I.J. Information Technology and Computer Science, **2023, 5, 1-11**
Published Online on October 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2023.05.01

Sample of Groups: A New Strategy to Find a Representative Point for Each Undisclosed Cluster

Wallace A. Pinheiro*
Center for Systems Development, Brazilian Army, Brazil
E-mail: wallaceapinheiro@gmail.com
ORCID iD: https://orcid.org/0000-0001-7076-8785
*Corresponding Author

Ana B. S. Pinheiro
University of Brasilia, Department of Tropical Medicine, Brazil
E-mail: absapienza@gmail.com
ORCID iD: https://orcid.org/0000-0002-2766-3141

Received: 14 May 2023; Revised: 23 July 2023; Accepted: 28 August 2023; Published: 08 October 2023

Abstract: Some problems involving the selection of samples from undisclosed groups are relevant in various areas such as health, statistics, economics, and computer science. For instance, when selecting a sample from a population, well-known strategies include simple random and stratified random selection. Another related problem is selecting the initial points corresponding to samples for the K-means clustering algorithm. In this regard, many studies propose different strategies for choosing these samples. However, there is no consensus on the best or most effective approaches, even when considering specific datasets or domains. In this work, we present a new strategy called the Sample of Groups (SOG) Algorithm, which combines concepts from grid, density, and maximum distance clustering algorithms to identify representative points or samples located near the center of the cluster mass. To achieve this, we create boxes with the right size to partition the data and select the representatives of the most relevant boxes. Thus, the main goal of this work is to find quality samples or seeds of data that represent different clusters. To compare our approach with other algorithms, we not only utilize indirect measures related to K-means but also employ two direct measures that facilitate a fairer comparison among these strategies. The results indicate that our proposal outperforms the most commonly used algorithms.

Index Terms: Clustering, Grouping, Similarity, Sampling, Grid.

1. Introduction

When dealing with a dataset composed of undisclosed groups with unknown features, accurately separating these groups and identifying a unique representative element for each group is a complex task. In the field of health research, sampling individuals from specific groups is a critical undertaking. Sometimes, although several features of these groups are available, determining which features should be used to accurately distinguish them remains a challenge. One solution to automate this process is to select the relevant features using strategies such as Principal Component Analysis (PCA) [1, 2]. Subsequently, clustering algorithms can utilize these features to generate clusters and identify the samples within each group.

When considering different data sources and distributions, the K-means algorithm, one of the most commonly used clustering algorithms, typically achieves an average allocation accuracy of around 50% to 60% for elements within their respective clusters. If we employ a naive strategy that randomly selects one element for each cluster, then the sample selection will be limited by the outcomes of K-means.

Conversely, some approaches attempt to find samples for each cluster in the dataset without performing clustering as a preliminary step. These methods, known as initialization methods in the field of clustering, can enhance the results of certain clustering algorithms by providing seeds that guide the initial steps of the clustering process.

In computer science, problems of this nature are typically addressed through unsupervised classification or clustering approaches. K-means is one of the most widely used algorithms for grouping data and requires the number of clusters as an input parameter. K-means employs seeds as initial points, which act as references for locating each cluster. Subsequently, it computes the centroids of the clusters. K-means iteratively refines its centroids until one or more of these centroids remain unchanged. It's important to note that this strategy may not perform well for non-Voronoi distributions [3, 4]. While several methods have been proposed to find seeds, a consensus on the best method has not been reached [5, 6].

Additionally, it is relevant to consider the time required to obtain the samples and to ensure that they truly belong to different groups (avoiding the repetition of elements related to the same cluster). In K-means, this effect can impact the number of iterations needed to find the final centroids and, consequently, reduce the execution time of the clustering process. These characteristics allow for the calculation of direct measures related to seed quality. Given these considerations, there is a need for direct measures in this context. Therefore, we propose two direct measures to compare different strategies for finding representative elements of clusters: the number of centroids in distinct clusters divided by the number of clusters and the distance between each centroid and the center of mass of the nearest distinct related cluster (one distinct cluster per centroid). In this paper, we demonstrate that these additional measures offer a valuable way to compare these strategies. Additionally, the improvement in clustering algorithms that use seeds can be evaluated through indirect measures, such as the Adjusted Rand Index (ARI) [7] and Maximal HITS (MHITS), which will be further explained in this paper.

The rest of this paper is organized as follows: the next section provides an overview of related works, Section 3 describes the proposed solution, Section 4 introduces two new direct measures, Section 5 discusses the indirect measures (MHITS), Section 6 presents the experiments, and Section 7 concludes the work and outlines future developments.

2. Related Works

There are various techniques available for data sampling [8-11]. One of these methods is cluster sampling, where clusters are created, and a set of elements is extracted from these clusters. Another approach involves selecting points from purportedly different clusters (not yet created) and repeating this process until the required number of samples from each cluster is obtained. In this context, initialization clustering techniques can be employed to find representative samples.

Computational efficiency and the fast convergence of K-means (i.e., the number of iterations) are relevant considerations when creating clustering seeds. However, these factors alone are insufficient to evaluate the quality of initialization algorithms [10]. Consequently, in this paper, we propose some measures that may be used to assess these algorithms.

In research on initialization methods for clustering, algorithms can be categorized into linear (often non-deterministic) and super linear (usually deterministic) initialization methods [10]. The most common initialization methods include Forgy [12], K-means++ initialization [13] (referred to as Initialization++ in this work), and Max-min strategies [14-16]. These strategies are commonly discussed in the literature on initialization methods. Consequently, we compare our proposal to these approaches.

However, there is no consensus on the best technique to use across different scenarios and applications [10, 17]. While there are numerous algorithms available for data clustering [17-20], K-means [21] currently stands as the most widely used algorithm that employs seeds. Typically, the Euclidean distance approach [22] is used to measure the distance between data points. This algorithm utilizes centroids to initiate the clustering process, highlighting the importance of initialization methods for K-means. Consequently, some authors use measures to evaluate improvements in K-means behavior when comparing different initialization methods.

An entropy-based initialization method is also utilized to provide seeds [11]. This algorithm has a quadratic complexity. Recently, the concept of maximizing the joint probability of pixel intensities with distance restriction criteria has also been employed to provide seeds [23]. According to the authors, the algorithm's performance depends on the optimal number of clusters in the dataset. In both works [11, 23], the algorithms were primarily evaluated based on the number of iterations, emphasizing that this factor alone is not a suitable parameter for evaluating initialization algorithms.

Other clustering strategies involve density-based and grid-based clustering algorithms. One popular density-based clustering algorithm is DBSCAN [24], which aims to group a central point and its neighbors when they exhibit high density relative to the occupied space while separating distant points as outliers. Variations of this algorithm have been proposed in recent years [25-27].

Regarding grid-based clustering algorithms, notable strategies include GRIDCLUS [28], STING [29], and CLIQUE [30]. These methods typically employ a grid structure to divide the data space into multidimensional boxes. The number of points within these boxes (density of the boxes) is used to identify clusters. Therefore, grid algorithms often utilize density to group data and are sometimes classified as density-based approaches. Both density-based and grid-based strategies generally do not require additional algorithms to find the initial points within clusters. However, the resulting groups may provide good samples. Advances over time have also contributed to the development of these algorithms [3, 31, 32].

3. Proposed Sample of Groups (SOG) Algorithm

As we can observe, K-means relies on strategies for selecting the initial multidimensional data points, often referred to as initial points, seeds, or group samples. Consequently, many authors evaluate the impact of these strategies on the effectiveness of K-means. We have also explored alternative approaches for clustering data that do not require the use of initial points within clusters.

Our proposed method combines insights from these different strategies to determine the initial points and introduces new measures to compare its results with other approaches. The process of finding the seeds involves several steps, which are described in the following subsections.

3.1. Create Boxes to Split Data

The first and arguably the most crucial step in our proposal is to determine the appropriate number of multidimensional boxes (or simply boxes). To accomplish this, we initially consider a subset of steps, outlined in items i. and ii.

i. Set the minimum number of boxes

We consider that the number of boxes is directly proportional to the data size (*dataSize*) and the number of data dimensions (*dimNumber*), but inversely proportional to the number of clusters (*clusterNumber*). With this in mind, we propose the following expression to calculate the minimum number of boxes (*minBoxesNumber*), where α represents an adjustment factor that aims to increase the number of boxes containing data, as empty data boxes will be discarded in future steps:

```
Algorithm 1: minBoxesNumberFunction(dataSize, dimNumber, clustersNumber)

    1. if (dataSize > dimNumber):
            a. minBoxesNumber = α*(dataSize*dimNumber)/clustersNumber
    2. if (dataSize <= dimNumber):
            b. minBoxesNumber = α*(dataSize*dataSize)/clustersNumber
```

In this work, we consider $\alpha = 10$ to maintain a low number of boxes while ensuring a minimum number of final boxes that may contain a significant number of points. The concept of relevant boxes, which refers to boxes that contain a relevant number of points, will be discussed in upcoming subsections.

ii. Set the adjusted dimensions number and the minimum number of markers per dimension

Considering that the boxes have the same size and their sides have equal lengths, it is possible to determine the space delimited by each box by calculating the side length. We view the side lengths as markers in each dimension. Initially, we calculate the minimum number of markers per dimension (*minMarkersPerDim*). This number can be estimated using the equation:

$$minMarkersPerDim = \sqrt[dimNumber]{minBoxesNumber} \qquad (1)$$

However, we acknowledge that many dimensions may not be relevant for discovering data clusters. Therefore, instead of using *dimNumber*, we apply a logarithmic function to reduce the number of dimensions, using *minBoxesNumber* as the input parameter. In this case, the adjusted dimension number (*adjDimNumber*) is calculated as follows:

$$adjDimNumber = log(minBoxesNumber) \qquad (2)$$

We ensure that *minMarkersPerDim* keeps the total number of boxes approximately equal to *minBoxesNumber* by setting *minMarkersPerDim* to 10. Since our goal is to group data considering at least 2 dimensions, we set the minimum value of *adjDimNumber* to 2. The maximum value of *adjDimNumber* is limited by the dimension number of the original data.

AdjDimNumber represents the number of dimensions effectively used. To select these dimensions, we employ the standard deviation as a criterion for relevance. Dimensions with higher standard deviations offer greater separation among data points, making it easier to correctly partition the data into boxes and, consequently, separate them into distinct clusters. By considering the relevant dimensions and the number of markers per dimension, we can create the boxes used to partition the data. Additionally, the proposed heuristics ensure that the number of dimensions and the number of boxes remain relatively low, enabling good algorithm performance.

3.2. Select Fuller Boxes

To select fuller boxes, we employ certain criteria, which include:

- Discarding boxes that do not contain any data;
- Considering the average amount of data (*avgDataOfBoxes*) and the standard deviation of data (*stdDataOfBoxes*) among the remaining boxes. We only retain boxes that have a number of data points (*numberOfDataInBox*) satisfying the following formula:

$$numberOfDataInBox >= avgDataOfBoxes + 2 * stdDataOfBoxes \tag{3}$$

The objective is to retain only boxes with a higher number of data points, which is equivalent to having a higher data density. These selected boxes will provide the initial points.

3.3. Select the Representatives of the Most Relevant Boxes

In this step, we calculate the distances between all the filtered boxes created in the previous step. We use the centers of these boxes as reference points and calculate the distances between the selected boxes using the Euclidean distance in the relevant dimensions chosen earlier. It's important to note that the number of selected boxes is typically a small fraction of the total number of data points, which makes the calculations between the boxes relatively fast, enhancing the algorithm's efficiency, as often seen in grid clustering strategies.

In addition to distance, we also consider the sum of the number of data points (pointsNumber) inside the compared boxes as a relevance factor. The idea is to assign 80% of the relevance to the distance between the boxes and 20% of the relevance to the sum of the number of points contained in both boxes. To achieve this, we maintain a list of relations among boxes, which includes:

$$relation = \left(distance;\ pointsNumber;\ idBox1;\ idBox2\right) \tag{4}$$

$$relations\ = \left[relation1,\ relation2,\ \ldots,\ relationN\right] \tag{5}$$

Then, we sort the relations based on their distances in ascending order, with higher distances indicating higher relevance. After sorting, we create segments within the relations where the distances range from 20% of the value of the highest distance in the segment. Within each segment, we sort again based on the sum of the number of points from the compared boxes, with a higher number of points indicating higher relevance.

The algorithm's concept is that, after proper sorting, the first pair of relations contains the two most relevant boxes. The next relevant box, extracted from the relations, must form pairs with all the relevant boxes already discovered, and so on, until the number of relevant boxes matches the desired number of clusters. Based on this concept, we apply the following algorithm to select the most relevant boxes:

```
Algorithm 2: relevantBoxesFunction(relations, clusterNumber)
    1.  relevantBoxes = []
    2.  count = 0
    3.  while (clustersNumber > size (relevantBoxes) and count < size (relations) ):
            a.  count = 0
            b.  candidates = [ [ ] [ ] ]
            c.  for relation in relations:
               i. if size(relevantBoxes) == 0:
                    pointsInRelevantBoxes = relation
                    break
               ii. if relation[idBox1] in pointsInRelevantBoxes and relation[idBox2] not in pointsInRelevantBoxes:
                     if relation[idBox2] not in candidates:
                        candidates[relation[idBox2]] = []
                     candidates[relation[idBox2]].append(relation[idBox1])
                     if (size(candidates[relation[idBox2]]) == size(pointsInRelevantBoxes)):
                        pointsInRelevantBoxes.append([relation[idBox2]])
                        break
               iii. if relation[idBox2] in pointsInRelevantBoxes and relation[idBox1] not in pointsInRelevantBoxes:
                     if relation[idBox1] not in candidates[]:
                        candidates[relation[idBox1]] = []
                     candidates[relation[idBox1]].append(relation[idBox2])
                     if (size(candidates[relation[idBox1]]) == size(pointsInRelevantBoxes)):
                        pointsInRelevantBoxes.append([relation[idBox1]])
                        break
               iv. count = count + 1
    4. return pointsInRelevantBoxes
```

Finally, we select the nearest element to the centers of the selected boxes using the Euclidean distance. These selected elements serve as representatives of the clusters.

Fig. 1 illustrates an example of the results obtained from the main steps of the proposed algorithm, as well as the outcomes of some competitor algorithms. The example considers 15 different clusters and 1500 points in two dimensions. The figure consists of four charts: the upper-left chart illustrates the "create boxes to split data" step, the upper-right chart displays the "select fuller boxes" step, the lower-left chart presents the "select representatives of the most relevant boxes" step, and the lower-right chart showcases the representatives obtained using some competitor algorithms (Forgy, Initialization++, Max-min).

It is evident that the proposed algorithm selects one representative point from each cluster. Forgy and Initialization++ do not always achieve the same results (these algorithms are not deterministic, and the results may vary across different executions). Max-min also obtains one representative from each group. However, on average, the points selected by SOG are closer to the center of mass associated with these groups. In this paper, we propose two direct measures to more accurately assess the outcomes of these different approaches.

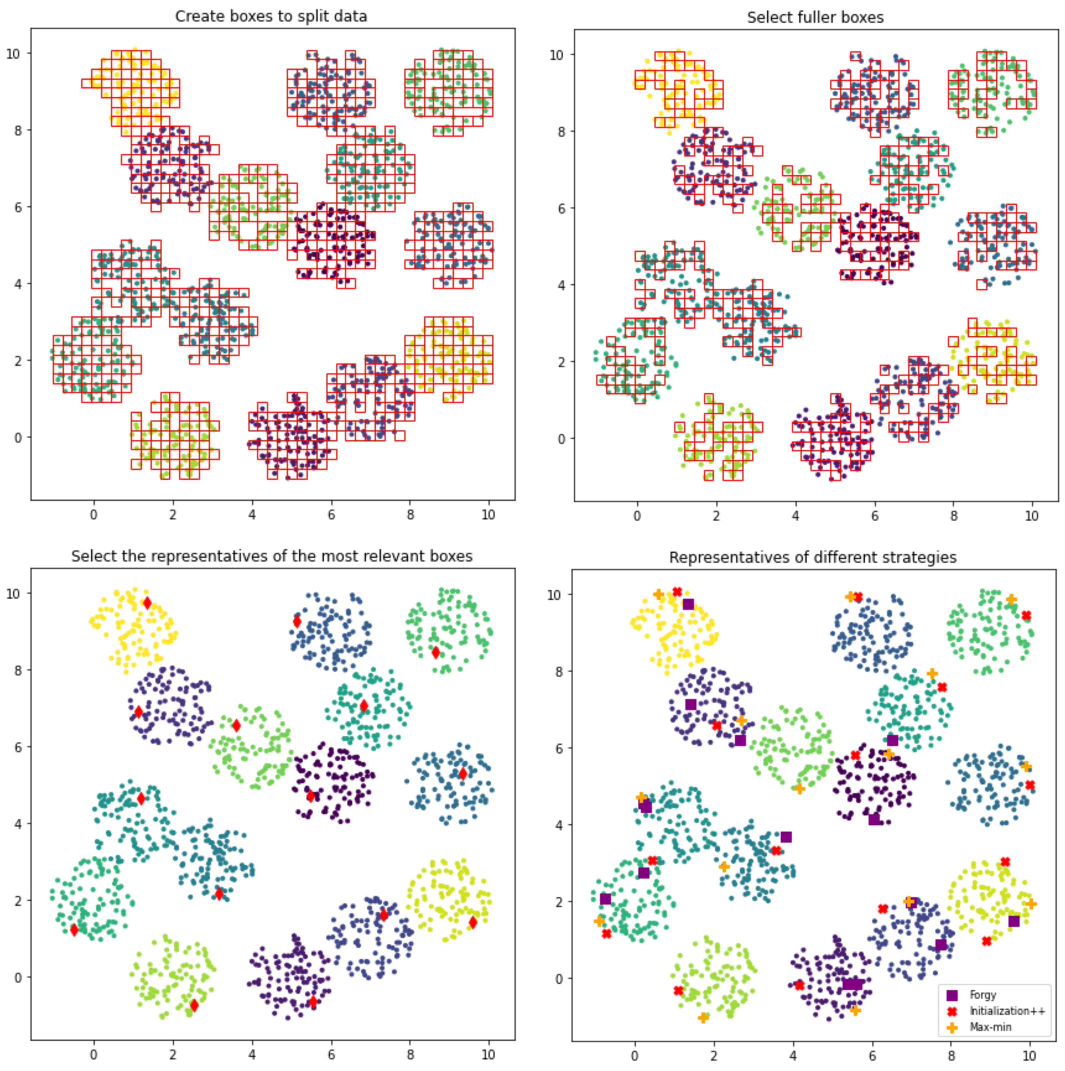

Fig.1. Steps of the proposed algorithm and the results of some other algorithms

4. Proposed Direct Measures

To compare strategies that involve selecting one point or sample from each distinct group within a given dataset, where these groups are not provided beforehand, several questions can be considered to assess their effectiveness, including:

- Are the selected points truly from different groups, or are there instances where points belong to the same group?
- Are the selected points located near the center of mass of their respective groups?

These questions are crucial as they enable us to evaluate the effectiveness of strategies in selecting elements from distinct and well-defined groups, as well as determining if the selected points are positioned close to the center of mass of their respective groups. To assess these aspects, we propose two measures, which will be discussed in the following subsections.

4.1. Rate of Right Points (RRP)

This measure evaluates the number of points that belong to different clusters out of the total number of clusters. It ranges from values between 0 and 1, where values closer to 1 indicate more effective strategies. The following expression formalizes this measure:

$$rateOfRightPoints = numberOfPointsInDistinctClusters \,/\, clustersNumber \tag{6}$$

4.2. Distance to the Center of Mass (DCM)

This measure calculates the average distance of each point, which is expected to represent a cluster, to the likely closest center of mass of a cluster. When the algorithm selects points that are closer to the center of mass for each cluster, it indicates more relevant results. Therefore, this measure is executed in two phases.

i. First Phase

- Calculate the center of mass for each cluster;
- If the selected points belong to the same cluster as the center of mass, calculate the distance between the selected points and the center of the cluster. (Multiple points may correspond to the same center of mass, and some centers of mass may not be associated with any point);
- Sum up the lowest distances. Each selected distance is associated with a different cluster.

ii. Second Phase

- Select the points that are not used in the first phase distances, as well as the clusters without any point. These elements are referred to as second-phase points and clusters;
- For each center of mass, calculate the distance to all second-phase points, creating relations between centers of mass, points, and distances;
- Sort the relations in ascending order based on the distance and obtain the lowest distances for each distinct pair of centers of mass and clusters. (Each element of a pair can contribute only once to the final relation of distances);
- Sum up all the lowest distances from both the first and second phases;
- Divide the result by the total number of clusters to obtain the average distance.

It is important to note that the first phase aims to achieve the minimum average distance by ensuring that the selected points belong to different clusters, with only one point per cluster. If this criterion is not met, the unused points and clusters are transferred to the second phase. In the second phase, the points that are already associated with other clusters are connected to the center of mass of different clusters. However, since the second phase does not calculate all possible combinations of distances (only the minimum distances that appear in a sorted relation list), which is an NP problem, it may not lead to a global optimum result for points that are not directly associated with the correct clusters. Thus, strategies that provide a higher number of points related to different clusters tend to yield better results. Additionally, any local optimum results obtained can be compensated by considering the Rate of Right Points (RRP), as proposed previously.

For most conventional cluster distributions, the Distance to Center Mass (DCM) measure indicates the centrality of the points. Values closer to 0 indicate better results.

5. Maximal HITS

One of the metrics used in this work to evaluate the elements grouped in the clusters is called Maximal HITS (MHITS). MHITS aims to maximize the matches between the elements in the clusters formed by an algorithm and the elements in the reference clusters that have been previously defined. It is an indirect measure to assess the quality of seeds or samples provided by our algorithm, as it evaluates the final result of a clustering algorithm.

For example, if an algorithm assigns the clusters [a, b, c, d, e, f] as follows: [a = cluster 2, b = cluster 2, c = cluster 2, d = cluster 2, e = cluster 1, f = cluster 1], and the reference clusters are [a = group 1, b = group 1, c = group 1, d = group 2, e = group 2, f = group 2], we can compare the two sets of clusters.

If we consider that cluster 1 corresponds to group 1 and cluster 2 corresponds to group 2, then the number of elements correctly identified by the algorithm is 1 (element d). However, if we consider that cluster 1 corresponds to group 2 and cluster 2 corresponds to group 1, then the number of elements correctly identified by the algorithm is 5. Therefore, the maximum number of elements correctly identified is 5, and the MHITS is calculated by dividing this number by the total number of elements.

In this case, the MHITS is given by: MHITS = maximum number of elements correctly identified / total number of elements = 5/6 = 0.833.

6. Experiments

We conducted a comparison of our proposal with three different approaches: Forgy, Initialization++, and Max-min. The comparison was performed using a set of 70 datasets obtained from the UCR Time Series Classification Archive [16], as shown in Table 1. These datasets comprise over 56,000 data series that were split into training and testing sets. To ensure reliable results, we executed non-deterministic algorithms three times for each dataset and considered the average of the results.

Table 1. Datasets used in the experiments (source: [16])

ACSF1	CricketX	GunPoint	OliveOil	SyntheticControl
Adiac	CricketY	GunPointAgeSpan	OSULeaf	ToeSegmentation1
ArrowHead	CricketZ	GunPointMale VersusFemale	PigAirwayPressure	Trace
Beef	DiatomSize Reduction	GunPointOld VersusYoung	Plane	TwoLeadECG
BeetleFly	DistalPhalanx OutlineAgeGroup	Haptics	PowerCons	TwoPatterns
BirdChicken	DistalPhalanx OutlineCorrect	InlineSkate	ProximalPhalanx OutlineAgeGroup	UwaveGesture Librar-yX
BME	DistalPhalanx TWCorrect	InsectEPG RegularTrain	Rock	UwaveGesture Li-braryY
Car	ECG200	ItalyPowerDemand	ScreenType	UwaveGesture LibraryZ
CBF	ECGFiveDays	Lightning2	SmoothSubspace	Wafer
Chinatown	FaceAll	Lightning7	SonyAIBO RobotSurface1	Wine
Chlorine Concentration	FaceFour	Mallat	SonyAIBO RobotSur-face2	WordSynonyms
CinCECGTorso	FacesUCR	MedicalImages	Strawberry	Worms
Coffee	Fish	MiddlePhalanx OutlineAgeGroup	SwedishLeaf	WormsTwoClass
Computers	Fungi	MoteStrain	Symbols	Yoga

In our experiments, we compared the results of each algorithm against the SOG algorithm using three direct measures (RRP, DCM, and Time Consumed to obtain the samples) and four indirect measures (ARI, MHITS, Number of Iterations, and Time Consumed using K-means). The graphics related to the number of datasets indicate the number of comparisons in which the competitor algorithm won, the SOG algorithm won, and the number of ties. We executed the algorithms for both training and testing datasets, resulting in a total of 140 results for each pair of comparisons (competitor against SOG). The graphics related to time present the average execution time for each algorithm across the datasets.

Fig. 2 illustrates a comparison of the selected algorithms against SOG based on the RRP measure. The first bar in blue for each comparison indicates the number of datasets where the comparison resulted in a tie. The second bar, shown in orange, represents the number of datasets where the competitor algorithm achieved better results. The third bar, displayed in yellow, indicates the number of datasets where SOG obtained better results. The results revealed that SOG outperformed the Forgy and Initialization++ algorithms. However, SOG achieved similar results as the Max-min algorithm in this analysis.

Fig. 3 illustrates the DCM measure applied to compare the algorithms against SOG, following the same strategy as the RRP measure. The results demonstrated that SOG consistently outperformed the other algorithms by providing points that were closer to the centers of mass.

Fig. 4 displays a comparison between SOG and each of the competitor algorithms in the context of K-means results using the MHITS measure. This indirect evaluation indicates that SOG consistently achieved better results when compared to the other algorithms, which can be attributed to the suitable selection of samples that were closer to the centers of mass, as verified previously.

Fig. 5 also presents an evaluation of K-means results, comparing the selected algorithms with SOG using ARI as the measure of analysis. SOG achieved superior results compared to Forgy and Initialization++. However, Max-min and SOG obtained equivalent results.

Fig. 6 displays an analysis of the number of iterations used by K-means, comparing the three competitor approaches with SOG. The proposed algorithm, SOG, allowed K-means to find the final centroids using a smaller number of iterations in all comparisons. It is worth noting that the Max-min strategy yielded similar results.

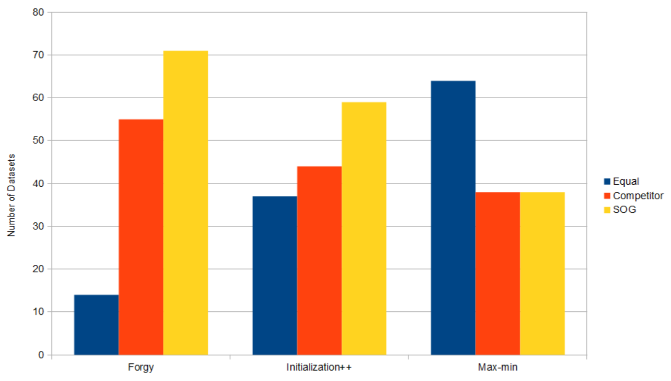

Fig.2. Comparison of algorithms and SOG over the datasets considering the RRP

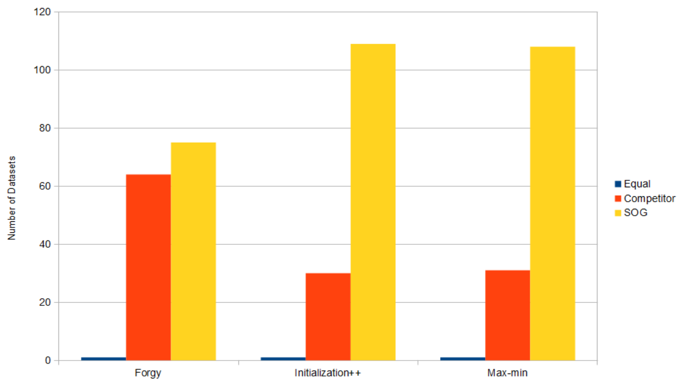

Fig.3. Comparison of algorithms and SOG over the datasets considering the DCM

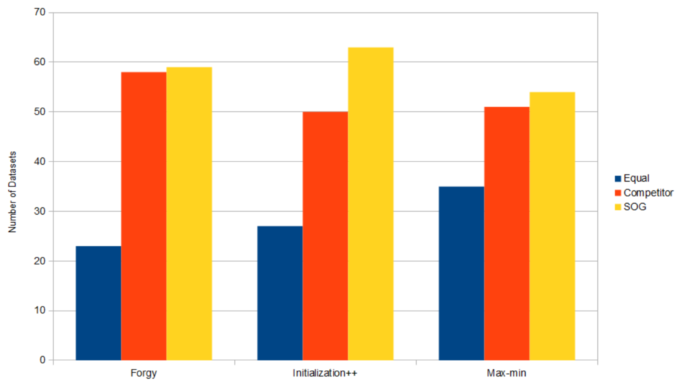

Fig.4. Comparison of algorithms and SOG over the datasets considering K-means and MHITs

Upon analyzing the results holistically, it is evident that the Sample of Groups (SOG) algorithm outperformed the competitor algorithms, consistently yielding higher-quality seeds. Particularly, the DCM measure demonstrated that SOG effectively identified seeds that were closer to the centers of mass of the clusters, indicating its effectiveness in this aspect. Overall, these findings highlight the superior performance and quality of the seeds generated by the SOG algorithm.

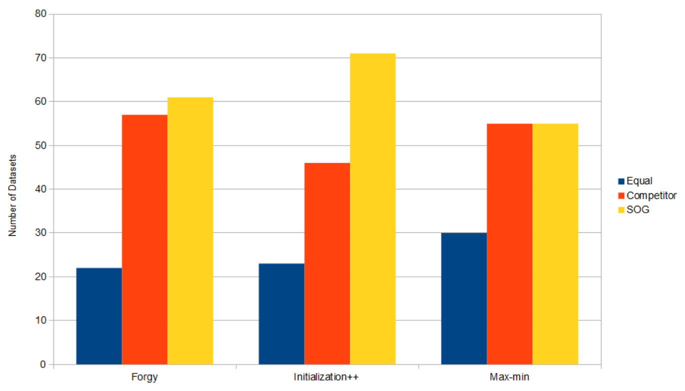

Fig.5. Comparison of algorithms and SOG over the datasets considering K-means and ARI

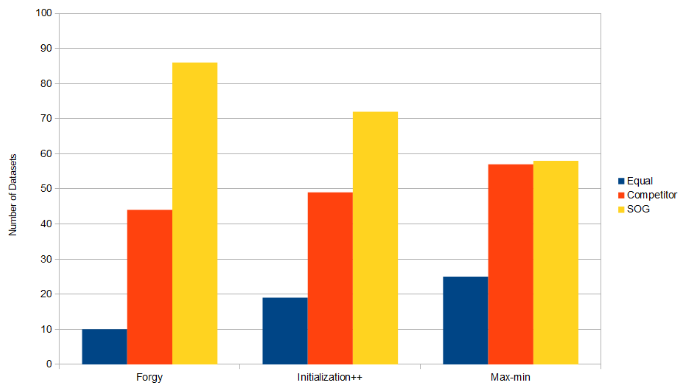

Fig.6. Comparison of algorithms and SOG over the datasets considering K-means and number of iterations

7. Conclusions and Future Works

This work presents a novel perspective on initialization algorithms, particularly in the context of K-means clustering. Instead of solely focusing on their role in initializing the clustering process, we propose viewing these algorithms as sample selectors for datasets with unknown clusters. The objective is to find one representative point from each distinct cluster. If necessary, these algorithms can be applied multiple times to collect more points and, if needed, cluster the selected samples rather than the entire dataset. In this process, we create boxes with the right size to split the data, and then we select the representatives of the most relevant boxes. The selection of good representative points through sample algorithms may, for instance, enhance the quality of the subsequent clustering process using other algorithms, such as K-means.

To properly evaluate this proposal, we introduce two new direct measures: the Rate of Points in the Right Clusters (RRP) and the Distance of the Selected Samples to the Center of the Cluster (DCM). These measures allow for a thorough analysis of sample algorithms, complementing traditional indirect measures that assess their impact on K-means results. In addition, we propose a novel strategy, the Sample of Groups (SOG) algorithm, which combines concepts from grid and density clustering strategies with distance-based considerations.

The experimental results demonstrate that our proposed SOG algorithm achieves superior performance when evaluated using our proposed measures as well as traditional indirect measures commonly used to assess these algorithms. The DCM measure deserves special attention in this regard, as it demonstrated the ability of the algorithm to find samples closer to the center of mass than competing algorithms.

Briefly, the main contributions of this article are:

- presentation of an algorithm that finds samples or seeds of data that represent different groups or clusters;
- proposals for measures to assess the quality of samples or seeds extracted from data that can be grouped, including: the *Rate of Right Points* (RRP), the *Distance to the Center of Mass* (DCM) , and the *Maximal HITS* (MHITS) that is an indirect measure; and
- an experiment that demonstrates the superiority of the proposed algorithm when compared to competing algorithms.

In future research, we aim to investigate the effects of multiple sampling compared to the strategy of selecting samples based on already clustered datasets using K-means and other algorithms. The objective is to determine the optimal scenarios for each of these strategies and explore their respective limitations.

References

[1] T. Velmurugan and T. Santhanam, “A Survey of partition based clustering algorithms in data mining: An experimental approach”, *Information Technology Journal*, vol. 10, no. 3, pp. 478-484, 2011.

[2] M. I. Akodj`enou-Jeannin, K. Salamatian, and P. Gallinari, “Flexible grid-based clustering”, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4702 LNAI, pp. 350-357, Springer Verlag, 2007.

[3] J. A. Hartigan and M. A. Wong, “Algorithm AS 136: A K-Means Clustering Algorithm”, *Applied Statistics*, vol. 28, no. 1, p. 100, 1979.

[4] P. Bhattacharjee and P. Mitra, “A survey of density based clustering algorithms”, *Frontiers of Computer Science*, vol. 15, pp. 1-27, 2021.

[5] M. Elfil and A. Negida, “Sampling methods in clinical research; an educational review”, *Archives of Academic Emergency Medicine*, vol. 7, no. 1, p. 52, 2019.

[6] D. Xu and Y. Tian, “A Comprehensive Survey of Clustering Algorithms”, *Annals of Data Science*, vol. 2, pp. 165-193, 2015.

[7] M. Ankerst, M. M. Breunig, H. P. Kriegel, and J. Sander, “OPTICS: Ordering Points to Identify the Clustering Structure”, *SIGMOD Record (ACM Special Interest Group on Management of Data)*, vol. 28, pp. 49-60, 1999.

[8] D. Arthur, D. Arthur, and S. Vassilvitskii, “K-means++: the advantages of careful seeding”, *In Proceedings of the 18th annual ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[9] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping Multidimensional Data: Recent Advances in Clustering*, pp. 25-71, Springer Berlin Heidelberg, 2006.

[10] M. E. Celebi, H. A. Kingravi, P. A. Vela, A comparative study of efficient initialization methods for the k-means clustering algorithm, Expert Systems with Applications, vol 40, no. 1, pp. 200-210, 2013.

[11] K. Chowdhury, D. Chaudhuri, A. K. Pal. An entropy-based initialization method of K-means clustering on the optimal number of clusters. Neural Computing and Applications, vol. 33, pp. 6965–6982, 2021.

[12] G. Balakrishnan and Z. Syed, “Scalable Personalization of Long-Term Physiological Monitoring: Active Learning Methodologies for Epileptic Seizure Onset Detection”, in *PMLR* (N. D. Lawrence and M. Girolami, eds.), vol. 22 of *Proceedings of Machine Learning Research (PMLR)*, (La Palma, Canary Islands), pp. 73-81, 2012.

[13] R. Q. A. Fernandes, W. A. Pinheiro, G. B. Xex´eo, and J. M. de Souza, “Path Clustering: Grouping in an Efficient Way Complex Data Distributions”, *Journal on Today’s Ideas - Tomorrow’s Technologies*, vol. 5, pp. 141-155, 2017.

[14] R. C. De Amorim, “An empirical evaluation of different initializations on the number of K-Means iterations”, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7629 LNAI, pp. 15-26, Springer, Berlin, Heidelberg, 2013.

[15] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli, “Euclidean Distance Matrices: Essential theory, algorithms, and applications”, *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 12-30, 2015.

[16] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, “The UCR Time Series Classification Archive”, 2018.

[17] M. Z. Rodriguez, C. H. Comin, D. Casanova, O. M. Bruno, D. R. Amancio, L. F. Costa, F. A. Rodrigues. Clustering algorithms: A comparative approach. PLoS ONE, vol. 14, no. 1, 2019.

[18] J. M. Peña, J. A. Lozano, and P. Larrañaga, “An empirical comparison of four initialization methods for the KMeans algorithm”, *Pattern Recognition Letters*, vol. 20, no. 10, pp. 1027-1040, 1999.

[19] W. Wang, J. Yang, and R. R. Muntz, “STING: A Statistical Information Grid Approach to Spatial Data Mining”, in *Proceedings of the 23rd International Conference on Very Large Data Bases*, VLDB ’97, (San Francisco, CA, USA), pp. 186-195, Morgan Kaufmann Publishers Inc., 1997.

[20] T. Sajana, C. M. Sheela Rani, and K. V. Narayana, “A survey on clustering techniques for big data mining”, *Indian Journal of Science and Technology*, vol. 9, pp. 1-12, 2016.

[21] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990.

[22] E. Forgy, “Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classification”, *Biometrics*, vol. 21, no. 3, pp. 768-769, 1965.

[23] K. Chowdhury, D. Chaudhuri, A. K. Pal, A.K, A. Samal. Seed selection algorithm through K-means on optimal number of clusters. Multimedia and Tools Applications, vol. 78, pp. 18617–18651, 2019.

[24] B. Yuan, W. Zhang, and Y. Yuan, “A max-min clustering method for k-means algorithm of data clustering”, *Journal of Industrial and Management Optimization*, vol. 8, pp. 565-575, 2012.

[25] B. Schelling, L. Miklautz, and C. Plant, “Non-linear Cluster Enhancement: Forcing Clusters into a compact shape”, in *24th European Conference on Artificial Intelligence*, 2020.

[26] J. M. Santos and M. Embrechts, “On the use of the adjusted rand index as a metric for evaluating supervised classification”, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5769 LNCS, pp. 175-184, Springer, Berlin, Heidelberg, 2009.

[27] E. Schikuta, “Grid-clustering: An efficient hierarchical clustering method for very large data sets”, in *Proceedings - International Conference on Pattern Recognition*, vol. 2, pp. 101-105, Institute of Electrical and Electronics Engineers Inc., 1996.

[28] P. Singh and P. A. Meshram, “Survey of density based clustering algorithms and its variants”, in *Proceedings of the International Conference on Inventive Computing and Informatics, ICICI 2017*, pp. 920-926, Institute of Electrical and Electronics Engineers Inc., 2018.

[29] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, “Automatic subspace clustering of high dimensional data for data mining applications”, *SIGMOD Record*, vol. 27, pp. 94-105, 1998.

[30] J. Lever, M. Krzywinski, and N. Altman, “Points of significance: Principal component analysis”, 2017.

[31] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis”, *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37-52, 1987.

[32] M. Ester, M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise”, *KDD-96*, pp. 226-231, 1996.

Authors’ Profiles

Wallace A. Pinheiro graduated in Electronic Engineering in 1998 at the Federal University of Rio de Janeiro (UFRJ). Master in Computer Systems in 2004 at the Military Engineering Institute (IME). Doctor in Systems and Computer Engineering in 2010 at UFRJ. From 2010 to 2014, He was a professor at the IME, working on the following themes: command and control, databases, data quality, and information retrieval. From 2014 to 2017, he worked at the Systems Development Center (CDS), an organization responsible for developing various systems used by the Brazilian Army. In 2018, he made a post-doctorate at UFRJ in Systems and Computer Engineering. From 2019 to 2022, he returned to work at the Systems Development Center (CDS), where he could apply his postdoctoral research. Currently, he is interested in data mining and artificial intelligence.

Ana B. S. Pinheiro graduated in Maternal and Child Nursing from the Federal University of Rio de Janeiro (2000). Post-graduated in Neonatal Intensive Care from Universidade Federal Fluminense (2003). Master in Tropical Medicine in the area of Molecular Biology from the University of Brasília (2019). From 2001 to 2004, she worked at the Army Health School. From 2004 to 2008, she worked at the Army Central Hospital. Besides, she worked in the following areas of the Military Polyclinic of Rio de Janeiro (2008 to 2014): Center for Study in Integrated Therapies; Commission for Infection Control, Control, and Combat against Dengue. Furthermore, she worked in the Army Health Directorate in the Preventive and Assistance Health Section (2015 to 2017) and at the Brasília Military Hospital in the area of physiotherapy (2017 to 2019). Currently, her interests involve: epidemiology, and information technologies focused on the health area.

How to cite this paper: Wallace A. Pinheiro, Ana B. S. Pinheiro, "Sample of Groups: A New Strategy to Find a Representative Point for Each Undisclosed Cluster", International Journal of Information Technology and Computer Science(IJITCS), Vol.15, No.5, pp.1-11, 2023. DOI:10.5815/ijitcs.2023.05.01