Complex Adaptive Systems Modeling

RESEARCH Open Access

Multiobjective clustering algorithm for complex data in learning management systems

Rabie A. Ramadan[1,2*], Majed Mohaia Alhaisoni[1] and Ahmed Y. Khedr[1,3]

*Correspondence: rabie@rabieramadan.org
[1] Department of Computer Science, University of Ha'il, Ha'il, Saudi Arabia
Full list of author information is available at the end of the article

Abstract

Learning Management Systems (LMS) is now an emergent technology where massive data are collected and requires handling. This data comes from different sources with multiple features which represents another complex paradigm. However, as part of business intelligence and decision support, this data needs to be classified and analyzed for the management, teachers, as well as students to make the appropriate decisions. Thus, one of the effective data analysis methods is clustering. However, LMS data encompasses multi-features, which are not sufficient to make appropriate decisions. Therefore, single feature clustering algorithms would not help LMS decision-makers. Consequently, multifeatured/multiobjective clustering algorithms could be one of the proposed solutions. Thus, looking at different multiobjective clustering algorithms as compared to the LMS nature of data, those algorithms do not satisfy the clustering purpose. In addition, the LMS data could be huge, complex, and sequential algorithms would not help as well. Thus, this paper is a step forward towards clustering LMS data for better decision making. The paper proposes a new clustering framework based upon distributed systems and a new multiobjective algorithm for the purpose of LMS clustering. The algorithm has been examined experimentally in order to answer some of the questions that help taking decision based upon LMS collected data.

Keywords: Multiobjective, Clustering, LMS, K-means

Introduction

Most conventional clustering methods can handle only numeric or non-numerical data, but many real-world datasets consist of a combination of both; this is called heterogeneous data. For instance, k-means algorithm has a limited functionality when it comes to handling heterogenous data. K-means is one of the famous clustering algorithms that is based on computing the Euclidean distance between vectors. At the same time vectors have to be homogenous. Mixed numerical and non-numerical data might require different approach (MacQueen 1967).

For heterogeneous clustering, there are two different types of algorithms including algorithms that directly cluster heterogenous data and the ones that cluster heterogenous data based on some functionalities. For the first type of algorithms, several

Springer Open

algorithms can be generally divided into two types for heterogeneous clustering, i.e. (1) those clustering directly heterogeneous data and (2) those clustering heterogeneous data based on the functionality. The variable-and entropy-based clustering algorithm (CAVE) (Hsu et al. 2011; Hsu and Chen 2007) hierarchical clustering such as similarity agglomeration (SBAC) (Li and Biswas 2002), and extended self-organized maps (Hsu et al. 2011) are examples on the first class. SBAC could be applied for numerical and nonnumerical data measuring data density based on Goodall (1966). In the meantime, the expanded map (Hsu et al. 2011) and CAVE (Hsu and Chen 2007) built hierarchal distances that apply to heterogenous data. Hierarchical clustering is however computer-intensive and not suitable for high-dimensional datasets. The centroid-based clustering is less computationally intensive and effective in contrast with hierarchical clustering.

For k-means to compute the Euclidean and Hamming distance, it uses a dissemblance measurement for both numerical and non-numeric data. Both distances are integrated as centroid for the heterogeneous data (Huang 1997). However, adjusting and modifying the weights for hamming distance must be done manually. Although k-means can work on both Hamming and Euclidean distances, it is not straight forward to combine them linearly because they are different types of distances.

Therefore, in order to resolve this issue, it has been suggested to combine fuzzy c-means with Gauss-multinomial distribution (KLFCM-GM) (Chatzis 2011). This method uses a comprehensive measurement of differences between the negative Loglikehood feature of the Gaussian distribution and the fuzziness elements to cluster mixed data. Through taking the negative leg of the probability density function, KL-FCM-GM enables no merge between the Euclidean and Hamming distances. The KL-FCM-GM however still needs a parameter to manage the fuzziness. Furthermore, for numerical values of certain data sets, the assumption of a Gaussian multivariant distribution may be inappropriate. In 2013, an enhanced clustering protocol for k-prototypes was proposed based upon fuzzy k-prototypes (Chatzis 2011). In this algorithm, the weight for each feature (numerical or non- numerical) can be optimized in this way. However, the enhanced Kprototypes still need the combination distances of Hamming and Euclidean to measure their disparity. Other types of heterogeneous data cluster algorithms are applied with the functional transformation, in order to unify the data features and then use cluster algorithms with a single feature (numerical or non-numerical).

As an example, SpectralCAT (David and Averbuch 2012) is based on transforming numeric features into non-numeric features in order to handle heterogenous data clustering. Such transformation might not be appropriate because it removes the original data distance and may result in loss of information (Hsu et al. 2011). Another classical method is Functional Calibration (FC) where non-numeric features are transformed into numerical features (Flach 2012). FC is a supervised algorithm that uses the probability distributions of class labels. However, clustering is usually unsupervised problem that does not use class label information. Consequently, FC is not suitable for heterogenous data clustering.

This paper proposes a distributed framework for multifeature clustering. In addition, it proposes a new multifeatured clustering algorithm. The proposed algorithm has three main functions including clustering heterogenous data, handling data with multi features/objectives, and handling large-scale data. The distributed framework tends to divide the data into different sets for large-scale data clustering while the clustering algorithm considers the multifeature classification. Some of the previous work proposed to recover the previous clustering algorithms limitations are presented in (Alqurashi and Wang 2018; Ayad and Kamel 2008; Kang et al. 2016; Aggarwal et al. 2001; Fred and Jain 2002; Liu et al. 2014; Zhong et al. 2015). However, the main purpose of those algorithms is to increase the clustering outcomes in contrast to the basic clustering algorithms. On the other hands, various state-of-the-art algorithms that can be grouped into the following:

- The simple approach: It is used for explicitly recasting the initial input clustering solutions as discussed in (Fischer and Buhmann 2003; Gionis et al. 2005).
- Feature-based solutions: Such solutions consider the cluster of categorical data such as taking the ensemble question into account (Boulis and Ostendorf 2004; Nguyen and Caruana 2007; Cristofor and Simovici 2002).
- Methods based on pair similarity: Such methods consider the inter-relationship between objects. They consider the frequency of the clustering solutions as given in (Monti et al. 2003). Cluster Similarity Partitioning Algorithm (CSPA) (Strehl and Ghosh 2002; Costa et al. 2004) is the graph-based approach that very efficiently divides the unstructured graph. Hypergraph Partitioning Algorithm (HGPA) constructs a hypergraph for identifying k connected components from the clustering solutions where objects are classified as hyperedges, vertices and clusters. Algorithms Meta Clustering Algorithm (MCLA) is another similarity type of algorithms where the clusters are represented by vertices and the edge weights represent similarity between such clusters. The authors of (Strehl and Ghosh 2002) and (Costa et al. 2004) introduced cluster-based similarity partitioning algorithm based a similarity matrix. Such matrix is considered as an indicator toe the partition fraction where targets are allocated to the same cluster. The generated matrix is then used partition the graph accordingly.
- Graph Based Clustering (Ji et al. 2012; Boulis and Ostendorf 2004; Nguyen and Caruana 2007): One of the graph-based algorithms is Prim's algorithm. It is used to find the minimum cost tree for a graph. Based on this approach, the final solution is reached and there were no additional parameters needed. The authors of (Mimaroglu and Aksehirli 2011) suggested alternative approach where the actual clustering and the number of clusters could be provided. However, a clustering reference has to be defined through simulation. Several other recent clustering methods have been proposed in (Alqurashi and Wang 2018; Zhong et al. 2015). In (Zhong et al. 2015), the authors propose multimedia-based optimization algorithm for two objective functions optimization simultaneously. The solution set involves automatic generation to the number of clusters. At the same time, the algorithm generates a set of solutions instead of single solution. This is useful when the clustering criteria are not understood.
- Hypergraph Partitioning Algorithm (HGPA): In HGPA, hypergraphs are considered for clustering the basic nodes. It is assumed that hyperlinks and vertices weights are the same for all nodes. The algorithm hypergraphs division is used to split the edges in k-connected components that might be identical. Also, the

authors of (Shah et al. 2015) and (Armano and Javarone 2013) presented Hypergraph Partitioning algorithms for categorizing samples of the data.

- Multi-objective Clustering with Automatic K Determination (MOCK) (Strehl and Ghosh 2002): MOCK optimizes two additional lenses based on cluster compactness and connectivity, which can automatically recognize the correct partition of a dataset with either hyper-spherical clusters or well-separated clusters. The algorithm returns a large number of different scoreboards using various scoreboards that approximate the Pareto front. It includes in particular a mechanism for automatically selecting the best partitions from the solution set based on an approximation of the Pareto front shape.
- Clustering is proved to be important and effective (Ianni et al. 2020) and one of the used technique is MapReduce in which it is also used for Big data clustering with some restrictions as given in (Heidari et al. 2019).

Although many clustering algorithms have been established over the years, most of these algorithms need two key parameters: the number of clusters and the clustering algorithm. As stated previously, it becomes very hard to estimate a relevant number of clusters that can be provided as a parameter where the number of clusters in the cluster algorithms may differ. One of the approaches discussed in (Mimaroglu and Aksehirli 2011) offers promising results and automatically finds the number of clusters. In this method, the weighted graph shows all clustering algorithms, with the vertices marked with the clusters, and the inter-cluster similarity is marked by the edge. The similarity between clusters basically encodes the similarity between the object level clusters.

The number of final clusters required by CSPA and HGPA is specified in (MacQueen 1967). While HGPA is the fastest algorithm, its degenerative effect of noise, clusters might not be accurate. Because it works at an objective level, CSPA doesn't scale very well. MOCK is one of the most powerful algorithms based on a multi-target genetic algorithm, but the cost is too much if the data is large. The strategies for selection are based on domain considerations that limit their application. Therefore, in contrast to the above highlighted algorithms, the proposed algorithm in this paper takes in consideration mitigating these limitations.

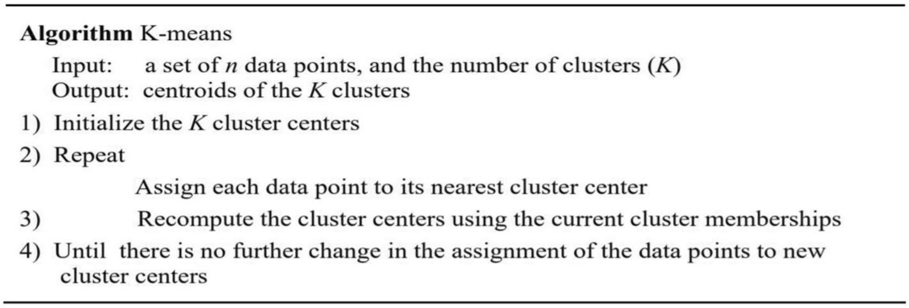
To prove the efficiency of the proposed multiobjective clustering algorithm, LMS data was utilized. LMS data will be used to prove the efficiency of the proposed multiobjective clustering algorithms. LMS data is considered as a complex as well as big data where it has a large number of features and extracting some of the data features for the management or even students to take the appropriate decision is not straight forward. Therefore, such complex data needs to be clustered, and single or even two objective clustering algorithms would not be adequate. Multiobjective clustering would be more suitable.

The paper is organized as follows: k-means algorithm is described in the following section, followed by the proposed multiobjective clustering framework; the proposed modified k-means algorithm for learning management systems (MKM) is also detailed; finally the experimental results are presented and evaluated.

Methods

K-means algorithm

The conventional k-means algorithm (MacQueen 1967) is the most famous clustering method. The algorithm is a simple reassessment procedure described in *algorithm k-means*.



```
Algorithm K-means
    Input:   a set of $n$ data points, and the number of clusters ($K$)
    Output:  centroids of the $K$ clusters
1) Initialize the $K$ cluster centers
2) Repeat
           Assign each data point to its nearest cluster center
3)         Recompute the cluster centers using the current cluster memberships
4) Until  there is no further change in the assignment of the data points to new
    cluster centers
```

In the dataset, $X = \{x_1, \ldots, x_n\}$ are the original n data points for clustering where in the k-means, n data points are partitioned into K sets. On the basis of the membership function m(cj|xi), the decision of assigning each data point xi to the nearest cluster center c_j. Therefore, the function returns one value of {0, 1} where $m(c_j \,|x_i) = 1$ if $j = \text{argmink} \,||x_i - \text{ck} \,||^2$; otherwise it is zero. Then, we can use all data points x_i in the cluster to determine the centroid of the cluster. The objective function J is to minimize the squared error, $J = I = 1{:}n \min j \,\{1..k\} \,||\, x_i - c_j \,||^2$. Each data point has equal importance in k-means in determining the cluster centroid. Thus, in the computation of cluster centers, the clustering algorithm must take into account a weight associated with each data point. So, weighted k-means could be a solution of the k-means algorithm. However, in our case, there is no need for the weighted k-means due to the nature of the LMS (Niazi and Temkin 2017; Alenezi 2018) data used in this paper.

Proposed multiobjective clustering framework

The proposed framework in this section is generic, and it is inspired by the previous work stated in the introduction section. The framework consists of two main blocks, which are preprocessing and clustering, as shown in Fig. 1. The preprocessing module tries to speed up the optimization process by dividing the large dataset into subsets. Dataset division could be based on random selection or based on simple clustering algorithms. Therefore, the generated sets are could be initial clusters. The generated clusters could be based on single objective clustering, as well. The output of this block is a certain number of sets where each set could have some of the populations that are close enough from each other. The clustering block contains distributing optimizers where they can work in parallel. Here, there are two options, as follows:

1. Each optimizer works on separate set of data.
2. Each optimizer selects, either randomly or again based on certain criteria, different records from the sets to work on.

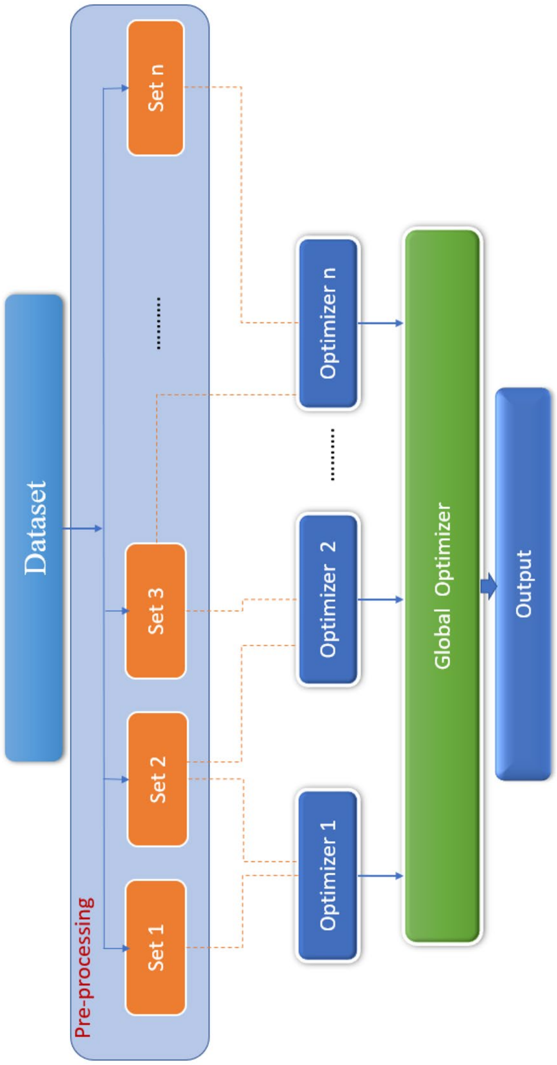


Fig. 1 Clustering framework

By the end of this process, the output of such optimizers is used as an input to a global optimizer for the final clustering results. Here, the global optimizer tries to merge some of the closest generated clusters using some of cluster indexes.

As can be seen in Fig. 1, the whole optimization process could be distributed processes developed on different processors. In addition, the framework enables the utilization of Cloud computing in case of large dataset.

Modified K-means algorithm for learning management systems (MKM)

The nature of data generated from the LMS differs from any other datasets. The LMS datasets is related to the teachers and students' performance. Also, there are many objectives including the time spent by the teacher on the system, uploaded data, the time spent by the student on the system, etc. Here, we propose an algorithm for data clustering in LMS. The algorithm tries to minimize the variance functions within each cluster and maximizing variance between clusters. Initially, in order to normalize the dataset records, we calculate the variance and the mean values for each record. This would represent the x value (mean) and y value |(variance) for each dataset record, respectively. Based on this generated data, K-means algorithm is used for clustering such data. By default, k-means will minimize the variance and the mean values within each cluster. However, we propose to use Silhouette score (Rousseeuw 1987) to maximize the variance between the clusters. Silhouette score shows how close a point is to the related cluster. i.e. For each point, calculate the average distance from the points in the nearest cluster minus the average distance from the points in its own cluster divided by the maximum between those distances.

Overall score is the average score per point. Silhouette score is restricted between 1 and − 1 and higher score means more distinct clusters. This scoring operation is used by the global optimizer.

Therefore, K-means is modified to work on two objective functions, the distance between cluster C and Silhouette score. For each point x, the distance between x and the cluster center is computed as well as the Silhouette scores provided that x belongs to one of the clusters. Pareto optimality is used to identify the nondominated set; based on the results of the Pareto optimality, x will be assigned to one of the clusters.

The algorithm steps could be summarized as follows:

Step 1: The dataset D is divided into a number of sets S. S may depend on the number of distributed machines or number of threads to be used.

Step 2: x value (mean) and y value |(variance) are computed for each dataset record.

Step 3: K-means clustering is applied to each set s $\in S$. K is selected either heuristically or based on the number of records in each set.

Step 4: At the global optimizers, Pareto optimality is applied to the clusters' centroids and nondominated centroids.

Step 5: for nondominated clusters, the distance between a point x and the cluster center is computed as well as the Silhouette scores between x and the nearest cluster center. Then, the K-means algorithm is used to re-cluster those points.

Step 6: A window W is used to extract the most effective clusters based on the required points, e.g. LMS questions. Pareto optimality could be applied once more for better results.

The complexity of the modified algorithm depends on two factors which are k-mean algorithm and the Pareto optimality section. The k-main algorithm complexity is $O(n^2)$ and the complexity of the Pareto optimal depends on the number of parameters. In the case of 2 parameters, it is estimated to be O(n) and this applies to all of our experiments in this paper, where n is the number of points. Therefore, the overall complexity of the proposed algorithm is $O(n) + O(n^2) = O(n^2)$.

Results

The proposed clustering algorithm has been experimented using Java platform. However, in order to show the robustness of the algorithm, genuine data has been collected from University of ha'il dataset, primarily, from Blackboard as a Learning management Systems (LMS). The collected data is almost a million record for students and teachers based on different semesters. Each record contains the following fields (see Table 1):

Research questions are summarized as follow:

A. Which courses that are the best based on all dataset features?
B. What are the courses with high exams (which exam, final or mid-term) grades where students were able to get good final grades?
C. Is there a relation between the total spent time on Blackboard and achieving good grades?
D. Have the number of items uploaded on Blackboard affect students' grades?

We believe that those questions could be answered through clustering. In this section, we cluster the dataset trying to answer the previous questions using k-means and Modified K-Means algorithms. This section also tries to examine the importance of the proposed algorithm for producing accurate results. In the k-means algorithm, each record

Table 1 Dataset abbreviations

Field	Meaning
Stud_ID	Student/Teacher ID
Course_ID	Course ID
TOTAL_LOGIN	Number of times a user logged in
Last_Login_Date	It is the last time a user logged into the Blackboard
Enrollments	The number of students enrolled in the class
Content_Items	The number of uploaded items to the class
Gradecenter _Columns	It is the number of fields in the grade center. This counts the number of assignments, quizzes, and other exams in the class
Assignments	This field record students grades in each assignment assigned to students e.g. Assg1, Assig 2,..etc
Tests	This field records the exams grades, e.g. Quiz 1, Quiz 2, midterm, and final exam
Total_Spent_Time_Hrs	The total time a student and a teacher spent on the Blackboard in hours
Total_Spent_Time_Mins	The total time a student and a teacher spent on the Blackboard in seconds

in the dataset is represented as a point. Therefore, the features are normalized; then the variance and the mean are computed for each record.

So, lets answer the questions one by one based on the two algorithms, k-means and MKM algorithms. However, due to the huge dataset, we display only a sample from the results.

Question A: Which courses that are the best based on all dataset features?

Based on the results gained from running both algorithms considering all of the features having the same weight. As can be seen in Fig. 2, we examined k-means for a limited number of records due to the dataset size. The algorithm was taken too much time to run 10,000 records and it was not able to run more than 50,000 records, especially when 5 iterations was required. Also, as can be noticed k-means algorithm is not really answering the question of this subsection. If we look at Fig. 2f, we can find that the top right corner contains large number of records that cannot be considered as a good answer to our question.

On the other hand, for MKM algorithm, we computed the mean and variance for each record. Therefore, a point (x, y) represents each record. Now, we have to decide on the number of sets where larger number of sets makes the clustering fast. We learned the lesson from our previous experience with k-means and decided to divide the dataset into 100 sets with 10,000 record/set with $k=2$. With distributed servers, cloud, or even threads, the whole dataset can be easily clustered.

Next, the centroids of the produced clusters are computed, and the k-means run on them for clustering. K is chosen to be 10 for the centroid of clusters, see Fig. 3. A window of three clusters were selected to enter the next stage which the Pareto optimality. Pareto optimality leaves us with top records out of the dataset which were only 10 records.

This experiment does say many other things where the students are clustered based on the dataset features. Now, we are able to classify students into categories that can separately studied, i.e. weak, intermediate, excellent students. Weak students could be identified from each class once we apply pareto optimality on each class where those dominated records represent the weak students.

To verify the validity and efficiency of the results, Silhouette score is computed for different number of K starting from 2 to 10 as shown in Fig. 4. As can be noticed, the score is more into the positive area and the best value is reached at $k=10$. This is an indictor to the effectiveness of the proposed algorithm in terms of clustering the data.

Similarly, we can answer the rest of questions easily as follows:

> Question B: The dataset is clustered based on exams only; here, we still need the MKM clustering where multiple exams are conducted, quizzes, midterm, and final. Therefore, MKM procedures effective answers to this question in terms of all exams. Based on our datasets, the algorithm was able to cluster students into different groups based on their different exams.
>
> Question C: Here we need to cluster the dataset based on two features only which are spent time and final grades. Here, the k-means could be enough; however, the dataset is too large to be clustered by k-means. Using our distributed algorithm, we were able to cluster the dataset into 3 clusters where k is selected to be

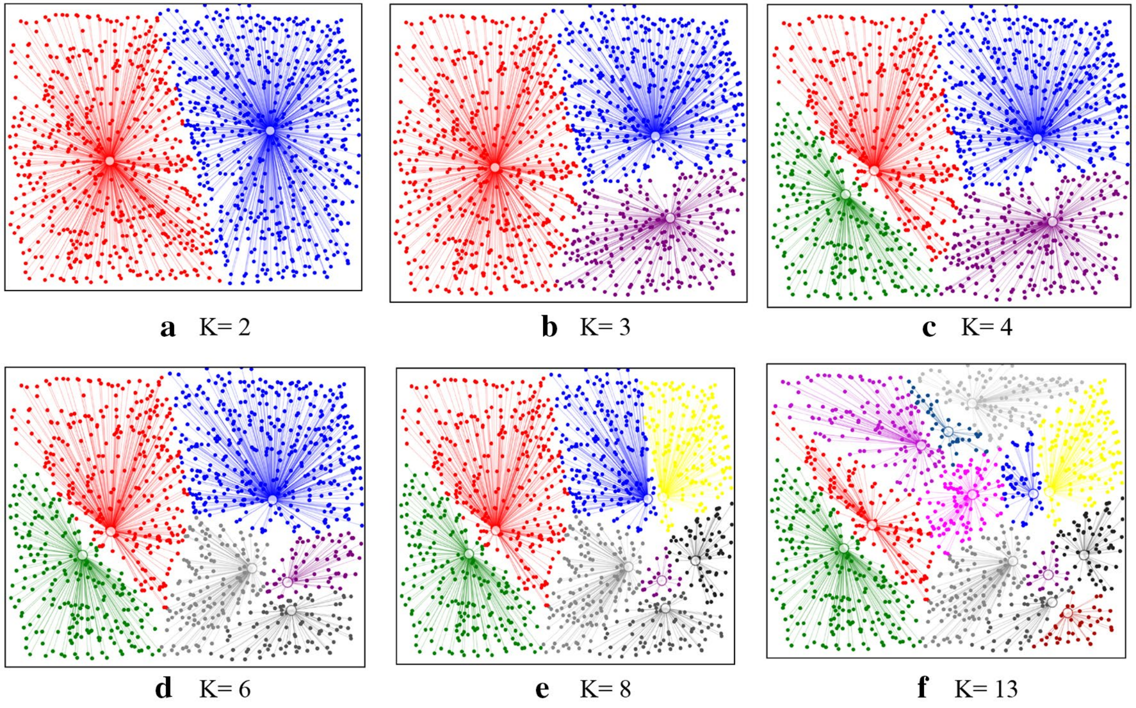


Fig. 2 K-means algorithm with different number of clusters

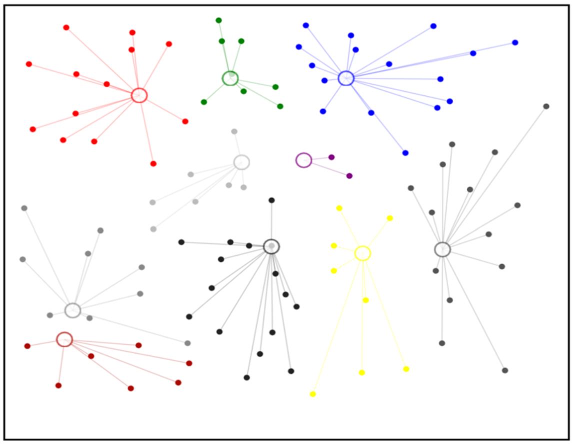


Fig. 3 Centroid clusters

3. Also, we tried with different values for k to verify the algorithm performance. It turns out, as we increase the k, the number of threads could be increased and the required time is minimized.

Question D: Again, here we cluster the dataset based on two features which are items uploaded and final grades. This ranks the students based on those two features. Here, the k-means could be enough; however, the dataset is too large to be clustered by k-means. Again, different experiments were conducted using our distributed algorithm and it clusters the data in reasonable time.

Discussion

With the emergent technologies in LMS, huge data are collected from different sources. Such data has multi features as well as complex structure. Therefore, data analysis is not easy, and it might take long time. By dividing the collected data into separate classes and handling each class separately, this makes it easy and quick to take a decision. Therefore, the first step is to divide the dataset into smaller sets and then process each set separately. Following this division, each subset is clustered based on the proposed multiobjective clustering algorithm. The global optimizer owns the final clustering. The proposed framework and the clustering algorithm are applied to millions of records from LMS data. Since the data has large number of features, the proposed framework and the algorithm is examined on the LMS data trying to answer some of the questions. Also, it is tested against k-means algorithm as well. The proposed algorithm is proved superior answering the required questions. In addition, it is a generic algorithm that can be used for other datasets as well.

The limitation of the proposed algorithm is two-fold as follows:

- With the large number of parameters, more processing capabilities might be required. So, running the distributed algorithm in local computers with multicores might not be possible.

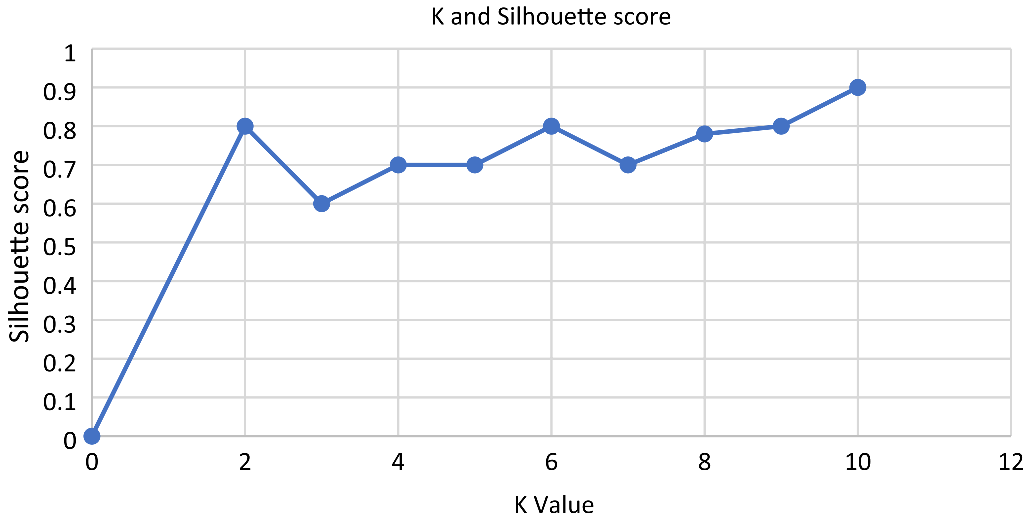


Fig. 4 K Vs Silhouette score

- The data used to test the proposed algorithm is homogenous. However, some big datasets might not be homogenous. Therefore, few adaptations have to be made to the proposed algorithm to fit the heterogeneity of the data.

Conclusions

This paper proposed a novel distributed clustering framework for complex multi-objective clustering. The proposed framework is generic to be used with any dataset. Also, the paper introduced k-means algorithms for learning management systems. However, due to the complexity of the data, it turns out that K-means was not able to accurately answer the LMS management questions. In addition, k-means is a sequential algorithm that can work on large-scale data sets. Therefore, a modified k-means algorithm is proposed for the purpose of LMS data multiobjective clustering. The distributed version of the algorithm was able to satisfy the LMS requirements and answer the required questions to help decision makers. University of Ha'il LMS dataset is used to test the performance of the proposed algorithm and to answer mainly four questions. Our future work is to examine the performance of the proposed algorithm on other datasets.

Acknowledgements
The authors would like to thank the Deanship of Scientific Research at University of Hail for their sponsorship to the project number 0161138.

Authors' contributions
The authors have equal contributions. All authors read and approved the final manuscript.

Funding
The project is sponsored by the University of Hail.

Availability of data and materials
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1] Department of Computer Science, University of Ha'il, Ha'il, Saudi Arabia. [2] Computer Engineering Department, Cairo University, Giza, Egypt. [3] Computer Engineering Department, Alazhar University, Cairo, Egypt.

Received: 31 January 2020 Accepted: 20 April 2020
Published online: 28 April 2020

References

Aggarwal G, Garg S, Gupta N (2001) Combining clustering solutions with varying number of clusters. IJCSI Int J Comput Sci Issues 11(1):240

Alenezi A (2018) Barriers to participation in learning management systems in Saudi Arabian Universities. Educ Res Int. https://doi.org/10.1155/2018/9085914

Alqurashi T, Wang W (2018) Clustering ensemble method. Int J Mach Learn Cybern 10:1227–1246

Armano G, Javarone MA (2013) Clustering datasets by complex networks analysis. Complex Adapt Syst Model 1(1):1–10. https://doi.org/10.1186/2194-3206-1-5

Ayad HG, Kamel MS (2008) Cumulative voting consensus method for partitions with variable number of clusters. IEEE Trans Pattern Anal Mach Intell 30(1):160–173

Boulis C, Ostendorf M (2004) Combining multiple clustering systems. In: Proceedings of the European conference on principles and practice of knowledge discovery in databases, pp 63–74

Chatzis SP (2011) A fuzzy c-means-type algorithm for clustering of data with mixed numeric and categorical attributes employing a probabilistic dissimilarity functional. Expert Syst Appl 38:8684–8689

Costa IG, Carvalho FAD, de Souto MCP (2004) Comparative analysis of clustering methods for gene expression time course data. Genet Mol Biol 27(4):623–631

Cristofor D, Simovici DA (2002) Finding median partitions using information-theoretical-based genetic algorithms. J Univ Comput Sci 8:153–172

David G, Averbuch A (2012) SpectralCAT: categorical spectral clustering of numerical and nominal data. Pattern Recognit 45:416–433

Fischer B, Buhmann JM (2003) Bagging for path-based clustering. IEEE Trans Pattern Anal Mach Intell 25(11):1411–1415

Flach P (2012) Machine learning: the art and science of algorithms that make sense of data. Cambridge University Press, Cambridge

Fred A, Jain A (2002) Evidence accumulation clustering based on the K-means algorithm. In: Structural, syntactic, and statistical pattern recognition, LNCS 2396. SpringerVerlag, pp 442–451

Gionis A, Mannila H, Tsaparas P (2005) Clustering aggregation. In: Proceedings of the international conference on data engineering, pp 341–352

Goodall DW (1966) A new similarity index based on probability. Biometrics 22:882–907

Heidari S, Alborzi M, Radfar R, Afsharkazemi MA, Ghatari AR (2019) Big data clustering with varied density based on MapReduce. J Big Data 6:77

Hsu C-C, Chen YC (2007) Mining of mixed data with application to catalog marketing. Expert Syst Appl 32(12–23):14

Hsu C-C, Lin S-H, Tai W-S (2011) Apply extended self-organizing map to cluster and classify mixed-type data. Neurocomputing 74(3832–3842):13

Huang Z (1997) Clustering large data sets with mixed numeric and categorical values. In: Proceedings of the 1st Pacific-Asia conference on knowledge discovery and data mining, Singapore, Singapore, 23–24 February, pp 21–34. 16

Ianni M, Masciari E, Mazzeo GM, Mezzanzanica M, Zaniolo C (2020) Fast and effective Big Data exploration by clustering. Future Gener Comput Syst 102:84–94. https://doi.org/10.1016/j.future.2019.07.077

Ji J, Pang W, Zhou C, Han X, Wang Z (2012) A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data. Knowl Based Syst 30:129–135

Kang Q, Liu S, Zhou MC, Li S (2016) A weight incorporated similarity-based clustering ensemble method based on swarm intelligence. Knowl Based Syst 104(C):156–164

Li C, Biswas G (2002) Unsupervised learning with mixed numeric and nominal data. IEEE Trans Knowl Data Eng 14:673–690

Liu S, Kang Q, An J, Zhou MC (2014) A weightincorporated similarity-based clustering ensemble method. In: Proceedings of the 11th IEEE international conference on networking, sensing and control, pp 719–724

MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. University of California Press, Oakland, CA, USA, pp 281–297

Mimaroglu S, Aksehirli E (2011) Diclens: divisive clustering ensemble with automatic cluster number. IEEE/ACM Trans Comput Biol Bioinform 99(2):408–420

Monti S, Tamayo P, Mesirov J, Golub T (2003) Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. Mach Learn 52(1/2):91–118

Nguyen N, Caruana R (2007) Consensus clusterings. In: Seventh IEEE international conference on data mining (ICDM 2007), pp 607–612

Niazi MA, Temkin A (2017) Why teach modeling & simulation in schools? Complex Adapt Syst Model 5(1):5–8. https://doi.org/10.1186/s40294-017-0046-y

Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20(C):53–65. https://doi.org/10.1016/0377-0427(87)90125-7

Shah MA, Abbas G, Dogar AB, Halim Z (2015) Scaling hierarchical clustering and energy aware routing for sensor networks. Complex Adapt Syst Model. https://doi.org/10.1186/s40294-015-0011-6

Strehl A, Ghosh J (2002) Cluster ensembles—a knowledge reuse framework for combining multiple partitions. J Mach Learn Res 3:583–617

Zhong C, Yue X, Zhang Z, Lei J (2015) A clustering ensemble: two-level-refined co-association matrix with pathbased transformation. Pattern Recognit 48(8):2699–2709

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.