A PROPOSED ALGORITHM FOR DETERMINING THE OPTIMAL NUMBER OF CLUSTERS

Markela Muca
Department of Applied Mathematics, Faculty of Natural Sciences,
University of Tirana, Albania
Gleda Kutrolli
Quant Department, ikubINFO Software Solutions ltd, Albania
Maksi Kutrolli
Department of Computer Sciences, Faculty of Information Technology,
"Aleksander Moisiu" University, Albania

Abstract

Data clustering is a data exploration technique that allows objects with similar characteristics to be grouped together in order to facilitate their further processing. The K-means algorithm is a popular data-clustering algorithm. However, one of its drawbacks is the requirement for the number of clusters, K, to be specified before the algorithm is applied. This paper first reviews existing methods for selecting the number of clusters for the algorithm. Factors that affect this selection are then discussed and an improvement of the existing k-means algorithm to assist the selection is proposed. The paper concludes with an analysis of the results of using cluster validation referring to some measures that are classified as internal and external indexes to determine the optimal number of clusters for the K-means algorithm. There are applied some stopping criterion referring to those indexes for evaluating a clustering against a gold standart.

Keywords: Clustering, K-means algorithm, optimal number of clusters, cluster validation

Introduction

Clustering is a standard procedure in multivariate data analysis. It is designed to explore an inherent natural structure of the data objects, where objects in the same cluster are as similar as possible and objects in different clusters are as dissimilar as possible. The equivalence classes induced by the clusters provide a means for generalizing over the data objects and their features. Clustering methods are applied in many domains, such as medical research, psychology, economics and pattern recognition. One of the most popular and efficient clustering methods is the K-means method (Hartigan & Wang, 1979; Lloyd, 1957; MacQueen, 1967) which uses prototypes (centroids) to represent clusters by optimizing the squared error function. The k-means problem is to partition data into k groups such that the sum of squared Euclidean distances to each group mean is minimized. However, the problem is NP-hard in a general Euclidean space, even when the number of clusters k is 2 (Aloise et al., 2009; Dasgupta and Freund, 2009), or when the dimensionality is 2 (Mahajan et al., 2009). The standard iterative k-means algorithm (Lloyd, 1982) is a widely used heuristic solution. The algorithm iteratively calculates the within-cluster sum of squared distances, modifies group membership of each point to reduce the withincluster sum of squared distances, and computes new cluster centers until local convergence is achieved. The time complexity of this standard k-means algorithm is O(*qknp*), where *q* is the number of iterations, *k* is the number of clusters, *n* is the sample size, and *p* is the dimensionality (Manning et al., 2008). The result of heuristic k-means clustering, heavily dependent on the initial cluster centers, isn't always optimal. Our algorithm restarts the procedure a number of times to mitigate the problem. The number of restarts for k-means to approach an optimal solution can be prohibitively high and the procedure stops when some specified criterion are reached. So in this way we apply an clustering validation refering to some measures or indexes that are clasified as internal and external validation. This paper proposes a method based on information obtained during the K-means clustering operation itself to select the number of clusters, K. The method employs an objective evaluation measure to suggest suitable values for K, thus avoiding the need for trial and error. The remainder of the paper consists of three sections. Section 1 reviews the existing k-means algorithm and its details. Section 2 analyses the factors influencing the selection of K and describes the proposed evaluation measure. Section 3 presents the results of applying the proposed algorithm to select the optimal K for different data sets (mainly quantitative data sets).

K-means Clustering

Clustering is the process of partitioning a group of data points into a small number of clusters. For instance, the items in a supermarket are clustered in categories (butter, cheese and milk are grouped in dairy products). Of course, this is a qualitative kind of partitioning. A quantitative approach would be to measure certain features of the products, say percentage of milk and others, and products with high percentage of milk would be grouped together. In general, we have *n* data points x_i, *i = 1...n* that have to be partitioned in *k* clusters. The goal is to assign a cluster to each data point. K-means is a clustering method that aims to find the positions μ_i, *i = 1*...k of the clusters that minimize the *distance* from the data points to the cluster. K-means clustering solves:

$$\arg min \sum_{i=1}^{k} \sum_{x \in c_i} d(x, \mu_i) = \arg min \sum_{i=1}^{k} \sum_{x \in c_i} \left|\left|x - \mu_i\right|\right|_2^2$$

where c_i is the set of points that belong to cluster *i*. The K-means clustering uses the square of the Euclidean distance $d(x, \mu_i) = \left|\left|x - \mu_i\right|\right|_2^2$. This problem is not trivial (in fact it is NP-hard), so the K-means algorithm only hopes to find the global minimum, possibly getting stuck in a different solution. In our case we are using Lloyd's algorithm which converges to a point, although it is not necessarily the minimum of the sum of squares. That is because the problem is non-convex and the algorithm is just a heuristic, converging to a local minimum. The algorithm stops when the assignments do not change from one iteration to the next.

Deciding the number of clusters

The number of clusters should match the data. An incorrect choice of the number of clusters will invalidate the whole process. In this paper we use cluster validation to locate the optimal number of clusters. Cluster validation consist on internal and external validation indexes that are used as criteria on printing the optimal number of clusters. At first, we start to consider the silhouette width and if this measure has a value under *0.51* then we conclude that our clustering structure referring to this measure is unstable. This is the case where we take in account the other internal indexes which are SSE and Dunn Index. If there still have problems than we reinitialize centroids and the procedure start from the beginning. In conclusion, we will display the optimal number of clusters and some detailed information including external validation about the resulting number.

Initializing the position of the clusters: Since the algorithm stops in a local minimum, the initial position of the clusters is very important. We start by an suggested initialization and if this ends up in unsatisfactory results then we continue with other random initialization of seed.

Internal and External measures

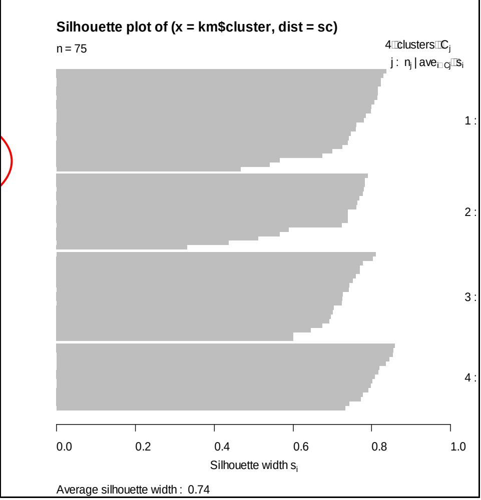
In this section, we illustrate the relationship between K-means clustering and validation measures. Generally speaking, there are two types of clustering validation techniques [1], [2], [6], [7] which are based on external criteria and internal criteria, respectively. The focus of this paper is on the evaluation of external clustering validation measures including Entropy, Purity [8] and internal clustering validation measures including Silhouette Width, Dunn Index, Sum of Squared Error [9], [10] and a combination of both of those indexes for K-means clustering to build and implement an algorithm that generates the optimal number of clusters.

Internal Measures

When a clustering result is evaluated based on the data that was clustered itself, this is called internal evaluation. These methods usually assign the best score to the algorithm that produces clusters with high similarity within a cluster and low similarity between clusters. One drawback of using internal criteria in cluster evaluation is that high scores on an internal measure do not necessarily result in effective information retrieval applications. Additionally, this evaluation is biased towards algorithms that use the same cluster model. For example, k-Means clustering naturally optimizes object distances, and a distance-based internal criterion will likely overrate the resulting clustering. Therefore, the internal evaluation measures are best suited to get some insight into situations where one algorithm performs better than another, but this shall not imply that one algorithm produces more valid results than another. The following methods can be used to assess the quality of clustering algorithm based on internal criterion:

The Silhouette index value detect if we have to do with an appropriate clustering or not is categorized as it is shown below:

Table 2.1.1. Categorization of Silhouette Width values.

Range of Silhouette Index	**Interpretation**	
0.71 - 1.0	A strong structure has been found.	
0.51 - 0.7	A reasonable structure has been found.	
0.26 - 0.5	The structure is weak and could be artificial. Try additional methods of data analysis.	
≤ 0.25	No substantial structure has been found.	

As you see, we find out a direct result in case of a stable structure given from silhouette width (which is circled in red), in other cases we examine the behavior of other indexes. In our case, silhouette width is *0.74.* For different number of clusters we look for that number that has maximum value of silhouette width. As you see in the silhouette plot the maximum value of silhouette width is reached for the corresponding number of clusters *4.*

Dunn's validation index is conceptually the simplest of the internal validation indices: it compares the size of the groups with the distances between groups. The further apart the groups, relative to their size, the larger the index and the "better" the clustering. This index, is computed as the ratio between the minimum distance between two clusters and the size of the largest cluster. So, we are looking for the maximum value of this index.

Sum of Squared Error (SSE) is the simplest and most widely used criterion measure for clustering. The SSE criterion function is suitable for cases in which the clusters form compact clouds that are well separated from one another (Duda et al., 2001). In this case, we are looking for the knee which is identified by the maximum absolute second derivative (MASD). MASD can be approximated with a central difference:

$$sd(i) = x(i+1) + x(i-1) - 2 * x(i).$$

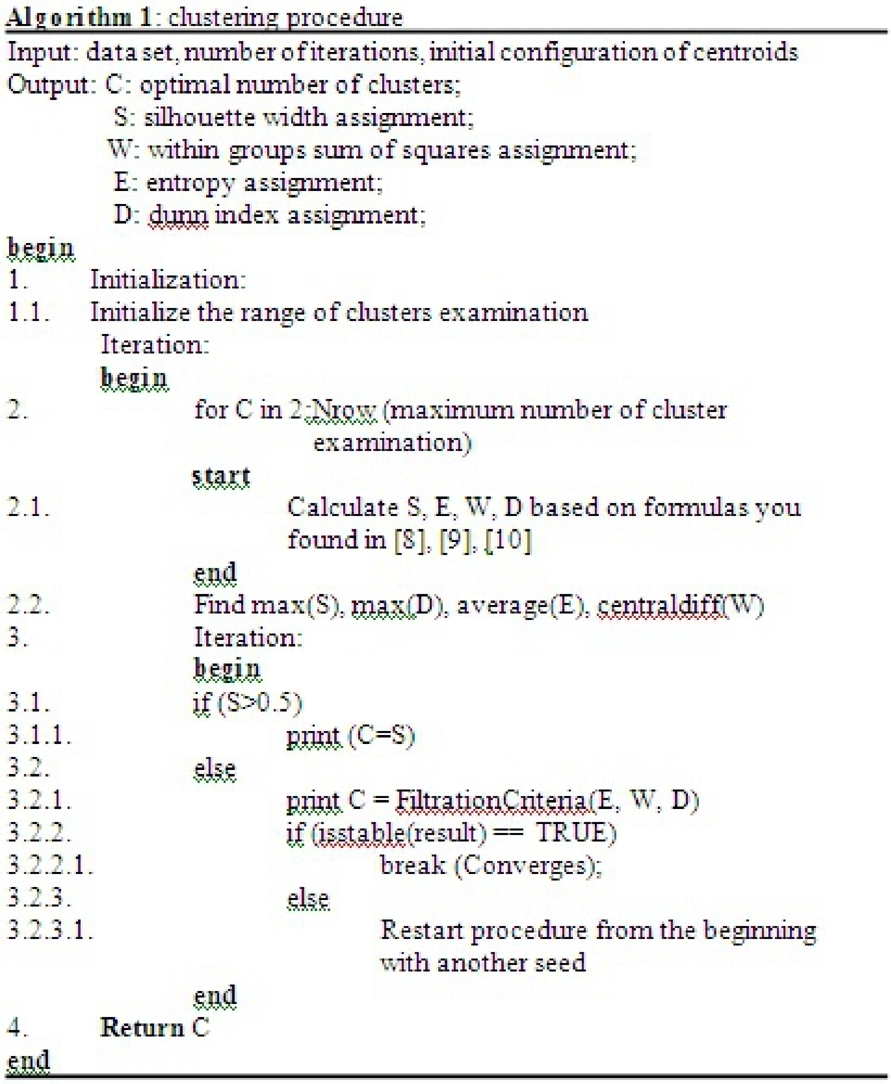
External Validation

External validation indices are used to measure the extent to which cluster labels affirm with the externally given class labels. The external validation measures are extremely useful in deducing the ambit to which the clustering structure is ascertained by a clustering algorithm that matches some external structure. This is compared to the individual designated class labels. External validation measures criteria evaluate the final clustering output result with respect to a pre designated structure. There are many external validation measures [1] but we focus on two external validation measures Purity and Entropy:

Entropy measures the purity of the clusters with respect to the given class labels. Thus, if all clusters consist of objects with only a single class label, the entropy is 0. However, as the class labels of objects in a cluster become more varied, the entropy increases. The entropy is negative measure, the lower the entropy the better clustering it is. The greater entropy means that the clustering is not good. So, we expect that every cluster should have low entropy to maintain the quality of our clustering. *Purity* is one of very primary validation measure to determine the cluster quality. The greater the value of purity indicates good clustering. So we expect that every cluster should have high purity to maintain the quality of our clustering.

Results

The clustering algorithm and validity indices were evaluated synthetically with generated data set. These data were generated by our data set generator (they can be random data sets or user's data sets). We begin with a simulated example in which there truly *4* clusters in the data and that exactly the optimal number of clusters that is obtained from our algorithm. The clustering procedure is shown in Algorithm 1.



```
Algorithm 1: clustering procedure
Input: data set, number of iterations, initial configuration of centroids
Output: C: optimal number of clusters;
          S: silhouette width assignment;
         W: within groups sum of squares assignment;
          E: entropy assignment;
          D: dunn index assignment;
begin
1.        Initialization:
1.1.      Initialize the range of clusters examination
           Iteration:
           begin
2.                   for C in 2:Nrow (maximum number of cluster
                            examination)
                     start
2.1.                         Calculate S, E, W, D based on formulas you
                             found in [8], [9], [10]
                     end
2.2.                 Find max(S), max(D), average(E), centraldiff(W)
3.                   Iteration:
                     begin
3.1.                 if (S>0.5)
3.1.1.                       print (C=S)
3.2.                 else
3.2.1.                       print C = FiltrationCriteria(E, W, D)
3.2.2.                       if (isstable(result) == TRUE)
3.2.2.1.                             break (Converges);
3.2.3.                       else
3.2.3.1.                             Restart procedure from the beginning
                                     with another seed
                     end
4.         Return C
end
```

Figure 1 shows the optimal number of clusters for our corresponding data set that is *4*. In this case we see that all indexes have converged in the same result. Also, you can see there a plot corresponding to the entropy. As you see, entropy is increased with increasing of number of clusters. Ranges of rapidly increasing entropy are inappropriate for the selection of stopping points because minor changes in the classifications cause major changes in the entropies, indicating that the information in classifications in those ranges is not well organized.

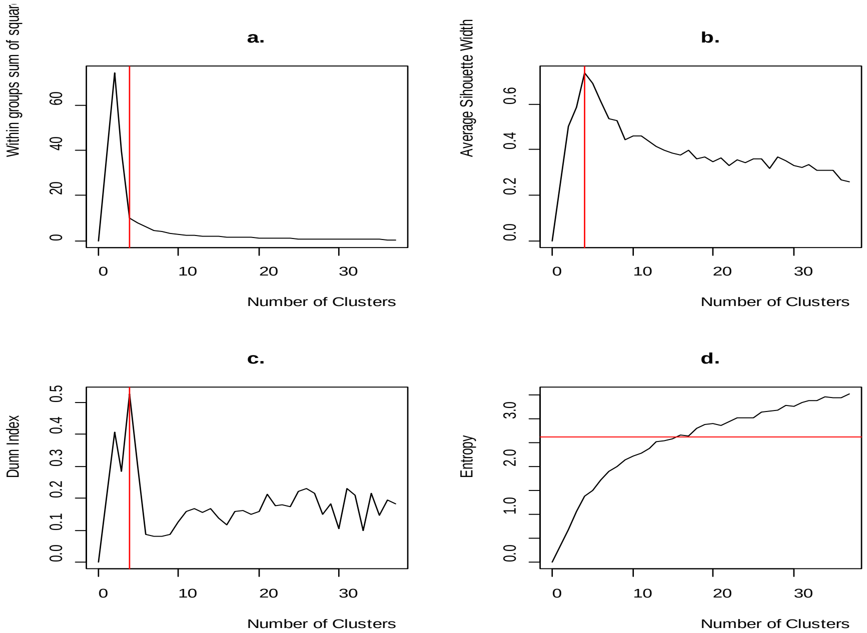


Figure 1. Plot the within groups sums of squares (a), Average Silhouette Width (b), Dunn Index (c) and Entropy (d) vs. the number of clusters extracted. The optimal number of clusters is 4.

In the figure 2 you will see a plot that inspects the centroids. This is an output that we generate after we have located the optimal number of clusters. As you see, all our row data are distributed in accordance with the optimal number of clusters. It is clear that our clustering is pure and stable but also we must emphasize that we need to continue our examination in each cluster separately to determine if we have obtained the proper number of clusters for each data set. For example, in cluster 2 it looks that some points are dispersed more than the others. Figure 3 visualize cluster quality and also to reinforce the results we have obtained from our advanced function.

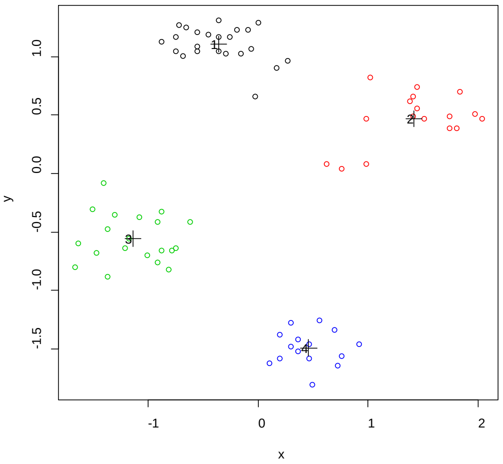


Figure 3. Dissimilarity plot with number of clusters = 4.

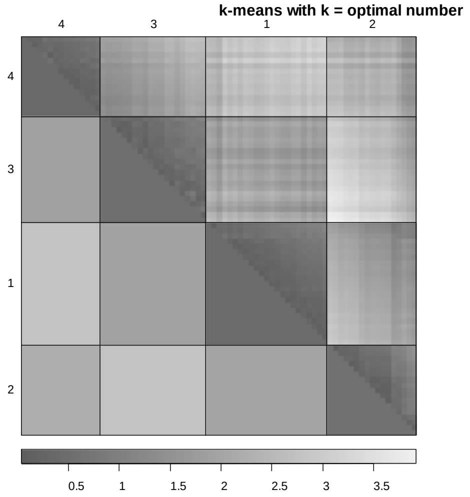


Figure 2. Distribution of our data set and the initialization of centroids used in case of the optimal number of clusters.

Objects belonging to the same cluster are displayed in consecutive order. The placement of clusters and the within cluster order is obtained by a seriation algorithm which tries to place large similarities/small dissimilarities close to the diagonal. Compact clusters are visible as dark squares (low dissimilarity) on the diagonal of the plot. As it is shown in the plot our clustering is reasonable and compact. To sum up, since K-means cluster analysis starts with *k* randomly chosen centroids, a different solution can be obtained each time the function is invoked. Our new advanced k-means function has an nstart option that attempts multiple initial configurations and reports on the best one. For example, adding nstart = 25 will generate 25 initial configurations. We *strongly* recommend always running *K*-means clustering with a large value of nstart, such as 20 or 50, since otherwise an undesirable local optimum may be obtained. Also, our clustering approach can be sensitive to the initial selection of centroids. When performing *K*-means clustering, in addition to using multiple initial cluster assignments, it is also important to set a random seed. This way, the initial cluster assignments in Step 1 can be replicated, and the *K*-means output will be fully reproducible.

Conclusion

Existing methods/algorithms of selecting the number of clusters for K-means clustering have a number of drawbacks. Also, the choice of a clustering algorithm and a validation index is not a trivial one. A new method to select the number of clusters for the K-means algorithm has been proposed in the paper. The new method is closely related to the approach of K-means clustering because it takes into account information reflecting the performance of the algorithm and also are applied several criteria referring to internal and external validation. The proposed method can suggest multiple values of clusters to users for cases when different clustering results could be obtained with various required levels of detail. The method could be computationally expensive if used with large data sets because it requires several applications of the K-means algorithm before it can suggest the optimal number of clusters. Further research is required to verify the capability of this method and other improvements are necessary to be applied. Clustering algorithms should be improved based on (a) the nature of the problem to solve, (b) characteristics of the objects to be analyzed, and (c) the size of the problem and computational power available. Thus, we need to add other criteria and filtrations concerning the type of dataset we will consider.

References:

B. Desgraupes, Clustering Indices, April 2013

Daniel Barbar´a, Yi Li, and Julia Couto. Coolcat: An entropy-based algorithm for categorical clustering. In *Proceedings of the 11th ACM International Conference on Information and Knowledge Management*, pages 582–589, 2002.

Rousseeuw, 1987. Rousseeuw, P.J., (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20, 53-65.

Duda, P. E. Hart and D. G. Stork, Pattern Classification, Wiley, New York, 2001.

Hui Xiong, Senior Member, IEEE, Junjie Wu, and Jian Chen, Fellow, IEEE, K-means Clustering Versus Validation Measures: A Data Distribution Perspective.

Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. Cluster validity methods: Part i. *SIGMOD Record*, 31(2):40–45, 2002.

A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1998.

Y. Zhao and G. Karypis. Criterion functions for document clustering: Experiments and analysis. *Machine Learning*, 55(3):311–331, June 2004.

F. Kovács C. Legány A. Babos. Cluster Validity Measurement Techniques.

Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. In *Workshop on Text Mining, the 6th ACM*