2017-07-20

An Analysis of the Application of Simplified Silhouette to the Evaluation of k-means Clustering Validity

Fei Wang
Technological University Dublin, d13122837@mytudublin.ie

Hector-Hugo Franco-Penya
Technological University Dublin, hector.franco@tudublin.ie

John D. Kelleher
Technological University Dublin, john.d.kelleher@tudublin.ie

See next page for additional authors

Recommended Citation

Franco-Penya, H. et al. (2017) An Analysis of the Application of Simplified Silhouette to the Evaluation of k-means Clustering Validity. *13th International Conference on Machine Learning and Data Mining MLDM 2017, July 15-20, 2017, New York, USA.*

An Analysis of the Application of Simplified Silhouette to the Evaluation of *k*-means Clustering Validity

Fei Wang[1,3], Hector-Hugo Franco-Penya[1], John D. Kelleher[1,3], John Pugh[2], and Robert Ross[1,3]

[1] School of Computing, Dublin Institute of Technology, Ireland
[2] Nathean Technologies Ltd. Dublin, Ireland
[3] ADAPT Research Centre
d13122837@mydit.ie

Abstract. Silhouette is one of the most popular and effective internal measures for the evaluation of clustering validity. Simplified Silhouette is a computationally simplified version of Silhouette. However, to date Simplified Silhouette has not been systematically analysed in a specific clustering algorithm. This paper analyses the application of Simplified Silhouette to the evaluation of k-means clustering validity and compares it with the k-means Cost Function and the original Silhouette from both theoretical and empirical perspectives. The theoretical analysis shows that Simplified Silhouette has a mathematical relationship with both the k-means Cost Function and the original Silhouette, while empirically, we show that it has comparative performances with the original Silhouette, but is much faster in calculation. Based on our analysis, we conclude that for a given dataset the k-means Cost Function is still the most valid and efficient measure in the evaluation of the validity of k-means clustering with the same k value, but that Simplified Silhouette is more suitable than the original Silhouette in the selection of the best result from k-means clustering with different k values.

Keywords: k-means, clustering validity, internal measures, Simplified Silhouette, Silhouette, Cost Function

1 Introduction

As a fundamental method in data mining and machine learning, clustering aims to partition data into homogeneous groups [13,1,8]. Unlike supervised machine learning methods, clustering does not require external labels as ground truth, but investigates the intrinsic structure and characteristics of data, and partitions data into clusters such that the data in the same cluster are more similar to each other than the data in other clusters. Clustering has been applied in many

This is the pre-print version of the paper. The final publication is available at link.springer.com.

domains, such as image and text analysis, biology and so on [8], and also noted as an important part of unsupervised learning in many data mining and machine learning text books [13,1,19,9].

Our research focuses on the application of clustering to the domain of Business Intelligence. The effective segmentation of customer data is a vital tool for commercial users. For this purpose, two specific characteristics need to be considered in the clustering: (1) there are a large proportion of categorical features in customer data; and (2) users don't have much a priori knowledge about clustering. In our previous research [17], we compared different methods for categorical data clustering, such as 1-of-k coding and k-prototypes. In this paper, we look at k-means clustering, and aim to find the best way to automate the selection of the best clustering result from a set of k-means clusterings with different parameter configurations.

k-means is one of the most widely used clustering algorithms due to its ease of implementation, simplicity, efficiency and empirical success [12]. There are two parameters that need to be set before the start of k-means clustering - the number of clusters k and the initial centroids. Given a fixed parameter configuration, k-means will output a fixed clustering result. However, because different parameter configurations usually lead to different clustering results, a single k-means clustering cannot guarantee the best clustering result. The common way to implement k-means is to run it multiple times with different parameter configurations and select the best one from all the clustering results. The process to find the best result is normally based on the evaluation of the clustering validity, that is, the goodness or quality of the clustering result for a dataset [19].

In this paper, we mainly analyse the application of an internal measure for evaluating the clustering validity - Simplified Silhouette - and compare it with other related measures in k-means clustering. We start with a brief introduction to the background of the evaluation of k-means clustering validity in Sect. 2, followed by a theoretical analysis in Sect. 3. In Sect. 4, we outline the design for our empirical analysis, and then in Sect. 5 present and analyse the experimental results. Finally, in Sect. 6 we draw conclusions and outline future work.

2 Background

Normally, there are three types of measures that can be used to evaluate clustering validity in empirical studies [19]: internal measures, external measures and relative measures. Internal measures are based on the intrinsic structure and characteristics of the dataset. External measures are based on labelled datasets such as the ground truth, and compare the clustering results with the existing labels to uncover how good the clustering is. Relative measures are used to compare different clustering usually with the same clustering algorithm but different parameter settings. Because clustering is usually used for the situation in which users do not have any labelled data, internal measures are the most generally

used measures for the evaluation of clustering validity in practice and therefore our research focus in this paper.

Internal measures are usually some indices designed to show the compactness and separation of data [19]. The compactness means that the data within the same cluster should be close to each other, and the separation means that the data in different clusters should be widely spaced. There are numerous different internal measures for the evaluation of clustering validity. Different measures show these two concepts in different ways.

First of all, for some clustering algorithms like k-means, the design of the algorithm aims to minimise a cost function. Intuitively, the cost function can be considered as an internal measure for evaluating the clustering validity of this specific algorithm. The k-means Cost Function [8] is defined as the sum of all the distances of each point to its cluster centroid. The process of k-means is designed specifically to reduce the Cost Function by centroid shifts and re-assignments of the data to its closest cluster until the Cost Function converges to a minimum (the optimum), so the convergence of the Cost Function is a monotonic process in k-means. Additionally, because the distances of each data point to its centroid have been calculated during the process of k-means, the calculation of the k-means Cost Function is only to sum up these distances, which requires few extra calculations. Therefore, we can consider the k-means Cost Function as the default internal measure for k-means and the clustering result with the smallest Cost Function as the best result or global optimum. However, for the evaluation of the validity of clustering with different k values, using the Cost Function measure is problematic because it tends to reduce as the k value increases and, consequently, the Cost Function measure has an intrinsic bias toward selecting the result with the largest k as the best result [8]. Therefore we have to use other internal measures.

In addition to the kind of internal measures designed specifically for a clustering algorithm like the k-means Cost Function, there are quite a lot of general internal measures that can be applied in the evaluation of the validity of a set of clustering algorithms: the Dunn index [3] adopts the maximum intra-cluster distance and the minimum inter-cluster distance, the Davies-Bouldin (DB) index [2] evaluates the dispersion of data based on the distances between cluster centroids, the C-index [19] takes the sum of a set of the smallest distances as the baseline, and the SD index [6] is defined based on the concepts of the average scattering for clusters and total separation between clusters.

Silhouette [11] analyses the distances of each data point to its own cluster and its closest neighbouring cluster (defined as the average distance of a data point to all the other data points in its own cluster and that to all the data points in the neighbouring cluster nearest to the data point). Different from most other internal measures, Silhouette is not only used for the evaluation of the validity of a full clustering, but also can be used for that of a single cluster or even a single data point to see if it is well clustered. The calculation of Silhouette starts from each data point, and the Silhouette value of a cluster or a full clustering is just the average of point Silhouette values for all the data involved. Regarding our

focus on customer segmentation, it is the advantage of Silhouette that it shows if each customer or customer cluster is well segmented.

Compared with that of the k-means Cost Function, the bias of Silhouette in k-means clustering toward selecting the result with the largest k as the best result exists only when the number of clusters is almost as big as the number of data points. In other situations they can be applied to evaluate the validity of k-means clustering with different k values and select the best result, which can be seen in the experimental results in Sect. 5.4.

Since it was created, Silhouette has become one of the most popular internal measures for clustering validity evaluation. In [18,10,15] it is compared with a set of other internal measures and proven to be one of the most effective and generally applicable measures. However, when Silhouette is applied in the evaluation of k-means clustering validity, many more extra calculations are required, and the extra calculations increase following a power law corresponding to the size of the dataset, because the calculation of the Silhouette index is based on the full pairwise distance matrix over all data. This is a challenging disadvantage of Silhouette. From this perspective, Silhouette needs to be simplified for k-means to improve its efficiency.

Simplified Silhouette was, to our knowledge, first introduced by Hruschka in [7], and used as one of the internal measures in his following research. It inherits most characteristics from Silhouette and therefore can be used in the evaluation of the validity of not only a full clustering but also a single cluster or a single data point. On the other hand, the distance of a data point to a cluster in Simplified Silhouette is represented with the distance to the cluster centroid instead of the average distance to all (other) data points in the cluster, just as in the k-means Cost Function.

However, Simplified Silhouette has not been systematically analysed or introduced to the evaluation of k-means clustering validity. In this paper, the application of Simplified Silhouette to k-means will be analysed and compared with that of the k-means Cost Function and the original Silhouette from both theoretical and empirical perspectives. The specific research targets are to solve these two questions:

1 Does Simplified Silhouette or the original Silhouette perform as well as the k-means Cost Function in the evaluation of k-means clustering validity?
2 Does Simplified Silhouette have competitive performances to the original Silhouette in the evaluation of k-means clustering validity?

In the next section, we will start with theoretical analysis of the mathematical relationships between Simplified Silhouette and the other two internal measures.

3 Theoretical Analysis

3.1 Mathematics Expressions

Let $X = \{X_1, X_2, ..., X_n\}$ be a set of n data points. $X_i\,(1 \leq i \leq n)$ is one of the data points, which can be represented as $[x_{i,1}, x_{i,2}, ..., x_{i,m}]$, where m is the

number of features. Given the set of data points X, an integer $k\,(2 \leq k \leq n)$ and k initial centroids in the domain of X, the k-means algorithm aims to find a clustering of X into k clusters such that it minimises the k-means Cost Function, which is defined as the sum of the distances from a data point to the centroid of the cluster it is assigned to as follows:

$$CF(X, C) = \sum_{l=1}^{k} \sum_{i=1}^{n} w_{i,l} d_E(X_i, C_l) \tag{1}$$

where $d_E(\cdot, \cdot)$ is the squared Euclidean distance, which is the most commonly used distance for clustering [14], $C = \{C_1, C_2, ..., C_k\}$, which is a set of cluster centroids after clustering, and $w_{i,l}$ is the indicator function, which equals to 1 when X_i is in C_l and 0 when X_i is not in C_l. As defined, the smaller the cost function is, the better the corresponding k-means clustering result is, so it can be considered as the default internal measure for k-means clustering. However, for the evaluation of the validity of clustering with different k values, using the Cost Function measure is problematic because it tends to reduce as the k value increases. Therefore, we need other general internal measures like Silhouette.

The calculation of Silhouette doesn't use any representative of a cluster (such as the cluster centroids used by the Cost Function), but is based on the full pairwise distance matrix over all data. For a single data point X_i, its Silhouette value $sil(i)$ is calculated as:

$$sil(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{2}$$

where $a(i)$ is the distance of X_i to its own cluster, which is defined as the average distance of X_i to all the other data points in its own cluster h as[4]:

$$a(i) = \frac{\sum_{\substack{p=1 \\ p \neq i}}^{n} w_{p,h} d_E(X_i, X_p)}{n_h - 1} \tag{3}$$

where n_h is the number of data points in the cluster h. $b(i)$ is the distance of X_i to its closest neighbouring cluster, which is defined as the average distance of X_i to all the data points in its closest neighbouring cluster as:

$$b(i) = \underset{l \neq h}{minimum} \frac{\sum_{p=1}^{n} w_{p,l} d_E(X_i, X_p)}{n_l} \tag{4}$$

The $sil(i)$ ranges from -1 to 1. When $a(i)$ is much smaller than $b(i)$, which means the distance of the data point to its own cluster is much smaller than that to other clusters, the $sil(i)$ is close to 1 to show this data point is well clustered. In the opposite way, the $sil(i)$ is close to -1 to show it is badly clustered.

[4] Here we assume that there are at least two different data points in the cluster. Otherwise, the $a(i)$ is set to be 0, and the $sil(i)$ will be 1.

The Silhouette value of a whole cluster or a full clustering is defined as the average value of $sil(i)$ across all the data involved, e.g. the Silhouette value for a full clustering Sil is defined as follows:

$$Sil = \frac{1}{n}\sum_{i=1}^{n} sil(i) \tag{5}$$

Therefore, the Silhouette value for a full clustering Sil also ranges from -1, which shows a very bad clustering, to 1, which shows a perfect clustering.

Simplified Silhouette adopts a similar approach as that of the original Silhouette, but simplifies the distance of a data point to a cluster from the average distance of X_i to all (other) data points in a cluster to the distance to the centroid of the cluster as follows:

$$a(i)^{'} = d_E(X_i, C_h) \tag{6}$$

$$b(i)^{'} = \underset{l \neq h}{minimum}\, d_E(X_i, C_l); \tag{7}$$

And the Simplified Silhouette value for a single data point $ss(i)$ is defined as:

$$ss(i) = \frac{b(i)^{'} - a(i)^{'}}{max\{a(i)', b(i)'\}} \tag{8}$$

In the same way, the Simplified Silhouette value for a full clustering SS is defined as:

$$SS = \frac{1}{n}\sum_{i=1}^{n} ss(i) \tag{9}$$

The Simplified Silhouette value also ranges from -1 to 1. -1 shows a very bad clustering, while 1 shows a perfect clustering.

3.2 Theoretical Comparison

In some sense, Simplified Silhouette can be considered as the medium between the k-means Cost Function and Silhouette, because it evaluates the distances of each data point to its own cluster and its closest neighbouring cluster as Silhouette, and adopts the centroids from the k-means Cost Function as the representatives of clusters. In this section, we compare these different internal measures from a mathematical perspective.

Firstly, because at the end of k-means clustering the distances of a data point to its closest neighbouring cluster centroid $b(i)^{'}$ is always greater than or equal to the distance to its own cluster centroid $a(i)^{'}$, $max\{a(i)^{'}, b(i)^{'}\}$ in (8) can be simplified to $b(i)^{'}$. The Simplified Silhouette value for a single data point can also be simplified as follows:

$$ss(i) = 1 - \frac{a(i)^{'}}{b(i)^{'}} \tag{10}$$

It can be easily found that $ss(i)$ is always greater than or equal to 0 after k-means, as well as $SS(i)$.

For the comparison with the Cost Function in (1), Simplified Silhouette for all data points in (9) can also be written as:

$$SS = 1 - \sum_{l=1}^{k} \sum_{i=1}^{n} w_{i,l} (\frac{1}{nb(i)^{'}} d_E(X_i, C_l)) \tag{11}$$

where $\frac{1}{nb(i)^{'}} d_E(X_i, C_l)$ can be considered as the weighted distance of X_i to the centroid of its cluster l, and $\sum_{l=1}^{k} \sum_{i=1}^{n} w_{i,l} (\frac{1}{nb(i)^{'}} d_E(X_i, C_l))$ can also be considered as the weighted Cost Function of k-means. The weight $\frac{1}{nb(i)^{'}}$ is the only difference from Cost Function as (1). With the weight, the distance of X_i to its closest neighbouring cluster is taken into account. Given the same Cost Function, when the weight gets larger, that is, the data points are far from the centroid of its closest neighbouring cluster, the weighted Cost Function gets smaller and Simplified Silhouette gets a larger value that is closer to 1 to present a good cluster. Otherwise, the weighted Cost Function gets larger and Simplified Silhouette gets a smaller value that is closer to 0 to present a bad cluster.

For the comparison with Silhouette, we firstly expand $a(i)$ in (3) by expanding the squared Euclidean distance as follows:

$$a(i) = \frac{\sum_{\substack{p=1 \\ p \neq i}}^{n} w_{p,h} (\sum_{j=1}^{m} (x_{i,j} - x_{p,j})^2)}{n_h - 1} \tag{12}$$

Similarly $a(i)^{'}$ in (6) can be expanded as follows:

$$a(i)^{'} = \frac{\sum_{j=1}^{m} (\sum_{p=1}^{n} w_{p,h} (x_{i,j} - x_{p,j}))^2}{n_h^2} \tag{13}$$

We look into the mathematical relationship between (12) and (13), and get the equality as follows:

$$a(i) = \frac{n_h}{n_h - 1} a(i)^{'} + \frac{\sum_{p=1}^{n} \sum_{q=1}^{n} w_{p,h} w_{q,h} d_E(X_p, Y_q)}{2n_h(n_h - 1)} \tag{14}$$

It is shown that the $a(i)$ adds a weight $\frac{n_h}{n_h - 1}$ that is greater than 1 into $a(i)^{'}$, and takes into account another factor - the sum of all the pairwise distances within its cluster with a weight $\frac{1}{2n_h(n_h - 1)}$, therefore is always bigger than $a(i)^{'}$.

Similarly, we can re-write $b(i)$ and $b(i)^{'}$ as follows:

$$b(i) = \underset{l \neq h}{minimum} \frac{\sum_{j=1}^{m} \sum_{p=1}^{n} w_{p,l} (x_{i,j} - x_{p,j})^2}{n_l} \tag{15}$$

$$b(i)^{'} = \underset{l \neq h}{minimum} \frac{\sum_{j=1}^{m} (\sum_{p=1}^{n} w_{p,l} (x_{i,j} - x_{p,j}))^2}{n_l^2} \tag{16}$$

where we denote $\frac{\sum_{j=1}^{m}\sum_{p=1}^{n} w_{p,l}(x_{i,j}-x_{p,j})^2}{n_l}$ as $D_E(X_i,l)$, the distance from a data point X_i to a cluster l that it does not belong to based on Silhouette, while $\frac{\sum_{j=1}^{m}(\sum_{p=1}^{n} w_{p,l}(x_{i,j}-x_{p,j}))^2}{n_l^2}$ as $D_E^{'}(X_i,l)$ based on Simplified Silhouette. Then

$$D_E(X_i,l) = D_E^{'}(X_i,l) + \frac{\sum_{p=1}^{n}\sum_{q=1}^{n} w_{p,l}w_{q,l}d_E(X_p,Y_q)}{2n_l^2} \tag{17}$$

It can be found easily that the $b(i)$ in Silhouette also takes into account one more factor than the $b(i)^{'}$ in Simplified Silhouette - the sum of all the pairwise distances within the corresponding cluster with a weight $\frac{1}{2n_l^2}$.

3.3 Complexity Analysis

Finally, we can analyse the complexity of the computation of these measures. From [16], the overall complexity of the computation of Silhouette is estimated as $O(mn^2)$, while that of Simplified Silhouette is estimated as $O(kmn)$. When k is much smaller than n, Silhouette is much more computationally expensive than Simplified Silhouette. In addition, during the process of k-means clustering, the distance of each data point to its cluster centroid has already been calculated in each iteration, which greatly reduces the calculation of both the Cost Function and Simplified Silhouette. Therefore, the Cost Function and Simplified Silhouette are much more efficient in the evaluation of k-means clustering validity.

3.4 Conclusions of Theoretical Analysis

In summary, from the theoretical comparison, we can conclude that Simplified Silhouette is an internal measure with features related with both the k-means Cost Function and the original Silhouette:

1 It considers more than the k-means Cost Function by additionally bringing in the distance of each data point to its closest neighbouring cluster;
2 It also simplifies Silhouette by ignoring within-cluster pairwise distances.

Therefore, we can consider Simplified Silhouette as a variant of Silhouette for k-means clustering. In the experimental analysis, we will compare the time consumed by different measures, and most importantly, verify the performance of Simplified Silhouette compared with k-means Cost Function and the original Silhouette so that to find out if these mathematical differences lead to performance differences.

4 Experimental Design

4.1 Research Targets

The experimental analysis is designed to evaluate the performances of these three internal measures, the k-means Cost Function, Silhouette and Simplified Silhouette, in the evaluation of k-means clustering validity and to answer specifically the two research questions proposed at the end of Sect. 2.

For the evaluation of the validity of clustering with the same k value, we take the k-means Cost Function as the default measure, and aim to find out if Silhouette or Simplified Silhouette can perform as well as the Cost Function. On the other hand, for the evaluation of the validity of clustering with different k values, we evaluate Silhouette and Simplified Silhouette to find out if Simplified Silhouette has comparative performances to the original Silhouette so that it can be used safely instead of the original Silhouette.

4.2 Datasets

This experiment adopts four real world datasets and four synthetic datasets. The real world datasets are all famous numeric datasets from the UC Irvine Machine Learning Repository (`http://archive.ics.uci.edu/ml/`): Iris, Glass, Wine and Yeast. The labels in these datasets are subjectively labelled only for some specific purposes, so they cannot reflect exactly the intrinsic structure inside the data or the ground-truth k value. Therefore, we ignore the labels of these four datasets in the experiment. The other four datasets are generated artificially for clustering, and hence the labels in them can be used in the evaluation. The first two synthetic datasets are the Dim032 and Dim064 datasets from [5] with 32 dimensions and 64 dimensions respectively. The other two synthetic datasets are the S1 and S3 datasets from [4], which have only two dimensions but many more instances. The clusters in S1 are separated widely from each other, while those in S3 are more compact. We select these different datasets in order to evaluate the internal measures in different situations. Detailed information about the datasets is summarised in Table 1. As discussed above, we only know the desired k values of the four synthetic datasets.

Table 1: Experiment Datasets

No.	**Dataset**	**#Instances**	**#Dimensions**	**Desired k**
1	Iris	150	4	None
2	Glass	214	9	None
3	Wine	178	13	None
4	Yeast	1484	8	None
5	Dim032	1024	32	16
6	Dim064	1024	64	16
7	S1	5000	2	15
8	S3	5000	2	15

4.3 Experimental Process

As introduced in Sect. 1, the common way to implement k-means is to run it multiple times with different parameter configurations and select the best result.

In this paper, for each dataset we run k-means with the k values ranging from 2 to 30, and for each k value, we run it 30 times with the initial centroids randomly selected from the dataset[5]. As each run of k-means usually takes multiple iterations to process, we keep records of all the clustering labels and the cluster centroids of each iteration, and consider the clustering labels of each iteration in each k-means run as a clustering result. Then we calculate the internal measures of all these clustering results. In this way, these measures are based on not only good clustering results after the convergence of the Cost Function, but also the clustering results during the k-means process that are not very good. Based on these different clustering results, the evaluation of our three measures are more comprehensive and reasonable.

In the experimental process, there are some detailed features that are worth mentioning. Firstly, all the data is normalised to z-score [19] before input into k-means clustering to keep the balance among features with different magnitudes. Secondly, we use a different method to deal with empty clusters in the k-means process. The common way to deal with empty clusters is to re-run the k-means by re-selecting the initial centroids. In order to generate different clustering results for the evaluation, the way we adopt in this work is to find the closest data point to the centroid of each empty cluster, and assign the closest data point to the corresponding empty cluster to form a non-empty cluster.

Based on the total inventory of results accumulated, we then make the following four evaluations:

1 An evaluation of the measures in each iteration of each run of k-means;
2 An evaluation of the measures in each run of k-means;
3 An evaluation of the measures in the selection of the best result across all the 30 clustering results with each fixed value of k;
4 An evaluation of the measures in the selection of the overall best result from the best results selected for all the 29 k values.

5 Experimental Results

In this section we detail the results of the four evaluations outlined in the last section, and analyse the performances of the three measures - the k-means Cost Function (CF), Silhouette (Sil) and Simplified Silhouette (SS).

5.1 Evaluation in Each Iteration

Firstly, we look at the performances of the three internal measures in each iteration. As discussed in Sect. 2, the Cost Function of k-means is defined as

[5] In preparing our experiments we tested two different initialisation methods for k-means, a random initialisation and a well-known algorithm k-means++. However, we found that the initialisation method made no difference in our results so in this paper we just report the results using the random initialisation.

the default measure for k-means and it decreases monotonically during the k-means process. Therefore, the Cost Function value in the next iteration cannot be larger than that in the previous iteration. However, both Silhouette and Simplified Silhouette are designed for general clustering, so they may not represent the validity of k-means exactly. Table 2 shows the number of iterations in which the k-means Cost Function increases of all the iterations for each dataset, and that in which Silhouette or Simplified Silhouette decreases. Note that a smaller value of the k-means Cost Function indicates a better clustering result, while a bigger value of Silhouette or Simplified Silhouette indicates a better clustering result. The percentages with the parentheses around indicate the proportions of these kinds of iterations in corresponding total iteration numbers.

Table 2: Evaluation of Iterations

Dataset	**Total #Iterations**	**#Iterations with Increasing CF**	**#Iterations with Decreasing Sil**	**#Iterations with Decreasing SS**
Iris	5685	0	303 (5.33%)	473 (8.32%)
Glass	7548	0	776 (10.28%)	709 (9.39%)
Wine	5588	0	248 (4.44%)	278 (4.97%)
Yeast	23923	0	5211 (21.78%)	3888 (16.25%)
Dim032	3570	0	119 (3.33%)	93 (2.61%)
Dim064	3342	0	70 (2.09%)	21 (0.63%)
S1	17256	0	5832 (33.80%)	6457 (37.42%)
S3	27329	0	7585 (27.75%)	8234 (30.13%)

From Table 2, we can see the Cost Function decreases monotonically as expected, but neither Silhouette nor Simplified Silhouette increases monotonically (although both of them increase in most cases). Based on the definition, the clustering result always gets better along iterations of each run of k-means. Therefore the evaluations of k-means clustering validity with both Silhouette and Simplified Silhouette are inaccurate in some iterations, so we can see neither Silhouette nor Simplified Silhouette performs as well as the Cost Function.

Meanwhile, also from Table 2 we see that there is not much difference between the numbers of iterations with decreasing Silhouette and Simplified Silhouette values, which indicates these two measures perform similarly in the evaluation in iterations.

5.2 Evaluation in Each Run of k-means

As stated in Sect. 5.1, Silhouette and Simplified Silhouette may be inaccurate in the evaluation of clustering validity in individual iterations. Therefore, for Silhouette and Simplified Silhouette the k-means process may be not a monotonically converging process, and in the last iteration of k-means where the minimum of the Cost Function is always found, the Silhouette or Simplified Silhouette value

may be not the best value in the k-means process. We get the results just as we expect: for all datasets, there are always clustering with the last Silhouette or Simplified Silhouette smaller than the best value (due to the limitation of space, the details are not included in the paper). Similarly, we can see that neither Silhouette nor Simplified Silhouette performs as well as the Cost Function.

Even though it may not result in the best Silhouette or Simplified Silhouette value, the last iteration is always taken as the end of a k-means clustering based on its definition. Therefore, the result in the last iteration is always taken as the final clustering result of the k-means clustering in further steps of the experiment.

5.3 Evaluation in the Selection of the Best Result from Clustering with Each Fixed Value of k

For each fixed value of k, we compare 30 k-means clustering results and select the best one among them based on our three internal measures. Table 3 shows the number of k values with which the same best result is selected from clustering, based on every pair of two measures or all three measures.

Table 3: Evaluation of the Selection of the Best Result from Clustering with Each Fixed Value of k

Dataset	**Total #k Values**	**#k Values - Sil and SS**	**#k Values - Sil and CF**	**#k Values - SS and CF**	**#k Values - All Measures**
Iris	29	23	10	9	8
Glass	29	15	9	5	3
Wine	29	18	13	11	9
Yeast	29	12	6	6	4
Dim032	29	16	13	8	8
Dim064	29	19	17	14	12
S1	29	23	7	7	6
S3	29	22	4	3	3

Silhouette and Simplified Silhouette can select the same best result for most k values, but only for a small number of k values, they can select the same best result as the Cost Function. Similarly, we can see that neither of them can perform as well as the Cost Function. Although Silhouette performs a little better than Simplified Silhouette in this case, there is not much difference.

Based on these results as well as the results in above sections, we can conclude that the k-means Cost Function is the only one among these three internal measures that can accurately evaluate k-means clustering validity. Therefore, the best clustering result for each k value is selected based on the k-means Cost Function in further steps of the experiment.

5.4 Evaluation in the Selection of the Overall Best Result

The selection of the overall best result from all the best results selected for each k value is the last step of the experiment. Table 4 shows the k values corresponding to the overall best results selected based on each internal measure for each dataset. It is shown that the k-means Cost Function is problematic in the evaluation of the validity of k-means clustering with different k values, and tends to select the result with the largest k value as the overall best result, therefore as we discussed, it is not suitable for this case.

Table 4: Evaluation of the Selection of the Overall Best Result from the Best Clustering Results Selected for Different k Values Based on Different Measures

Dataset	**Desired k Value**	**Corresponding k for CF**	**Corresponding k for Sil**	**Corresponding k for SS**
Iris	Unknown	29	2	2
Glass	Unknown	30	2	2
Wine	Unknown	29	3	3
Yeast	Unknown	30	7	8
Dim032	16	29	16	16
Dim064	16	30	20	20
S1	15	30	15	15
S3	15	30	15	15

On the other hand, Silhouette and Simplified Silhouette select the same overall best result for almost all datasets. For three of the four synthetic datasets that are designed for clustering, both measures can select the results with the desired k values. For the dataset Dim064, they also select the result with the same value. It is common to select results with the non-desired k value based on conditions like this because the initial centroids are randomly selected to generate a variety of clustering results[6]. From this perspective, we can conclude that Simplified Silhouette has competitive performance to Silhouette in the selection of the overall best result.

5.5 Evaluation of Correlations between Internal Measures

We also evaluate the Pearson correlations between Silhouette and Simplified Silhouette. For each dataset and each k value, the distinct pairs of these two measures are extracted from the results. From Fig. 1 and Fig. 2, it is shown that there is highly positive correlation between Silhouette and Simplified Silhouette in an overwhelming majority of situations.

[6] If other methods like k-means++ are used for selecting the initial centroids, it is very likely to get all the desired k values for all the synthetic datasets.

Correlations between Sil and SS

K	Iris	Glass	Wine	Yeast	Dim032	Dim064	S1	S3
2	0.993294	0.992193	0.99702	0.943325	-0.12825	0.300981	0.941844	0.971279
3	0.995385	0.923887	0.996318	0.894682	0.778223	0.425153	0.942118	0.955456
4	0.967166	0.544803	0.961703	0.81958	0.738165	0.439857	0.912224	0.991052
5	0.980247	0.948815	0.951209	0.727211	0.613938	0.948636	0.933215	0.978902
6	0.91578	0.893141	0.973678	0.690819	0.849891	0.858353	0.962757	0.97548
7	0.928572	0.957998	0.959664	0.757024	0.85729	0.967433	0.975869	0.97462
8	0.96134	0.943394	0.90954	0.849662	0.91182	0.954233	0.980063	0.960327
9	0.85654	0.903842	0.70277	0.848189	0.910435	0.966105	0.981642	0.98846
10	0.968438	0.913511	0.924346	0.903346	0.916598	0.958255	0.980504	0.985758
11	0.922668	0.870661	0.928736	0.949788	0.968223	0.980685	0.979796	0.98705
12	0.981977	0.943115	0.875796	0.910057	0.959173	0.960018	0.973384	0.991971
13	0.901127	0.879644	0.92124	0.920059	0.980426	0.965945	0.979321	0.98883
14	0.910819	0.865278	0.831538	0.944559	0.966271	0.961901	0.975649	0.988905
15	0.953423	0.89128	0.90954	0.937084	0.971514	0.980536	0.98125	0.986448
16	0.944367	0.862471	0.876525	0.938838	0.958231	0.977597	0.985526	0.989248
17	0.982379	0.918348	0.935388	0.927498	0.968129	0.963791	0.982856	0.99135
18	0.935021	0.861701	0.905783	0.931496	0.962147	0.950856	0.982589	0.991791
19	0.907564	0.844418	0.91705	0.933302	0.920502	0.922081	0.984666	0.988965
20	0.986338	0.847514	0.818287	0.92105	0.977352	0.961491	0.980543	0.990079
21	0.894858	0.926713	0.832134	0.929169	0.946484	0.916429	0.984255	0.989441
22	0.919939	0.92103	0.899059	0.924478	0.900888	0.933575	0.982873	0.979498
23	0.953855	0.868697	0.841784	0.942225	0.956174	0.904049	0.986931	0.98348
24	0.941008	0.936139	0.85987	0.934342	0.920883	0.914759	0.983635	0.980152
25	0.958949	0.881833	0.88931	0.940603	0.925157	0.887353	0.983654	0.982337
26	0.858996	0.869662	0.847474	0.960428	0.934942	0.761293	0.982828	0.98087
27	0.95194	0.914826	0.87712	0.950632	0.920293	0.782278	0.985089	0.98381
28	0.885049	0.917608	0.874958	0.931141	0.932947	0.881241	0.985711	0.976128
29	0.902892	0.918648	0.901555	0.948535	0.898227	0.924445	0.989047	0.983812
30	0.829004	0.902836	0.867191	0.936504	0.906922	0.800268	0.991359	0.977056

-1 0 1

Fig. 1: Correlations between Sil and SS

P-value for Correlation Testing between Sil and SS

K	Iris	Glass	Wine	Yeast	Dim032	Dim064	S1	S3
2	8.78E-09	2.22E-16	3.44E-10	0	0.541225	0.184905	0	0
3	1.97E-09	3.35E-13	0	0	0.000143	0.027055	-4.44E-16	0
4	0	0.001045	7.76E-10	4.44E-16	0.000308	0.016956	0	0
5	4.44E-16	4.44E-16	0	0	0.00085	0	0	0
6	8.95E-10	0	0	8.88E-16	7.61E-08	3.36E-10	0	0
7	1.91E-14	0	0	0	4.86E-11	0	0	0
8	5.94E-11	0	2.22E-16	0	0	0	0	0
9	2.04E-10	2.22E-16	2.22E-09	0	0	0	0	4.44E-16
10	4.44E-16	0	0	0	0	0	0	0
11	1.57E-13	0	2.22E-16	0	0	0	0	0
12	0	0	8.88E-16	0	0	0	0	0
13	6.82E-14	0	0	0	0	-4.44E-16	0	0
14	1.05E-10	0	0	0	0	0	-4.44E-16	0
15	0	4.44E-16	2.22E-16	0	0	0	0	0
16	4.88E-15	2.22E-16	0	2.22E-16	0	0	0	0
17	0	0	4.44E-16	0	0	0	0	0
18	0	0	0	4.44E-16	4.44E-16	0	0	0
19	8.88E-16	2.22E-16	0	0	0	0	0	0
20	2.22E-16	0	4.44E-16	0	0	-4.44E-16	0	0
21	1.01E-07	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	2.22E-16
23	1.33E-15	0	0	0	2.22E-16	0	0	0
24	-4.44E-16	0	0	0	0	0	2.22E-16	0
25	0	0	6.66E-16	0	0	0	0	0
26	1.59E-10	0	0	0	0	0	0	0
27	0	0	2.22E-16	0	2.22E-16	0	0	0
28	3.58E-13	0	0	0	0	0	0	4.44E-16
29	1.18E-10	0	0	0	0	0	0	0
30	4.24E-10	0	0	0	0	2.22E-16	2.22E-16	4.44E-16

0 0.01 1

Fig. 2: P-value for Correlation Testing between Sil and SS

5.6 Evaluation of Time Consumed

Finally, we compare the time consumed in the calculation of each internal measure. Figure 3 shows the time consumed (with *ms* as unit) for the datasets S1, which is the dataset with the most instances. It is shown that the time consumed by Silhouette is much more than that by the Cost Function or Simplified Silhouette, and the differences can be orders of magnitude in size. Similar results are found for other datasets. The rough time consumed in calculation may not reflect the genuine efficiency of algorithms exactly, but from the commercial perspective, it is meaningful to notice that the implementation of the Cost Function and Simplified Silhouette is generally much faster than Silhouette.

6 Conclusion

In this paper we have analysed the application of Simplified Silhouette to the evaluation of k-means clustering validity, and compared it with two other internal measures: the k-means Cost Function and the original Silhouette from both theoretical and empirical perspectives.

Theoretically, Simplified Silhouette has a mathematical relationship with both the k-means Cost Function and the original Silhouette. It brings in additionally the distance of each data point to its closest neighbouring cluster to the k-means Cost Function, but simplifies Silhouette by ignoring within-cluster pairwise distances.

Empirically, we can make the following conclusions:

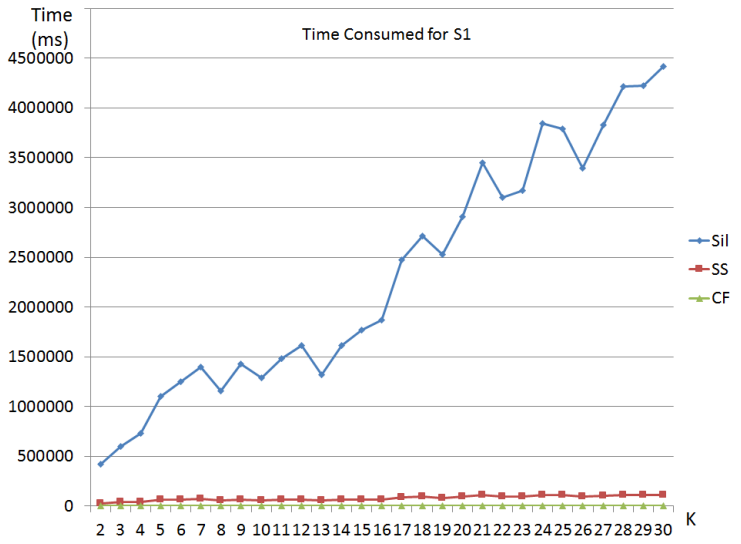

Fig. 3: Time Consumed - S1

1 Neither Simplified Silhouette nor the original Silhouette can perform as well as the k-means Cost Function in the evaluation of the validity of k-means clustering with the same k value but different initial centroids;
2 Simplified Silhouette has competitive performances to the original Silhouette in the evaluation of k-means validity and is much faster in the calculation;

Therefore, the most suitable method to automate the selection of the best k-means result is using the k-means Cost Function firstly to select the best result for each k value and then using Simplified Silhouette to select the overall best result from the best results for different k values.

Due to the limitation of time and resources, Simplified Silhouette has not been fully explored in this paper, e.g. the actual industrial datasets are not available. On the other hand, this is an attempt to evaluate the internal measures for a specific clustering algorithm. Specific methods should be evaluated, selected and even designed for specific algorithms or conditions, rather than always a same set of general methods for all the situations.

Acknowledgement. The authors wish to acknowledge the support of Enterprise Ireland through the Innovation Partnership Programme SmartSeg 2. The authors also wish to acknowledge the support of the ADAPT Research Centre. The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Funds.

References

1. Bishop, C.M.: Pattern recognition. Machine Learning (2006)
2. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE transactions on pattern analysis and machine intelligence (2), 224–227 (1979)
3. Dunn, J.C.: Well-separated clusters and optimal fuzzy partitions. Journal of cybernetics 4(1), 95–104 (1974)
4. Fränti, P., Virmajoki, O.: Iterative shrinking method for clustering problems. Pattern Recognition 39(5), 761–775 (2006)
5. Franti, P., Virmajoki, O., Hautamaki, V.: Fast agglomerative clustering using a k-nearest neighbor graph. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(11), 1875–1881 (2006)
6. Halkidi, M., Vazirgiannis, M., Batistakis, Y.: Quality scheme assessment in the clustering process. In: European Conference on Principles of Data Mining and Knowledge Discovery. pp. 265–276. Springer (2000)
7. Hruschka, E.R., de Castro, L.N., Campello, R.J.: Evolutionary algorithms for clustering gene-expression data. In: Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on. pp. 403–406. IEEE (2004)
8. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern recognition letters 31(8), 651–666 (2010)
9. Kelleher, J.D., Mac Namee, B., D'Arcy, A.: Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies. MIT Press (2015)
10. Moulavi, D., Jaskowiak, P.A., Campello, R.J., Zimek, A., Sander, J., et al.: Density-based clustering validation. In: SDM. pp. 839–847. SIAM (2014)
11. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics 20, 53–65 (1987)
12. Steinley, D.: K-means clustering: a half-century synthesis. British Journal of Mathematical and Statistical Psychology 59(1), 1–34 (2006)
13. Theodoridis, S., Koutroumbas, K., et al.: Pattern recognition. (1999)
14. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63(2), 411–423 (2001)
15. Tomasini, C., Emmendorfer, L., Borges, E.N., Machado, K.: A methodology for selecting the most suitable cluster validation internal indices. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing. pp. 901–903. ACM (2016)
16. Vendramin, L., Campello, R.J., Hruschka, E.R.: Relative clustering validity criteria: A comparative overview. Statistical Analysis and Data Mining 3(4), 209–235 (2010)
17. Wang, F., Franco, H., Pugh, J., Ross, R.: Empirical comparative analysis of 1-of-k coding and k-prototypes in categorical clustering (2016)
18. Xiong, H., Li, Z.: Clustering validation measures. (2013)
19. Zaki, M.J., Meira Jr, W.: Data mining and analysis: fundamental concepts and algorithms. Cambridge University Press (2014)