Global Optimal and Minimal Solutions to K-means Cluster Analysis

Ruming Li [1], **Xiu-Qing Li** [2], **and Guixue Wang** [3*]
[1,3]Key Laboratory of Biorheological Science and Technology (Chongqing University), Ministry of Education; Bioengineering College of Chongqing University, Chongqing, 400044, China
[2]Molecular Genetics Laboratory, Potato Research Centre, Agriculture and Agri-Food Canada, Fredericton, New Brunswick, E3B 4Z7 Canada

Abstract - *The K-means clustering method has been widely used in nonhierarchical cluster analysis of multi-dimensional data sets. Chronic problems with the K-means clustering algorithms since the 1960s have been the local minima of clustering results, computationally expensive iterations, and suboptimal solutions with large-size data sets. Their usabilities are controversial and risky. Without using traditional heuristics, we announced our milestone solutions to K-means cluster analysis in a novel global partitioning model. They were developed to overcome all of these problems and make the K-means algorithm truly optimal. These solutions innovated traditional algorithms and introduced new methods for rationally partitioning a data set from the globality and integrity of data structure and object relationships. The theories and techniques involved were experimentally proven by all successful implementations. The globally optimal results confirmed that the breakthrough was achieved in rapidly classifying any type/size of data sets into any number of disjoint clusters with a global minimum of total error sum of squares (TESS).*

Keywords: cluster analysis, K-means clustering, global optimal partitioning, clustering error and quality, global optimization and minimization, global minimum TESS.

1 Introduction

The *K*-means clustering method is a major nonhierarchical or partitional classification technique that has been most commonly used in information processing and exploratory data analysis such as statistics, informatics, phylogenetics, gene clustering, data mining, pattern or trend recognition, image segmentation, and machine learning. Particularly *K*-means clustering is more efficient in processing of a massive data than hierarchical clustering that is computationally expensive on a very large data set. *K*-means clustering is used to divide a set of objects (aka, entities, cases, observations, data points, samples, or items) into *K* subsets or clusters (aka, classes, groups or partitions). The separated clusters are disjoint and the members in a cluster are sufficiently close or similar to each other and sufficiently distant or dissimilar to non-members in other clusters [1-3]. The number *K* of such clusters can be either known a priori or pre-set by the algorithm or pre-defined by users. The clustering error is measured by a total error sum of squares (TESS) in statistics and the criterion of optimal clusterings takes the least TESS that expresses the minimized clustering error. Suppose that x is an arbitrary data point in the m-dimensional data space and let l be the number of clusters and n be the number of objects (data points) in a cluster, the least objective function TESS is defined as:

$$\text{Minimum} \sum_{k=1}^{l} \sum_{i=1}^{n} \sum_{j=1}^{m} \left(x_{kij} - \bar{x}_{kj} \right)^2$$

where $\bar{x}_{kj}$ is the j-th component of a mean vector or centroid for the k-th cluster. When the least TESS is satisfied, it translates into the accurate (most reasonable) cluster membership that is achieved for each of clusters. That is, all members within a cluster are closest or most similar to each other and most distant or dissimilar to non-members otherwise.

Traditionally, the partitional clustering method operates on the various algorithms that are unexceptionally based upon the *K*-center model or centrotype in each cluster. In the *K*-means model, the centrotype is the arithmetic mean vector or cluster centroid (barycenter). The primary *K*-means procedure starts from initial *K* seeds or cluster centers and then uses an iterative refinement heuristic with centroids. Unfortunately this algorithm terminates with a local convergence and does not definitely find the globally optimal cluster configuration corresponding to the objective function minimum. The main reasons are that it operates on the center-based model and is inevitably sensitive to the initial cluster centers aside from its local assignment of data points. The algorithm can be run multiple times to reduce these effects but there is no guarantee that it should converge to a global minimum even if a stopping criterion is met. What is worse, this would bring another issue that such algorithmic iterations cause a heavy computational load.

The *K*-means algorithm has been being improved since the year 1967 [4-6]. The significant progress was made in the early 21st century with a couple of global *K*-means algorithms being a new paradigm [7-10]. These modified algorithms introduced or adopted the typical incremental approach to clustering that dynamically adds one cluster center at a time. Although these algorithms still could not get rid of the center-based model in which issues associated with cluster centers and iterations

remained, they provided better heuristics that were approximating global optimal solutions [11-13].

For the larger number of clusters to partition and a huge-size data set, all the heuristic algorithms based on the *K*-center model would have the undesirable performance for an optimal clustering solution. Especially, the overwhelming demand for such a desired solution quality through an improved *K*-means algorithm is that a globally minimized TESS must be first of all satisfied.

Then what is the perfect way of partitioning a given data set into disjoint clusters that realizes a global optimization and ends up with the highest quality of clusterings? What is the ultimately minimum TESS reached from such *K*-means clusterings no matter what size a data set has? Our methodology and algorithm devised took care of these issues and worked around all these problems from the perspectives of data integrity, global partitioning, implementation efficiency and outcome robustness. They changed conventional thinking and abandoned the *K*-center model [14]. The breakthrough was obtained in that it is a multidisciplinary solution instead of the pure mathematical development. This particularly involves application of computational intelligence, informatics, combinatorics, logic, operations research, and statistics to the algorithmic or programmatic solutions. In this study, all underlying concepts and theories were addressed and experimentally proven. The objective function TESS that the algorithm should minimize was implemented by our software ParCluster (Partition Cluster for short). For convenience of study, the above notation used is effective thereafter. Also, out of consideration for space and simplicity, we have to use small-sized yet poorly-clusterable sample data for demonstration purposes, but their principles are universally extensible.

2 Methods

2.1 Data clustering error definitions

In *K*-means clustering, the relative importance or individual weight of the clustering error for a single member in a cluster can be quantified and expressed as an error sum of squares (SESS) in statistics. It is the summation of squared differences between each variable and its mean in the centroid for an *m*-dimensional data point; that is,

$$\text{SESS} = \sum_{j=1}^{m} \left(x_j - \bar{x}_j \right)^2$$

where the square-root of SESS is the Euclidean distance. The greater SESS a single member has, the farther it is away from its centroid and also the more distant or dissimilar to other members in a cluster (i.e., the weaker membership). Likewise, for all members in a cluster, the aggregated SESSs are termed as CESS with respect to the clustering error resulted from the entire cluster. That is,

$$\text{CESS} = \sum_{i=1}^{n} \sum_{j=1}^{m} \left(x_{ij} - \bar{x}_j \right)^2 .$$

In the special case of a singleton cluster that contains one member only, CESS equals zero.

2.2 Data extraction techniques and criteria

There are two techniques, one is SESS-based and the other is CESS-based, and three criteria to extract data and assign it to another cluster. The first data extraction criterion is defined as: If a member object in a cluster has a larger SESS than it does when placed in another cluster, it should be grabbed from its cluster and allocated to that cluster. The second data extraction criterion is defined as: Compute the difference between the SESS of each member object in a cluster and the resulting SESS when it is placed in another cluster. If these differences are positive (i.e., if the first criterion is met), choose the largest one and its corresponding object should be grabbed and allocated as such. The third data extraction criterion is defined as: If a member object in a cluster is placed in another cluster, the sum of resulting CESSs for the two current clusters is smaller than it is for the two previous clusters. Then that member object should be grabbed and allocated as such.

2.3 Data partitioning theory

For a given objects (data points) in the *m*-dimensional data space or Euclidean space, they have a data structure consisting of integral parts of all members. For this reason, all of their data points cannot be individually, separately or locally treated and manipulated in terms of global optimization. That is, each partitioning of data points must be performed under a global setting where each point dynamically coordinates with any other points [15], which collectively contributes to minimiza- tion of the total clustering error. Since the classic *K*-center model starts with *K*-partitions and fails to provide a data-dependent setting, it is trapped in a local setting from the very beginning and never bails out. Instead we introduce a novel, unbiased, data-driven global partitioning model for *K*-means clustering.

In our new model, each partitioning of data points is performed with all other points being taken into account. It is a straightforward classification process without using an iterative refinement heuristic with centroids. What it does is starting from a bipartition with a minimized TESS. Then further partition one of clusters, which has the largest SESS, into a tripartition and so forth until a specified number of clusters are partitioned. For each cluster partitioned throughout the clustering, its CESS is guaranteed to be a minimum. Thereby, each level of clusterings is guaranteed to have a minimized TESS, so is the TESS when a final level of clusterings is done. The rationale is that theoretically those clusters having established, firm, or close memberships are not necessarily re-partitioned and only the cluster having a global maximum SESS needs to do so. It is termed "partitional cluster". All partitions and assignments of data points follow a global maximal difference law throughout. This law requires that a maximum "ESS" due to the maximal difference always take precedence over others. However, when a cluster having the second-to-maximum SESS has the larger CESS than does the one having the maximum SESS, both need to be minimized in TESS. This protects the result from being

non-minimized when their SESSs differ so little. Specially, if there are clusters having tie maximum SESSs, all of them are subject to TESS minimization. In addition, one or more data points from other clusters may be dynamically extracted to join the new cluster (reshuffled) and orchestrated to lead to a global optimization. As the number of clusters increases, more of the closest members (patterns) get isolated (recognized) from clusters. Then the clustering process becomes increasingly simpler and easier, no matter what size a data set may have [11]. The general data flow directions and order in the partitioning process were illustrated in Fig. 1.

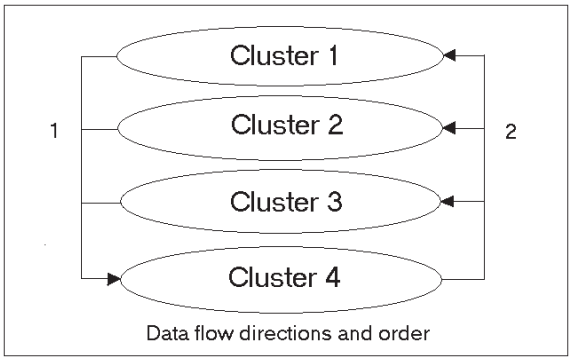

Fig. 1. A general view of the partitioning process from the existing Clusters 1, 2, and 3 to the newly-generated Cluster 4. The data flow among them starts from 1 and ends up with 2.

The principal procedure and algorithms are outlined as belows.

1. Bipartition all data points into two clusters starting with a maximum SESS.
2. Partition one of clusters into two sub-clusters with a global maximum SESS.
3. Re-partition that cluster as SESS-based and CESS-weighted tripartitions.
4. Re-partition that cluster as tri-partitions based on global object relationships.
5. Re-partition that cluster as tri-partitions based on the dichotomous evaluations.
6. Post-partition all such-obtained clusterings, each being globally optimized.
7. Keep track of all TESSs yielded and take a minimum as the convergent end.
8. Repeat Steps 2-7 for a specified number of clusters until they are partitioned.

Step 1 produces the primitive level of bipartition. Step 2 is the partition using a global maximum SESS from a partitional cluster as a seed. Step 3 refers to the partition using a global maximum SESS+CESS as the criterion of a partitional cluster; this step is skipped if Step 2 uses the same partitional cluster. Step 4 refers to the partition using deviators from a partitional cluster that have relationships with objects in other clusters. Step 5 refers to the partition taking a maximum SESS from a maximum SESS+CESS partitional cluster and taking a maximum SESS unbiasedly from any other clusters. This two-way value taking is called dichotomous evaluation. Step 6 refers to the partition using a global TESS minimizer after all the above tasks are done, which is a way to produce a guaranteed global minimum. Step 7 records all TESSs of optimized clusterings and takes a minimum as the optimum reached for that level. Step 8 iterates the above procedure till the given partitions are made.

The idea behind the theory is to follow the global maximal difference law and to gain an equilibrium at which each cluster gets the least CESS and forms a stable membership from an initial level of clusterings. Then break the equilibrium, regain it, break it again, and so on until the objective function TESS in a final level of *K*-means clusterings is minimized. The bottom line is that a hard-to-reach global optimum can be achieved by a guaranteed optimization from a primitive level of clusterings up to the last one. That is, break the harder big problem into the smaller one that is easier to optimize. Each level of globally optimized results makes the last global minimum reached.

2.4 The context of pivotal-point theorem

For a given *m*-dimensional data set, assuming that it is partitioned into multiple clusters, there may be some data points that cannot be allocated normally from cluster to cluster based on the data extraction criteria. This is caused by the properties of a couple of data points having very close or similar component (i.e., variable) values as well as complementary values. The extreme case is that they have all the equal component values across *m*-dimensional variables. That is, some data points may be identical or duplicate, especially in low dimensional data sets. Complementary values are those component values that have small-for-large values for one variable and large-for-small values for another variable. They are exemplified as follows:

	Variable 1	Variable 2
Data point 1:	3.0	4.0
Data point 2:	4.0	3.0
Mean vector:	3.5	3.5

where Point 1 is smaller than Point 2 for Variable 1 and larger than Point 2 for Variable 2 in a *2*-dimensional data set and hence the net effect is a decrease in difference between the two points, as their contributions to the mean vector have the same value (3.5). This will make such complementary values behave like close-component data points. All these properties make it difficult to partition such data points into a common cluster while they are separate or into different clusters while they have tight bonds within a cluster. It should be noted that these properties are also partially responsible for early clustering terminations against stopping criteria and locally converged minima. To solve the problems with data points being of such properties, the solution is to follow the pivotal-point theorem.

Pivotal-point theorem: The data points possessing all close, equal or complementary components in *K*-means clustering are defined as and behave like pivotal points. By playing a pivotal

role, they can preclude data points from being further partitioned based on the data extraction criteria, or terminating data point allocation among clusters hinges on them. When one (or more) of these pivotal points is (are) assumed (forced) to be placed in a target cluster, the further partitioning of data points may be allowed to proceed. This can finally lead to a global optimization of *K*-means clustering and its TESS minimization if a given condition is met. The condition met requires that, based on the second data extraction criterion, a data point with the largest negative difference, instead of the largest positive one, be chosen as the pivotal point to extract. After pivotal point(s) is (are) extracted and assumed to be placed in a target cluster, *K*-means clustering will tend to global optimization and TESS minimization if and only if either of the following scenarios applies:

1) For one pivotal point, it will cause an increase in TESS first but its extraction and placement may induce other point(s) to become extractable, which finally, collectively results in the less overall TESS.

2) For multiple pivotal points, their successive extractions and placements in a target cluster yield the smaller and smaller absolute value of the negative difference until it turns to positive one. Also this may induce other points to be extractable and results in a net decrease in TESS.

There is a constraint that, when such an absolute value becomes larger, the extraction of pivotal points stops. This makes the algorithm non-greedy but enable the global optimization and minimization. When pivotal points become extractable, we say that further partitioning data points is allowed to proceed; that is also to say, a balance due to local optimality is broken or local "convergency".is passed or skipped. This is the most significant contribution of the theorem to realizing a global optimality for the *K*-means solution. Particularly, to handle equal-component pivotal points (identical objects), they are grouped, grabbed, and moved as one unit across clusters to ensure inseparability, integrity, and efficiency throughout the clustering.

Again, the concept underlying the theorem is following the gain-and-break law of the aforementioned cluster membership equilibrium. Here the equilibrium is the early stopped partitioning process or local or near-optimal "convergence" of TESS.

2.5 Global maximal SESS method

If a member object has the global maximal SESS in a cluster, it contributes the most to the total clustering error (TESS) and also results in the maximal CESS for that cluster. Thus, if that incompatible object is eliminated from that cluster, it must be the right candidate member to generate a new cluster for a bipartition. That is, take that object as an initial seed and use it to start partitioning of data points.

2.6 Global object relationship method

In addition to the above core techniques used in *K*-means clustering, another key technique is required to continuously isolate the closest members from a cluster. Generally, the members joined in a cluster are arranged in the order of inherent relationship (closeness or similarity) [12,14]. The member(s) last joined in a cluster may have the weakest membership but have a global object relationship, which is (are) deviator(s) from that cluster. It is (they are) the initiator(s) for generating a new cluster.

For a tripartition or a higher level of clusterings, use the global object relationship method or the CESS-weighted global maximal SESS method. For a guaranteed global minimal TESS, the deviators are calculated from their memberships or exhaustively searched within a partitional cluster.

3 Results and discussion

The clustering criterion to judge if an ultimately minimal TESS is "converged" from our *K*-means clustering adopts the outcome of executing a brute-force algorithm that computes the TESSs for all possible combinations of data points for a given number of clusters. That is, if one of all the possible TESSs computed is a minimum, it is taken as a reference standard or benchmark for testing if our minimal TESS is the ultimate minimum for a global optimal solution. Moreover, all the real or simulation data sets used for the study were noisier, tougher, and more challenging than general data sets, as they contained hard-to-split pivotal points. This is especially true for a whole- number and/or extremely low dimensional data set. With such ill-clusterable data, they were nevertheless computationally harder to cluster by the K-means algorithms that suffered from apparent errors despite their small sizes as shown in this study. In other words, our successful acquisition of the maximum reduction of the clustering error is effected by all the devised computational laws and global minimizer theories. Through them, all data clustering should result in global minima, which are not necessarily data size-related.

Table 1. A *four*-dimensional (4 variables) real data set of 17 real-number objects (data points) that contains pivotal and close membership points* and has a moderately low dimensionality.

Objects	Variable 1	Variable 2	Variable 3	Variable 4
lau	0.38	626.5	601.3	605.3
ccu	0.18	654.0	647.1	641.8
bhu	0.07	677.2	676.5	670.5
ing	0.09	639.9	640.3	636.0
com	0.19	614.7	617.3	606.2
smm	0.12	670.2	666.0	659.3
bur	0.20	651.1	645.2	643.4
gln	0.41	645.4	645.8	644.8
pvu	0.07	683.5	682.9	674.3
sgu	0.39	648.6	647.8	643.1
abc	0.21	650.4	650.8	643.9
pas	0.24	637.0	636.9	626.5
lan	0.09	641.1	628.8	629.4
plm	0.12	638.0	627.7	628.6
tor	0.11	661.4	659.0	651.8
dow	0.22	646.4	646.2	647.0
lbu	0.33	634.1	632.0	627.8

*The values for Variables 2, 3 and 4 are in close proximity across both objects and variables, which are hard to split from among their data points as they have a poor clusterability for such a data set.

Table 2. A demonstration of the global optimal K-means clustering technique following the pivotal-point theorem and using the *4*-dimensional real data set (*size = 17*) in Table 1 and the global maximum SESS method in our natural, data-driven global partition model.

Bipartition	The first level of clusterings	Data point partitioning	To proceed[¶]	TESS
Cluster 1 = Cluster 2 =	1,2,3,4,5,6,7,8,10,11,12,13,14,15,16,17 9 ← It has the largest SESS and is a solely unbiased seed.	Point 9 is grabbed and used to build Cluster 2.	Allowed (4165.18)	13043
Cluster 1 = Cluster 2 =	1,2,4,5,6,7,8,10,11,12,13,14,15,16,17 9,3	Point 3 is grabbed and allocated to Cluster 2.	Allowed (3413.59)	9630
Cluster 1 = Cluster 2 =	1,2,4,5,7,8,10,11,12,13,14,15,16,17 9,3,6	Point 6 is grabbed and allocated to Cluster 2.	Allowed (1784.07)	7846
Cluster 1 = Cluster 2 =	1,**2**,4,5,7,8,**10**,**11**,12,13,14,16,17 9,3,6,**15**	Point 15 is grabbed and allocated to Cluster 2.	Stopped (665.09)	7181*
Cluster 1 = Cluster 2 =	1,2,4,5,7,8,10,12,13,14,16,17 9,3,6,15,**11**	Point 11 is assumed to be placed in Cluster 2.	Assumed (-580 .78)	7761
Cluster 1 = Cluster 2 =	1,4,5,7,8,10,12,13,14,16,17 9,3,6,15,**11**,**2**	Point 2 is assumed to be placed in Cluster 2.	Assumed (-280.35)	8042
Cluster 1 = Cluster 2 =	1,4,5,7,8,12,13,14,16,17 9,3,6,15,**11**,**2**,**10**	Point 10 is assumed to be placed in Cluster 2.	Assumed (-139.77)	8181
Cluster 1 = Cluster 2 =	1,4,5,7,8,12,13,14,17 9,3,6,15,11,2,10,16	Point 16 is grabbed and allocated to Cluster 2.	Allowed (131.21[$])	8050
Cluster 1 = Cluster 2 =	1,4,5,8,12,13,14,17 9,3,6,15,11,2,10,16,7	Point 7 is grabbed and allocated to Cluster 2.	Allowed (387.62)	7663
Cluster 1 = Cluster 2 =	1,4,5,12,13,14,17 9,3,6,15,11,2,10,16,7,8 ←The last joined member	Point 8 is grabbed and allocated to Cluster 2.	Stopped (497.45)[§]	7165[#]

*The first reached TESS (7181) is a local minimum when a regular stopping criterion is early met.
[#]The second reached TESS (7165) is the global minimum when the pivotal-point theorem applies.
[$]The smaller and smaller absolute value of the negative difference until turn to positive one (131.21).
[§]The values in the brackets are the amounts by which the clustering error or TESS is reduced.
[¶]To proceed with data point partitioning according to the data extraction criteria on the CESS basis.

Table 3. A demonstration of the global optimal K-means clustering technique following the pivotal-point theorem and using the *4*-dimensional real data set (*size = 17*) in Table 1 and the global object relationship method as well as taking the data extraction criteria to proceed with data point partitioning on the CESS basis by which all three clusters are ingeniously, properly and robustly generated.

Tripartition	The second level of clusterings	Data point partitioning	To proceed	TESS
Cluster 1 = Cluster 2 = Cluster 3 =	1,4,5,12,13,14,17 9,3,6,15,11,2,10,16,7 8 ← It is a deviator, and a natural, data-driven initiator.	Point 8 is grabbed and used to generate Cluster 3.	Allowed	6775
Cluster 1 = Cluster 2 = Cluster 3 =	1,5,12,13,14,17 9,3,6,15,11,2,10,16,7 8,4	Point 4 is grabbed and allocated to Cluster 3.	Allowed	6359
Cluster 1 = Cluster 2 = Cluster 3 =	1,5,12,13,14,17 9,3,6,15,11,2,10,7 8,4,16	Point 16 is grabbed and allocated to Cluster 3.	Allowed	5993
Cluster 1 = Cluster 2 = Cluster 3 =	1,5,12,13,14,17 9,3,6,15,11,2,7 8,4,16,10	Point 10 is grabbed and allocated to Cluster 3.	Allowed	5535
Cluster 1 = Cluster 2 = Cluster 3 =	1,5,12,13,14,17 9,3,6,15,11,2 8,4,16,10,7	Point 7 is grabbed and allocated to Cluster 3.	Allowed	4921
Cluster 1 = Cluster 2 = Cluster 3 =	1,5,12,13,14,17 9,3,6,15,11 8,4,16,10,7,2	Point 2 is grabbed and allocated to Cluster 3.	Allowed	4192
Cluster 1 = Cluster 2 = Cluster 3 =	1,5,12,13,14,17 9,3,6,15 8,4,16,10,7,2,11	Point 11 is grabbed and allocated to Cluster 3.	Allowed	3162
Cluster 1 = Cluster 2 = Cluster 3 =	1,5,12,13,14,17 9,3,6 8,4,16,10,7,2,11,15	Point 15 is grabbed and allocated to Cluster 3.	Stopped	2959*
Cluster 1 = Cluster 2 = Cluster 3 =	1,5,14 9,3,6 8,4,16,10,7,2,11,15,**12**,**13**,**17**	Points 12, 13, and 17 are assumed to be successively placed in Cluster 3.[$]	Assumed	3417
Cluster 1 = Cluster 2 = Cluster 3 =	1,5 9,3,6 8,4,16,10,7,2,11,15,12,13,17,14	Point 14 is grabbed and allocated to Cluster 3.	Allowed	3048

Cluster 1 =	1,5			
Cluster 2 =	9,3,6,15	Point 15 is grabbed and allocated back to Cluster 2.	Stopped	2858[#]
Cluster 3 =	8,4,16,10,7,2,11,12,13,17,14			

[*]The first reached TESS (2959) is a local minimum when the regular stopping criterion is early met.
[#]The second reached TESS (2858) (same result as the brute-force) is the global minimum when the pivotal-point theorem applies.
[$]Like 11,2,10 in Table 2, they are assumed to be consecutively placed in Cluster 3 without showcasing the process again to save space.

Table 4. A larger-size *2*-dimensional (V) sample data set of 50 whole-number objects (O) (data points) that contains some hard-to-partition pivotal points and has an extremely low dimensionality, and the SESS of each object was computed.

O#	V1	V2	SESS	O#	V1	V2	SESS
1	3	6	0.08	*26*	4	7	2.08
2	4	9	10.88	*27*	5	0	36.88
3	1	6	4.88	*28*	4	4	3.88
4	0	5	10.88	*29*	2	9	11.68
5	2	7	2.88	*30*	0	8	15.08
6	3	8	4.88	*31*	1	2	19.28
7	7	9	24.68	*32*	2	3	9.28
8	5	3	11.08	*33*	3	4	3.28
9	6	4	11.08	*34*	4	5	1.28
10	2	8	6.28	*35*	5	6	3.28
11	3	7	1.48	*36*	0	1	33.28
12	3	5	0.68	*37*	1	5	5.48
13	6	6	7.88	*38*	2	4	4.68
14	4	5	1.28	*39*	3	3	7.88
15	3	9	10.28	*40*	4	2	15.08
16	1	4	8.08	*41*	3	6	0.08
17	2	3	9.28	*42*	4	6	0.68
18	0	6	10.28	*43*	2	5	2.08
19	5	9	13.48	*44*	2	7	2.88
20	3	8	4.88	*45*	4	9	10.88
21	1	7	6.28	***46***	6	0	**41.48**
22	3	9	10.28	*47*	7	10	32.08
23	2	6	1.48	*48*	4	9	10.88
24	5	6	3.28	*49*	6	8	12.68
25	3	4	3.28	*50*	5	8	8.08

Max O# = 46 Max SESS = 41.48 TESS = 473.99

***Proof of pivotal-point theorem*:** A real data set with the typical characteristics of pivotal points was given in Table 1 and the pivotal-point theorem was experimentally verified and justified in Table 2. From there, the data points 11, 2, and 10 were identified as pivotal points (bold ones) that were responsible for early meeting a stopping criterion and locally reached minimum (7181). The reason is that there are close relationships between the data points of 15 and 11, 11 and 2, and 2 and 10; when any couple of such points gets separated, the two clusters to which they belong reach an equilibrium or "dead point". That is, both the points make both the clusters less differential such that their abilities to grab a point from the opposite are balanced. When this equilibrium is broken by assuming a pivotal point to be placed in a target cluster in a given condition, it leads to "convergence" at the global minimum (7165) (same result as the brute-force). Therefore, the highest quality of clusterings is attained with the correct cluster membership {1,4,5,12,13,14,17} and {9,3,6,15,11,2,10, 16,7,8} rather than with {1,2,4,5,7,8,10, 11,12,13,14,16,17} and {9,3,6,15}.

The central idea behind the theorem is making the "less differential" clusters differential. Note that the workings from this proof are universally extensible and applicable to big data as well with no exceptions. This theorem also turns out to be one of our most important findings and data partitioning theories that make a global optimal *K*-means solution possible.

Table 3 exhibited the stepwise process of the second level of clusterings as the number of clusters to be partitioned rose to 3. The partitioning processes of the higher levels of clusterings are analogous to this. When a data set size grows, it works the same without any less optimization and minimization as one may think. This is determined by the globally interactive property and integrity of data structure and object relationships (Tables 4 and 5). Take a notice that the initial cluster memberships used for the tripartition was inherited from the result from the bipartition in Table 2. It should be pointed out that not all the partitioning processes have to exploit the pivotal-point theorem; it is employed wherever necessary. As shown in Table 3, two groups (patterns) of the closest members {9,3,6,15} and {1,5} got isolated (recognized) from clusters, which would make the next-level clustering process much simpler and easier.

Table 5. A demonstration of the properness and robustness of a member object with the global maximum SESS when used as an initial seed to start partitioning of data points as compared to any other objects with the smaller SESS than it.

O#	SESS	TESS	$TESS^2$	O#	SESS	TESS	$TESS^2$
46	**41.48**	474	**431.67**	*7*	24.68	474	448.82
27	36.88	474	436.37	*31*	19.28	474	454.33
36	33.28	474	440.04	*30*	15.08	474	458.61
47	32.08	474	441.27	*40*	15.08	474	458.61

Note: $TESS^2$ is the TESS resulted from exclusion of an object from the initial cluster {1,2,3, • • •,50}, which is inversely proportional to its parental TESS.

As shown in Table 4, the maximum SESS (41.48) of the initial cluster was computable in terms of statistical error that revealed the identity of a member object responsible for the maximum error contribution. As a result of a collection of data points and their integral memberships, this maximum error component is always identifiable whatever a data set size would be. And it is always effective, proper, and robust for that to be used for the maximum reduction of the clustering error when eliminated. In Table 5, only Object 46 with the global maximum SESS (41.48) resulted in the maximum reduction of TESS brought down from 474 to 431.67 when taken off. This was effected by following the global maximal difference law everywhere. Here we only take the maximal difference (41.48) from among all the SESSs, which always brings in TESS minimization whose functionality is unrelated to a data size.

Table 6. A demonstration of the properness and robustness of either the global maximal SESS object or deviators with the global object relationship when used to start partitioning of data points ($CESS^2$ is the reduced CESS).

K	Partitional Cluster	Partition Seeds	CESS	$CESS^2$
2	Initial cluster	46	473.99	431.67
3	Cluster 2	46 or 36,18,4	138.35	80.69
4	Cluster 3	46 or 36,31	80.69	43.09
5	Cluster 3	36 or 14,4,28	48.19	34.27
---	------	------	------	------

Table 7. A *2*-dimensional sample data set of 20 whole-number objects (O) (data points) that contains some hard-to-partition pivotal points* and has an extremely low dimensionality.

O#	V1	V2	O#	V1	V2	O#	V1	V2	O#	V1	V2
1	3	6	*6*	**3**	**8**	*11*	3	7	*16*	1	4
2	4	9	*7*	7	9	*12*	**3**	**5**	*17*	2	3
3	1	6	*8*	**5**	**3**	*13*	6	6	*18*	0	6
4	0	5	*9*	6	4	*14*	4	5	*19*	5	9
5	2	7	*10*	2	8	*15*	3	9	*20*	**3**	**8**

*The bold values are either identical or complementary components and some of the other data points are very close or similar to each other, thus being tougher than real data sets in the partitioning.

Table 8. A comparison of the TESSs reached from the *K*-means clustering of the *2*-dimensional sample data set (*size = 20*) in Table 7 using the brute-force benchmark ($BFB^{\$}$), our global optimal parti- tioning (GOP) and the software $SPSS^{\#}$ solutions.

K	BFB	GOP	SPSS	*K*	BFB	GOP	SPSS
2	86.20	86.20	86.53	*11*	5.00	5.00	6.24
3	50.62	50.62	52.55	*12*	4.00	4.00	4.34
4	33.23	33.23	33.23	*13*	3.16	3.16	3.50
5	24.66	24.66	26.13	*14*	2.50	2.50	3.00
6	17.83	17.83	24.40	*15*	2.00	2.00	2.16
7	13.40	13.40	17.13	*16*	1.50	1.50	1.50
8	9.06	9.06	9.06	*17*	1.00	1.00	1.00
9	7.16	7.16	8.07	*18*	0.50	0.50	0.50
10	6.00	6.00	7.24	*19**	0.00	0.00	0.00

*The data set is supposed to be partitioned into at most 19 clusters because of the two identical objects (data points) that should go in one cluster due to their zero difference.

#Taken from the statistic software SPSS output (using Sum of Error Mean Squares × d.f. and membership information). SPSS implements the typically iterative *K*-means clustering algorithm.

$BFB is used as the compelling evidence of the global minima only, but not as a practical solution, as getting it is prohibitively time-consuming (by the day, week or longer run time, depending on the number of clusters and data size).

Likewise, the deviators are computable for all higher levels of clusterings (K>2). They are available in a global, natural, unbiased, and data-driven setting without artificial operations. That is, they are produced due to the context where they do not belong to any existing clusters, thereby being eliminated and automatically classified as a new sub-cluster. Table 6 gave the 5 top levels of clusterings and their CESSs had maximal amounts of reduction. When all of these workings are fulfilled, a correct new cluster is set up and each level of clusterings has been optimized.

Table 9. A 3-dimensional real-world data set (*size = 16*)*

O#	V1	V2	V3	O#	V1	V2	V3
1	50	50	9	*9*	40	40	5
2	28	9	4	*10*	50	50	9
3	17	15	3	*11*	50	50	5
4	25	40	5	*12*	50	50	9
5	28	40	2	*13*	40	40	9
6	50	50	1	*14*	40	32	17
7	50	40	9	*15*	50	50	9
8	50	40	9	*16*	50	50	1

*The data set contains multiple hard-to-partition repeated data points.

Table 10. A comparison of the TESSs reached from the *K*-means clustering of the *3*-dimensional real-world data set (*size = 16*) in Table 9 using the brute-force benchmark (BFB), our global optimal partitioning (GOP) and the SPSS* solutions.

K	BFB	GOP	SPSS	*K*	BFB	GOP	SPSS
2	1759.33	1759.33	1790.85	*7*	103.85	103.85	103.86
3	743.71	743.71	959.33	*8*	27.66	27.66	29.80
4	422.85	422.85	460.45	*9*	17.00	17.00	18.67
5	286.85	286.85	381.43	*10*	8.00	8.00	10.67
6	182.85	182.85	272.12	*11*	0.00	0.00	0.00

*Taken from the statistical software SPSS output TESS (using Sum of Error Mean Squares × d.f.).

Table 11. A comparison of the TESSs reached from the *K*-means clustering of the *2*-dimensional sample data set (*size = 50*) in Table 4 using the brute-force benchmark (BFB), our global optimal parti- tioning (GOP) and the software SPSS.

K	B&G*	SPSS	*K*	B&G	SPSS	*K*	B&G	SPSS
2	258.644	264.912	*15*	21.249	24.290	*28*	6.499	7.172
3	181.576	181.576	*16*	18.649	20.502	*29*	5.833	6.657
4	140.777	150.512	*17*	16.983	19.338	*30*	5.166	5.680
5	104.386	116.280	*18*	15.483	18.336	*31*	4.500	5.149
6	80.477	81.840	*19*	14.249	17.329	*32*	4.000	4.662
7	67.478	74.261	*20*	13.083	15.900	*33*	3.500	3.995
8	56.477	63.714	*21*	12.083	14.500	*34*	3.000	3.328
9	47.482	51.332	*22*	11.083	11.592	*35*	2.500	2.670
10	40.938	46.520	*23*	10.083	11.718	*36*	2.000	2.156
11	35.714	41.886	*24*	9.250	11.440	*37*	1.500	1.664
12	30.599	35.074	*25*	8.499	9.825	*38*	1.000	1.176
13	27.199	28.897	*26*	7.833	8.160	*39*	0.500	0.500
14	24.166	25.848	*27*	7.166	7.498	*40*	0.000	0.000

*Since BFB equals GOP, we use B&G to stand for their shared values.

The computational results from all the numerical experiments were correspondingly equal between the brute-force and our algorithm. The sample data sets and their globally reached or "converged" results were demonstrated in Tables 7, 8, 9, 10, and 11, respectively.

Table 12. A comparison of the globally optimal values reached by the *K*-means clustering based on the global optimal partitioning (GOP) technology and those obtained from the best-known global or near global solutions* (listed in ascending order of magnitude for the number *K* of clusters partitioned).

Iris Plant			Heart Disease		
K	Known optima	GOP *K*-means	*K*	Known optima	GOP *K*-means
2	152.348	152.3479517603579	***2***	**598900**	598939.9625573959
3	78.851	78.85144142614601	***5***	327970	327542.51402046264
4	57.228	57.228473214285714	***10***	202220	200302.90233807705
5	46.446	46.44618205128205	***15***	147710	146988.31833135852
6	39.040	39.03998724608724	***20***	117780	116877.55985613653
7	34.298	34.298229665071766	***25***	102130	98512.82346652425
8	29.989	29.988943950786055	***30***	88795	85564.07854828662
9	27.786	27.786092417308087	***40***	68645	66402.38570451768
10	25.834	25.834054819972504	***50***	55894	54092.01516688864
Liver Disorders			Ionosphere		
K	Known optima	GOP *K*-means	*K*	Known optima	GOP *K*-means
2	423980	423980.8837969465	***2***	2419.4	2419.364807189691
5	218260	218255.95536164998	***5***	1891.5	1889.7165311526685
10	127680	127416.55683351347	***10***	1569.4	1550.0535842095453
15	97474	96756.18094954843	***15***	1401.4	1355.1803607937275
20	81820	80044.73383914215	***20***	1271.4	1213.6066254685106
25	70419	67845.66696431948	***25***	1148.6	1095.1795601937533
30	61143	58967.039226129564	***30***	1046.9	990.6881120696817
40	47832	46582.47583897826	***40***	856.58	815.3967562367158
50	39581	37530.131872894075	***50***	702.58	671.9685317355181

*They are the known theoretical values derived from the cluster function [8,9]. For Iris plant data set, its best-known optima are also the optimal values derived from the cluster function by the known global minimizer [8,9].

For the capability of our global optimal partitioning (GOP) technology that can be extended (generalized) to the scenarios of big data, four well-known benchmark data sets from the UCI machine learning repository (http://archive.ics.uci.edu/ml/machine-learning-databases/) were used to verify the power of the GOP-based *K*-means solutions. In Table 12, Iris data set has 150 instances (objects) of 4 attributes (variables) each. Heart disease data set has 297 instances of the first 13 attributes each. Liver disorders data set has 345 instances of the first 6 attributes each. Ionosphere data set has 351 instances of the first 34 attributes each. Note that Heart disease data set was obtained by removing those instances with a missing attribute from Cleveland raw data. All the above GOP-based optimal values were output from our software ParCluster v.2.0 and were much lower than or equal to the known optima. There is one slight discrepancy in the bold known optima of Heart disease data set as compared to ours. This is beyond explanation since these known optima are theoretical values and may not be the exact ones. To our best knowledge, all reported clustering errors from the multi-start *K*-means, global *K*-means, fast global *K*-means, modified global *K*-means, and efficient global *K*-means algorithms are greater than or far from even the above known optima. Practically, the clustering results from the GOP *K*-means should be treated as real, exact global optima against the theoretical criterion.

The optimized and TESS-minimized clustering result is of great importance in that its correct cluster memberships formed provide accurate information whereas the local- or near-optimal ones give an artifact or reflect wrong information. A cluster membership is very sensitive to clustering errors, which will make a quite difference in combination of objects. This is critical as the misrepresented object relationships could lead to serious consequences and hence is risky.

In particular, this global optimization and minimization would make genes accurately clustered in gene clustering from next-generation sequencing data. This also requires that all the alignment data with base or amino acid sequences be transformed into a similarity data for each of clusterings on which *K*-means clustering is based. A paradigm of its internal data structure is given in Table 13.

Table 13. An alignment data set with genes (G) and bases (B)

O#	B	B	B	B	B	B	B	B	B	B	B	B	---
G1	A	T	G	T	A	C	A	A	A	T	C	A	---
G2	A	T	G	A	A	C	T	G	C	A	G	C	---
G3	A	T	G	A	T	T	A	T	C	A	A	T	---
---	---	---	---	---	---	---	---	---	---	---	---	---	---

Some of the clustering results in the numerical experiments are illustrated in Fig. 2, Fig. 3, and Fig. 4, which are output from our software ParCluster v.2.0.

4 Conclusions and future work

Our global partitioning model underlying the global optimization and exact minimization algorithm provides a milestone approach to true solutions for *K*-means clustering. All computational results in terms of TESS from our algorithmic breakthrough turned out to be the global optima, namely the global minima. We managed to test that there must not be the smaller TESS than what is called minimum not only by exhaustive search but also by the integrated global minimizer theories (separate publication). Our approach proved to produce the lowest value as compared with any other algorithms or software and this difference tended to be greater as the number of clusters and data size became larger (as shown in Tables 8, 10, 11 and 12). Our GOP algorithm also ends up with the unique result or a 100% reproducible cluster membership no matter how many times it operates. These global minimum solutions make it realistic for the *K*-means clustering technique to be reliably and robustly applicable to information processing and data analysis. This gives the confidence that the highest clustering quality or accurate (the most reasonable) cluster membership is achieved for each of clusters. It means using this GOP technology will

make *K*-means clustering results free of risk and no longer controversial. Besides the least TESS that indicates the minimized clustering error, our algorithm has no size issue of a huge data set although it takes more run time with the complexity $O(kn^2m)$. And it also has no initial cluster centers and computational expenses resulted from the iterative two-step refinements in traditional and heuristic solutions. With our proven and mature techniques, the human dream comes true that one is able to classify any type/size of data sets into any number of disjoint clusters with a global minimum TESS. Preferably with the rationale behind GOP, data clustering with the arbitrary number *K* of clusters is capable of producing all the clustering results from *2* up to *K* for a one-time computation. Especially using its resumed clustering feature can inherit previous results and continue with partitioning to the next number of clusters *K* without having to start over from *K* = *2*. It also works well for clusters of arbitrary shape (e.g., non-convex type) and any size, and never gets empty clusters; noisy data and outliers can be isolated from the clustering as soon as possible, thereby eliminating any influence on the final clustering quality [16].

These desired properties/functionalities of nonhierarchical or partitional clustering technique have been expected and pursued since the year 1957 or 1967 or the earliest 1955. The ground-breaking approach and its innovative algorithm we developed put an end to this period of history lacking a global optimal and minimal solution to *K*-means cluster analysis. The Java-based software ParCluster v.2.0 built on the GOP *K*- means technology is available upon request (rli@alumni.lsu.edu) or from some web sites. Please refer to the supplementary material for more details and the pseudocode–based algorithm will be separately published for space reasons.

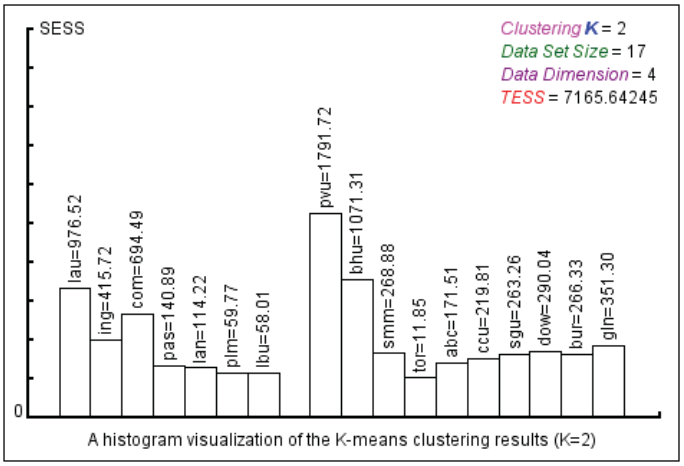

Fig. 2. A graphic view of the clustering result of bipartition (*K*=2) in Table 1 where the highest column of histogram represents the object (pvu) having the global maximum SESS in Cluster 2 (right). The object having the second-to-maximum SESS is bhu. Both obviously contribute the most error to the CESS of Cluster 2 and also to the TESS of all Clusters, making Cluster 2 a "partitional cluster" for the next-level clustering. Furthermore, those members having the similar heights of columns formed and showed closer or compatible relationships.

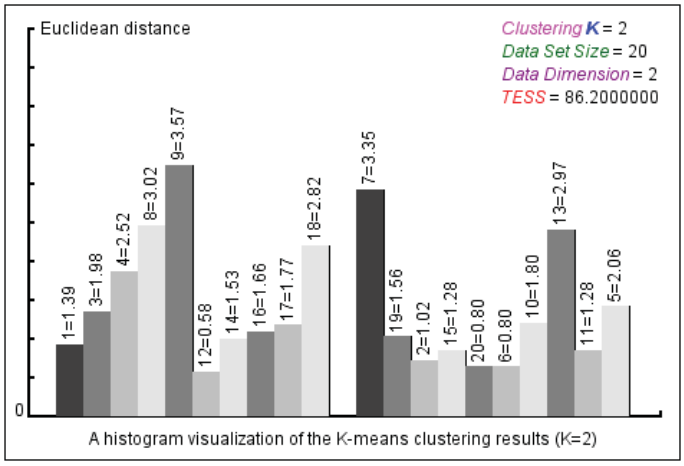

Fig. 3. The Euclidean distances about *K* cluster means were computed and displayed to indicate the logical distance of each object (data point) in the Euclidean space from Table 7. A full and shiftable recognition of the patterns of cluster memberships is enabled, no matter how huge a graph might be. Here Object 9 in Cluster 1 (left) is responsible for the global maximum SESS and is most distant from its cluster center.

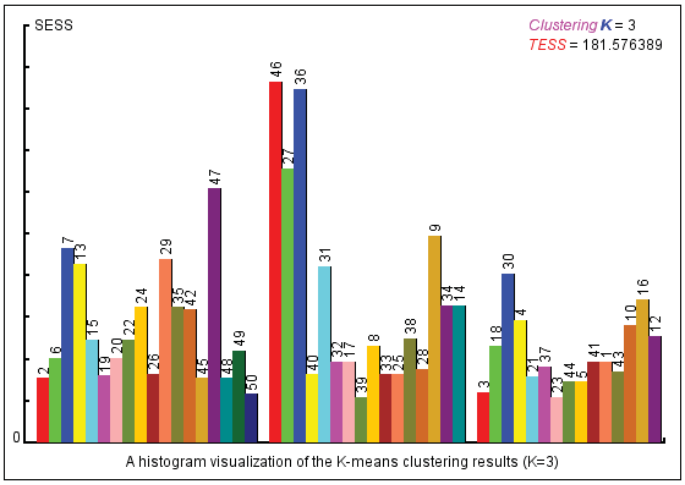

Fig. 4. A full view of tripartition of the *2*-dimensional data set (*size* = *50*) in Table 4, which is partially visible by shifting (moving) the histogram or zoomable. The red column is the first member object for each cluster. The objects 46 and 36 in Cluster 2 (middle) stand out as the globally first and second maximum SESSs respectively, apparently making their cluster a "partitional cluster" for the next-level clustering (*K*=4).

From the scientific viewpoint, it is harder to achieve a global optimality without the cost of computational complexity. Nevertheless, further improving its time complexity (for cost efficiency) and space complexity (for less intermediate data storage) merits consideration if the prerequisite of a resultant global minimum is met.

The next clustering problems to be resolved are what the optimal number of clusters would be and what all the possible different combinations (memberships) of data points would be for a given number of clusters under the same global minimum TESS. This would reveal more information about all of their potential memberships, relationships, and patterns in the identical condition.

5 Supplementary materials

The reader is referred to the on-line Supplementary materials for more details about the GOP *K*-means technology, technical appendices, additional demonstrations, and sample optimal and minimal solutions.

6 Authors' information

Bio of Ruming Li
A visiting professor at Chongqing University, China, received his BS and MS in China and two MSs and PhD in USA in applied statistics and quantitative genetics. Conducting research in bioinformatics, computational biology, genomics, proteomics, phylogenetics, applied mathematics, data structure, (Big) data science and analytics, data / information mining, pattern recognition, numerical analysis, algorithmic solutions and analytics, logic analytics, operations research, computer programming solution, and scientific software development.

Bio of Xiu-Qing Li
The research scientist of molecular genetics at Agriculture and Agri-Food Canada, received his B.S. in China and Doctorate d'Etat in France. Carrying out research in genome biology, bioinformatics, genotyping, molecular breeding, and taxonomic /homologous classification of DNA fingerprints.

Bio of Guixue Wang
The professor and Dean at Bioengineering College of Chongqing University, China, received his B.S., M.S., and doctorate in China. Engaging in teaching and research in biotechnology, bioinformatics, quantitative genetics, cellular and molecular bioengineering, cardiovascular biomechanics and biomaterials, biorheological science, and mechano-developmental biology.

7 Acknowledgements

This research was supported in part by Projects of China (2009ZX08009-109B) and National 111 Project (B06023).

8 References

[1] Lloyd, S. P. Least squares quantization in PCM. Technical Note, Bell Laboratories. Published in 1982 in IEEE Transactions on Information Theory **28**, 128–137, 1957, 1982.

[2] MacQueen, J. B. Some methods for classification and analysis of multivariate observations. (Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, Univ. of California Press), **1**, 281-297, 1967.

[3] Hartigan, J. A. Clustering Algorithms. (Wiley, New York, ed. 1), 1975.

[4] Jain, A. K. Data clustering: 50 years beyond K-means. Pattern Recognition Letters. **31** (8) 651-666, 2010.

[5] Hamerly, G. & Elkan, C. Alternatives to the k-means algorithm that find better clusterings. (Proceedings of the 11th international conference on Information and knowledge management), 600-607, 2002.

[6] Kanungo, T. , et al. An efficient k-means clustering algorithm: Analysis and implementation. IEEE Transac- tions on Pattern Analysis and Machine Intelligence, **24**, 881-892, 2002.

[7] Likas, A., Vlassis, N. & Verbeek, J. J. The global k-means clustering algorithm. Pattern Recognition **36**, 451-461, 2003.

[8] Hansen, P., Ngai, E., Cheung, B. K. & Mladenovic, N. Analysis of global k-means, an incremental heuristic for minimum sum-of-squares clustering. J. of Classification **22**, 287-310, 2005.

[9] Bagirov, A. M. Modified global k-means algorithm for minimum sum-of-squares clustering problems. Pattern Recognition **41**, 3192-3199, 2008.

[10] Daniel Aloise, Pierre Hansen, Leo Liberti An improved column generation algorithm for minimum sum-of-squares clustering. Mathematical Programming **131**(1-2) 195-220, 2012.

[11] Bakar, Z. A., Deris, M. M. & A. C. Alhadi, Performance analysis of partitional and incremental clustering. (SNATI 2005), ISBN: 979-756-061-6, 2005.

[12] Wilkin, G. A. & Huang, X. K-means clustering algorithms: Implementation and comparison. (2nd IMSCCS), 133-136, 2007.

[13] Chakraborty, S. & Nagwani, N. K. Analysis and study of incremental K-means clustering algorithm. (HPAGC, CCIS), **169**, 338-341, 2011.

[14] Kumar, A., Sinha, R., Bhattacherjee, V., Verma, D. S. & Singh, S. Modeling using K-means clustering algorithm. (1st Int'l Conf. on Recent Advances in Information Technology), 554-558, 2012.

[15] Charikar, M., Chekuri, C., Feder, T. & Motwani, R. Incremental clustering and dynamic information retrieval. SIAM J. Comput. **33**, 1417-1440, 2004.

[16] Mimaroglu, S. and Erdil, E. Obtaining better quality final clustering by merging a collection of clusterings. Bioinformatics **26** (20) 2645-2646, 2010.