Author(s): Hämäläinen, Joonas; Jauhiainen, Susanne; Kärkkäinen, Tommi

Title: Comparison of Internal Clustering Validation Indices for Prototype-Based Clustering

Year: 2017

Version:

Please cite the original version:

Hämäläinen, J., Jauhiainen, S., & Kärkkäinen, T. (2017). Comparison of Internal Clustering Validation Indices for Prototype-Based Clustering. Algorithms, 10(3), Article 105. https://doi.org/10.3390/a10030105

algorithms

MDPI

Article

Comparison of Internal Clustering Validation Indices for Prototype-Based Clustering

Joonas Hämäläinen *,†, Susanne Jauhiainen † and Tommi Kärkkäinen †

Faculty of Information Technology, University of Jyvaskyla, P.O. Box 35, FI-40014 Jyvaskyla, Finland; susanne.m.jauhiainen@student.jyu.fi (S.J.); tommi.karkkainen@jyu.fi (T.K.)

* Correspondence: joonas.k.hamalainen@jyu.fi

† These authors contributed equally to this work.

Received: 13 July 2017; Accepted: 1 September 2017; Published: 6 September 2017

Abstract: Clustering is an unsupervised machine learning and pattern recognition method. In general, in addition to revealing hidden groups of similar observations and clusters, their number needs to be determined. Internal clustering validation indices estimate this number without any external information. The purpose of this article is to evaluate, empirically, characteristics of a representative set of internal clustering validation indices with many datasets. The prototype-based clustering framework includes multiple, classical and robust, statistical estimates of cluster location so that the overall setting of the paper is novel. General observations on the quality of validation indices and on the behavior of different variants of clustering algorithms will be given.

Keywords: prototype-based clustering; clustering validation index; robust statistics

1. Introduction

Clustering aims to partition a given dataset (a set of observations) into groups (clusters) that are separated from other groups in a twofold manner: observations within a cluster are similar to each other and dissimilar to observations in other clusters [1]. Diverse sets of clustering approaches have been developed over the years, e.g., density-based, probabilistic, grid-based, and spectral clustering [2]. However, the two most common groups of crisp (here, we do not consider fuzzy clustering [3]) clustering algorithms are partitional and hierarchical clustering [4]. Hierarchical clustering constructs a tree structure from data to present layers of clustering results, but because of the pairwise distance matrix requirement, the basic form of the method is not scalable to a large volume of data [5]. Moreover, many clustering algorithms, including hierarchical clustering, can produce clusters of arbitrary shapes in the data space, which might be difficult to interpret for knowledge discovery [6].

The two aims of clustering for K groups in data are approached in the partitional algorithms, most prominently in the classical K-means [4,7], by using two main phases: initial generation of K prototypes and local refinement of the initial prototypes. The initial prototypes should be separated from each other [4,8]. Lately, the K-means++ algorithm [9], where the random initialization is based on a density function favoring distinct prototypes, has become the most popular variant to initialize the K-means-type of an algorithm. Because the prototype refinement acts locally, we need a globalization strategy to explore the search space. This can be accomplished with repeated restarts through initial prototype regeneration [10] or by using evolutionary approaches with a population of different candidate solutions [11].

One can utilize different error (score) functions in partitional clustering algorithms [12]. Mean is the statistical estimate of the cluster prototype in K-means and the clustering error is measured with the least-squares residual. This implies the assumption of spherically symmetric, normally distributed data with Gaussian noise. These conditions are relaxed when the cluster prototype is replaced, e.g., with a robust location estimate [13–15]. The two simplest robust estimates of location

are median and spatial median, whose underlying spherically symmetric distributions are uniform and Laplace distributions, respectively. If the type of data is discrete, for instance, an integer variable with uniform quantization error [16], then the Gaussian assumption is not valid. Median, given by the middle value of the ordered univariate sample (unique only for odd numbers of points [17]), can, like the mean, be estimated from the marginal distribution being inherently univariate. The spatial median, on the other hand, is truly a multivariate, orthogonally equivariant location estimate [18]. These location estimates and their intrinsic properties are illustrated and more thoroughly discussed in [17,19]. The median and spatial median have many attractive statistical properties, especially since their so-called breakdown point is 0.5, i.e., they can handle up to 50% of contaminated and erroneous data.

In a typical unsupervised scenario, one does not possess any prior knowledge of the number of clusters K. Finding the best possible representation of data with K groups is difficult because the number of all possible groupings is the sum of Stirling numbers of the second kind [19]. Defining validation measures for clustering results has been, therefore, a challenging problem that different approaches have tried to overcome [20–25]. The quality of a clustering result can be measured with a Clustering Validation Index (CVI). The aim of a CVI is to estimate the most appropriate K based on the compactness and separation of the clusters. Validation indices can be divided into three categories [26]: internal, external, and relative. An external validation index uses prior knowledge, an internal index is based on information from the data only, and in a relative CVI, multiple clustering results are compared. A comprehensive review of clustering validation techniques up to 2001 was provided in [27]. There exists also alternative approaches for determining the number of clusters, e.g., by measuring the stability of the clustering method [28] or using multiobjective evolutionary approaches [11,29].

In this paper, we continue the previous work reported in [30] by focusing on a comparison of the seven best internal CVIs, as identified in [30] and augmented by [22]. The earlier comparisons, typically reported when suggesting novel CVIs, only include K-means as the partitional clustering algorithm [22,30–36]. Here, this treatment is generalized by using multiple statistical estimates as a cluster prototype and to define the clustering error, under the currently most common initialization strategy as proposed in [9] (which is also generalized). Note that prototype-based clustering can also be conducted with an incremental fashion [37–39]. However, here were restrict ourselves on the batch versions of the algorithms, which can be guaranteed to converge in a finite number of iterations (see Section 2.1). The definitions of the considered validation indices are also extended and empirically compared with K-means, K-medians, and K-spatialmedians (using spatial median as a prototype estimate) clustering results for a large pool of benchmark datasets. According to our knowledge, there exists no previous work that compares CVIs with multiple different distance metrics. Our aim is to sample the main characteristics of the indices considered and to identify what indices most reliably refer to ground truth values of the benchmark datasets. Note that by their construction, all CVIs considered here can also be used to suggest the number of clusters in hierarchical clustering.

The structure of the article is as follows. After this introductory section, we describe generalized prototype-based clustering, discuss its convergence, and also present the generalized versions of cluster initialization and indices in Section 2. Our experimental setup is described in Section 3, and the results are given and discussed in Section 4. Finally, conclusions are drawn in Section 5.

2. Methods

In this section, we introduce and analyze all the necessary formulations for clustering and cluster validation indices.

2.1. General Prototype-Based Clustering and Its Convergence

As described above, prototype-based partitional clustering algorithms comprise two main phases. First, they start with an initial partition of the data, and second, the quality of this partition is

improved by a local search algorithm during the search phase. The initial partition can be obtained based on many different principles [4,8], but a common strategy is to use distinct prototypes [9]. Most typically, the globalization of the whole algorithm is based on random initialization with several regenerations [10]. Then, the best solution with the smallest clustering error is chosen as the final result. The iterative relocation algorithm skeleton for prototype-based partitional clustering is presented in Algorithm 1 [12,16].

Algorithm 1: Prototype-based partitional clustering algorithm.

Input: Dataset and the number of clusters K.
Output: Partition of dataset into K disjoint groups.
Select K points as the initial prototypes;
repeat
1. Assign individual observation to the closest prototype;
2. Recompute the prototype with the assigned observations;

until *the partition does not change*;

As stated in [17] (see also [19]), the different location estimates for a cluster prototype arise from different l_p-norms to the q-th power as the distance measure and the corresponding clustering error function; mean refers to $\|\cdot\|_2^2$ ($p = q = 2$), median is characterized by $\|\cdot\|_1^1$ ($p = q = 1$), and the spatial median is given by $\|\cdot\|_2^1$ ($p = 2, q = 1$). Hence, generally the repetition of Steps 1 and 2 from the search phase of Algorithm 1 locally minimize the following clustering error criterion:

$$\mathcal{J}(\{\mathbf{b}_k\}) = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathbf{C}_k} \|\mathbf{x}_i - \mathbf{b}_k\|_p^q. \tag{1}$$

Here $\{\mathbf{b}_k\}$, $k = 1, \ldots, K$ denote the prototype vectors to be determined and $\{\mathbf{x}_i\}_{i=1}^N$, $\mathbf{x}_i \in \mathbb{R}^n$ refers to the given set of n-dimensional observations. The interpretation of (1) is that each observation $\mathbf{x}_i$ is assigned to cluster k with the closest prototype on the l_p-norm:

$$\mathbf{C}_k = \{1 \leq k \leq K \mid \|\mathbf{x}_i - \mathbf{b}_k\|_p \leq \|\mathbf{x}_i - \mathbf{b}_{k'}\|_p \quad \forall k \neq k'\}.$$

Hence, as noted in [40], another more compact way of formalizing the clustering error criterion reads as

$$\mathcal{J}(\{\mathbf{b}_k\}) = \sum_{i=1}^{N} \min_{k=1,\ldots,K} \|\mathbf{x}_i - \mathbf{b}_k\|_p^q, \tag{2}$$

which more clearly shows the *nonsmoothness* of the clustering problem, because the min-operator is not classically differentiable (see [17] and references therein). This observation gives rise to a different set of clustering algorithms that are based on nonsmooth optimization solvers [41].

However, despite the nonsmoothness of the error function, it can be shown that the search phase of Algorithm 1 decreases the clustering error, ensuring local convergence of the algorithm in finite many steps. We formalize this in the next proposition. The proof here is a slight modification and simplification of the more general treatment in [19], Theorem 5.3.1, along the lines of the convergence analyses in different problem domains, as given in [42–44].

Proposition 1. *The repeated Steps 1 and 2 of Algorithm 1 decrease the clustering error function (2). This guarantees convergence of the algorithm in finite many steps.*

Proof. Let us denote by superscript t the current iterates of the prototypes $\{\mathbf{b}_k^t\}$ with the initial candidates for $t = 0$. If assignments to clusters and to the closest cluster prototypes do not change,

we are done, so let us assume that the repeated step 1 in Algorithm 1 has identified at least one $1 \leq j \leq N$ such that, for $\mathbf{x}_j \in \mathbf{C}_k^t$, there exists a better prototype candidate:

$$\|\mathbf{x}_j - \mathbf{b}_k^t\|_p > \|\mathbf{x}_j - \mathbf{b}_{k'}^t\|_p \quad \text{for some } \; k' \neq k. \tag{3}$$

Then, a direct computation, using monotonicity of the function $\|\cdot\|^q$ for $q = \{1, 2\}$ and reflecting the change in the assignments, gives

$$\begin{aligned} \mathcal{J}(\{\mathbf{b}_k^t\}) = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathbf{C}_k^t} \|\mathbf{x}_i - \mathbf{b}_k^t\|_p^q &= \sum_{k=1}^{K} \left(\sum_{\substack{\mathbf{x}_i \in \mathbf{C}_k^t \\ i \neq j}} \|\mathbf{x}_i - \mathbf{b}_k^t\|_p^q + \|\mathbf{x}_j - \mathbf{b}_k^t\|_p^q \right) \\ &> \sum_{k=1}^{K} \left(\sum_{\substack{\mathbf{x}_i \in \mathbf{C}_k^t \\ i \neq j}} \|\mathbf{x}_i - \mathbf{b}_k^t\|_p^q + \|\mathbf{x}_j - \mathbf{b}_{k'}^t\|_p^q \right) = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathbf{C}_k^{t+1}} \|\mathbf{x}_i - \mathbf{b}_k^t\|_p^q \\ &\geq \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathbf{C}_k^{t+1}} \|\mathbf{x}_i - \mathbf{b}_k^{t+1}\|_p^q = \mathcal{J}(\{\mathbf{b}_k^{t+1}\}). \end{aligned} \tag{4}$$

Here, the last inequality follows from the repeated Step 2 of Algorithm 1 and from the optimization-based definitions of mean/median/spatial median as minimizers of the l_p^q-norm [17] over a dataset:

$$\sum_{\mathbf{x}_i \in \mathbf{C}_k^{t+1}} \|\mathbf{x}_i - \mathbf{b}_k^{t+1}\|_p^q = \min_{\mathbf{b} \in \mathbb{R}^n} \sum_{\mathbf{x}_i \in \mathbf{C}_k^{t+1}} \|\mathbf{x}_i - \mathbf{b}\|_p^q \leq \sum_{\mathbf{x}_i \in \mathbf{C}_k^{t+1}} \|\mathbf{x}_i - \mathbf{b}_k^t\|_p^q \quad \text{for all } \; k. \tag{5}$$

Because (4) and (5) are valid for any reallocated index j satisfying (3), we conclude that the clustering error strictly decreases when a reallocation for a set of observations occurs in Algorithm 1.

To this end, because there exists only a finite number of possible sets $\mathbf{C}_k^t$, we must have $\mathbf{C}_k^{t+1} = \mathbf{C}_k^t$ after a finite number of steps $t \mapsto t + 1$. This ends the proof. □

The K-means++ initialization method utilizes squared Euclidean distance-based probabilities to sample initial prototypes from the data points. In Algorithm 2, this initialization strategy is generalized for varying l_p-norms to the q-th power. Note that Algorithm 2 is the same as the K-means++ initialization algorithm when $p = q = 2$. In order to be successful, one needs to assume in Algorithm 2 that the dataset $\{\mathbf{x}_i\}_{i=1}^N$ has at least K distinct data points, which is a natural and reasonable assumption. In Step 2, the probability for each point $\mathbf{x}_i$ to be selected as the next initial prototype is proportional to the distance to the closest already selected prototype divided by the value of the clustering error Function (2) for the already selected prototypes. Clearly, in each iteration, the most distant points (with respect to the previously selected initial prototypes) have the highest probability of being selected.

Algorithm 2: General K-means++-type initialization.

Input: Dataset $\{\mathbf{x}_i\}_{i=1}^N$ and the number of clusters K.
Output: Initial prototypes $\{\mathbf{b}_k\}_{k=1}^K$.
1. Select $\mathbf{b}_1 = \mathbf{x}_i$ uniformly randomly, $i = 1, \ldots, N$;
for $k = 2$, $k = k + 1$, $k \leq K$ **do**
    2. Select $\mathbf{b}_k = \mathbf{x}_i$ with probability $\dfrac{\min_{j=1,\ldots,k-1} \|\mathbf{x}_i - \mathbf{b}_j\|_p^q}{\mathcal{J}(\{\mathbf{b}_j\}_{j=1}^{k-1})}$, $i = 1, \ldots, N$;
end

2.2. Cluster Validation Indices

From now on, for a given number of clusters K, we denote, by $\{\mathbf{c}_k\}$ and $\{\mathbf{C}_k\}$, $k = 1, \ldots, K$, the best prototypes and divisions obtained after a fixed number of repeated applications of Algorithm 1. When $K = 1$, we denote the prototype, i.e., the mean, median, or spatial median, of the whole data with m. Moreover, we let $\mathcal{J}_K$ denote the corresponding clustering error over the whole data and $\mathcal{J}_K^k = \sum_{\mathbf{x}_i \in \mathbf{C}_k} \|\mathbf{x}_i - \mathbf{c}_k\|_p^q$ to refer to the corresponding final within-cluster errors.

Cluster validation considers the quality of the result of a clustering algorithm, attempting to find the partition that best fits the nature of the data. The number of clusters, given as a parameter for many clustering algorithms (such as the ones presented in Section 2.1), should be decided based on the natural structure of the data. Like the best clustering solution, the number of clusters is also not always clear and many 'right' answers can exist (see, e.g., [19], Figure 5). The number can also depend on the resolution, i.e., whether the within- and between- cluster separabilities are considered globally or locally. Here, we focus on the CVIs based on the internal criteria.

The validation indices measure how well the general goal of clustering—high similarity within clusters and high separability between clusters—is achieved. These are considered with measures of within-cluster (*Intra*) and between-cluster (*Inter*) separability, for which lower and higher values are better, respectively. Normally, a division between *Intra* and *Inter* is made and the optimal value is at the minimum or maximum, based on the order of the division.

In Table 1, the best internal validation indices (as determined for the K-means-type of clustering in [30], augmented with [22]) are introduced in a general fashion for the l_p^q-norm setting. All except one of the indices have been modified in such a way that the optimal number of clusters can be found at the minimal value. Only the Wemmert–Gançarski index, which has a unique pattern of the general formula, is given in the original form, where the maximum value indicates the optimal number of clusters. In Table 1, if the formula that combines Intra and Inter depends on the generated clusters, then this is indicated with the corresponding parameters.

Table 1. Internal cluster validation indices.

Name	Notation	Intra	Inter	Formula
KCE [30]	KCE	$K \times \mathcal{J}_K$		*Intra*
WB-index [22]	WB	$K \times \mathcal{J}_K$	$\sum_{k=1}^{K} n_k \|\mathbf{c}_k - m\|_p^q$	$\frac{Intra}{Inter}$
Calinski–Harabasz [24]	CH	$(K-1) \times \mathcal{J}_K$	$(N-K) \times \sum_{k=1}^{K} n_k \|\mathbf{c}_k - m\|_p^q$	$\frac{Intra}{Inter}$
Davies–Bouldin [23]	DB	$\frac{1}{n_k}\mathcal{J}_K^k + \frac{1}{n_{k'}}\mathcal{J}_K^{k'}$	$\|\mathbf{c}_k - \mathbf{c}_{k'}\|_p^q$	$\frac{1}{K}\sum_{k=1}^{K} \max_{k \neq k'} \frac{Intra(k,k')}{Inter(k,k')}$
Pakhira, Bandyopadhyay, and Maulik [45]	PBM	$K \times \mathcal{J}_K$	$\max_{k \neq k'}(\|\mathbf{c}_k - \mathbf{c}_{k'}\|_p^q) \times \mathcal{J}_1$	$\left(\frac{Intra}{Inter}\right)^2$
Ray–Turi [25]	RT	$\frac{1}{N} \times \mathcal{J}_K$	$\min_{k \neq k'} \|\mathbf{c}_k - \mathbf{c}_{k'}\|_p^q$	$\frac{Intra}{Inter}$
Wemmert– Gançarski ([46])	WG	$\|\mathbf{x}_i - \mathbf{c}_k\|_p^q$	$\min_{k \neq k'} \|\mathbf{x}_i - \mathbf{c}_{k'}\|_p^q$	$\frac{1}{N}\sum_{k=1}^{K} max\left(0,\ n_k - \sum_{i \in I_k} \frac{Intra(i)}{Inter(i)}\right)$

As can be seen from the formulas of the indices, there are a lot of similarities in how different indices measure the within- and between-cluster separability. For example, the clustering error straightforwardly measures the similarity within clusters and, therefore, almost all of the indices include it in their measure of *Intra*. In case of between-cluster separability, it is common to measure, for instance, the distance between cluster prototypes or between cluster prototypes and the whole data prototype. The rationale behind the index structures and their more detailed descriptions can be found in the original articles.

2.3. On Computational Complexity

The computational complexity of the prototype-based clustering is $\mathcal{O}(RTKNn)$, where T is the number of iterations needed for convergence and R is the chosen number of repetitions of joint Algorithms 1 and 2. As the clustering itself is quite time-consuming, especially with large datasets, it is only natural to also consider the complexity of the indices to avoid excessively complex computations. Here, the KCE index has an advantage in not requiring any extra calculation after the clustering solution has been obtained. For the indices that go through the prototypes once, here the WB and CH that measure the prototype distances to the whole data prototype, the complexity is $\mathcal{O}(Kn)$. Indices that measure the distances between all the prototypes, such as DB, PBM, and RT, have complexity $\mathcal{O}(K^2n)$.

In our tests, WG is the index with the highest complexity, $\mathcal{O}(KNn)$, going through the whole data, comparing points and the prototypes. A commonly used and generally well performing index, Silhouette (see, e.g., [30,33]), goes through the whole data twice when calculating its values and therefore its complexity is $\mathcal{O}(N^2n)$. With large datasets of at least hundreds of thousands of observations, this might be even more complex than the clustering task itself, with the chosen values of R and K and the observed value of T (see Figure A1) in Section 4. If computationally more involved indices would be used, also application of more complex clustering algorithms should be considered. Therefore, Silhouette was excluded from our tests.

2.4. About Earlier Validation Index Comparisons

There has been a lot of research on cluster validation, including comparisons of different validation indices and clustering algorithms. Often when a new index is proposed, the work also includes a set of comparisons that conclude that the new index is the best one. In [34], eight common CVIs were compared and with 5% additional noise, different densities, and skewed distributions, most indices were able to find the correct number of clusters. However, only three of them were able to recognize close subclusters. In their tests, S_Dbw was the only CVI that suggested the correct number of clusters for all datasets. In our previous tests in [30], the S_Dbw also recognized the close subclusters in Sim5D2, but it did not perform that well in general.

Often no single CVI has a clear advantage in every context, but each is best suited to a certain kind of data. This was also the conclusion in [33], where 30 different indices with 720 synthetic and 20 real datasets were compared. However, a group of about 10 indices were found to be the most recommendable, including Silhouette, Davies–Bouldin * and Calinski–Harabasz at the top. Also, in the earlier extensive comparison [47], where 30 indices were compared, the authors suggested that if different datasets were used for testing, the order of the indices would change but the best ones—including Calinski–Harabasz, Duda–Hart, and the C-index—would still perform well.

3. Experimental Setup

Next, we test the indices in Table 1 with different distance measures, i.e., with different prototype-based clustering algorithms. The index values are calculated with the distance corresponding to the clustering method used, i.e., city-block with the K-medians, squared Euclidean with the K-means, and Euclidean with the K-spatialmedians. All datasets were scaled to the range of $[-1, 1]$. All the tests were run on MATLAB (R2014a), where a reference implementation on both the validation indices and the general clustering Algorithm 1 with the initialization given in Algorithm 2 were prepared. The impact of the necessary amount of repetitions of Algorithms 1 and 2 was tested with multiple datasets (*S*-sets, *Dim*-sets, *A1*, *Unbalance*), comparing the clustering error and the cluster assignments. With 100 repetitions, the minimum clustering error and the corresponding cluster assignments were stabilized and an appropriate clustering result was found. This result with the minimum clustering error was used for computing the CVI value.

To test the indices, we used the basic benchmark datasets described in detail in [48], with the two other synthetic datasets http://users.jyu.fi/~jookriha/CVI/Data/ as given in [30] (see Figure 1).

These benchmark sets are synthetic datasets, suggested for use when testing any algorithm dealing with clustering spherical or Gaussian data. Here, we restrict ourselves to the benchmarks with at most 20 clusters, because the interpretation and knowledge discovery from the clustering results with a large number of prototypes might become tedious [6,49]. Therefore, the number of clusters was also tested with $K = 2 - 25$.

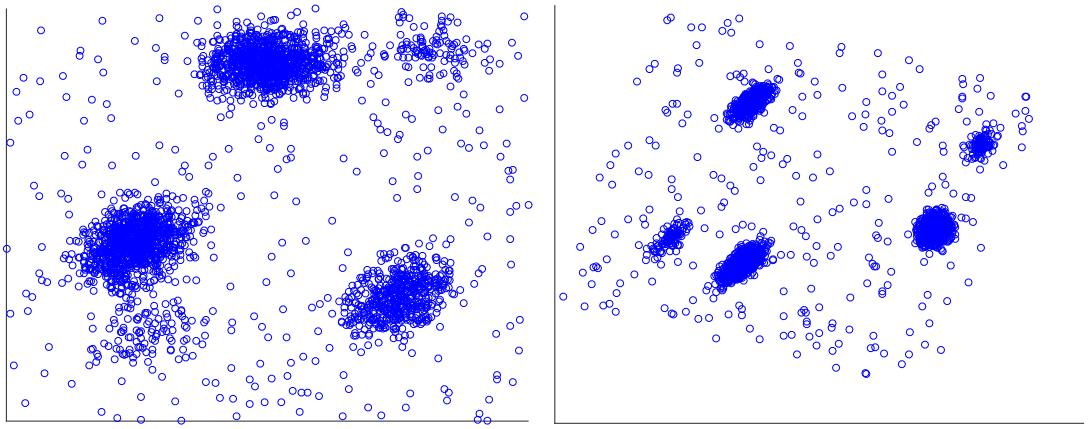


Figure 1. Scatter plots of *Sim5* datasets.

In addition, we use six real datasets that include *Steel Plates*, *Ionosphere*, and *Satimage (Train)* from the UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets.html *Iris* and *Arrhythmia* from MATLAB's sample datasets, https://se.mathworks.com/help/stats/_bq9uxn4.html, and the *USPS* dataset. https://www.otexts.org/1577 A summary of all these datasets can be seen in Table 2. For these real datasets, even if there are class labels provided, we do not compare the clustering results with the class information because of the completely unsupervised scenario with the internal validation indices. Moreover, the classes need not correspond to the clusters determined by the prototype-based clustering algorithm, since the probability density functions of the classes are not necessarily spherically symmetric. For these datasets, we therefore only study and compare the stability of the suggestions on the numbers of clusters for different indices.

Table 2. Description of datasets.

Data	Size	Dimensions	Clusters	Description
S	5000	2	15	Varying overlap
G2	2048	1–1024	2	Varying overlap and dimensionality
DIM	1024	32–1024	16	Varying dimensionality
A	3000–7500	2	20–50	Varying number of clusters
Unbalance	6500	2	8	Both dense and sparse clusters
Birch	100,000	2	1–100	Varying structure
Sim5	2970	2–10	5	Small subclusters close to bigger ones
Data	**Size**	**Dimensions**	**Classes**	**Description**
Iris	150	4	3	Three species of iris
Arrhythmia	452	279	13	Different types of cardiac arrhythmia
Steel Plates	1941	27	7	Steel plates faults
Ionosphere	351	34	2	Radar returns from the ionosphere
USPS	9298	256	10	Numeric data from scanned handwritten digits
Satimage (Train)	6435	36	6	Satellite images

Finally, our datasets include the following: *A1*, with 20 spherical clusters and some overlap; the *S*-sets, including four datasets with 15 Gaussian clusters, and cluster overlap increasing gradually from *S1* to *S4*; *Dim*-sets, including six datasets with 16 well-separated clusters in high-dimensional space and dimensions varying from 32 to 1024; a subset of *Birch2*, including 19 datasets

with 2–20 clusters with their centroids on a sine curve; *Unbalance*, with eight clusters in two separate groups, the one having three dense and small clusters and the other five more sparse and bigger clusters; and a subset of *G2*, with 20 datasets—the lowest and highest overlap in ten different dimensions from 1 to 1024. Finally, together with the six real datasets, we had altogether 62 datasets in our tests.

4. Results

In this section, we provide the results of the clustering validation index tests with K-medians, K-means, and K-spatialmedians clustering. Results for the synthetic datasets are combined in Table A1, where each cell includes the results for all three methods with 'cb' referring to the city-block distance ($p = q = 1$), 'se' to the squared Euclidean distance ($p = q = 2$), and 'ec' to the Euclidean distance ($p = 2,\ q = 1$). In addition, the convergence properties of the clustering algorithms are compared for varying K values.

4.1. CVIs for Synthetic Datasets

For the *Dim*-sets, results were equal (and correct) in all dimensions from 32 to 1024 and for the *G2*-sets results were equal (and correct) from dimension eight upwards. Therefore, these have been excluded from Table A1. Correct suggestions for the number of clusters are marked in bold.

The most challenging synthetic datasets seem to be *Unbalance*, *Sim5D2*, and *Sim5D10*. Only a few indices were able to recognize the correct number and no single index managed to solve both the *Unbalance* and the *Sim* sets.

As concluded in the previous studies (see Section 2.4), different indices seem to work better with different kinds of datasets. However, there are also a lot of differences in the general performances of the tested CVIs. The overall success rates for the CVIs, i.e., for how many of the datasets (%) it gave correct suggestions, can be seen in Table 3, listed separately for each distance measure.

Table 3. Right suggestions for the 56 synthetic datasets (number of right suggestions/number of datasets).

Index	City-Block	Squared Euclidean	Euclidean
KCE	85.7%	87.5%	85.7%
WB	87.5%	80.4%	87.5%
CH	58.9%	85.7%	57.1%
DB	60.7%	60.7%	62.5%
PBM	87.5%	64.3%	89.3%
RT	60.7%	58.9%	64.3%
WG	94.6%	96.4%	92.9%

In conclusion, the WG index outperforms all the other indices in all three distance measures and clustering approaches. WB and KCE also perform very well in general. For some indices, the performances vary between different distances; for example, CH works very well with the squared Euclidean distance, while PBM clearly works better with city-block and Euclidean distances. As a whole, the recommendation for the use of indices is as follows: for the original K-means, WG, KCE, and CH are the three best indices, and for the robust variants with K-medians and K-spatialmedians, WG, PBM, and WB have the highest success rate.

4.2. CVIs for Real Datasets

As mentioned, here we only observe and compare the stability of the clustering results. The results are combined in Table 4. With real datasets, a typical behavior of the internal validation indices is the suggestion of only a small number of clusters. This is especially true for KCE and CH, even if we know that there are observations from multiple classes, with the class boundaries having unknown

forms and shapes. Different from the other indices, the results of DB seem to deviate a lot, with high variability over the datasets. The same can happen for RT, with squared Euclidean distance.

Table 4. Internal CVIs results for real datasets.

cb, se, ec	KCE	WB	CH	DB	PBM	RT	WG
Iris	2, 3, 2	2, 3, 2	2, 3, 2	2, 2, 2	3, 22, 3	2, 2, 2	2, 2, 2
Arrhythmia	2, 2, 2	2, 5, 2	2, 2, 2	25, 24, 17	2, 14, 2	2, 25, 3	2, 25, 25
Steel	2, 2, 2	3, 5, 2	2, 2, 2	7, 3, 7	3, 7, 2	2, 2, 3	3, 2, 3
Ionosphere	2, 2, 2	2, 2, 2	2, 2, 2	11, 23, 2	3, 20, 3	4, 4, 4	4, 2, 2
USPS	2, 2, 2	2, 4, 2	2, 2, 2	2, 18, 11	2, 4, 4	2, 12, 7	2, 2, 7
Satimage (Train)	2, 3, 2	3, 6, 2	2, 3, 2	3, 3, 3	3, 6, 3	3, 3, 3	3, 3, 3

Based on the observed behavior, the most stable, and therefore the recommended indices, seem to be WB, PBM, and WG. However, when the data is of higher dimension without a Gaussian structure, the curse-of-dimensionality [50] might be the reason for the basic suggestions of a low number of clusters. Therefore, to obtain more fine-tuned clustering results and validation index evaluations, it might be necessary to use the prototype-based clustering in a hierarchical manner, as suggested in [16,51]. For example, many indices suggested three clusters for the *Sim5* datasets, but after a further partitioning of the data, new index values could be calculated for those three clusters separately, and the correct division into five clusters altogether could be revealed.

4.3. Convergence

For each repetition, the number of iterations needed for convergence, T, was saved. Median values of T for synthetic datasets were: 19 for K-medians, 19 for K-means, and 21 for K-spatialmedians. K-spatialmedians requires slightly more iterations than K-means and K-medians. In practice, the effect of the total running time between $T = 19$ and $T = 21$ is negligible. For real datasets, median values of T are again similar: 15 for K-medians, 17 for K-means, and 16 for K-spatialmedians. K-means performs slightly worse than the robust K-medians and K-spatialmedians for real datasets. It is known that K-means is sensitive to noise, which could explain these results. Overall, based on the median values of T, there seems to be no practical difference between the convergence of K-medians, K-means, and K-spatialmedians.

We plotted and analyzed the median of T as a function of K for each dataset separately in order to compare the convergence characteristics of K-means, K-medians, and K-spatialmedians. The most relevant plots are shown in Figure A1. Even though the median values of T are close to each other in general, there are some interesting differences with multiple datasets.

From Figure A1, we can observe that the robust location estimates, median and spatial median, clearly converge faster than the mean for the *S4* dataset, which has highly overlapping clusters. For the *USPS* dataset, K-medians seems to converge slightly faster than K-means. In the figure, plots for datasets *b2-sub-5*, *b2-sub-10*, *b2-sub-15*, and *b2-sub-20* show that K-means converges faster than K-medians and K-spatialmedians when K is smaller than the number of clusters in the dataset. This might be because the mean location estimate is more sensitive to movement towards the center of multiple combined clusters than the robust location estimates since outlying points within a cluster affect it more than the robust location estimates. The plots of datasets *G2-2-10* and *G2-64-10* demonstrate how the curse-of-dimensionality greatly affects K-medians when compared to K-means and K-spatialmedians. K-medians converge faster than K-means and K-spatialmedians in the 2-dimensional case; however, from the 64-dimensional case onward the reverse is observed.

5. Conclusions

Tests for a representative set of previously qualified internal clustering validation indices with many datasets, for the most common prototype-based clustering framework with multiple statistical

estimates of cluster location, were reported here. This study further confirmed the conclusions in the previous index comparisons that no single CVI dominates in each context, and some indices are better suited to different kinds of data. Therefore, it is recommended to utilize multiple indices when doing cluster analysis. However, in our tests with the synthetic datasets, the WG index outperformed other indices in all distance measures used, also showing stable behavior with real datasets. It found the correct number of clusters for all datasets, except the *Sim5* with different sized clusters and overlap. Due to this high performance, the WG index would be worth studying more in the future. In addition, PBM was an index with a high and stable performance, along with WB for robust clustering and KCE and CH for K-means. The correct number of clusters in the *Sim5* datas was only found with WB and KCE for K-means and PBM with robust estimates. Based on these tests, DB and RT are not recommended. In general, the indices seem to perform worse with higher cluster overlap and some of them (CH, DB, RT) fail more often when the number of clusters grows.

Here, we also extended previous index comparisons using K-means clustering, with robust K-medians and K-spatialmedians clusterings. Just like the indices, the different distance measures also seem to work differently with different datasets and, in addition, some indices seem to work better with different distances. For instance, for the synthetic datasets, the PBM index clearly works better with city-block and Euclidean distances. Therefore, the usage of robust K-medians and K-spatialmedians could bring added value when trying to find the optimal number of clusters present in the data.

In addition, the convergence properties of the clustering algorithms were compared. Based on the experiments, the number of iterations needed for convergence varies for different datasets between the methods, e.g., K-means requires more iterations than the robust clustering methods for noisy datasets, while K-medians is most affected by an increase in dimensionality. Moreover, K-means++-type initialization strategy seems to work well with city-block and Euclidean distance.

Moreover, as many indices often suggest quite low numbers of clusters, due to clusters being seemingly close to each other, when observed globally in a high-dimensional space, the hierarchical application of a partitional prototype-based clustering algorithm is recommended, in order to improve recognition of possible clusters at different resolution levels.

Acknowledgments: The authors would like to thank the editor and reviewers for their insightful suggestions, which improved the paper.

Author Contributions: Tommi Kärkkäinen conceived and designed this research; Joonas Hämäläinen and Susanne Jauhiainen performed the experiments; Joonas Hämäläinen, Susanne Jauhiainen, and Tommi Kärkkäinen wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

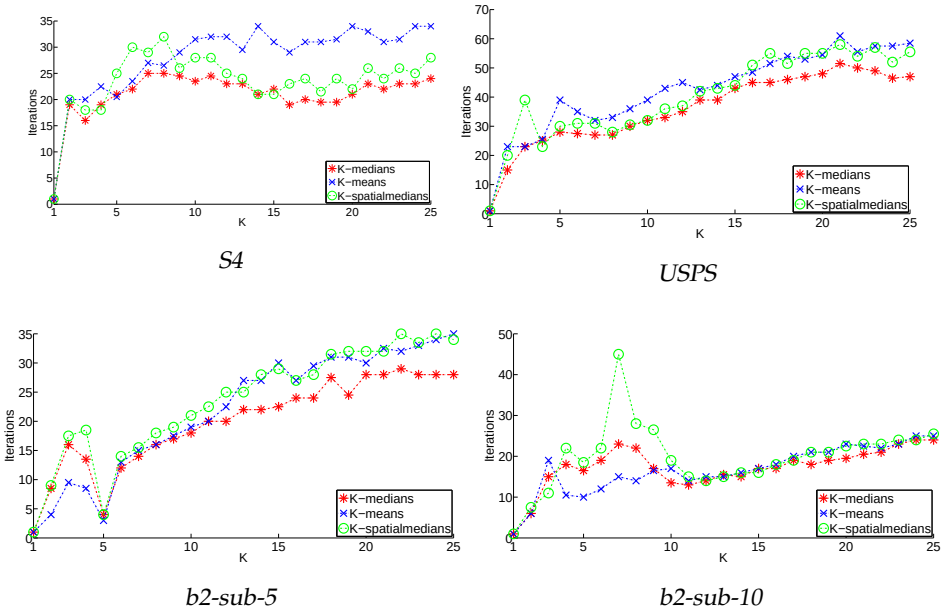
Appendix A

Table A1. Internal CVIs results for synthetic datasets.

cb, se, ec	KCE	WB	CH	DB	PBM	RT	WG
Unbalance	5, 20, 5	5, 20, 5	5, 20, 5	**8**, **8**, **8**	5, 25, 5	**8**, 2, **8**	**8**, **8**, **8**
a1	2, **20**, 2	2, **20**, 2	2, **20**, 2	**20**, 19, 16	6, 24, 6	2, 18, 17	**20**, **20**, **20**
Sim5D10	3, **5**, 3	3, **5**, 3	3, 3, 3	3, 3, 3	**5**, 10, **5**	3, 3, 3	3, 3, 3
Sim5D2	3, **5**, 3	3, **5**, 3	3, 3, 3	3, 3, 3	**5**, 16, **5**	3, 3, 3	3, 3, 3
S1	2, **15**, 2	**15**, **15**, **15**	2, **15**, 2	**15**, **15**, **15**	**15**, **15**, **15**	**15**, **15**, **15**	**15**, **15**, **15**
S2	2, **15**, 2	2, **15**, 3	2, **15**, 2	**15**, **15**, **15**	**15**, **15**, **15**	**15**, **15**, **15**	**15**, **15**, **15**
S3	2,**15**, 2	2, **15**, 2	2, **15**, 2	7, 13, **15**	4, **15**, 4	4, 4, **15**	**15**, **15**, **15**
S4	2, **15**, 2	2, **15**, 3	2, **15**, 2	17, 17, **15**	5, 23, 5	17, 13, **15**	16, **15**, 16
DIM032	**16**, **16**, **16**	**16**, **16**, **16**	**16**, **16**, **16**	**16**, **16**, **16**	**16**, **16**, **16**	**16**, **16**, **16**	**16**, **16**, **16**
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
DIM1024	**16**, **16**, **16**	**16**, **16**, **16**	**16**, **16**, **16**	**16**, **16**, **16**	**16**, **16**, **16**	**16**, **16**, **16**	**16**, **16**, **16**

Table A1. *Cont.*

cb, se, ec	KCE	WB	CH	DB	PBM	RT	WG
b2-sub-2	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, 20, **2**	**2**, **2**, **2**	**2**, **2**, **2**
b2-sub-3	**3**, **3**, **3**	**3**, **3**, **3**	**3**, **3**, **3**	**3**, **3**, **3**	**3**, **3**, **3**	**3**, **3**, **3**	**3**, **3**, **3**
b2-sub-4	**4**, **4**, **4**	**4**, **4**, **4**	**4**, **4**, **4**	**4**, **4**, **4**	**4**, **4**, **4**	**4**, **4**, **4**	**4**, **4**, **4**
b2-sub-5	**5**, **5**, **5**	**5**, **5**, **5**	**5**, **5**, 2	**5**, **5**, **5**	**5**, **5**, **5**	**5**, **5**, **5**	**5**, **5**, **5**
b2-sub-6	**6**, **6**, **6**	**6**, **6**, **6**	2, **6** ,2	5, **6**, 5	**6**, **6**, **6**	5, **6**, **6**	**6**, **6**, **6**
b2-sub-7	**7**, **7**, **7**	**7**, **7**, **7**	2, **7**, 2	5, 6, 6	**7**, 14, **7**	2, 2, 2	**7**, **7**, **7**
b2-sub-8	**8**, **8**, **8**	**8**, **8**, **8**	2, **8**, 2	6, 6, 7	**8** ,17, **8**	2, 2, 2	**8**, **8**, **8**
b2-sub-9	**9**, 19, **9**	**9**, 19, **9**	2, **9**, 2	6, 7, 8	**9**, 21, **9**	2, 7, 7	**9**, **9**, **9**
b2-sub-10	**10**, 21, **10**	**10**, 21, **10**	2, 21, 2	8, 8, 9	**10**, 25, **10**	8, 8, 8	**10**, **10**, **10**
b2-sub-11	**11**, 23, **11**	**11**, 23, **11**	2, 23, 3	9, 9, 10	**11**, 24, **11**	9, 9, 8	**11**, **11**, **11**
b2-sub-12	**12**, 25, **12**	**12**, 25, **12**	2, 25, 3	10, 10, 11	**12**, 25, **12**	10, 10, 9	**12**, **12**, **12**
b2-sub-13	**13**, **13**, **13**	**13**, 24, **13**	2, **13**, 2	11, 11, 12	**13**, 24, **13**	11, 11, 11	**13**, **13**, **13**
b2-sub-14	**14**, **14**, **14**	**14**, **14**, **14**	2, **14**, 2	12, 12, 13	**14**, **14**, **14**	12, 12, 12	**14**, **14**, **14**
b2-sub-15	**15**, **15**, **15**	**15**, **15**, **15**	2, **15**, 2	13, 13, 14	**15**, **15**, **15**	13, 13, 13	**15**, **15**, **15**
b2-sub-16	**16**, **16**, **16**	**16**, **16**, **16**	2, **16**, 2	14, 14, 15	**16**, **16**, **16**	14, 14, 14	**16**, **16**, **16**
b2-sub-17	**17**, **17**, **17**	**17**, **17**, **17**	2, **17**, 2	15, 15, 16	**17**, **17**, **17**	15, 2, 2	**17**, **17**, **17**
b2-sub-18	**18**, **18**, **18**	**18**, **18**, **18**	2, **18**, 2	15, 15, 16	**18**, **18**, **18**	2, 2, 2	**18**, **18**, **18**
b2-sub-19	**19**, **19**, **19**	**19**, **19**, **19**	2, **19**, 2	16, 16, 17	**19**, **19**, **19**	2, 2, 2	**19**, **19**, **19**
b2-sub-20	**20**, **20**, **20**	**20**, **20**, **20**	2, **20**, 2	2, 2, 2	**20**, 21, **20**	2, 2, 2	**20**, **20**, 2
G2-1-10	**2**, 25, **2**	**2**, 25, **2**	**2**, 25, **2**	**2**, **2**, **2**	25, 25, 25	**2**, **2**, **2**	**2**, **2**, **2**
G2-1-100	**2**, 25, **2**	**2**, 25, **2**	**2**, 25, **2**	22, 25, 22	21, 25, 21	3, 3, 3	**2**, **2**, **2**
G2-2-10	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, 20, **2**	**2**, **2**, **2**	**2**, **2**, **2**
G2-2-100	**2**, **2**, **2**	**2**, 22, **2**	**2**, **2**, **2**	21, 19, 25	8, 23, **2**	10, 7, 7	**2**, **2**, **2**
G2-4-10	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**
G2-4-100	**2**, **2**, **2**	**2**, 3, **2**	**2**, **2**, **2**	**2**, 17, 23	**2**, 6, **2**	**2**, 16, 16	**2**, **2**, **2**
G2-8-10	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**
G2-8-100	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
G2-1024-10	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**
G2-1024-100	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**	**2**, **2**, **2**



S4 *USPS*

b2-sub-5 *b2-sub-10*

Figure A1. *Cont.*

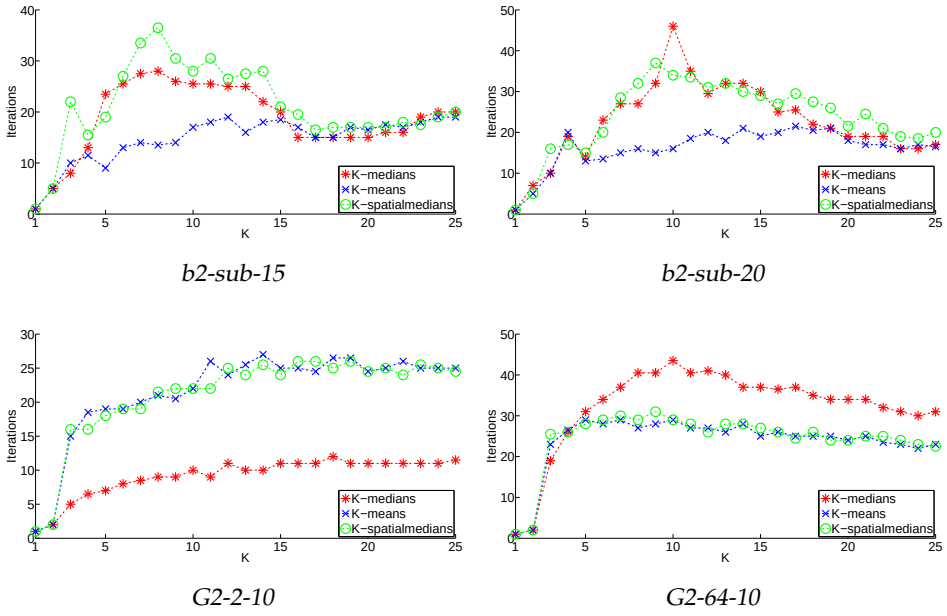


Figure A1. Median of the number of iterations needed for convergence with varying *K*.

References

1. Jain, A.K.; Murty, M.N.; Flynn, P.J. Data clustering: A review. *ACM Comput. Surv.* **1999**, *31*, 264–323.
2. Aggarwal, C.C.; Reddy, C.K. *Data Clustering: Algorithms and Applications*; CRC Press: New York, NY, USA, 2013.
3. Xie, X.L.; Beni, G. A validity measure for fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 841–847.
4. Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666.
5. Zaki, M.J.; Meira, W., Jr. *Data Mining and Analysis: Fundamental Concepts and Algorithms*; Cambridge University Press: New York, NY, USA, 2014.
6. Saarela, M.; Hämäläinen, J.; Kärkkäinen, T. Feature Ranking of Large, Robust, and Weighted Clustering Result. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Jeju, Korea, 23–26 May 2017; pp. 96–109.
7. Lloyd, S. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **1982**, *28*, 129–137.
8. Khan, S.S.; Ahmad, A. Cluster center initialization algorithm for K-modes clustering. *Expert Syst. Appl.* **2013**, *40*, 7444–7456.
9. Arthur, D.; Vassilvitskii, S. K-means++: The advantages of careful seeding. In Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, USA, 7–9 January 2007; pp. 1027–1035.
10. Xu, R.; Wunsch, D. Survey of clustering algorithms. *IEEE Trans. Neural Netw.* **2005**, *16*, 645–678.
11. Hruschka, E.R.; Campello, R.J.; Freitas, A.A.; de Carvalho, A.C.P.L.F. A survey of evolutionary algorithms for clustering. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2009**, *39*, 133–155.
12. Han, J.; Kamber, M.; Tung, A. Spatial Clustering Methods in Data Mining: A Survey. In *Geographic Data Mining and Knowledge Discovery*; Miller, H., Han, J., Eds.; CRC Press: Boca Raton, FL, USA, 2001.
13. Huber, P.J. *Robust Statistics*; John Wiley & Sons Inc.: New York, NY, USA, 1981.
14. Rousseeuw, P.J.; Leroy, A.M. *Robust Regression and Outlier Detection*; John Wiley & Sons Inc.: New York, NY, USA, 1987; p. 329.
15. Hettmansperger, T.P.; McKean, J.W. *Robust Nonparametric Statistical Methods*; Edward Arnold: London, UK, 1998; p. 467.
16. Saarela, M.; Kärkkäinen, T. Analysing Student Performance using Sparse Data of Core Bachelor Courses. *J. Educ. Data Min.* **2015**, *7*, 3–32.

17. Kärkkäinen, T.; Heikkola, E. Robust Formulations for Training Multilayer Perceptrons. *Neural Comput.* **2004**, *16*, 837–862.
18. Croux, C.; Dehon, C.; Yadine, A. The *k*-step spatial sign covariance matrix. *Adv. Data Anal. Classif.* **2010**, *4*, 137–150.
19. Äyrämö, S. Knowledge Mining Using Robust Clustering. Ph.D. Thesis, Jyväskylä Studies in Computing 63, University of Jyväskylä, Jyväskylä, Finland, 2006.
20. Shannon, C.E. A mathematical theory of communication. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **2001**, *5*, 3–55.
21. Strehl, A.; Ghosh, J. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **2002**, *3*, 583–617.
22. Zhao, Q.; Fränti, P. WB-index: A sum-of-squares based index for cluster validity. *Data Knowl. Eng.* **2014**, *92*, 77–89.
23. Davies, D.L.; Bouldin, D.W. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *PAMI-1*, 224–227.
24. Caliński, T.; Harabasz, J. A dendrite method for cluster analysis. *Commun. Stat. Theory Methods* **1974**, *3*, 1–27.
25. Ray, S.; Turi, R.H. Determination of number of clusters in k-means clustering and application in colour image segmentation. In Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques, Calcutta, India, 27–29 December 1999; pp. 137–143.
26. Rendón, E.; Abundez, I.; Arizmendi, A.; Quiroz, E.M. Internal versus external cluster validation indexes. *Int. J. Comput. Commun.* **2011**, *5*, 27–34.
27. Halkidi, M.; Batistakis, Y.; Vazirgiannis, M. On clustering validation techniques. *J. Intell. Inf. Syst.* **2001**, *17*, 107–145.
28. Kuncheva, L.I.; Vetrov, D.P. Evaluation of stability of k-means cluster ensembles with respect to random initialization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 1798–1808.
29. Handl, J.; Knowles, J. An evolutionary approach to multiobjective clustering. *IEEE Trans. Evolut. Comput.* **2007**, *11*, 56–76.
30. Jauhiainen, S.; Kärkkäinen, T. A Simple Cluster Validation Index with Maximal Coverage. In Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESAINN 2017), Bruges, Belgium, 26–28 April 2017; pp. 293–298.
31. Kim, M.; Ramakrishna, R. New indices for cluster validity assessment. *Pattern Recognit. Lett.* **2005**, *26*, 2353–2363.
32. Maulik, U.; Bandyopadhyay, S. Performance evaluation of some clustering algorithms and validity indices. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 1650–1654.
33. Arbelaitz, O.; Gurrutxaga, I.; Muguerza, J.; Pérez, J.M.; Perona, I. An extensive comparative study of cluster validity indices. *Pattern Recognit.* **2013**, *46*, 243–256.
34. Liu, Y.; Li, Z.; Xiong, H.; Gao, X.; Wu, J. Understanding of internal clustering validation measures. In Proceedings of the 2010 IEEE 10th International Conference on.Data Mining (ICDM), Sydney, Australia, 13–17 December 2010; pp. 911–916.
35. Agrawal, K.; Garg, S.; Patel, P. Performance measures for densed and arbitrary shaped clusters. *Int. J. Comput. Sci. Commun.* **2015**, *6*, 338–350.
36. Halkidi, M.; Vazirgiannis, M. Clustering validity assessment: Finding the optimal partitioning of a data set. In Proceedings of the IEEE International Conference on Data Mining (ICDM 2001), San Jose, CA, USA, 29 November–2 December 2001; pp. 187–194.
37. Lughofer, E. A dynamic split-and-merge approach for evolving cluster models. *Evol. Syst.* **2012**, *3*, 135–151.
38. Lughofer, E.; Sayed-Mouchaweh, M. Autonomous data stream clustering implementing split-and-merge concepts—Towards a plug-and-play approach. *Inf. Sci.* **2015**, *304*, 54–79.
39. Ordonez, C. Clustering binary data streams with K-means. In Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. San Diego, CA, USA, 13 June 2003; pp. 12–19.
40. Bagirov, A.M.; Yearwood, J. A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems. *Eur. J. Oper. Res.* **2006**, *170*, 578–596.
41. Karmitsa, N.; Bagirov, A.; Taheri, S. *MSSC Clustering of Large Data using the Limited Memory Bundle Method*; Discussion Paper; University of Turku: Turku, Finland, 2016.

42. Kärkkäinen, T.; Majava, K. Nonmonotone and monotone active-set methods for image restoration, Part 1: Convergence analysis. *J. Optim. Theory Appl.* **2000**, *106*, 61–80.
43. Kärkkäinen, T.; Kunisch, K.; Tarvainen, P. Augmented Lagrangian Active Set Methods for Obstacle Problems. *J. Optim. Theory Appl.* **2003**, *119*, 499–533.
44. Kärkkäinen, T.; Kunisch, K.; Majava, K. Denoising of smooth images using L^1-fitting. *Computing* **2005**, *74*, 353–376.
45. Pakhira, M.K.; Bandyopadhyay, S.; Maulik, U. Validity index for crisp and fuzzy clusters. *Pattern Recognit.* **2004**, *37*, 487–501.
46. Desgraupes, B. "ClusterCrit: Clustering Indices". R Package Version 1.2.3., 2013. Available online: https://cran.r-project.org/web/packages/clusterCrit/ (accessed on 6 September 2017).
47. Milligan, G.W.; Cooper, M.C. An examination of procedures for determining the number of clusters in a data set. *Psychometrika* **1985**, *50*, 159–179.
48. Fränti, P.; Sieranoja, S. K-means properties on six clustering benchmark datasets. *Algorithms* **2017**, submitted.
49. Saarela, M.; Kärkkäinen, T. Do country stereotypes exist in educational data? A clustering approach for large, sparse, and weighted data. In Proceedings of the 8th International Conference on Educational Data Mining (EDM 2015), Madrid, Spain, 26–29 June 2015; pp. 156–163.
50. Verleysen, M.; François, D. The Curse of Dimensionality in Data Mining and Time Series Prediction. In Proceedings of the International Work-Conference on Artificial Neural Networks (IWANN), Cadiz, Spain, 14–16 June 2005; Volume 5, pp. 758–770.
51. Wartiainen, P.; Kärkkäinen, T. Hierarchical, prototype-based clustering of multiple time series with missing values. In Proceedings of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2015), Bruges, Belgium, 22–24 April 2015; pp. 95–100.