Are Cluster Validity Measures (In)valid?

Marek Gagolewski[a,c,1], Maciej Bartoszuk[b], Anna Cena[b]

[a] *Deakin University, School of Information Technology, Geelong, VIC 3220, Australia*
[b] *Warsaw University of Technology, Faculty of Mathematics and Information Science, ul. Koszykowa 75, 00-662 Warsaw, Poland*
[c] *Systems Research Institute, Polish Academy of Sciences ul. Newelska 6, 01-447 Warsaw, Poland*

Abstract

Internal cluster validity measures (such as the Caliński–Harabasz, Dunn, or Davies–Bouldin indices) are frequently used for selecting the appropriate number of partitions a dataset should be split into. In this paper we consider what happens if we treat such indices as objective functions in unsupervised learning activities. Is the optimal grouping with regards to, say, the Silhouette index really meaningful? It turns out that many cluster (in)validity indices promote clusterings that match expert knowledge quite poorly. We also introduce a new, well-performing variant of the Dunn index that is built upon OWA operators and the near-neighbour graph so that subspaces of higher density, regardless of their shapes, can be separated from each other better.

Keywords: clustering methodology, cluster validity index, Dunn index, nearest neighbours (NNs), ordered weighted averaging (OWA) operator, no free lunch

1. Introduction

An internal cluster validity index (CVI for short; see, e.g., [2, 29, 42, 43, 58]) is – in theory – a measure of how well a given partitioning of a dataset reflects the underlying structure of the modelled domain. CVIs are frequently employed as tools for selecting the appropriate number of clusters a dataset should be segmented into [43]. By re-applying some algorithm (e.g.,

[*1] Corresponding author; email: m.gagolewski@deakin.edu.au

k-means or spectral methods), one can determine the splits into 2-, 3-, 4-, . . . , disjoint and nonempty subsets, compute the corresponding CVIs, and select the partition that maximises a chosen utility measure.

Here we shall focus on the other popular use case thereof. Some practitioners utilise CVIs to compare the outputs of different algorithms on the same dataset. Is the partition that the average linkage method returned better than that yielded by the DBSCAN algorithm (provided that they are of equal cardinality)? Similarly, researchers use CVIs for evaluating new clustering algorithms the same way: a new method X^{++} produces partitions that have a higher average Caliński–Harabasz indices (on some benchmark datasets) than procedures X, Y, and Z, thus "proving" its superiority. However, we would like to call this methodology into question, especially because CVIs in general constitute an extremely diverse set of measures.

Thus, we shall be interested in determining which of the popular CVIs are particularly suitable or unsuitable for judging the quality of different partitions of the same cardinality. Does a high value of a CVI make sense at all? Can it really be treated as an indicator of a useful clustering result?

To address these questions, we shall find the partitions that yield the highest possible index values, for a large number of datasets and CVIs. In other words, we will treat each CVI as an objective function to be maximised over the *whole* space of *all* possible clusterings.

Our assumption here is that a cluster validity measure can only be considered meaningful whenever it is maximised at the partitions closely resembling the reference ones. Otherwise stated, good CVIs should promote results that agree with expert knowledge.

In the course of our study, which we of course detail in the sequel, we have discovered that this is often very much not the case – see Figures 1 and 2 for two quite representative graphical examples. It turns out that some CVIs promote highly overlapping groupings while other ones work better as outlier detectors. One should thus not uncritically believe that a high value of, e.g., a generalised Dunn index *GDunn_d2_D1* (see Section 2) is better than a lower one; at the bottom-right subfigures we see that this index promotes some rather random partitions as "best".

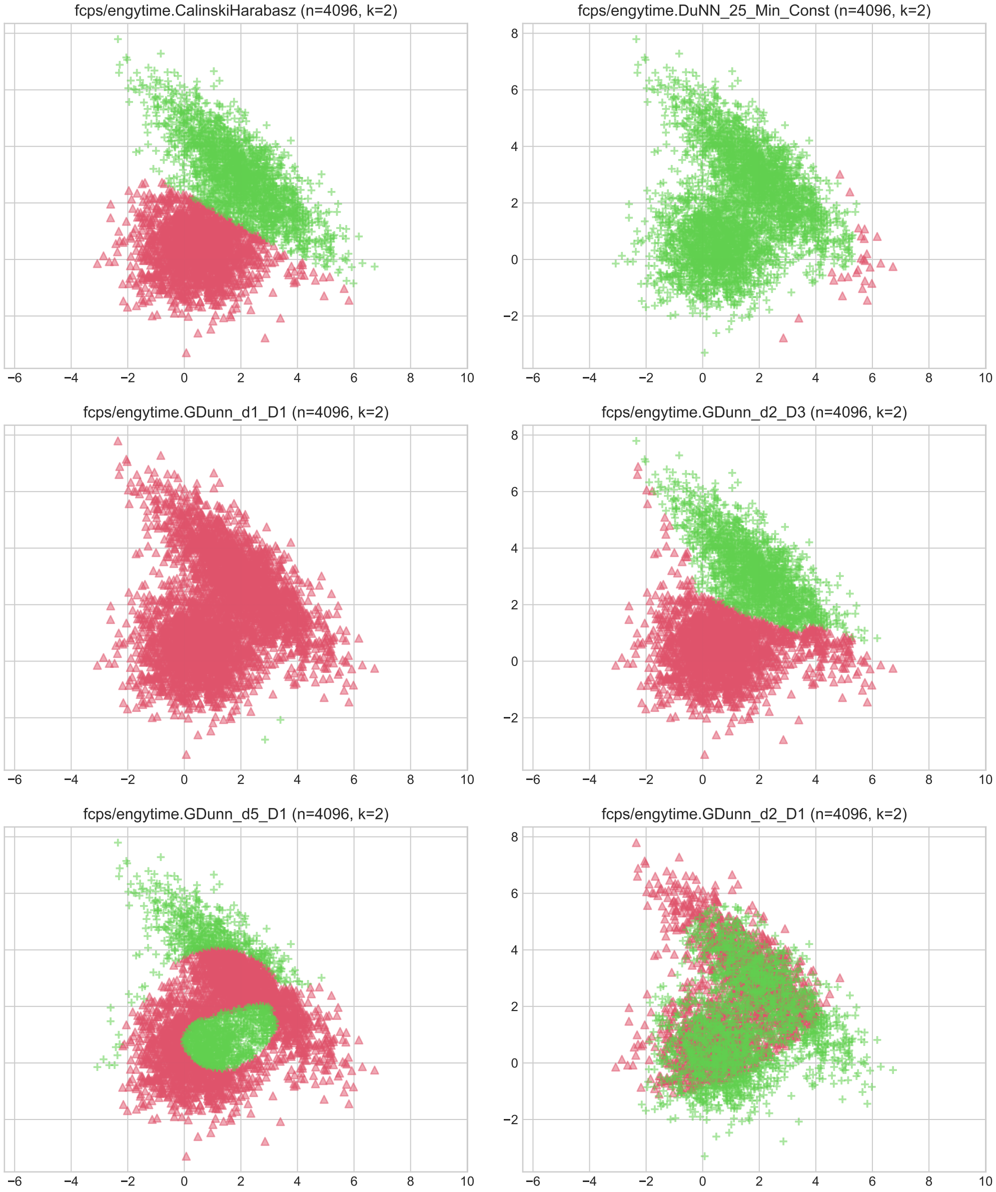

Figure 1: *fcps/engytime* dataset: Optimal clusters as seen by 6 different cluster validity indices (see Section 2 for more details). The reference partitions consist of two (clearly visible if viewed in colour) Gaussian blobs; the Caliński–Harabasz index (top-left subfigure) identifies them quite correctly. Some indices promote very peculiar, overlapping groupings (e.g., *GDunn_d2_D1*), other ones should rather be employed as outlier detectors (like *GDunn_d1_D1*).

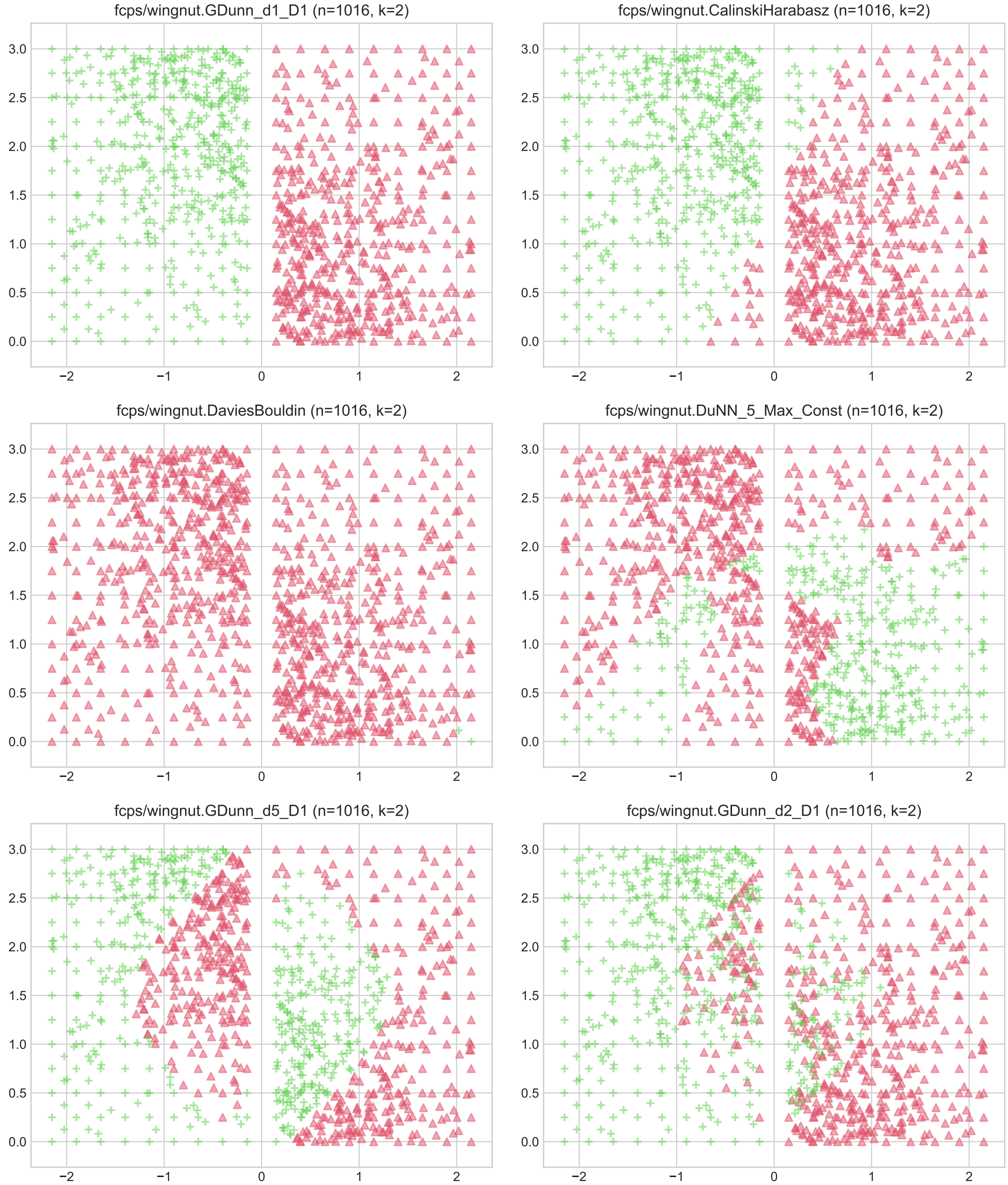

Figure 2: *fcps/wingnut* dataset: Optimal clusters as seen by 6 different cluster validity indices (discussed in Section 2). The two point clouds are well-separable; *GDunn_d1_D1* (top-left subfigure) identified them correctly. However, certain CVIs favour some rather unusual cluster shapes instead.

Let us note that papers introducing new CVIs are plentiful and new ones are being published on a regular basis, see, e.g., [39, 40] for some recent examples. More often than not, such indices are designed to be “the best” for a given particular situation and/or they aim to “eliminate” certain deficiencies with the previous measures. Because of this, the number of CVIs to choose from can be overwhelming, see Section 2 and, e.g., [56, 58], for up-to-date overviews.

This is why comprehensive, in-depth comparative studies are so important. There are quite a few interesting surveys of various properties of cluster validity measures in different contexts and settings, but we have only found [59] somewhat methodologically relevant to the task at hand. There, the performance of 8 CVIs was studied but the authors’ focus seems slightly shifted towards the analysis of the stability of the solutions generated by what they call differential-evolution–particle-swarm-optimisation algorithms, which of course do not necessarily guarantee finding patterns that are optimal in the eye of a specific cluster validity measure.

Further, in [2] and [42], the authors have shown that most of the CVIs work well with spherical clusters but fail in other types of data. A CVI may fail to assign the highest evaluation to the partition that fits the data best, e.g., the reference one. The conclusion is that there is no single internal cluster validation index that outperforms the other indices everywhere. Similar conclusions were reached in [7], where a model-based study of the correlations of CVIs with carefully chosen error rates across the outputs of different clustering algorithms was conveyed. For more studies in similar spirit, see [14, 33, 38, 43]. Note again that our task is to find the optimal partition under the guidance of a given CVI (amongst the set of all partitions), and not to assess a fixed set of particular clusterings.

We are well aware of the fact that the sheer act of optimising of CVIs is, overall, not a new idea. For instance, the within-cluster sum of squares (WCSS), which is the basis for the Caliński–Harabasz index is used as the objective function in the k-means [41] algorithm.

Other algorithms employing some goodness-of-split measures include, e.g., fuzzy c-means [5] that features a smoothened variant of WCSS, ITM (Information-Theoretic-MST; [46]) which applies a divisive scheme over an Euclidean minimum spanning tree to optimise an information-theoretic criterion, finding Gaussian mixtures via expectation-maximisation, and the generalisation of the Ward linkage in the form of the Lance–Williams formulae [35].

Nevertheless, as n data items enjoy $\Omega(k^n)$ possible k-partitionings (Stirling number of the second kind), the problem of optimising a general CVI over the whole search space is very difficult computationally (provably hard in the case of the said WCSS, see [1, 25], amongst many others).

Because of this, many algorithms which are defined as minimisers/maximisers of some CVI, use a variety of simplifications, approximations, or heuristics, e.g., they optimise the objective over reduced search spaces or apply some greedy strategies. For instance, to find a partition optimising the aforementioned WCSS, Ward in [57] suggested to build a hierarchy of clusters in an agglomerative way, and Edwards and Sforza in [19] as well as Caliński and Harabasz in [8] proposed to employ some divisive schemes. However, such heuristics are usually limited to specific CVIs; in this paper we would like to go far beyond that.

Also notice that in the literature we may of course find numerous techniques constructed as a combination of different metaheuristics-based optimisation procedures and chosen cluster validity measures playing the role of objective functions, see, e.g., [13, 30, 34, 48, 52, 61]. Their respective authors often claim that this way they create "new" clustering algorithms (e.g., optimise WCSS using particle swarms vs by means of differential evolution), but it is semantically not quite appropriate. Rather, they should be thought of as ways to test the performance of the optimisation algorithms themselves (with CVIs computed over particular datasets serving as benchmark objectives as in [31]).

Due to the intrinsic difficulty of optimising CVIs, it is no wonder that such a comprehensive study as the current one has not been performed yet. Interestingly, it will turn out that the methodology we have employed, despite its still being based on some approximations, is sufficient for achieving our goal.

Also it is worth mentioning that, up to date, the variety of studies of the various aspects of CVIs (which we shall review in the next section), was quite limited because of the lack of a larger, standardised benchmark set batteries. Most studies considered few datasets, either synthetic, or inherently difficult to partition (such as the UCI [17] datasets). Luckily, thanks to the recent notable efforts by the authors of [21, 28, 55] and our new battery that aggregates and extends them [24], studies such as this one finally become possible. Furthermore, we propose a unique approach where we account for the fact that a dataset can exhibit multiple equally valid clusterings (as discussed also in [11]).

Let us cast a glance at the structure of this paper. In Section 2 we review some of the most notable cluster validity indices. Furthermore, we propose a new, wide class of CVIs generalising the Dunn index which is based on the notion of ordered weighted averaging (OWA) operators and near-neighbour (NN) graphs. In Section 3 we describe the methodology we have applied in order to answer our main research question, including the description of the benchmark datasets used, the approach to identify the partitions that are optimal from the perspective of a given cluster validity measure, and ways to determine the extent to which they agree with expert knowledge. In Section 4 we present the results of our empirical study, e.g., explore the relationship between cluster compactness or separability and what is considered a good clustering by experts. Also, we perform a cluster analysis of clustering algorithms to determine which methods are most similar to each other. We conclude the paper in Section 5.

2. Cluster validity indices

Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ denote the input dataset comprised of n points in a d-dimensional Euclidean space, with $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,d})$ denoting the i-th point, $i \in [1:n] = \{1, 2, \dots, n\}$.

We shall be looking for a partition of $\mathbf{X}$ into $k \geq 2$ nonempty, mutually disjoint clusters, with k fixed in advance. Note that a k-partition $\{X_1, \dots, X_k\}$ of a set $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ can be encoded by means of a surjection $C : [1:n] \overset{\text{onto}}{\to} [1:k]$, where $C(i) \in [1:k]$ gives the cluster number of the i-th point. Let us denote the set of all such possible mappings with $\mathcal{C}_k$.

For the sake of clarity and simplicity, we will only be focused on cluster validity indices based on Euclidean distances between all pairs of points, $\|\mathbf{x}_i - \mathbf{x}_j\|$, or the input points and some other pivots, such as their corresponding cluster centroids, $\|\mathbf{x}_i - \boldsymbol{\mu}_j\|$, where $\mu_{j,l} = \frac{1}{|X_j|}\sum_{\mathbf{x}_i \in X_j} x_{i,l}$. The fixation of the distance metric is not at all restrictive, as various transformations can be applied onto $\mathbf{X}$ at the data pre-processing stage, including variable selection, standardisation, outlier removal, feature engineering (by means of spectral/kernel-based methods), etc., see [15, 16, 44], amongst others.

We shall consider 52 different internal cluster validity indices like $I : \mathcal{C}_k \to$

$\mathbb{R}$. Apart from the most popular, classical CVIs, we also bring forth our own proposal.

2.1. CVIs based on cluster centroids

1,2) BallHall, CalińskiHarabasz. Let $\boldsymbol{\mu}$ denote the centroid of the whole $\mathbf{X}$. The two following indices are based on within-cluster sum of squares (WCSS), which itself can be rewritten in terms of the squared Euclidean distances between the points and their respective centroids.

The Ball–Hall index [3] is the WCSS weighted by the cluster cardinality:

$$\mathrm{BallHall}(C) = -\sum_{i=1}^{n} \frac{1}{|X_{C(i)}|} \|\mathbf{x}_i - \boldsymbol{\mu}_{C(i)}\|^2. \tag{1}$$

Note the minus that accounts for the fact that in Section 3.3 we want all the indexes be maximised.

Then the Caliński–Harabasz index [8, Eq. (3)] ("variance ratio criterion") is given by:

$$\mathrm{Cali\acute{n}skiHarabasz}(C) = \frac{n-k}{k-1} \frac{\sum_{i=1}^{n} \|\boldsymbol{\mu} - \boldsymbol{\mu}_{C(i)}\|^2}{\sum_{i=1}^{n} \|\mathbf{x}_i - \boldsymbol{\mu}_{C(i)}\|^2}. \tag{2}$$

It may be shown that the task of minimising the (unweighted) WCSS is equivalent to maximising the Caliński–Harabasz index. Hence, this index is precisely the objective function in k-means [41] and the algorithms by Ward, Edwards and Cavalli-Sforza, etc., see [8, 19, 57].

3) DaviesBouldin. The Davies–Bouldin [12, Def. 5] index also refers to the notion of cluster centroids. It is given as the average similarity between each cluster and its most similar counterpart (note the minus sign again):

$$\mathrm{DaviesBouldin}(C) = -\frac{1}{k} \sum_{i=1}^{k} \left(\max_{j \neq i} \frac{s_i + s_j}{m_{i,j}} \right), \tag{3}$$

where s_i is the dispersion of the i-th cluster: if $|X_i| > 1$, it is given by $s_i = \frac{1}{|X_i|} \sum_{\mathbf{x}_u \in X_i} \|\mathbf{x}_u - \boldsymbol{\mu}_i\|$ and otherwise we set $s_i = \infty$. Furthermore, $m_{i,j}$ is the intra-cluster distance, $m_{i,j} = \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|$. In [12], other choices of s_i and $m_{i,j}$ are also suggested; here, we choose the most popular setting (used, e.g., in [2]).

2.2. Silhouettes

4, 5) Silhouette, SilhouetteW. In [54, Sec. 2], Rousseeuw proposes the notion of a silhouette as a graphical aid in cluster analysis.

Denote the average dissimilarity between the i-th point and all other points in its own cluster with:

$$a_i = \frac{1}{|X_{C(i)}| - 1} \sum_{\mathbf{x}_u \in X_{C(i)}} \|\mathbf{x}_i - \mathbf{x}_u\| \tag{4}$$

and the average dissimilarity between the i-th point and all other entities in the "closest" cluster with:

$$b_i = \min_{j \neq C(i)} \left(\frac{1}{|X_j|} \sum_{\mathbf{x}_v \in X_j} \|\mathbf{x}_i - \mathbf{x}_v\| \right). \tag{5}$$

Then the *Silhouette* index is defined as the average silhouette score:

$$\mathrm{Silhouette}(C) = \frac{1}{n} \sum_{i=1}^{n} \frac{b_i - a_i}{\max\{a_i, b_i\}}, \tag{6}$$

with convention $\pm\infty/\infty = 0$.

The same paper also defines what we call here the *SilhouetteW* index, being the mean of the cluster average silhouette widths:

$$\mathrm{SilhouetteW}(C) = \frac{1}{k - s} \sum_{i=1}^{n} \frac{1}{|X_{C(i)}|} \frac{b_i - a_i}{\max\{a_i, b_i\}}, \tag{7}$$

where s is the number of singletons. Note that *SilhouetteW*, just like *BallHall*, employs weighting by cluster cardinalities.

2.3. Generalised Dunn indices

6–20) GDunn_dX_DY. In [18, Eq. (3)], Dunn proposed an index defined as the ratio between the smallest between-cluster distance and the largest cluster diameter. It has been generalised by Bezdek and Pal in [6] as:

$$\mathrm{GDunn}(C) = \frac{\min_{i \neq j} d\left(X_i, X_j\right)}{\max_i D\left(X_i\right)}. \tag{8}$$

The numerator measures the between-cluster separation whilst the denominator quantifies the cluster compactness.

Function d was assumed in [6] one of:

- $d_1(X_i, X_j) = \mathrm{Min}\left(\{\|\mathbf{x}_u - \mathbf{x}_v\| : \mathbf{x}_u \in X_i, \mathbf{x}_v \in X_j\}\right)$,
- $d_2(X_i, X_j) = \mathrm{Max}\left(\{\|\mathbf{x}_u - \mathbf{x}_v\| : \mathbf{x}_u \in X_i, \mathbf{x}_v \in X_j\}\right)$,
- $d_3(X_i, X_j) = \mathrm{Mean}\left(\{\|\mathbf{x}_u - \mathbf{x}_v\| : \mathbf{x}_u \in X_i, \mathbf{x}_v \in X_j\}\right)$,
- $d_4(X_i, X_j) = \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|$,
- $d_5(X_i, X_j) = \frac{|X_i|\,\mathrm{Mean}(\{\|\mathbf{x}_u - \boldsymbol{\mu}_i\| : \mathbf{x}_u \in X_i\}) + |X_j|\,\mathrm{Mean}\left(\{\|\mathbf{x}_v - \boldsymbol{\mu}_j\| : \mathbf{x}_v \in X_j\}\right)}{|X_i| + |X_j|}$.

Bezdek and Pal in [18] considered also d_6 based on the Hausdorff metric but this will be omitted here as it turned out too slow to compute.

On the other hand, D was chosen amongst:

- $D_1(X_i) = \mathrm{Max}\left(\{\|\mathbf{x}_u - \mathbf{x}_v\| : \mathbf{x}_u, \mathbf{x}_v \in X_i\}\right)$,
- $D_2(X_i) = \mathrm{Mean}\left(\{\|\mathbf{x}_u - \mathbf{x}_v\| : \mathbf{x}_u, \mathbf{x}_v \in X_i\}\right)$,
- $D_3(X_i) = \mathrm{Mean}\left(\{\|\mathbf{x}_u - \boldsymbol{\mu}_i\| : \mathbf{x}_u \in X_i\}\right)$.

There are 15 different combinations of the possible numerators and denominators in our study, hence 15 different CVIs, which we will denote as *GDunn_dX_DY*. In particular, *GDunn_d1_D1* gives the original Dunn [18] index.

2.4. CVIs based on near-neighbour graphs

Let $\mathrm{NN}_M(i) = \{j_1, \dots, j_M\}$ denote the set of the i-th point's M nearest neighbours, $0 < \|\mathbf{x}_i - \mathbf{x}_{j_1}\| < \dots < \|\mathbf{x}_i - \mathbf{x}_{j_M}\|$ (assuming there are no tied distances, otherwise, some small random noise can be added).

21–50) DuNN_OWAs_OWAc. Note that the original Dunn index (denoted *GDunn_d1_D1* above) can be viewed as:

$$\mathrm{Dunn}(C) = \frac{\mathrm{Min}\left(\{\|\mathbf{x}_i - \mathbf{x}_j\| : C(i) \neq C(j)\}\right)}{\mathrm{Max}\left(\{\|\mathbf{x}_i - \mathbf{x}_j\| : C(i) = C(j)\}\right)}. \tag{9}$$

Here we propose the following generalisation of the above – a generalised Dunn-type index based on the notion of the M-near-neighbour graph and ordered weighted averaging [60] operators – convex combinations (weighted sums) of ordered inputs. Namely:

$$\mathrm{DuNN}(C) = \frac{\mathrm{OWA}_s\left(\{\|\mathbf{x}_i - \mathbf{x}_j\| : C(i) \neq C(j), i \in \mathrm{NN}_M(j) \text{ or } j \in \mathrm{NN}_M(i)\}\right)}{\mathrm{OWA}_c\left(\{\|\mathbf{x}_i - \mathbf{x}_j\| : C(i) = C(j), i \in \mathrm{NN}_M(j) \text{ or } j \in \mathrm{NN}_M(i)\}\right)}. \tag{10}$$

As a measure of cluster separation we aggregate the ordered between-point distances but only provided that they are part of the near-neighbour graph. This will enable us to take the local point density into account and detect well-separable clusters of even quite sophisticated shapes. In a similar manner, cluster compactness will be based on the nearest neighbours as well.

Below we shall study pairs of OWA_s and OWA_c chosen amongst:

- Min,
- Max,
- Mean,
- $\mathrm{SMin}_\delta(q_1, q_2, \dots, q_z) = \sum_{i=1}^{z} w_{i,z} q_{(i)}$, with $w_{i,z} = \frac{\psi(i;z,\delta)}{\sum_{j=z-3\delta+1}^{z} \psi(j;z,\delta)}$ for $i > z - 3\delta$ and 0 otherwise ("smooth minimum"),
- $\mathrm{SMax}_\delta(q_1, q_2, \dots, q_z) = \sum_{i=1}^{z} w_{i,z} q_{(i)}$, with $w_{i,z} = \frac{\psi(i;z,\delta)}{\sum_{j=1}^{3\delta} \psi(j;1,\delta)}$ for $i \leq 3\delta$ and 0 otherwise ("smooth maximum"),

where $q_{(1)} \geq q_{(2)} \geq \cdots \geq q_{(z)}$ and $\psi(\cdot; \mu, \sigma)$ denotes the probability density function of the normal distribution with expectation μ and standard deviation σ, see also [10].

For instance, *DuNN_25_SMin:5_Max* denotes a generalised Dunn index based on each point's 25 nearest neighbours. It uses SMin_5 as a separation measure (computed over a subset of $25n$ distances restricted to the pair of points belonging to different clusters) and Max as a measure of compactness (the remainder of the $25n$ distances comprised of point pairs belonging to the same clusters). Moreover, we will study indices like *DuNN_25_Mean_Const*, where the denominator is fixed at 1.

In the sequel we will consider $M = 5$ and $M = 25$. In order to keep the number of cases within reasonable limits, we will restrict ourselves to 30 different CVIs of this type (see Table 3 for a complete listing).

51,52) WCNN_M. The within-cluster near-neighbours (WCNN) index is parametrised by $M \geq 1$. It aims to reflect how many nearest neighbours of every point actually belong to the very same cluster:

$$\mathrm{WCNN}(C) = \frac{|C(i) = C(j) : j \in \mathrm{NN}_M(i)|}{nM}. \tag{11}$$

Ideally, $\mathrm{WCNN}(C) = 1$. Hence, this is a measure of how well the clusters are separated from each other.

Additionally, to prevent the formation of small clusters, we will assume $\mathrm{WCNN}(C) = -\infty$ whenever there is a cluster of cardinality $\leq M$. Similarly as above, we shall consider $M \in \{5, 25\}$.

3. Method

3.1. What is a valid cluster validity index?

As we have proclaimed in the introduction, our key assumption in this paper is that a meaningful cluster validity measure I should be high whenever it is asked to assess the quality of one of the reference partitions, and lower if it is applied on other clusterings. In other words, useful CVIs should encourage the results that agree with expert knowledge.

In order to be able to answer our main research question, i.e., which cluster validity measures are valid, we need the following components:

- benchmark data sets for evaluating the methods (Section 3.2),
- a procedure for finding the partition that maximises a given CVI on each dataset (Section 3.3),
- a measure for quantifying the degree of agreement between what a CVI thinks is a good partition vs what experts have to say on this matter (Section 3.4).

3.2. Benchmark Datasets

We shall use an extensive battery of clustering benchmarks [24][1], which not only combines data that have already been used in a number of studies [17, 20, 21, 28, 32, 55], but also features new test sets.

[1] Available at `https://github.com/gagolews/clustering_benchmarks_v1`.

Table 1: Benchmark datasets studied, see [24] and https://github.com/gagolews/clustering_benchmarks_v1 for their visual depictions; l gives the number of reference partitions and ks denote their possible cardinalities.

	dataset	ks	l
1.	*fcps/atom*	2	1
2.	*fcps/chainlink*	2	1
3.	*fcps/engytime*	2	2
4.	*fcps/hepta*	7	1
5.	*fcps/lsun*	3	1
6.	*fcps/target*	2, 6	2
7.	*fcps/tetra*	4	1
8.	*fcps/twodiamonds*	2	1
9.	*fcps/wingnut*	2	1
10.	*graves/dense*	2	1
11.	*graves/fuzzyx*	2, 4, 5	6
12.	*graves/line*	2	1
13.	*graves/parabolic*	2, 4	2
14.	*graves/ring*	2	1
15.	*graves/ring_noisy*	2	1
16.	*graves/ring_outliers*	2, 5	2
17.	*graves/zigzag*	3, 5	2
18.	*graves/zigzag_noisy*	3, 5	2
19.	*graves/zigzag_outliers*	3, 5	2
20.	*other/chameleon_t4_8k*	6	1
21.	*other/chameleon_t5_8k*	6	1
22.	*other/hdbscan*	6	1
23.	*other/iris*	3	1
24.	*other/iris5*	3	1
25.	*other/square*	2	1
26.	*sipu/aggregation*	7	1
27.	*sipu/compound*	4, 5, 6	5
28.	*sipu/flame*	2	2
29.	*sipu/jain*	2	1
30.	*sipu/pathbased*	3, 4	2
31.	*sipu/r15*	8, 9, 15	3
32.	*sipu/spiral*	3	1
33.	*sipu/unbalance*	8	1
34.	*uci/ecoli*	8	1
35.	*uci/ionosphere*	2	1
36.	*uci/sonar*	2	1
37.	*uci/statlog*	7	1
38.	*uci/wdbc*	2	1
39.	*uci/wine*	3	1
40.	*uci/yeast*	10	1
41.	*wut/circles*	4	1
42.	*wut/cross*	4	1
43.	*wut/graph*	10	1
44.	*wut/isolation*	3	1
45.	*wut/labirynth*	6	1
46.	*wut/mk1*	3	1
47.	*wut/mk2*	2	1
48.	*wut/mk3*	3	1
49.	*wut/mk4*	3	1
50.	*wut/olympic*	5	1
51.	*wut/smile*	4, 6	2
52.	*wut/stripes*	2	1
53.	*wut/trajectories*	4	1
54.	*wut/trapped_lovers*	3	1
55.	*wut/twosplashes*	2	1
56.	*wut/windows*	5	1
57.	*wut/x1*	3	1
58.	*wut/x2*	3	1
59.	*wut/x3*	4	1
60.	*wut/z1*	3	1
61.	*wut/z2*	5	1
62.	*wut/z3*	4	1

Most importantly, each benchmark dataset comes with a set of $l \geq 1$ reference labels that were assigned by experts. The case $l > 1$ reflects the situation where there might be multiple valid/plausible/useful partitions (compare, e.g., [11]); we are dealing with an unsupervised learning problem after all.

The original benchmark battery consists of 79 data instances, however 16 datasets are accompanied by labels that yield $n(k-1) > 50{,}000$; they were omitted for their computation would be too lengthy (namely: *mnist/digits*, *mnist/fashion*, *other/chameleon_t7_10k*, *other/chameleon_t8_8k*, *sipu/a1*, *sipu/a2*, *sipu/a3*, *sipu/birch1*, *sipu/birch2*, *sipu/d31*, *sipu/s1*, *sipu/s2*, *sipu/s3*, *sipu/s4*, *sipu/worms_2*, *sipu/worms_64*). Also *uci/glass* has been removed as one of its 25-near-neighbour graph's connected components was too small for the NN-based methods to succeed. This leaves us with 62 datasets in total, see Table 1.

Further, all columns of 0 variance were removed and a tiny amount of noise (Gaussian with $\mu = 0$ and σ equal to 10^{-6} of each column's sample standard deviation) was added so as to assure the uniqueness of the clustering results.

3.3. Finding optimal partitions (w.r.t. a given CVI)

From now on we assume that the reader is familiar with the basics of the language of mathematical programming, see, e.g., [37, 49] for a comprehensive overview.

For a predefined $\mathbf{X}$ and k, let us fix a cluster validity measure $I : \mathcal{C}_k \to \mathbb{R}$. Without loss in generality, we assume that the higher the I, the more useful the partition. This is because we can always take $I := -I$, as we have done with the Ball–Hall and Davies–Bouldin indices above.

For a given k-partition C, let NEIGHBOURS(C) denote the set of all surjections like $C' : [1:n] \overset{\text{onto}}{\to} [1:k]$ with $C'(i) \neq C(i)$ for some i and $C'(j) = C(j)$ for all $j \neq i$. In other words, it is the set of all k-partitions that can be obtained from C by relocating a single point to some other cluster.

We are interested in finding a partition which is a solution to the optimisation problem:

$$\underset{C \in \mathcal{C}_k}{\text{maximise}}\, I(C), \tag{12}$$

i.e., $C^* \in \mathcal{C}_k$ such that $I(C^*) \geq I(C)$ for all $C \in \mathcal{C}_k$.

Remark 1. *The solution to* (12) *is not unique; clusterings are defined up to a permutation of the cluster numbers (IDs). For example, a 2-partition of a 4-ary set encoded like* $(C(1), C(2), C(3), C(4)) = (1, 1, 2, 1)$ *is semantically equivalent to* $(2, 2, 1, 2)$*. Moreover, it might happen that a dataset exhibits a number of equally good splits. This is exactly the case when we apply WCNN_M on datasets whose* M*-near-neighbour graphs are disconnected and the number of connected components is greater than* k*. For instance, assuming* Y_1, Y_2, Y_3 *are disconnected, in this setting the 2-partition* $\{Y_1 \cup Y_2, Y_3\}$ *is as good as* $\{Y_1, Y_2 \cup Y_3\}$*.*

In general, the combinatorial optimisation problem (12) is extremely difficult to solve in practice. Enumerating all the possible solutions is virtually impossible as the number of possible partitions is equal to the Stirling number of the second kind, $S(n, k) = \frac{1}{k!} \sum_{i=0}^{k} (-1)^i \binom{k}{i} (k - i)^n$ which is $O(k^n)$, and note that in our case $2 \leq k \ll n$.

In this paper, however, we shall make *reasonable efforts* towards finding the maximum of the objective (12). In essence, for each dataset we will generate dozens of "interesting" partitions (using existing state-of-the art clustering algorithms and evolutionary-based heuristics, see Section 3.5) each of which we shall then try to improve with an expansive variant of the steepest ascent hill climbing (with tabu [27] search-like memoisation) that itself guarantees to land in a local maximum of the objective (12).

The maximum of the objective (12) will be sought by means of the following variant of the hill climbing scheme.

Algorithm 1. *With* $\{C_1, \ldots, C_m\}$ *let us denote the set of initial candidate solutions (see Section 3.5 for more details) ordered in such a way that* $I(C_1) \geq \cdots \geq I(C_m)$*. For brevity of notation, we assume that* $I(\emptyset) = -\infty$*.*

```
In: $I$ :, $C_1, C_2, \ldots, C_m$, $P \in \mathbb{N}$;
  1. $\mathcal{T} = \emptyset$;                                   (a "tabu" list)
  2. $C^* = C_1$;                                         (best solution so far)
  3. for $C = C_1, C_2, \ldots, C_m$ do:                 ($I(C_1) \geq \cdots \geq I(C_m)$)
     3.1. $p = 1$;
     3.2. $C^+ = \emptyset$;
     3.3. for each $C' \in$ NEIGHBOURS($C$) do:
        3.3.1. if $C' \notin \mathcal{T}$ and $I(C') > I(C^+)$, then $C^+ = C'$;
```

```
    3.4. if $C^+ = \emptyset$ then continue to step 3;      (cannot improve further)
    3.5. $\mathcal{T} = \mathcal{T} \cup \{C^+\}$;                        (never visit $C^+$ again)
    3.6. $C = C^+$;
    3.7. if $I(C) > I(C^*)$, then $C^* = C$, else $p = p + 1$;
    3.8. if $p \le P$, then go to step 3.2;  (try to improve current $C^+$ next)
4. return $C^*$;
```

It is easily seen that the algorithm guarantees that the solution returned cannot be further improved by relocating an individual point to a different cluster. Hence, the return value is definitely a local maximum, however there is of course no guarantee that the identified optimum is global. As we argue below, though, it will turn out sufficient for our purposes.

We shall set the upper bound for the number of iterations without improvement, P, to 250 (we have rarely seen any improvements beyond 100, though). This allows for the procedure to explore the area around the candidate solutions quite broadly. Note that the $\mathcal{T}$ set, which guarantees that no partition is considered twice, is shared across all the iterations so that the visited subspace is even broader. Hence, the search is more comprehensive than if we had restarted the whole procedure independently for each $C_1, \dots, C_m$ and then chose the best amongst the identified local maxima.

Remark 2. *Note that the number of points in* NEIGHBOURS(C) *is* $O(n(k-1))$. *Overall, the procedure for certain CVIs can be sped up by computing* $I(C^+)$ *incrementally based on* $I(C')$ *and the knowledge of which point is being relocated to which cluster. For instance, the Silhouette index only requires an* $O(nk)$ *update instead of a full recompute worth of* $O(n^2)$ *time. The CVIs we have considered gave the time complexity of steps 3.1–3.8 most often lying between* $O(n^2k^2)$ *and* $O(n^3k^2)$ *(with constants* d *and* M *having some obvious influence as well). The typical size of* $\mathcal{T}$ *at the end of the algorithm's run (i.e., the number of executions of step 3.5) when started from* $m = 5$ *random points was 1000–2000.*

3.4. Measuring similarity to reference partitions

For a given benchmark dataset $\mathbf{X}$, let $C_1^\$, C_2^\$, \dots, C_l^\$$ be the reference partitions and $k_1, \dots, k_l$ be their respective cardinalities.

Note that any clustering method $\mathfrak{c}$ (for example, the maximiser of the Caliński–Harabasz index or the Ward linkage) can be thought of as a function that takes $\mathbf{X}$ and k_i on input and yields a k_i-partition of $\mathbf{X}$ on output, i.e., $\mathfrak{c}(\mathbf{X}, k_i) \in \mathcal{C}_{k_i}$.

We will use the adjusted Rand index (ARI) [36, 53] to measure the similarity between $\mathfrak{c}(\mathbf{X}, k_i)$ and $C_i^{\$}$. Recall that two equivalent partitions yield ARI equal to 1. Moreover, two "independent" (see [26] for discussion) clusterings have the expected ARI of 0. Negative ARIs will be replaced with 0 for better interpretability of the results.

In order to quantify the quality of the method $\mathfrak{c}$ on $\mathbf{X}$, we will evaluate its outputs against all the available reference labellings and choose the highest ARI in result:

$$Q_{\mathbf{X}}(\mathfrak{c}) = \max\left\{\mathrm{ARI}\left(\mathfrak{c}(\mathbf{X}, k_1), C_1^{\$}\right), \dots, \mathrm{ARI}\left(\mathfrak{c}(\mathbf{X}, k_l), C_l^{\$}\right)\right\}. \tag{13}$$

This is to account for the fact that there might be many equally valid partitions and the (unsupervised) method $\mathfrak{c}$ should be rewarded if it identifies one of them (does not matter which one).

It is worth noting that only 13 datasets have $l > 1$, 11 of which come with reference labellings that do not have identical cardinalities. Also, some reference partitions include noise points – these were excluded during the computations of the ARIs (after the output of $\mathfrak{c}$ was determined, as none of the clustering methods studied features a noise point detector). Overall, this validation methodology conforms with [24].

3.5. Candidate solutions

To generate the list of candidate (initial) solutions used in Algorithm 1, we will apply many different clustering algorithms on each dataset, including:

- the most popular hierarchical clustering methods (single, average, Ward, centroid, complete linkage),
- Genie [23] (with different thresholds),
- information-theoretic algorithms (ITM [46] as well as IcA and GIc [9]),
- other methods in the well-established `sklearn` [50] package for Python: k-means, Gaussian mixtures, spectral clustering with different kernels, Birch (with a range of parameter values).

This gives 87 different combinations of algorithms and their setups. In Section 4 we provide some technical details about their implementations. Note that 12 of them will constitute the baseline in our empirical study below.

Also, we shall utilise the following heuristic solvers:

- particle swarm optimisation (via R package `pso` [4]),
- "global" optimisation by differential evolution [51] (`DEoptim` [45] in R).

They pinpoint local maxima based on 3–5 restarts from different initial candidate solutions. Both of them search over the continuous space $\mathbb{R}^{(Vk)\times d}$ in such a way that the clusters are represented by means of Vk vantage points. V vantage points represent one cluster (empirically, we have determined $V = 5$ be a good compromise between quality and speed). In every iteration, each point is assigned to its closest vantage point, and, as a consequence, to a cluster which is represented by this vantage point. This approach allows to determine clusters of more sophisticated shapes than when simply $V = 1$ is utilised (as with $V > 1$ we consider different unions of cells in a Voronoi diagram).

Additionally, to broaden the search space even further, we will pick 5 partitions completely at random, i.e., each $C \in \mathcal{C}_k$ being such that $C(1), \dots,$ $C(n)$ being independent random variables from the discrete uniform distribution on $[1 : k]$. Nevertheless, starting from a random partition never turned out better than an assisted initialisation based on one of the aforementioned candidate solutions.

Most importantly, as each dataset comes with a set of reference labels given by experts, these shall be considered as well.

Overall, for each dataset, we have obtained $m \simeq 100$ different clusterings[2], however, quite often there were duplicated entries, hence the effective m was in the range 30–50.

An ideal index, if it existed, would be 100% concordant with expert labels. That is, it would be impossible for the hill climbing method to improve them any further. Note that when we maximise I, the reference partitions are always amongst the initial candidate solutions which are fed to Algorithm 1. Therefore, our procedure guarantees that all the good combinations of indices and datasets must be identified. If the hill climbing method converges to a different solution, it means that I promotes some points that are less compatible with experts' opinion.

[2] All results are available at `https://github.com/gagolews/clustering_results_v1`.

Let us stress that neither I itself, nor the procedure for maximising it, is “aware” of the existence of any external labels: only $\mathbf{X}$ and k are input to $\mathfrak{c}$, not $C_i^{\$}$; this is still an unsupervised learning method. Algorithm 1 converges where it converges; the starting points are plentiful and there is a great variety of them. Reference partitions are only used at the final evaluation stage.

4. Experiments

4.1. Implementation

Experiments, data analysis, and visualisation tasks were performed using Python 3.8.6 (PyPI packages: `numpy` 1.19.0, `scipy` 1.5.1, `pandas` 1.0.3, `matplotlib` 3.3.3, `seaborn` 0.11.1) and R 4.0.3 (CRAN packages: `DEoptim` 2.2-5 [45], `pso` 1.0.3 [4]).

The correctness of our C++ implementations[3] of the cluster validity indices was verified against R packages `clusterCrit` 1.2.8 and `clusterSim` 0.49-2 (wherever applicable). Our library turned out significantly faster than the two reference ones. Moreover, we allowed for the computing of the indices incrementally (as mentioned above), which was particularly beneficial in terms of the run-time of Algorithm 1.

Overall, the computations took ca. 3 months of computing time with the use of 2 computer clusters (within the allocated resource limits we have been granted by ICM UW/PL-Grid and the School of IT at Deakin University).

4.2. Which index best agrees with expert knowledge?

Let us proceed with the evaluation of the agreement between the cluster validity indices and expert knowledge.

4.2.1. GDunn_dX_DY

We first focus on the 15 generalised Dunn indices [6]. To recall, *GDunn* indices are defined as the ratio of cluster separation (d) and compactness (D).

Figure 3 gives the box-and-whisker plots for the Adjusted Rand indices across the benchmark datasets studied. We clearly see that d_1, i.e., the pairwise minimum distance (used in the original Dunn index) outperforms

[3] Available at `https://github.com/gagolews/optim_cvi`.

the other measures. In this scenario, the Wilcoxon signed-rank test does not find the choice of D significant ($\alpha = 0.05$).

Also, the measures based on d_5 are significantly worse than all other ones. Perhaps it would be better if d_5 was defined as the average *squared* point-centroid distance, not just average raw distance; recall that a centroid is the point that minimises exactly the square of the Euclidean metric.

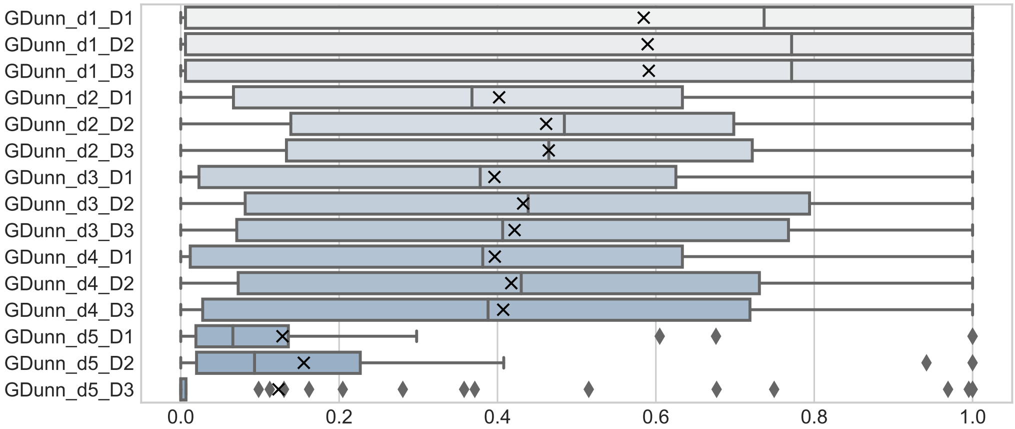

Figure 3: ARI: Generalised Dunn indices. We see that the choice of the cluster compactness measure D is rather negligible. On the other hand, separation measure $d_1 = \text{Min}$ performs best whilst d_5 (averaged distance to cluster centres) is subpar.

4.2.2. DuNN

Figure 4 shows the empirical distribution of ARIs in the case of the near-neighbour versions of the Dunn index.

For a fixed separation measure OWA_s, OWA_c equal to Min and Const is never significantly worse (one-sided Wilcoxon test, $\alpha = 0.05$) than Max and Mean. Moreover, there is no significant difference between Min and Const, therefore, applying Ockham's razor, we conclude that the cluster compactness could be omitted whatsoever (at least as far as our selection of aggregation functions is concerned).

Setting OWA_c at Const, interestingly, *DuNN_25_SMin:5_Const* significantly outperforms all the variants except *DuNN_5_Mean_Const*.

Moreover, *DuNN_25_Min_Const* is better than *DuNN_5_Min_Const*. Also note that the behaviour of *Max* or its smoothened version is particularly poor.

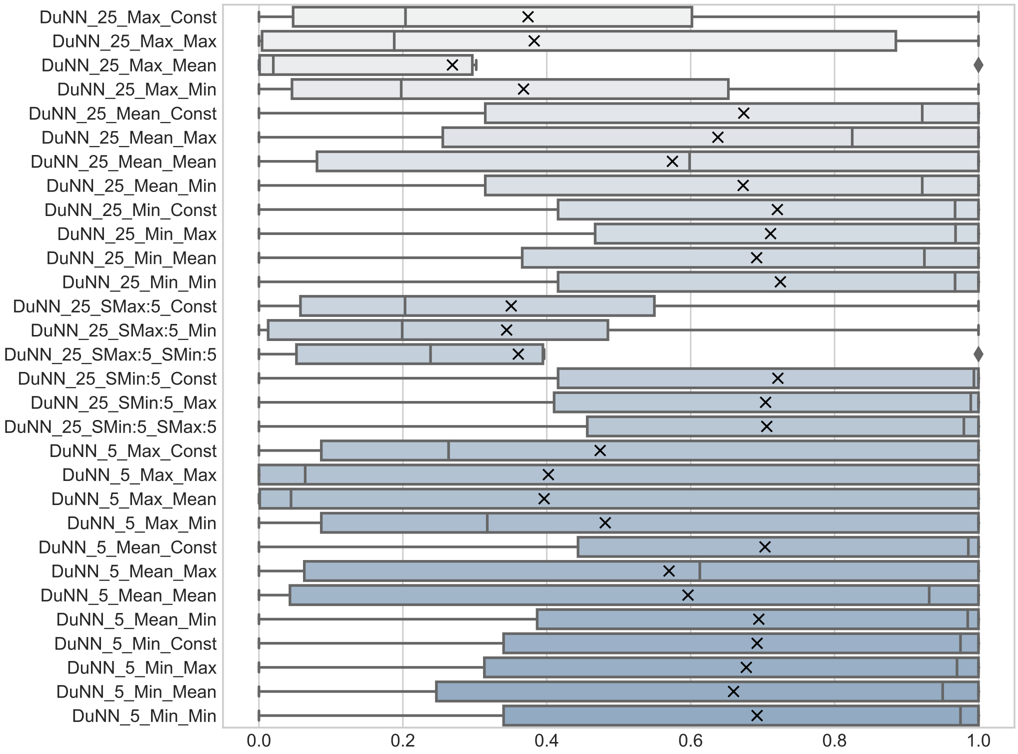

Figure 4: ARI: Near-neighbour-based DuNN indices. Disabling the use of a compactness measure whatsoever (*_Const*) might be preferred. Also, it is better to have the closest pairs of points from different clusters as far away from each other as possible.

4.2.3. Other methods

As a base line, the above and remaining CVIs will be compared against the outputs of 12 clustering algorithms:

1–5) *Average, Centroid, Complete, Ward, Single* – classical agglomerative hierarchical clustering algorithms;

6–9) *Genie_G0.1, Genie_G0.3, Genie_G0.5, Genie_G0.7* – the robust hierarchical clustering algorithm Genie that we have proposed in [23], with different thresholds for the Gini index of the inequity in cluster sizes;

10) *ITM* – greedy divisive minimiser of an information theoretic criterion over minimum spanning trees [46];

11) *GaussMix* – expectation-maximisation (EM) for Gaussian mixtures with 100 restarts and each cluster having its own covariance matrix;

12) *KMeans* – Lloyd-like k-means algorithm with 10 restarts (note that this is a heuristic to optimise the Caliński–Harabasz index/within-cluster sum of squares).

Their implementations are included in Python packages (available via PyPI; see their respective API documentation for more details on algorithms and default values of their parameters in place) `fastcluster` 1.1.26 (*Average, Centroid, Complete, Ward*; [47]), `genieclust` 0.9.4 (*Genie_G0.x, Single*; [22]), `sklearn` 0.23.1 (*GaussMix, KMeans*; [50]). Moreover, the implementation of *ITM* [46] is available from GitHub[4].

Tables 2 and 3 give the basic summary statistics on the empirical distribution of the ARIs across the 62 benchmark datasets and all the methods studied. Moreover, Figure 5 displays the boxplots.

We observe what follows:

- Our Genie algorithm [22] outperforms other methods. Note that it is significantly faster than most other algorithms as it is based on the minimum spanning tree of the pairwise distance graph.

- The lesser-known ITM method [46] performs relatively well.

- The near-neighbour-based *DuNN* indices that we have proposed in this paper are much better than *GDunn*.

- The difference between *DuNN_25_SMin:5_Const* and *WCNN_25* is insignificant.

- Overall, the algorithms based on the near-neighbourhood (minimum spanning trees can be considered a variant thereof, in some sense), seem to be much more valid for clustering tasks.

- Next in line are Gaussian mixtures that detect clusters of specific (spherical) shapes.

- Of course, k-means give similar results to the Caliński–Harabasz optimiser as the former is a heuristic to optimise the latter as the objective. Also *Ward* is a greedy agglomerative maximiser of the same objective.

[4] See `https://github.com/amueller/information-theoretic-mst`; git commit 178fd43.

- Most other cluster validity measures seem to promote the clusterings that are not concordant to expert knowledge which calls their relevance into question.

- *SilhouetteW* and *BallHall* – the weighted-by-cluster-cardinality versions of *Silhouette* and *CalińskiHarabasz*, respectively, perform worse than their unadjusted counterparts.

- The poor performance of some methods may be partially explained by the inequality of the cluster sizes they output – some of them are prone to generating few very large clusters and a number of very small ones (perhaps even being singleton objects). This includes *Single* (median Gini index of the cluster sizes=0.85), *DaviesBouldin* (0.95), and *SilhouetteW* (0.98). On the other hand, *Dunn_25_Max_Mean* (median Gini index of the cluster sizes=0.09), *Dunn_5_Max_Mean* (0.09), *CalińskiHarabasz* (0.13), *KMeans* (0.15), *ITM* (0.17), and *Genie_G0.1* (0.17) produced the least *skewed* partition sizes. However, let us note that this is not necessarily an accurate predictor of the clustering quality (see also [22] for discussion).

- Note that some datasets are inherently hard to cluster (the outputs of no algorithm matches the reference partition well); these include *uci/sonar* (max ARI=0.036), *uci/yeast* (max ARI=0.181), and *uci/ionosphere* (max ARI=0.401).

4.3. Clustering of clustering algorithms

Let us perform an interesting exercise where we determine a grouping of the clustering methods by means of the overall similarity of the results they generate on all the benchmark datasets. This way, we will know which methods (and CVIs) are "semantically" similar to each other.

We have computed the AR indices between all pairs of label vectors generated by all the methods (this time, reference/expert labels were not used). We used the mean, the median, or the 3rd quartile of (1.0 − ARI) to obtain a single number that summarises the "distance" between the algorithms.

We have applied the agglomerative hierarchical clustering algorithm with complete linkage so that the resulting dendrograms, which are depicted in Figure 6, are more interpretable (this will give us the maximal aggregated dissimilarities between all methods in a cluster). Note that we will be only interested in groups of algorithms that have small pairwise distances.

The majority vote of the results obtained by means of all the three dissimilarity measures gives the following "consensus" clusters, where the algorithms have quite high overall degree of similarity:

- *CalińskiHarabasz, KMeans*,
- *DuNN_25_Mean_Const, DuNN_25_Mean_Min*,
- *DuNN_5_Min_Const, DuNN_5_Min_Min*,
- *DuNN_25_Min_Const, DuNN_25_Min_Min, DuNN_25_SMin:5_Const*,
- *DuNN_5_Mean_Const, DuNN_5_Mean_Min*,
- *GDunn_d1_D1, GDunn_d1_D2, GDunn_d1_D3, Single.*

Let us note that:

- The above indicates again that in our task, the denominator (compactness measure) in all the generalisations of the Dunn index has no significant impact on the results. However, when CVIs are applied for the purpose of selecting the optimal number of clusters, the conclusions could be much different.
- *GDunn* and *DuNN* are not so similar, despite they both generalise the same index, *Dunn*. On the other hand, the latter (to recall, it is based on $d_1 = \mathrm{Min}$) is similar to *Single* linkage (which is determined through a greedy (agglomerative) consumption of the nearest pairs of points; it can be computed based on a minimum spanning tree).
- *Ward* is quite similar to *CalińskiHarabasz*, and *KMeans*, which was to be expected as they tend to optimise the same objective function.

5. Conclusion and Future Work

We have studied whether cluster validity indices really promote partitions that reflect what experts judge as meaningful. While some measures could still be considered relevant in the task of selecting the right number of clusters, it is better not to treat them as objective functions for identifying good partitions. This is particularly the case with the Davies–Bouldin, Silhouette, Ball–Hall, and the no-near-neighbour-based versions of the generalised Dunn index.

In the future, we will verify the usability of our new near-neighbour-based generalisations of the Dunn index in the problem of choosing the meaningful number of clusters. Their advantage is that they take into account the locality of the input points as well as the relative density of the points' distribution. Certainly, more combinations of OWA operators should be studied.

Acknowledgements

This research was supported by the Australian Research Council Discovery Project ARC DP210100227 as well as the PL-Grid Infrastructure.

Conflict of interest

The authors declare that they have no conflict of interest to disclose.

References

[1] D. Aloise, A. Deshpande, P. Hansen, P. Popat, NP-hardness of Euclidean sum-of-squares clustering, Machine Learning 75 (2009) 245–248.

[2] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J.M. Pérez, I. Perona, An extensive comparative study of cluster validity indices, Pattern Recognition 46 (2013) 243–256.

[3] G. Ball, D. Hall, ISODATA: A novel method of data analysis and pattern classification, Technical Report AD699616, 1965.

[4] C. Bendtsen, pso: Particle Swarm Optimization, 2012. R package version 1.0.3; `https://CRAN.R-project.org/package=pso`.

[5] J. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering algorithm, Computers & Geosciences 10 (1984) 191–203.

[6] J. Bezdek, N. Pal, Some new indexes of cluster validity, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 28 (1998) 301–315.

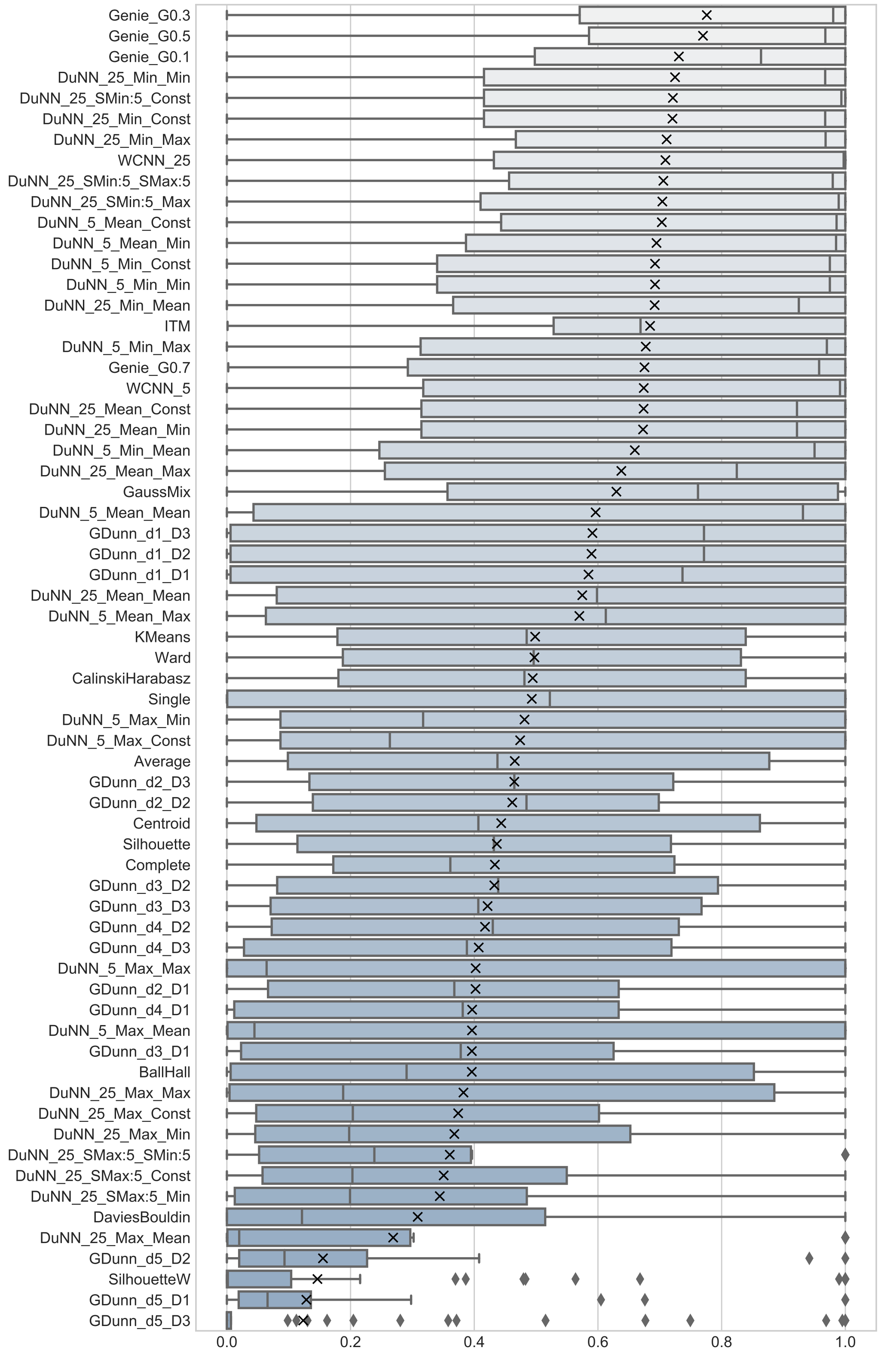

Figure 5: ARI: All methods (ordered by the average ARI). The Genie algorithm outperforms other clustering approaches. Near-neighbour-based cluster validity measures reflect expert knowledge quite well.

Table 2: ARI: Basic summary statistics; part I.

Method	Mean	St.Dev.	Q1	Median	Q3
Average	0.47	0.38	0.10	0.44	0.88
Centroid	0.44	0.39	0.05	0.41	0.86
Complete	0.43	0.33	0.17	0.36	0.72
GaussMix	0.63	0.38	0.36	0.76	0.99
Genie_G0.1	0.73	0.32	0.50	0.86	1.00
Genie_G0.3	0.78	0.29	0.57	0.98	1.00
Genie_G0.5	0.77	0.32	0.59	0.97	1.00
Genie_G0.7	0.68	0.38	0.29	0.96	1.00
ITM	0.68	0.28	0.53	0.67	1.00
KMeans	0.50	0.35	0.18	0.48	0.84
Single	0.49	0.47	0.00	0.52	1.00
Ward	0.50	0.35	0.19	0.50	0.83
BallHall	0.40	0.40	0.01	0.29	0.85
CalińskiHarabasz	0.49	0.35	0.18	0.48	0.84
DaviesBouldin	0.31	0.37	0.00	0.12	0.51
Silhouette	0.44	0.37	0.11	0.43	0.72
SilhouetteW	0.15	0.29	0.00	0.00	0.10
WCNN_25	0.71	0.38	0.43	1.00	1.00
WCNN_5	0.67	0.40	0.32	0.99	1.00
GDunn_d1_D1	0.58	0.44	0.01	0.74	1.00
GDunn_d1_D2	0.59	0.45	0.01	0.77	1.00
GDunn_d1_D3	0.59	0.45	0.01	0.77	1.00
GDunn_d2_D1	0.40	0.34	0.07	0.37	0.63
GDunn_d2_D2	0.46	0.32	0.14	0.48	0.70
GDunn_d2_D3	0.46	0.33	0.13	0.46	0.72
GDunn_d3_D1	0.40	0.35	0.02	0.38	0.63
GDunn_d3_D2	0.43	0.36	0.08	0.44	0.79
GDunn_d3_D3	0.42	0.36	0.07	0.41	0.77
GDunn_d4_D1	0.40	0.36	0.01	0.38	0.63
GDunn_d4_D2	0.42	0.36	0.07	0.43	0.73
GDunn_d4_D3	0.41	0.36	0.03	0.39	0.72
GDunn_d5_D1	0.13	0.20	0.02	0.07	0.14
GDunn_d5_D2	0.16	0.19	0.02	0.09	0.23
GDunn_d5_D3	0.12	0.28	0.00	0.00	0.01

Table 3: ARI: Basic summary statistics; part II.

Method	Mean	St.Dev.	Q1	Median	Q3
DuNN_5_Max_Const	0.47	0.44	0.09	0.26	1.00
DuNN_5_Mean_Const	0.70	0.38	0.44	0.99	1.00
DuNN_5_Min_Const	0.69	0.39	0.34	0.97	1.00
DuNN_25_Max_Const	0.37	0.39	0.05	0.20	0.60
DuNN_25_Mean_Const	0.67	0.38	0.31	0.92	1.00
DuNN_25_Min_Const	0.72	0.36	0.42	0.97	1.00
DuNN_25_SMax:5_Const	0.35	0.39	0.06	0.20	0.55
DuNN_25_SMin:5_Const	0.72	0.37	0.42	0.99	1.00
DuNN_5_Max_Min	0.48	0.43	0.09	0.32	1.00
DuNN_5_Mean_Min	0.69	0.38	0.39	0.99	1.00
DuNN_5_Min_Min	0.69	0.39	0.34	0.97	1.00
DuNN_25_Max_Min	0.37	0.39	0.05	0.20	0.65
DuNN_25_Mean_Min	0.67	0.38	0.31	0.92	1.00
DuNN_25_Min_Min	0.72	0.35	0.42	0.97	1.00
DuNN_25_SMax:5_SMin:5	0.36	0.38	0.05	0.24	0.39
DuNN_25_SMax:5_Min	0.34	0.39	0.01	0.20	0.48
DuNN_5_Max_Max	0.40	0.47	0.00	0.06	1.00
DuNN_5_Mean_Max	0.57	0.44	0.06	0.61	1.00
DuNN_5_Min_Max	0.68	0.40	0.31	0.97	1.00
DuNN_25_Max_Max	0.38	0.41	0.00	0.19	0.89
DuNN_25_Mean_Max	0.64	0.39	0.26	0.82	1.00
DuNN_25_Min_Max	0.71	0.36	0.47	0.97	1.00
DuNN_25_SMin:5_Max	0.70	0.37	0.41	0.99	1.00
DuNN_25_SMin:5_SMax:5	0.71	0.37	0.46	0.98	1.00
DuNN_5_Max_Mean	0.40	0.47	0.00	0.04	1.00
DuNN_5_Mean_Mean	0.60	0.44	0.04	0.93	1.00
DuNN_5_Min_Mean	0.66	0.41	0.25	0.95	1.00
DuNN_25_Max_Mean	0.27	0.42	0.00	0.02	0.30
DuNN_25_Mean_Mean	0.57	0.43	0.08	0.60	1.00
DuNN_25_Min_Mean	0.69	0.38	0.37	0.92	1.00

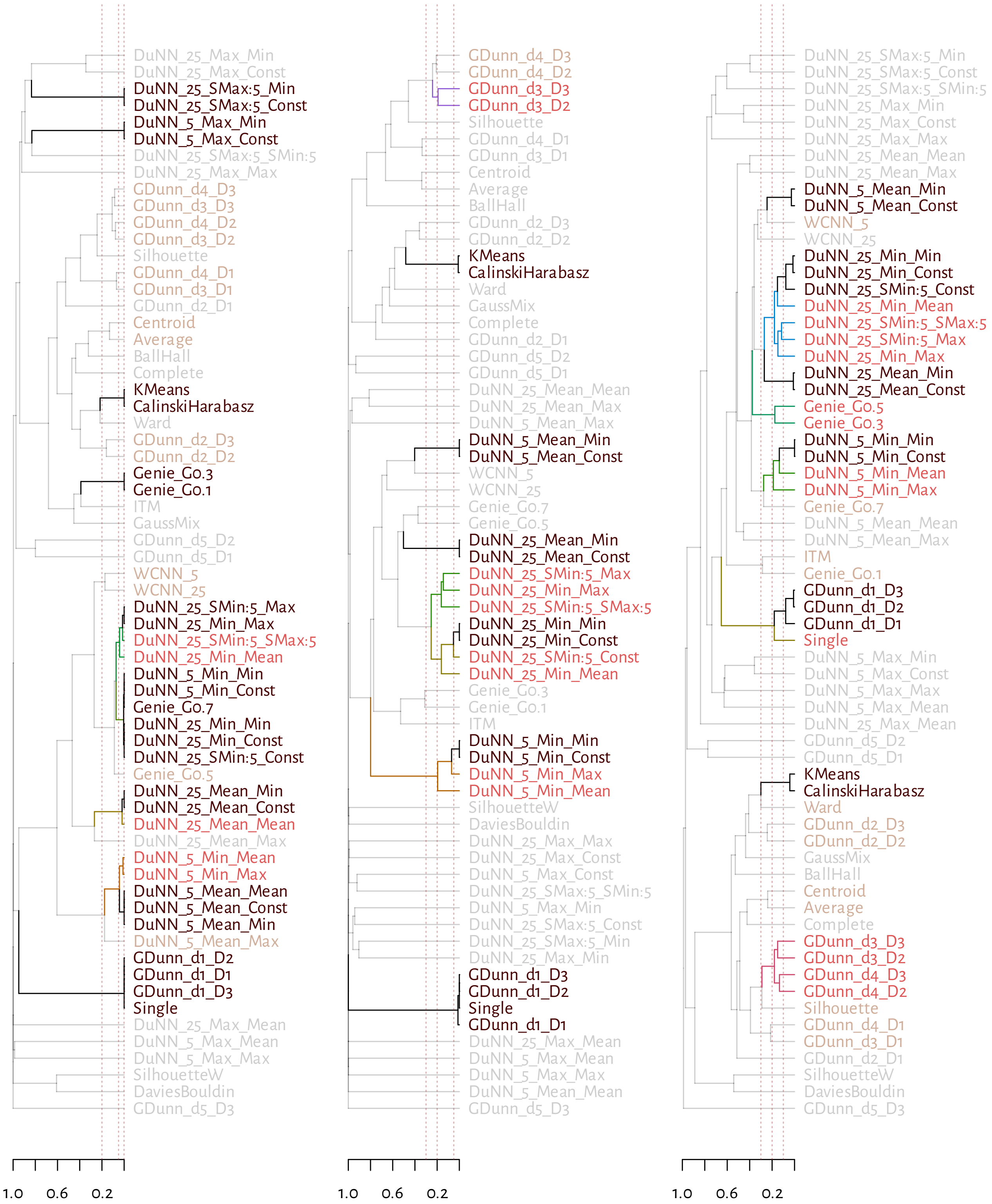

Figure 6: Complete linkage with Median(1.0-ARI), Q3(1.0-ARI), Mean(1.0-ARI), respectively. Only clusters that are formed at low dissimilarity levels (branches with connectors in the right parts of each figure) can be considered meaningful.

[7] M. Brun, C. Sima, J. Hua, J. Lowey, B. Carroll, E. Suh, E.R. Dougherty, Model-based evaluation of clustering validation measures, Pattern Recognition 40 (2007) 807–824.

[8] T. Caliński, J. Harabasz, A dendrite method for cluster analysis, Communications in Statistics 3 (1974) 1–27.

[9] A. Cena, Adaptive hierarchical clustering algorithms based on data aggregation methods, Ph.D. thesis, Systems Research Institute, Polish Academy of Sciences, 2018. In Polish.

[10] A. Cena, M. Gagolewski, Genie+OWA: Robustifying hierarchical clustering with OWA-based linkages, Information Sciences 520 (2020) 324–336.

[11] S. Dasgupta, V. Ng, Single data, multiple clusterings, in: Proc. NIPS Workshop Clustering: Science or Art? Towards Principled Approaches, 2009. `http://clusteringtheory.org`.

[12] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI–1 (1979) 224–227.

[13] K.G. Dhal, A. Das, S. Ray, J. Gálvez, Randomly attracted rough firefly algorithm for histogram based fuzzy image clustering, Knowledge-Based Systems 216 (2021) 106814.

[14] E. Dimitriadou, S. Dolnicar, F. Leisch, A. Weingessel, More insight into clustering: Comparison of cluster algorithms and evaluation of indexes for determining the correct number of clusters, Methods of Psychological Research 4 (1999) 65–66.

[15] D.T. Dinh, V.N. Huynh, S. Sriboonchitta, Clustering mixed numerical and categorical data with missing values, Information Sciences 571 (2021) 418–442.

[16] M. Du, R. Wang, R. Ji, X. Wang, Y. Dong, ROBP a robust border-peeling clustering using Cauchy kernel, Information Sciences 571 (2021) 375–400.

[17] D. Dua, C. Graff, UCI Machine Learning Repository, 2021. http://archive.ics.uci.edu/ml.

[18] J. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, Journal of Cybernetics 3 (1974) 32–57.

[19] A.W.F. Edwards, L.L. Cavalli-Sforza, A method for cluster analysis, Biometrics 21 (1965) 362–375.

[20] P. Fränti, R. Mariescu-Istodor, C. Zhong, XNN graph, Lecture Notes in Computer Science 10029 (2016) 207–217.

[21] P. Fränti, S. Sieranoja, K-means properties on six clustering benchmark datasets, Applied Intelligence 48 (2018) 4743–4759.

[22] M. Gagolewski, genieclust: Fast and robust hierarchical clustering, SoftwareX 15 (2021) 100722.

[23] M. Gagolewski, M. Bartoszuk, A. Cena, Genie: A new, fast, and outlier-resistant hierarchical clustering algorithm, Information Sciences 363 (2016) 8–23.

[24] M. Gagolewski, et al., Benchmark suite for clustering algorithms – version 1, 2020. https://github.com/gagolews/clustering_benchmarks_v1, doi:10.5281/zenodo.3815066.

[25] M. Garey, D. Johnson, H. Witsenhausen, The complexity of the generalized Lloyd–Max problem, IEEE Transactions on Information Theory 28 (1982) 255–256.

[26] A.J. Gates, Y.Y. Ahn, The impact of random models on clustering similarity, Journal of Machine Learning Research 18 (2017) 1–28.

[27] F. Glover, Future paths for integer programming and links to artificial intelligence, Computers & Operations Research 13 (1986) 533–549.

[28] D. Graves, W. Pedrycz, Kernel-based fuzzy clustering: A comparative experimental study, Fuzzy Sets and Systems 161 (2010) 522–543.

[29] M. Halkidi, Y. Batistakis, M. Vazirgiannis, On clustering validation techniques, Journal of Intelligent Information Systems (2001) 107–145.

[30] R. Isimeto, C. Yinka-Banjo, C.O. Uwadia, D.C. Alienyi, An enhanced clustering analysis based on glowworm swarm optimization, in: 2017 IEEE 4th International Conference on Soft Computing Machine Intelligence (ISCMI), pp. 42–49.

[31] M. Jamil, X.S. Yang, A literature survey of benchmark functions for global optimization problems, International Journal of Mathematical Modelling and Numerical Optimisation 4 (2013).

[32] G. Karypis, E. Han, V. Kumar, CHAMELEON: Hierarchical clustering using dynamic modeling, Computer 32 (1999) 68–75.

[33] M. Kim, R. Ramakrishna, New indices for cluster validity assessment, Pattern Recognition Letters 26 (2005) 2535–2363.

[34] R. Kuo, Y. Zheng, T.P.Q. Nguyen, Metaheuristic-based possibilistic fuzzy k-modes algorithms for categorical data clustering, Information Sciences 557 (2021) 1–15.

[35] G. Lance, W. Williams, A general theory of classification sorting strategies: 1. Hierarchical systems, Computer Journal (1967) 373–380.

[36] H. Lawrence, A. Phipps, Comparing partitions, Journal of Classification 2 (1985) 193–218.

[37] J. Lee, A First Course in Combinatorial Optimisation, Cambridge University Press, 2011.

[38] H. Li, S. Zhang, X. Ding, C. Zhang, P. Dale, Performance evaluation of cluster validity indices (cvis) on multi/hyperspectral remote sensing datasets, Remote Sensing 8 (2016).

[39] S. Liang, D. Han, Y. Yang, Cluster validity index for irregular clustering results, Applied Soft Computing 95 (2020) 106583.

[40] Y. Liu, Y. Jiang, T. Hou, F. Liu, A new robust fuzzy clustering validity index for imbalanced data sets, Information Sciences 547 (2021) 579–591.

[41] S. Lloyd, Least squares quantization in PCM, IEEE Transactions on Information Theory 28 (1957 (1982)) 128–137. Originally a 1957 Bell Telephone Laboratories Research Report; republished in 1982.

[42] U. Maulik, S. Bandyopadhyay, Performance evaluation of some clustering algorithms and validity indices, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 1650–1654.

[43] G.W. Milligan, M.C. Cooper, An examination of procedures for determining the number of clusters in a data set, Psychometrika 50 (1985) 159–179.

[44] G. Mishra, A.K. Kar, A.C. Mishra, S.K. Mohanty, M. Panda, SEND: A novel dissimilarity metric using ensemble properties of feature space for clustering numerical data, Information Sciences 574 (2021) 279–296.

[45] K. Mullen, D. Ardia, D. Gil, D. Windover, J. Cline, DEoptim: An R package for global optimization by differential evolution, Journal of Statistical Software 40 (2011) 1–26.

[46] A. Müller, S. Nowozin, C. Lampert, Information theoretic clustering using minimum spanning trees, in: Proc. German Conference on Pattern Recognition, 2012. https://github.com/amueller/information-theoretic-mst.

[47] D. Müllner, fastcluster: Fast hierarchical, agglomerative clustering routines for R and Python, Journal of Statistical Software 53 (2013) 1–18.

[48] S.J. Nanda, G. Panda, A survey on nature inspired metaheuristic algorithms for partitional clustering, Swarm and Evolutionary Computation 16 (2014) 1–18.

[49] J. Nocedal, S.J. Wright, Numerical Optimization, Springer, 2006.

[50] F. Pedregosa, et al., Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[51] K.V. Price, R.M. Storn, J.A. Lampinen, Differential Evolution – A Practical Approach to Global Optimization, Springer-Verlag, 2006.

[52] R. Qaddoura, H. Faris, I. Aljarah, An efficient evolutionary algorithm with a nearest neighbor search technique for clustering analysis, Ambient Intell Human Comput (2020).

[53] M. Rezaei, P. Fränti, Set matching measures for external cluster validity, IEEE Transactions on Knowledge and Data Engineering 28 (2016) 2173–2186.

[54] P.J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics 20 (1987) 53–65.

[55] A. Ultsch, Clustering with SOM: U*C, in: Workshop on Self-Organizing Maps, WSOM 2005, 2005, pp. 75–82.

[56] A. Vij, P. Khandnor, Validity of internal cluster indices, in: International Conference on Computational Systems for Sustainable Solutions, pp. 388–395.

[57] J.H. Ward Jr., Hierarchical grouping to optimize an objective function, Journal of the American Statistical Association 58 (1963) 236–244.

[58] Q. Xu, Q. Zhang, J. Liu, B. Luo, Efficient synthetical clustering validity indexes for hierarchical clustering, Expert Systems with Applications 151 (2020) 113367.

[59] R. Xu, J. Xu, D.C. Wunsch, A comparison study of validity indices on swarm-intelligence-based clustering, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42 (2012) 1243–1256.

[60] R.R. Yager, On ordered weighted averaging aggregation operators in multicriteria decision making, IEEE Transactions on Systems, Man, and Cybernetics 18 (1988) 183–190.

[61] S. Zhu, L. Xu, E.D. Goodman, Evolutionary multi-objective automatic clustering enhanced with quality metrics and ensemble strategy, Knowledge-Based Systems 188 (2020) 105018.