculates the compactness and separation values of the cluster based on calculating the compactness and separation values for each data separately. In other words, it calculates the compactness and separation values of the cluster over the distance of data, independent of parameters such as the cluster center because, in non-spherical clusters, the distance of the data to the closest data is more important than its distance to the cluster center. As can be seen in the example given in figure below, the closest data in the cluster that “it belongs to” is used when calculating the compactness value for the data x. Similarly, the separation value of x is calculated by the distance to the closest data of the cluster that “it does not belong”.

CoSeD—Compactness and Separation Value of a Data): the CoSeD can be described as the compactness and separation value of any data, we calculated the weight of each data that is *WKDE* according to obtained KDE value

CS index (65) is similar in concept to the DB and D indices.

### EDGING DISTANCE

centre\_from\_data

If n = len(data) (size of the subset passed in), computing the n x n pairwise matrix is O(n^2 d) (each pair distance costs d operations). Summation and argmin are O(n^2) dominated by the same.

Time complexity: O(n^2 d)

k\_nearest\_neighbors(data, visited, query\_point, n\_neighbours=3)

* Distance vector: O(n d).
* np.argsort(distances) sorts n values: O(n log n).
* Selecting & masking visited: O(n).

Time complexity: O(n d + n log n)

edging\_distance(X, start, end, n\_neighbours=5, lookahead=10, debug=False)

* start\_id, end\_id discovery: np.where(np.all(X == start,axis=1)) is O(n d) once.
* loop iteration:
  + np.count\_nonzero(visited) used several times → each call O(n) (this happens every loop in your code).
  + Either k\_nearest\_neighbors(...) (cost O(n d + n log n) per call) **or** np.where(visited==False) (O(n)).
  + Distance computation from the selected nearest points to end: O(|nearest| \* d) (≤ O(n d) worst-case).
  + visited[nearest\_points\_ids] = True and path.append are up to O(|nearest|) each.

Worst-case iteration count: the while loop can in principle run up to O(n) iterations (e.g., you visit or mark a small number each time). In the pathological/worst case, each iteration invokes a full k\_nearest\_neighbors call and several O(n) scans.

Worst-case per iteration cost: O(n d + n log n + n) ≈ O(n d + n log n).

Worst-case total cost (current code): O(n \* (n d + n log n)) = O(n^2 d + n^2 log n).

**ed\_silhouette\_score**

1. cluster\_means For each cluster i of size n\_i, centre\_from\_data is O(n\_i^2 d). Summed over clusters this is O(sum\_i n\_i^2 d). Worst case (one big cluster): O(n^2 d). Balanced clusters: ≈ O(n^2 d / k).
2. points\_to\_clusters This does n \* k calls to edging\_distance. Using the worst-case cost from (3), each call can be O(n^2 d). That gives O(k \* n \* n^2 d) **= O(k n^3 d)** worst-case.
3. All later operations (computing intra/inter cluster distances, silhouette coefficients) are O(n k) or O(n) and negligible versus the dominant term.

Dominant term: O(k n^3 d)

**ed\_davies\_bouldin\_score**

1. cluster\_means: same as before O(sum m\_i^2 d) ⇒ worst O(n^2 d).
2. cluster\_distances double loop for i in range(k): for j in range(k): cluster\_distances[i,j] = edging\_distance(data, cluster\_means[i], cluster\_means[j]) — up to O(k^2) calls to edging\_distance, each O(n^2 d) ⇒ O(k^2 n^2 d).
3. cluster\_scatter = [ mean(edging\_distance(data, sample, cluster\_means[i]) for sample in cluster\_i ) for i in range(k) ] — this is one edging\_distance call per sample (since each sample is compared with its cluster mean). So n calls each O(n^2 d) ⇒ O(n^3 d).
4. The DB index combine step is O(k^2).

Dominant term: O(n^3 d) (from cluster scatter)

**ed\_calinski\_harabasz\_score**

1. cluster\_means & overall\_mean: O(n^2 d) each (via centre\_from\_data) → O(n^2 d).
2. between\_cluster\_ss = sum(edging\_distance(cluster\_mean[i], overall\_mean) \* count\_i) — k calls to edging\_distance: O(k n^2 d).
3. within\_cluster\_ss = sum(edging\_distance(sample, cluster\_mean[i]) for all samples) — n calls, each O(n^2 d) ⇒ O(n^3 d).

Dominant term: O(n^3 d) from within-cluster distances. So ed\_calinski\_harabasz\_score = O(n^3 d) (worst case).

## General observations about indices

CHS does not work well for highly irregular shapes and can be influenced by outliers (17), DBS is sensitive to the parameter choice of the clustering algorithm, to non-Euclidean distance metrics, and to overlapping clusters (17), and SS may not be optimal in scenarios with noise and outliers, it requires a distance metric and for large datasets the distance matrix computation may be expensive (17).

While, the sum of squared errors (SSE) may also be a potential clustering metric, it has been shown to have limited usefulness in non-globular clusters (67).

The I index was found to be a better option than DB and D (62).

However, SS demonstrated excellent performance in a benchmark of validity indices (21). SS, DBS and CHS are still used today in the evaluation of clustering performance (68). Moreover, an extensive comparative study has shown that SS has the best results (20) and it indicated that some of the best performing clustering metrics are indeed SS, CHS and DBS (20).

Another study (16) showed that CH obtained the highest performance, with S and DB being top ranked as well (others top ranked SF). DB has been found as the most using CVI for meta-heuristic-based clustering algorithms (21).

Another study found that S obtained the best performance regardless of the amount of overlap, CH had a good performance on linearly separable data, while DB and GD indices had an adequate performance, yet all CVIs performed poorly on non-linearly separable datasets due to the inherent assumptions of data distribution (11).

A study (3) of 68 internal CVIs on 21 synthetic and real datasets based on the K-means clustering algorithm found that the RS, H, DB, and COP indices were top-performing with XB and CH having an acceptable evaluation of performance and the S index had the most consistent behaviour across datasets. It was also shown that only 14 out of 68 CVIs are actually invariant to the number of clusters (3) raising the question whether CVIs are actually useful in determining the unknown number of clusters.

The DBCV (47) index has been found to outperform (69) other indices such as, CDbw (63), DCSI (70), and VIASCKDE (64), for non-convex and density-based (such as those obtained by DBSCAN) clusters. Although all these indices degrade with an increase in noise (69), DBCV has been found to be the most reliable/consistent and it has been shown to not be affected by the spatial arrangement of clusters (69).

**Needless to say, there does not seem to be a consensus in the literature.**

## Datasets

Table X – Dataset information

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Characteristics** | **# Samples** | **# Dimensions** | **# Cluster (cluster size)** | **REF** |
| **Name** | **Characteristics** |  |  |  | **REF** |
| A1-3 | -Varying # cluster | 3000/5250/7500 | 2 | 20/30/50  (150) | (71,72) |
| S1-4 | Overlap (increasing overlap) | 5000 | 2 | 15  (333) | (73) |
| Unbalance | Imbalance | 6500 | 2 | 8  (3x 2000,  5x 100) | (4) |
| G2 10-90 | Overlap  (increasing overlap) | 2048 | 2 | 2  (2x 1024) | (72,74) |
|  |  |  |  |  |  |

## Measures

Imbalance ratio as majority\_class\_size / minority\_class\_size

|  |
| --- |
| FUNCTION imbalance\_ratio(X, labels):  counts = count occurrences of each distinct label in labels  max\_count = maximum value in counts  min\_count = minimum value in counts  IF min\_count == 0:  RETURN +infinity  RETURN max\_count / min\_count |

Overlap ratio as fraction of points whose nearest \*other\* center ≤ slack \* dist\_to\_own\_center:

|  |
| --- |
| FUNCTION overlap\_ratio(X, labels, slack = 1.2):  unique\_labels = ordered list of distinct labels  centers = empty list  FOR each label IN unique\_labels:  points = subset of X with that label  center = mean vector of points  append center TO centers  N = number of rows in X  overlapping\_count = 0  FOR i FROM 1 TO N:  p = row i of X  own\_idx = index of label of p in unique\_labels  dist\_to\_own = EuclideanDistance(p, centers[own\_idx])  dist\_to\_nearest\_other = +infinity  FOR j FROM 1 TO length(centers):  IF j == own\_idx: CONTINUE  d = EuclideanDistance(p, centers[j])  IF d < dist\_to\_nearest\_other:  dist\_to\_nearest\_other = d  IF dist\_to\_nearest\_other <= slack \* dist\_to\_own:  overlapping\_count = overlapping\_count + 1  overlap\_rate = overlapping\_count / N  RETURN overlap\_rate |

# Results

## A

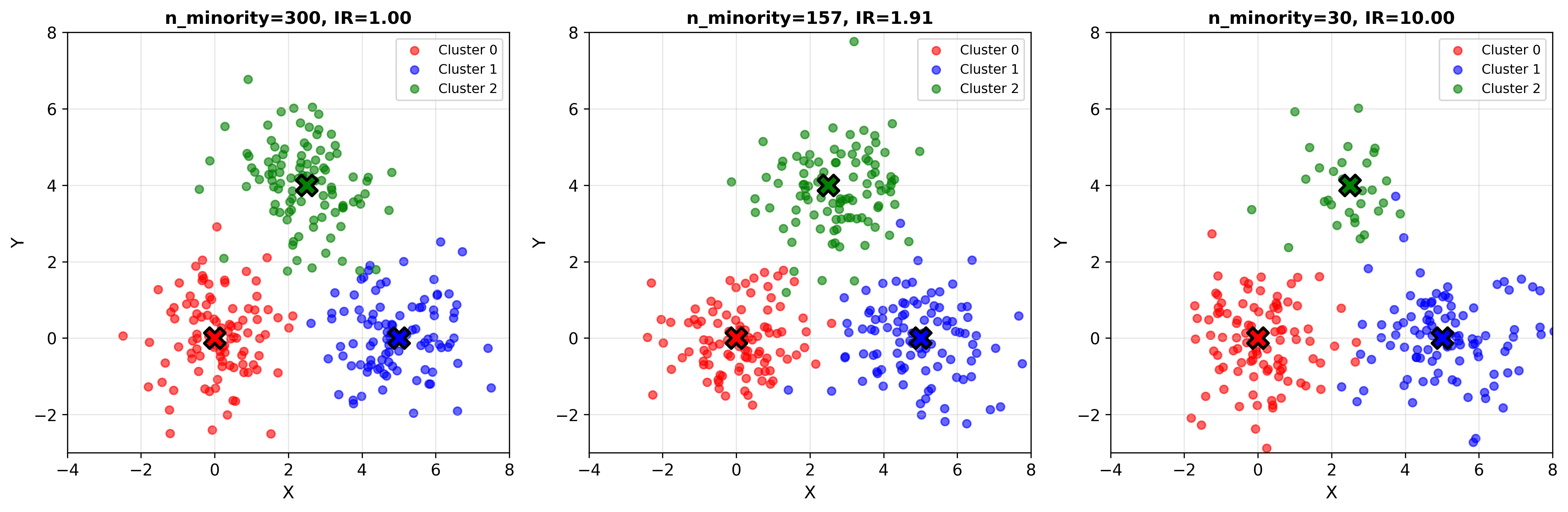
DBCV makes analysis with correlation to ARI, should we?

## Can we use Silhouette to find the number of clusters

Let us take a simple dataset and vary the amount of imbalance by regenerating a dataset with 3 clusters varying the number of points of a single cluster (green).

We can easily view that the Imbalance Ratio measure computes correctly the amount of imbalance as the size of minority class decreases, the imbalance ratio increases.





We can see that the Silhouette Score is able to correctly evaluate the true labels with a higher score for most cases of imbalance. However, for heavily imbalanced datasets, where close sparse clusters are present it may have erroneous evaluation as shown:

A diagram of red and blue dots

AI-generated content may be incorrect.

Let us take a simple dataset and vary the amount of overlap between two out of 3 clusters

We can easily view that the Overlap Ratio measure computes correctly (some variations due to regeneration of clusters) the amount of overlap as the distance between clusters decreases, the overlap ratio increases.

A graph with a line

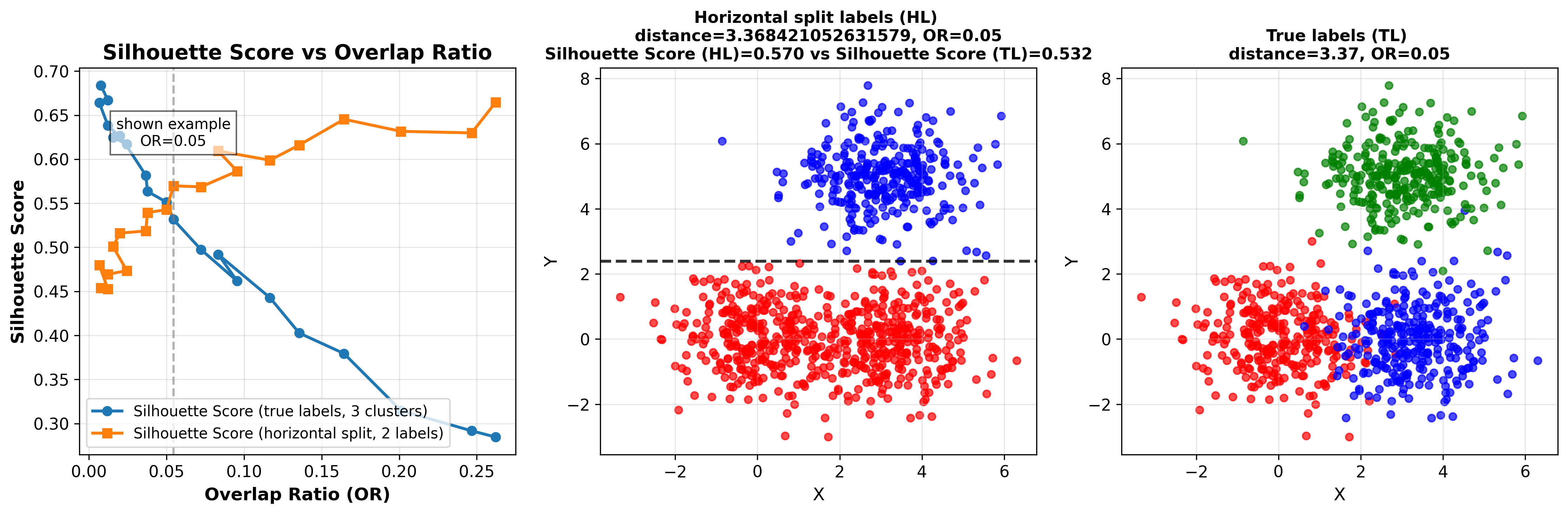
AI-generated content may be incorrect.

Examples based on various amounts of overlap between red and blue clusters:

A graph with colored dots

AI-generated content may be incorrect.

There is a point at which even though some clustering algorithms would obtain a decent result regardless of the overlap at which a Silhouette analysis would be unable to correctly determine the number of clusters.



## Errors of external metrics

Analysis on unbalance(4)-like dataset.

A graph of a graph with red and green lines

AI-generated content may be incorrect.

Almost perfect score with random labels on the clusters with a small count.

A screenshot of a graph

AI-generated content may be incorrect.

A group of colored dots

AI-generated content may be incorrect.

Studies (3,47) have used external metrics to validate the results of internal metrics through correlations or other methods, however due to the erroneous assessments of imbalanced clusters, it raises the question whether these results can be fully taken into account.

Q: If internal metrics have an issue with overlap and not imbalance, yet external have with imbalance not overlap? How does this affect the statement?

Extending external CVIs for balanced evaluations:

|  |
| --- |
| FUNCTION BalancedCVI(CVI, true\_labels, pred\_labels):  classes ← unique values in true\_labels  IF size(classes) = 1:  RETURN 1.0  class\_scores ← empty list  FOR each c IN classes:  mask ← (true\_labels == c)  true\_binary ← mask as {0,1}  pred\_subset ← pred\_labels[mask]  most\_common\_cluster ← argmax\_count(pred\_subset)  pred\_binary ← (pred\_labels == most\_common\_cluster) as {0,1}  score\_c ← CVI(true\_binary, pred\_binary)  append score\_c TO class\_scores  RETURN mean(class\_scores) |

This approach handles dataset imbalance by turning the global clustering comparison into a set of independent, per-class evaluations and then treating every class as equally important. Instead of scoring the clustering on all samples at once—where large classes contribute many more sample-level comparisons and therefore dominate the final index, drowning errors on smaller classes—it converts each true class into a binary task (in-class vs out-of-class), finds the predicted cluster that best captures that class, computes the chosen external CVI for that binary pair, and records one score per true class. Several options are available for the aggregation of per-class scores: macro averaging, weighted averaging, harmonic, etc. In this work, we present the macro averaging, a simple arithmetic mean of per-class scores, in which each class contributes the same amount to the overall metric regardless of how many samples it contains.

Nevertheless, there are trade-offs. Equal weighting makes the metric sensitive to noise in very small classes (a single mislabel can swing that class’s score), and choosing only the single most-overlapping predicted cluster may hide situations where a true class is split across several predicted clusters (also known as over-clustering). These are inherent consequences of the local, per-class scoring strategy as it prioritizes per-class fairness over global, sample-frequency-driven accuracy. We recommend the use of this approach when the evaluation is required to reflect per-class performance (for fairness, rare-class detection, or importance of small groups).

## Estimating the ‘correct’ number of clusters

Q: Using ‘best partition’ as reference not ground truth (3), is it better?

This analysis plainly shows that the best partition as defined by a CVI can be incorrect, thus, using the best partition as a reference to validate the performance of CVIs may include even more errors in the evaluation. In this work, we have chosen the ground truth as reference with the aim of finding the CVIs that are capable of assigning the highest score to the ground truth for a plethora of diverse datasets, providing an empiric proof of their performance.

A close-up of a graph

AI-generated content may be incorrect.

Maybe it’s our predefined labels, let us look at how actual labels from clustering algorithms perform.

# Conclusions

CVIs are sensitive to the number of clusters, perhaps too sensitive to use them as validation for the choice of this parameter.

Internal CVIs tend to be sensitive to cluster overlap resulting in erroneous performance evaluations.

External CVIs tend to be sensitive to cluster imbalance resulting in erroneous performance evaluations.

Bibliography

1. Linoff GS, Michael J. A. Berry. Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management. 3rd ed. Wiley Publishing; 2011. 888 p.

2. Halkidi M, Batistakis Y, Vazirgiannis M. On Clustering Validation Techniques. J Intell Inf Syst. 2001 Dec 1;17(2):107–45.

3. Todeschini R, Ballabio D, Termopoli V, Consonni V. Extended multivariate comparison of 68 cluster validity indices. A review. Chemom Intell Lab Syst. 2024 Aug 15;251:105117.

4. Rezaei M, Fränti P. Set Matching Measures for External Cluster Validity. IEEE Trans Knowl Data Eng. 2016 Aug;28(8):2173–86.

5. Theodoridis S, Koutroumbas K. Chapter 16 - Cluster Validity. In: Theodoridis S, Koutroumbas K, editors. Pattern Recognition (Fourth Edition) [Internet]. Boston: Academic Press; 2009 [cited 2025 Nov 13]. p. 863–913. Available from: https://www.sciencedirect.com/science/article/pii/B9781597492720500189

6. Guha S, Rastogi R, Shim K. CURE: an efficient clustering algorithm for large databases. SIGMOD Rec. 1998 Jun 1;27(2):73–84.

7. Ramze Rezaee M, Lelieveldt BPF, Reiber JHC. A new cluster validity index for the fuzzy *c*-mean. Pattern Recognit Lett. 1998 Mar 1;19(3):237–46.

8. Dasgupta S. The hardness of k-means clustering.

9. Deshpande A, Louis A, Singh AV. arXiv.org. 2018 [cited 2025 Nov 13]. On Euclidean $k$-Means Clustering with $\alpha$-Center Proximity. Available from: https://arxiv.org/abs/1804.10827v3

10. Mahajan M, Nimbhorkar P, Varadarajan K. The planar k<math><mi is="true">k</mi></math>-means problem is NP-hard. Theor Comput Sci. 2012 Jul 13;442:13–21.

11. José-García A, Gómez-Flores W. A survey of cluster validity indices for automatic data clustering using differential evolution. In: Proceedings of the Genetic and Evolutionary Computation Conference [Internet]. New York, NY, USA: Association for Computing Machinery; 2021 [cited 2025 Nov 6]. p. 314–22. (GECCO ’21). Available from: https://doi.org/10.1145/3449639.3459341

12. Falkenauer E. Genetic Algorithms and Grouping Problems. USA: John Wiley & Sons, Inc.; 1998. 238 p.

13. Gagolewski M, Bartoszuk M, Cena A. Are cluster validity measures (in) valid? Inf Sci. 2021 Dec 1;581:620–36.

14. Aloise D, Deshpande A, Hansen P, Popat P. NP-hardness of Euclidean sum-of-squares clustering. Mach Learn. 2009 May 1;75(2):245–8.

15. Garey M, Johnson D, Witsenhausen H. The complexity of the generalized Lloyd - Max problem (Corresp.). IEEE Trans Inf Theory. 1982 Mar;28(2):255–6.

16. Ikotun AM, Habyarimana F, Ezugwu AE. Benchmarking validity indices for evolutionary K-means clustering performance. Sci Rep. 2025 Jul 1;15(1):21842.

17. Hassan BA, Tayfor NB, Hassan AA, Ahmed AM, Rashid TA, Abdalla NN. From A-to-Z review of clustering validation indices. Neurocomputing. 2024 Oct 7;601:128198.

18. Maulik U, Bandyopadhyay S. Performance evaluation of some clustering algorithms and validity indices. IEEE Trans Pattern Anal Mach Intell. 2002 Dec;24(12):1650–4.

19. Milligan GW, Cooper MC. An examination of procedures for determining the number of clusters in a data set. Psychometrika. 1985 Jun 1;50(2):159–79.

20. Arbelaitz O, Gurrutxaga I, Muguerza J, Pérez JM, Perona I. An extensive comparative study of cluster validity indices. Pattern Recognit. 2013 Jan 1;46(1):243–56.

21. Ikotun AM, Habyarimana F, Ezugwu AE. Cluster validity indices for automatic clustering: A comprehensive review. Heliyon. 2025 Jan 30;11(2):e41953.

22. Xu R, Xu J, Wunsch DC. A Comparison Study of Validity Indices on Swarm-Intelligence-Based Clustering. IEEE Trans Syst Man Cybern Part B Cybern. 2012 Aug;42(4):1243–56.

23. Dimitriadou E, Dolničar S, Weingessel A. An examination of indexes for determining the number of clusters in binary data sets. Psychometrika. 2002 Mar 1;67(1):137–59.

24. MacQueen J. Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics [Internet]. University of California Press; 1967 [cited 2025 Sep 3]. p. 281–98. Available from: https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Proceedings-of-the-Fifth-Berkeley-Symposium-on-Mathematical-Statistics-and/chapter/Some-methods-for-classification-and-analysis-of-multivariate-observations/bsmsp/1200512992

25. Ardelean ER, Portase RL, Potolea R, Dînșoreanu M. A path-based distance computation for non-convexity with applications in clustering. Knowl Inf Syst. 2025 Feb 1;67(2):1415–53.

26. Brun M, Sima C, Hua J, Lowey J, Carroll B, Suh E, et al. Model-based evaluation of clustering validation measures. Pattern Recognit. 2007 Mar 1;40(3):807–24.

27. Kim M, Ramakrishna RS. New indices for cluster validity assessment. Pattern Recognit Lett. 2005 Nov 1;26(15):2353–63.

28. Li H, Zhang S, Ding X, Zhang C, Dale P. Performance Evaluation of Cluster Validity Indices (CVIs) on Multi/Hyperspectral Remote Sensing Datasets. Remote Sens. 2016 Apr;8(4):295.

29. Said AB, Hadjidj R, Foufou S. Cluster validity index based on Jeffrey divergence. Pattern Anal Appl. 2017 Feb 1;20(1):21–31.

30. ElMorshedy MM, Fathalla R, El-Sonbaty Y. Feature Transformation Framework for Enhancing Compactness and Separability of Data Points in Feature Space for Small Datasets. Appl Sci. 2022 Jan;12(3):1713.

31. Fränti P, Sieranoja S. K-means properties on six clustering benchmark datasets. Appl Intell. 2018 Dec 1;48(12):4743–59.

32. Halkidi M, Batistakis Y, Vazirgiannis M. On Clustering Validation Techniques. J Intell Inf Syst. 2001 Dec 1;17(2):107–45.

33. Caliński T, JA H. A Dendrite Method for Cluster Analysis. Commun Stat - Theory Methods. 1974 Jan 1;3:1–27.

34. Davies DL, Bouldin DW. A Cluster Separation Measure. IEEE Trans Pattern Anal Mach Intell. 1979 Apr;PAMI-1(2):224–7.

35. Rosenberg A, Hirschberg J. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In 2007. p. 410–20.

36. Rendón E, Abundez I, Arizmendi A, Quiroz EM. Internal versus External cluster validation indexes. 2011;5(1):8.

37. Rousseeuw PJ. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math. 1987 Nov 1;20:53–65.

38. cvi: A Python package for both batch and incremental cluster validity indices. [Internet]. [cited 2025 Nov 4]. Available from: https://github.com/AP6YC/cvi

39. Xie XL, Beni G. A validity measure for fuzzy clustering. IEEE Trans Pattern Anal Mach Intell. 1991 Aug;13(8):841–7.

40. Rawashdeh M, Ralescu A. Center-Wise Intra-Inter Silhouettes. In: Hüllermeier E, Link S, Fober T, Seeger B, editors. Scalable Uncertainty Management. Berlin, Heidelberg: Springer; 2012. p. 406–19.

41. Bezdek JC, Pal NR. Some new indexes of cluster validity. IEEE Trans Syst Man Cybern Part B Cybern. 1998 Jun;28(3):301–15.

42. Dunn JC. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. J Cybern. 1973 Jan;3(3):32–57.

43. Yang MS, Wu KL. A new validity index for fuzzy clustering. In: 10th IEEE International Conference on Fuzzy Systems (Cat No01CH37297) [Internet]. 2001 [cited 2025 Nov 4]. p. 89–92 vol.1. Available from: https://ieeexplore.ieee.org/document/1007254

44. Moshtaghi M, Bezdek JC, Erfani SM, Leckie C, Bailey J. Online Cluster Validity Indices for Streaming Data [Internet]. arXiv; 2018 [cited 2025 Nov 4]. Available from: http://arxiv.org/abs/1801.02937

45. Zhao Q, Xu M, Fränti P. Sum-of-Squares Based Cluster Validity Index and Significance Analysis. In: Kolehmainen M, Toivanen P, Beliczynski B, editors. Adaptive and Natural Computing Algorithms. Berlin, Heidelberg: Springer; 2009. p. 313–22.

46. Thieu NV. PerMetrics: A Framework of Performance Metrics for Machine Learning Models. J Open Source Softw. 2024 Mar 9;9(95):6143.

47. Moulavi D, Jaskowiak PA, Campello RJGB, Zimek A, Sander J. Density-Based Clustering Validation. In: Proceedings of the 2014 SIAM International Conference on Data Mining (SDM) [Internet]. Society for Industrial and Applied Mathematics; 2014 [cited 2025 Nov 4]. p. 839–47. (Proceedings). Available from: https://epubs.siam.org/doi/10.1137/1.9781611973440.96

48. Ester M, Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. Portland, Oregon: AAAI Press; 1996. p. 226–31. (KDD’96).

49. Dunn† JC. Well-Separated Clusters and Optimal Fuzzy Partitions. J Cybern. 1974 Jan 1;4(1):95–104.

50. Thompson M, Duda RO, Hart PE. Pattern Classification and Scene Analysis. In: Leonardo [Internet]. 1974 [cited 2025 Nov 7]. p. 370. Available from: https://www.jstor.org/stable/1573081?origin=crossref

51. Zhao Q, Xu M, Fränti P. Sum-of-Squares Based Cluster Validity Index and Significance Analysis. In: Kolehmainen M, Toivanen P, Beliczynski B, editors. Adaptive and Natural Computing Algorithms. Berlin, Heidelberg: Springer; 2009. p. 313–22.

52. Milligan GW, Cooper MC. An examination of procedures for determining the number of clusters in a data set. Psychometrika. 1985 Jun 1;50(2):159–79.

53. Beale EML. Euclidean Cluster Analysis. Scientific Control Systems Limited; 1969. book.

54. Ball G, Hall DJ. ISODATA, A NOVEL METHOD OF DATA ANALYSIS AND PATTERN CLASSIFICATION. In 1965 [cited 2025 Nov 4]. Available from: https://www.semanticscholar.org/paper/ISODATA%2C-A-NOVEL-METHOD-OF-DATA-ANALYSIS-AND-Ball-Hall/7dac28166b91d89ef6c38cf4fbb9f647b1d73c61

55. Hartigan JA. Clustering Algorithms. 99th ed. USA: John Wiley & Sons, Inc.; 1975. 351 p.

56. Galmiche N. PyCVI: A Python package for internal Cluster Validity Indices, compatible with time-series data. J Open Source Softw. 2024 Oct 8;9(102):6841.

57. Saitta S, Raphael B, Smith IFC. A Bounded Index for Cluster Validity. In: Perner P, editor. Machine Learning and Data Mining in Pattern Recognition. Berlin, Heidelberg: Springer; 2007. p. 174–87.

58. Halkidi M, Vazirgiannis M, Batistakis Y. Quality Scheme Assessment in the Clustering Process. In: Zighed DA, Komorowski J, Żytkow J, editors. Principles of Data Mining and Knowledge Discovery. Berlin, Heidelberg: Springer; 2000. p. 265–76.

59. Halkidi M, Vazirgiannis M. Clustering validity assessment: finding the optimal partitioning of a data set. In: Proceedings 2001 IEEE International Conference on Data Mining [Internet]. 2001 [cited 2025 Nov 4]. p. 187–94. Available from: https://ieeexplore.ieee.org/document/989517

60. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. J Mach Learn Res. 2011;12(85):2825–30.

61. Hubert LJ, Levin JR. A general statistical framework for assessing categorical clustering in free recall. Psychol Bull. 1976;83(6):1072–80.

62. Maulik U, Bandyopadhyay S. Performance evaluation of some clustering algorithms and validity indices. IEEE Trans Pattern Anal Mach Intell. 2002 Dec;24(12):1650–4.

63. Halkidi M, Vazirgiannis M. A density-based cluster validity approach using multi-representatives. Pattern Recognit Lett. 2008 Apr 15;29(6):773–86.

64. Şenol A. VIASCKDE Index: A Novel Internal Cluster Validity Index for Arbitrary-Shaped Clusters Based on the Kernel Density Estimation. Comput Intell Neurosci. 2022;2022(1):4059302.

65. Chou CH, Su MC, Lai E. A new cluster validity measure and its application to image compression. Pattern Anal Appl. 2004 Jul 1;7(2):205–20.

66. Gurrutxaga I, Albisua I, Arbelaitz O, Martín JI, Muguerza J, Pérez JM, et al. SEP/COP: An efficient method to find the best partition in hierarchical clustering based on a new cluster validity index. Pattern Recognit. 2010 Oct 1;43(10):3364–73.

67. Ansari Z, Azeem MF, Ahmed W, Babu AV. Quantitative Evaluation of Performance and Validity Indices for Clustering the Web Navigational Sessions [Internet]. arXiv; 2015 [cited 2025 Aug 7]. Available from: http://arxiv.org/abs/1507.03340

68. Wang Z, Ye C. Deep Clustering Evaluation: How to Validate Internal Clustering Validation Measures [Internet]. arXiv; 2024 [cited 2025 Sep 20]. Available from: http://arxiv.org/abs/2403.14830

69. Chicco D, Sabino G, Oneto L, Jurman G. The DBCV index is more informative than DCSI, CDbw, and VIASCKDE indices for unsupervised clustering internal assessment of concave-shaped and density-based clusters. PeerJ Comput Sci. 2025 Aug 29;11:e3095.

70. Gauss J, Scheipl F, Herrmann M. DCSI -- An improved measure of cluster separability based on separation and connectedness [Internet]. arXiv; 2025 [cited 2025 Nov 14]. Available from: http://arxiv.org/abs/2310.12806

71. Kärkkäinen I, Fränti P. A Dynamic local search algorithm for the clustering problem. In 2002 [cited 2025 Nov 13]. Available from: https://www.semanticscholar.org/paper/A-Dynamic-local-search-algorithm-for-the-clustering-K%C3%A4rkk%C3%A4inen-Fr%C3%A4nti/4c236eaf6a1db6fec5f032f0bceb3ffd54080aee

72. Gagolewski M. A framework for benchmarking clustering algorithms. SoftwareX. 2022 Dec 1;20:101270.

73. Fränti P, Virmajoki O. Iterative shrinking method for clustering problems. Pattern Recognit. 2006 May 1;39(5):761–75.

74. Fränti P, Mariescu-Istodor R, Zhong C. XNN Graph. In: Robles-Kelly A, Loog M, Biggio B, Escolano F, Wilson R, editors. Structural, Syntactic, and Statistical Pattern Recognition. Cham: Springer International Publishing; 2016. p. 207–17.

75. Kelly, Markelle, Longjohn, Rachel, Nottingham Kolby. Home - UCI Machine Learning Repository [Internet]. [cited 2024 Mar 23]. Available from: https://archive.ics.uci.edu/

# Data Availability

The datasets used in this work are openly available and can be found at:

* Synthetic datasets (72): <https://github.com/gagolews/clustering-data-v1/>
* Real datasets (75): <https://archive.ics.uci.edu/datasets>

# Code Availability

The code that supports the findings of this work was written in Python and is openly available at: <https://github.com/ArdeleanRichard/Clustering-Validity>/.

# Competing Interests

The authors have declared that no competing interests exist.

# Authors’ contributions

Conceptualization, E.-R.A. and R.L.P.; methodology, E.-R.A. and R.L.P.; software, E.-R.A. and R.L.P.; validation, E.-R.A. and R.L.P.; formal analysis, E.-R.A. and R.L.P.; investigation, E.-R.A. and R.L.P.; data curation, E.-R.A. and R.L.P.; writing—original draft preparation, E.-R.A. and R.L.P.; writing—review and editing, E.-R.A. and R.L.P.; visualization, E.-R.A. and R.L.P.; supervision, E.-R.A. and R.L.P.; project administration, E.-R.A. and R.L.P. All authors have read and agreed to the published version of the manuscript.

# Funding

This work is supported by the project "Romanian Hub for Artificial Intelligence-HRIA", Smart Growth, Digitization and Financial Instruments Program, MySMIS no. 334906.