Variable neighborhood search algorithm for k-means clustering

To cite this article: V I Orlov et al 2018 IOP Conf. Ser.: Mater. Sci. Eng. 450 022035

View the article online for updates and enhancements.

You may also like

- Integration K-Means Clustering Method and Elbow Method For Identification of The Best Customer Profile Cluster
  M A Syakur, B K Khotimah, E M S Rochman et al.

- PERFORMANCE ANALYSIS OF COMBINED METHODS OF GENETIC ALGORITHM AND K-MEANS CLUSTERING IN DETERMINING THE VALUE OF CENTROID
  Putra Adya Zizwan, Muhammad Zarlis and Erna Budhiarti Nababan

- K-Means Algorithm Performance Analysis With Determining The Value Of Starting Centroid With Random And KD-Tree Method
  Kamson Sirait, Tulus and Erna Budhiarti Nababan
Variable neighborhood search algorithm for k-means clustering

V I Orlov¹, L A Kazakovtsev¹,², I P Rozhnov¹, N A Popov¹ and V V Fedosov¹

¹Reshetnev Siberian State University of Science and Technology, 31, Krasnoyarsky Rabochy av., Krasnoyarsk, 660037, Russian Federation
²Krasnoyarsk State Agrarian University, 90, Mira av., Krasnoyarsk, 660049, Russian Federation

E-mail: levk@bk.ru

Abstract. We propose new algorithms of Greedy Heuristic Method for solving the classical problem of cluster analysis, k-Means, which allows us to obtain results with better objective function values in comparison with known algorithms such as k-Means and j-Means. Their comparative efficiency is proved by experiment on various data sets including multidimensional data of non-destructive rejection tests of electronic components for the space industry.

1. Introduction

Developing the methods of forecasting and providing electronic components performance under adverse external influences is a very important problem. One of the most important part of this problem is providing of resistance to thermal and radiation stresses [1, 2].

Traditional methods of ensuring resistance to impacts are applied under the assumption that the resistance of all electronic components in the batch is the same. However, the resistance of components in the batch is different and depends on the internal defects of the product.

The identification of the most significant of such defects is the purpose of the selective destructive physical analysis (DPA). At the same time, disseminating the results of DPA for the whole batch should proceed with caution. We need to be sure, that we are dealing with a single batch of electronic components, which are made of single party of raw material. Therefore, the identification of true production batches from presumably combined parties of the EEE components is one of the most important measures during tests [3].

The non-destructive tests consist of monitoring the current-voltage characteristics of Electrical, Electronic and Electro-mechanical (EEE) components. Test results tabulating and serve as data for analysis. Differences in results means differences in the performance characteristics of EEE components and its belonging to different production batch. The entire array of diagnostic test data is used to identify groups of components with homogeneous characteristics in the incoming batch. The range of the value of each parameter is too narrow to ascribe the component to a particular batch, but this is possible by the totality of parameters.

Thus, problem of identifying these homogeneous groups is a clustering problem.
The k-means problem is the most popular clustering model [4]. The aim of this problem is finding \( k \) cluster centers \( X_1\ldots X_k \) in a \( d \)-dimensional space, such that the sum of squared distances to known points \( A_i \) (clustered objects) is minimum.

\[
F(X_1\ldots X_k) = \arg\min_{X \in \mathbb{R}^{d}} \sum_{i=1}^{N} \min_{X \in \{X_1\ldots X_k\}} \|A_i - X\|^2,
\]

(1)

In conditions of the space industry, this problem must be solved with maximum accuracy, the result must be verifiable and multiple runs of the clustering algorithm must result in the same or very similar solution.

2. Known algorithms

The most popular algorithm for this problem is a procedure of the same name (k-means), also called ALA procedure (Alternating Location- Allocation) or Lloyd’s algorithm. This popular algorithm includes two repeated steps: allocating the objects to the cluster centers (an object is allocated to the nearest center, a subset of objects with the same nearest center is considered as a cluster) and recalculating the centers of each cluster. The number of clusters must be known.

Algorithm 1 (k-means). Required: data objects \( A_1\ldots A_N \), \( k \) initial cluster centers \( X_1\ldots X_k \).

1. For each center \( X_i \), form a cluster, i.e. a subset \( C_i \) of objects, for which this center is the nearest one.
2. For each cluster \( C_i \), calculate a new center \( X_i \). (i.e. solve the Weber problem).
3. Repeat Step 1 if any changes were made at Steps 1 and 2.

The idea of the k-mean algorithm was proposed in 1956 by Steinhaus [5], the algorithm was developed by Lloyd a year later, although his work [6] was published only in 1982. After McQueen’s paper [7], the algorithm became known as the k-means algorithm or Lloyd’s algorithm. Since then, the k-means algorithm, its improvement, modification and combination with other algorithms, has become the topic of many researchers’ work.

Alsabti et al [8] have proposed an efficient algorithm based on k-means using a pattern in a k-d tree. Nigam et al [9] has given an algorithm which uses labeled and unlabeled documents based on a classifier. Kanungo et al [10] have presented a filtering algorithm based on K-means clustering algorithm. Cheung [11] have developed a generalized K-means clustering algorithm, which give a correct results without previously known number of clusters. Xiaoli Cui [12] presented an algorithm based on k-means, which not use the whole dataset, but only representative points selecting by a sampling technique. Xiong et al [13] studied the effect of skewed data distribution on k-means clustering. They investigate how data distributions influences the performance of k-means clustering algorithm. Zhang et al [14] developed a simple and effective technique based on k-means clustering algorithm for classification and evaluation of NBA guards, including Euclidean distance as a measure of similarity. Wang et al [15] presented an improved k-means algorithm, which includes noise data filter and overcome the shortcomings of the traditional k-means clustering algorithm. Noise filter use the density based detection methods based on characteristics of noise data. R. V. Singh et al [16] provide a modified k-means algorithm based on the sensitivity of initial center of clusters. This algorithm separates the whole data space into different segments and calculates frequency of data points in each segment. The segment with maximum frequency of data point will contain the centroid of cluster at a maximum probability. Shi Na et al [17] presents a k-means algorithm, improved by storing calculated distance information in every iteration, which is to be used in the next iteration. Method avoids computing the distance of each data object to the cluster centers repeatedly and saves the running time. A similar approach is applied by Rani [18].

An important part of the k-means algorithm is the selection of the initial centers, which is often the subject of separate studies. Bhusare and Bansode [19] presents k-means algorithm in combination with Pillar algorithm. Pillar algorithm effectively chooses the initial centroids and improves accuracy of clusters. However, this algorithm has outlier problem leads to reduced performance. Improvement of the Pillar algorithm allowed to solve this problem. Kaur et al [20] also investigated the problem of the
choice of initial centers. Wang Shunye et al [21] uses dissimilarity matrix and Huffman tree for selection of initial centroid. Mahmud et al [22] has proposed a heuristic method to find better initial centroids and improve cluster accuracy. Their method includes merge sort and computing the average score. K.A. Abdul Nazeer et al [23] presents an improved k-means clustering algorithm, which includes special methods of the selection of initial centroids and assigning data points to the clusters. Speaking about solving of k-means problem, it is necessary to mention J-means algorithm, which has developed by Hansen and Mladenović [24]. This algorithm is considered one of the most effective and accurate algorithms for this problem and for p-median problem as well. The algorithm replaces the centroids with one from the data vectors (best one in terms of an object function), and then continues the search using the standard k-means.

It is possible to note the generality of clustering problems on the k-means model basis and the similar ones (the parameters are the coordinates of the cluster centers) with problems based on the mixture distribution model. Parameters of this kind of problems are the parameters of distribution (in fact, mathematical expectations are actually cluster centers) supplemented by probability distributions. In addition, the most popular algorithms for solving such problems – k-means and EM-algorithm are similar in structure: both are procedures with two alternating steps of cluster location and allocation. Finally, the objective functions have the multiple extrema. This makes it possible to use for the k-means problems the same methods of increasing the accuracy and stability of the solutions, which are successfully used in solving problems of separation of a mixture of distributions.

stashkov et al [25, 26] considered the application of the VNS-algorithm (variable neighborhood search) as an extended local search in combination with EM-algorithm. The essence of the VNS-algorithm [27] is that for a certain intermediate solution, the set of neighborhoods of this solution is determined. A solution with the best value of the objective function is searched in the next neighborhood of this set with the use of the corresponding algorithm of local search. If a solution is found, the intermediate solution is replaced by this new solution, and the search in the same neighborhood continues. If such a solution is not found, a new neighborhood of the search is chosen from the set of neighborhoods of the intermediate solutions.

The idea of the present paper is to apply new approaches of the greedy heuristic method [28] to the k-means problem in order to obtain the most accurate (in terms of the objective function) and stable result.

3. Algorithms of the Greedy Heuristics Method
A greedy agglomerative heuristic procedure for the k-means problem and analogous ones [25] consists of two steps. Suppose that there are two known (parent) solutions to the problem (the first of which, for example, is the best one from known), represented by the sets of centers of the clusters S.

Step 1. The sets of parent solutions have to be united. We obtain an intermediate infeasible (with an excessive number of clusters) solution.

Step 2. We make a gradual decrease in the number of distributions, removing distribution one by one in order to minimally deteriorate the value of the objective function.

Below, we describe the basic greedy heuristic algorithm of the Greedy Heuristic Method [25]:

**Algorithm 2.** Basic greedy agglomerative heuristic procedure.

Required: initial number of clusters \( K \), needed number of clusters \( k \).

1. Choose randomly an initial solution with \( K \) cluster centers \( S = \{X_1, \ldots, X_k\} \).
2. Run Algorithm 1 with initial solution \( S \), obtain the new (improved) solution \( S' \).
3. If \( K = k \) then STOP.
4. For each \( i \in \{1, K \} \) do:
   4.1. Obtain truncated set \( S' = S \setminus \{X_i\} \).
   4.2. Run Algorithm 1 with initial solution \( S' \). Here, only one iteration of Algorithm 1 is performed. Store the obtained value of the objective function (1) to variable \( F'_i \).
   4.3. Next iteration 4.
5. Choose index $i'' = \arg \max_{i'=1,2,...,k} F_i'$.
6. Obtain truncated set $S'' = S \setminus \{X_{i''}\}$, improve it with Algorithm 1, then go to Step 3.

The three procedures below are based on this basic procedure.

Algorithm 3. Greedy procedure with the partial merger #1.

Required: sets of cluster centers $S' = \{X'_1, ..., X'_k\}$ and $S'' = \{X''_1, ..., X''_k\}$.

1. For each $i'' \in \{1, k\}$ do:
   1.1. Merge set $S'$ with one element of set $S''$: $S = S' \cup \{X''_{i''}\}$
   1.2. Run Algorithm 2 from Step 2 with the initial solution $S$. Remember the obtained result (obtained set of cluster centers and the corresponding value of the objective function).
3. Return the best of the solutions (in accordance with the objective function value) obtained at Step 1.2.

The next algorithm requires more computations.

Algorithm 4. Greedy procedure with the full merger.

1. Merge sets $S = S' \cup S''$.
2. Run Algorithm 2 with the initial solution represented by $S$.
3. The last option is an algorithm which adds a randomly chosen part of the second set of cluster centers to the first set [28].

Algorithm 5. Greedy procedure with partial merger #2.

1. Choose random value $r' \in [0,1]$. Calculate $r=[2r' (k/2-2)] + 2$. Here, $[.]$ is the integer part.
2. Repeat $k$-times:
   2.1. Form randomly subset $S'''$ of $r$ elements of $S''$. Merge sets $S = S' \cup S'''$.
   2.2. Run Algorithm 2.1 with the initial solution $S$.
3. Return the best of the solutions obtained at Step 2.2.

The above procedures are local search procedures which search in a neighborhood of a known (“parent”) solution represented by set $S'$. In this process, the second solution $S''$ is a parameter of this neighborhood.

The following algorithm performs search in various neighborhoods with randomly chosen parameter.

Algorithm 6 (k-VNS).
1. Run Algorithm 1 from a randomly chosen initial solution and obtain solution $S$.
2. $s=s_{\text{start}}$ (number of the current neighborhood).
3. $i=0$, $j=0$ (number of idle iterations in the current neighborhood and overall number of idle iterations).
4. Run Algorithm 1 from a randomly chosen initial solution and obtain solution $S'$.
5. Depending on variable $s$ (values 1, 2 and 3 are allowed), run Algorithm 3, 4 or 5 with the initial solutions $S$ and $S'$, correspondingly. Thus, the integer variable $s$ and the $S'$ determine the neighborhood.
6. If the obtained result is better than $S$ (in terms of the objective function values) then replace $S$ with this new result; $i=0$, $j=0$; go to Step 5.
7. $i=i+1$.
8. If $i < i_{\text{max}}$ then go to Step 4.
9. $i=0$, $j=j+1$, $s=s+1$; if $s>3$ then $s=1$.
10. If $j > j_{\text{max}}$ or the time limit is exceeded then STOP. Otherwise, go to Step 5.

Parameter $i_{\text{max}}$ is the maximum number of idle iterations in the current neighborhood and $j_{\text{max}}$ is the maximum allowed number of neighborhood switchings. We ran our algorithm with $i_{\text{max}}=2k$, $j_{\text{max}}=2$.

The other important parameter is $s_{\text{start}}$, which determines the initial number of the neighborhood. In our computational experiments, we used all allowed values of this parameter. Depending on this value, algorithms are denoted as k-VNS1, k-VNS2, k-VNS3.
4. Experimental results

For computational experiments with our new algorithm, we used datasets of the UCI repository and data of non-destructive tests of electronic components. For each of the datasets, we performed 30 attempts of each algorithm. Algorithms k-means and j-means ran in multistart mode (tables 1-8).

Table 1. Results of experiments with dataset Ionosphere (10 clusters, 15 seconds, 30 attempts).

| Algorithm | Objective function value | Min     | Max     | Average | Std. dev. |
|-----------|-------------------------|---------|---------|---------|-----------|
| j-Means   |                         | 1 590.34| 1 598.83| 1 594.77| 2.41      |
| k-means   |                         | 1 590.93| 1 598.61| 1 595.28| 2.47      |
| VNS1      |                         | 1 586.38| 1 586.65| 1 586.52| 0.12      |
| VNS2      |                         | 1 586.38| 1 591.99| 1 588.00| 2.36      |
| VNS3      |                         | 1 586.38| 1 591.99| 1 587.24| 1.57      |

Table 2. Results of experiments with dataset Ionosphere (20 clusters, 15 seconds, 30 attempts).

| Algorithm | Objective function value | Min     | Max     | Average | Std.dev.  |
|-----------|-------------------------|---------|---------|---------|-----------|
| j-Means   |                         | 1 282.18| 1 299.13| 1 291.92| 4.83      |
| k-means   |                         | 1 286.30| 1 310.54| 1 301.98| 5.81      |
| VNS1      |                         | 1 239.16| 1 259.56| 1 246.39| 5.18      |
| VNS2      |                         | 1 243.94| 1 263.11| 1 252.26| 4.96      |
| VNS3      |                         | 1 238.53| 1 265.28| 1 252.53| 6.47      |

Table 3. Results of experiments with dataset Mopsi-Joensuu (20 clusters, 1200 seconds, 30 attempts).

| Algorithm | Objective function value | Min     | Max     | Average | Std.dev.  |
|-----------|-------------------------|---------|---------|---------|-----------|
| j-Means   |                         | 36.565  | 37.520  | 36.730  | 0.253     |
| k-means   |                         | 47.891  | 52.759  | 50.387  | 1.359     |
| VNS1      |                         | 36.565  | 36.565  | 36.565  | 0.000     |
| VNS2      |                         | 36.565  | 36.565  | 36.565  | 0.000     |
| VNS3      |                         | 36.565  | 36.565  | 36.565  | 0.000     |

Table 4. Results with Chess dataset (30 clusters, 150 seconds, 30 attempts).

| Algorithm | Objective function value | Min       | Max       | Average  | Std.dev. |
|-----------|-------------------------|-----------|-----------|----------|---------|
| j-Means   |                         | 8 021.22  | 8 102.13  | 8 060.24| 21.05   |
| k-means   |                         | 7 989.20  | 8 038.81  | 8 019.24| 13.68   |
| VNS1      |                         | 7 960.82  | 7 978.85  | 7 967.27| 4.99    |
| VNS2      |                         | 7 959.92  | 7 989.01  | 7 974.27| 8.61    |
| VNS3      |                         | 7 998.48  | 8 007.96  | 8 003.84| 3.51    |

Table 5. Results with Chess dataset (30 clusters, 1 hour, 30 attempts).

| Algorithm | Objective function value | Min       | Max       | Average  | Std.dev. |
|-----------|-------------------------|-----------|-----------|----------|---------|
| j-Means   |                         | 7 997.43  | 8 031.05  | 8 014.72| 10.71   |
| k-means   |                         | 7 970.88  | 8 005.28  | 7 990.12| 9.31    |
Table 6. Results with Europe dataset (30 clusters, 2 hours, 30 attempts).

| Algorithm  | Min       | Max       | Average   | Std.dev. |
|------------|-----------|-----------|-----------|----------|
| j-Means    | 7.51477E+12 | 7.60536E+12 | 7.56092E+12 | 29.764E+9 |
| k-means    | 7.54811E+12 | 7.57894E+12 | 7.56331E+12 | 13.560E+9 |
| VNS1       | 7.4918E+12  | 7.49201E+12 | 7.49185E+12 | 0.0735E+9  |
| VNS2       | 7.49488E+12 | 7.52282E+12 | 7.50082E+12 | 9.989E+9   |
| VNS3       | 7.4918E+12  | 7.51326E+12 | 7.49976E+12 | 9.459E+9   |
| VNS1-RND   | 7.51477E+12 | 7.60536E+12 | 7.56092E+12 | 29.764E+9 |

Table 7. Results with BIRCH3 dataset (100 clusters, 3 hours, 30 attempts).

| Algorithm  | Min       | Max       | Average   | Std.dev. |
|------------|-----------|-----------|-----------|----------|
| j-Means    | 3.76222E+13 | 3.7965E+13 | 3.77715E+13 | 0.116211E+12 |
| k-means    | 7.92474E+13 | 8.87404E+13 | 8.31599E+13 | 3.088140E+12 |
| VNS1       | 3.72537E+13 | 3.77474E+13 | 3.74703E+13 | 0.171124E+12 |
| VNS2       | 4.21378E+13 | 6.01082E+13 | 5.18976E+13 | 7.191247E+12 |
| VNS3       | 3.72525E+13 | 3.74572E+13 | 3.73745E+13 | 0.074315E+12 |

Table 8. Results with data of non-destructive tests of integrated circuits N5503XhM1-289 (10 clusters, 30 seconds, 30 attempts).

| Algorithm  | Objective function value | Average | Std. dev. |
|------------|--------------------------|---------|-----------|
| j-Means    | 43701.45                 | 12.22   |           |
| k-means    | 43722.19                 | 9.87    |           |
| k-VNS1     | 43689.39                 | 4.89    |           |

5. Conclusion
Results of our computational experiments (tables 1-8) show that our new algorithm k-VNS allows us to obtain better values of objective function.

In the case of large-scale datasets, the comparative efficiency of new algorithm increases. For the problems of detection of homogeneous production batches of electronic components, implementation of new algorithm allows to obtain the most stable and precise results which cannot be significantly improves by other known methods in comparable time. Thus, results of detecting homogeneous groups can be verified which is important for the space industry where each operation must be well-regulated and easily verifiable.

Acknowledgements
Results were obtained in the framework of the state task No. 2.5527.2017/8.9 of the Ministry of Education and Science of the Russian Federation.
References
[1] Vernov S N 1983 Model of a near-earth space (Moscow: MGU)
[2] Pease R L, Johnston A H and Azarevich J L 1988 Radiation testing of semiconductor devices for space electronics Proc. of the IEEE 76(11) 1510-26 doi:10.1109/5.90110
[3] Kononov V K, Malinin V G, Ospishechev D A and Popov V D 1994 Culling potentially unreliable integrated circuits using a radiation-stimulating method (St. Petersburg: RNII «Elektronstandart»)
[4] Farahani R 2009 Facility location: Concepts, models, algorithms and case studies ed R Z Farahani and M Hekmatfar (Berlin Heidelberg: Springer-Verlag) p 549
[5] Steinhaus H 1956 Sur la division des corps materiels en parties Bull. Acad. Polon. Sci. III(IV) 801-4
[6] Lloyd S P 1982 Least Squares Quantization in PCM IEEE Transactions on Information Theory 28 129-37
[7] MacQueen J B 1967 Some Methods of Classification and Analysis of Multivariate Observations Proc. of the 5th Berkeley Symp. on Mathematical Statistics and Probability 1 281-97
[8] Alsabti K, Ranka S and Singh V 1998 An efficient k-means clustering algorithm Proc. of IPPS/SPDP Workshop on High Performance Data Mining
[9] Nigam K, McCallum A K, Thrun S and Mitchell T 1999 Text Classification from Labeled and Unlabeled Documents using EM ACM journal of Machine Learning-Special issue on information retrieval
[10] Kanungo T, Mount D M, Netanyahu N S, Piatko C D, Silverman R and Wu A Y 2002 An efficient k-means clustering algorithm: analysis and implementation IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7) 881-92
[11] Cheung Y M 2003 k-Means: A new generalized k-means clustering algorithm Pattern Recognition Letters 24(15) 2883-93
[12] Xiaoli C et al 2014 Optimized big data K-means clustering using Map Reduce (New York: Springer Science and Business Media)
[13] Xiong H, Wu J and Chen J 2009 K-Means Clustering Versus Validation Measures: A Data-Distribution Perspective IEEE Transactions on Systems, Man, and Cybernetics 39(2) 318-31
[14] Zhang L, Lu F, Liu A, Guo P and Liu C 2016 Journal of Science and Engineering Applications 5(1)
[15] Wang J and Su X 2011 IEEE 3rd Int. Conf. 44-46
[16] Singh R V and Bhatia M P S 2011 IEEE Int. Conf. 717-21
[17] Shi Na, Liu Xumin and Guan Yong 2010 IEEE 3d Int. Symp. 63-7
[18] Sharmila Rani D and Shenbagamuthu V T 2017 Int. J. of Innovative Research in Computer and Communication Engineering 2(1)
[19] Bhusare B B and Bansode S M 2014 Int. J. of Advanced Research in Computer Engineering & Technology (IJARCET) 3(4)
[20] Kaur K, Singh Dhaliwal D and Kumar Vohra R 2013 Int. J. of Advanced Research in Computer Engineering & Technology (IJARCET) 2(11)
[21] Wang S 2013 Int. Conf. on Mechatronic Sciences, Electric Engineering and Computer (MEC) Shenyang, China IEEE
[22] Mahmud S, Rahman M and Akhtara N 2012 7th International Conference on Electrical and Computer Engineering IEEE doi:10.1109/icece.2012.6471633
[23] Abdul Nazeer K A and Sebastian M P 2009 Proc. of the World Congress on Engineering I WCE London U K
[24] Hansen P and Mladenović N 2001 Pattern Recognition 34(2) 405-13 doi:10.1016/s0031-203(99)00216-2
[25] Stashkov D V, Orlov V I, Nasyrov I R and Kazakovtsev L A 2017 Economics and Management of Control Systems 23(1.1) 185-93
[26] Stashkov D V, Gudyma M N, Kazakovtsev L A, Rozhnov I P and Orlov V I 2017 Reshetnev
readings 21 327-8

[27] Hansen P and Mladenovic N 2005 Variable Neighborhood Search Search Methodology ed E K Bruke and G Kendall (US: Springer) pp 211-38 doi: 10.1007/0-387-28356-0_8

[28] Kazakovtsev L A, Stupina A A and Orlov V I 2014 Modification of a genetic algorithm with greedy heuristic procedure for continuous location and classification problems Sistemy upravleniya i informatsionnyye tekhnologii 2(56) 35-9