RESEARCH ARTICLE

SAMRIDDHI Volume 14, Issue 1, 2022 Print ISSN: 2229-7111 Online ISSN: 2454-5767

Multi-level K-Means Density-based Flow Clustering Algorithm for Data Stream Clustering

Ankit K. Dubey[1]*, Rajendra Gupta[2], Satanand Mishra[3]

[1]Department of Computer Science, St. Aloysiu s College (Autonomous), Jabalpur, MP, India

[2]Department of Computer Science, Rabindranath Tagore University, Raisen, MP, India

[3]CSIR-Advanced Materials and Processes Research Institute (AMPRI), Bhopal, MP, India

Abstract

Data stream clustering is an active area of research that has recently emerged with the goal of discovering new knowledge from a large amount and variability of constantly generated data. In this context, many researchers have proposed different algorithm for unsupervised learning that clusters multiple data streams. There is a need for a more efficient and efficient data analysis method. This paper introduces a multi-level K-Means density-based flow clustering algorithm (MKDCSTREAM) for clustering problems. This approach proposes to view the problem of clustering as an optimization process hierarchy that follows different levels, from unrefined to subtle. In the clustering problem, for the solution, divide the problem into parts by following different levels to make the first clustering a coarser problem than calculated. Coarse problem clustering is mapped level by level and improves the clustering of the original problem by improving intermediate clustering using the general K-means algorithm. Compare the performance of the hierarchical approach with its single-tier approach using tests with a set of data-sets collected from different areas.

Keywords: Clustering, MKDCSTREAM, Multi-level K-Means, Unsupervised learning.

SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology (2022); DOI: 10.18090/samriddhi.v14i01.20

Introduction

The amount of data stored on your computer is growing at an alarming rate. However, it turned out to be very difficult to obtain useful information. In many cases, regular data research techniques and analysis tools are simply not suitable for meeting these growing demands for information.[1] Data clustering is one of the main tasks of data analysis: pattern identification, data densification, analysis of an image, and machine learning.[2] Our method first estimates multiple sub-clusters from the data stream and then applies task-specific clustering algorithms to these sub-clusters. Each subcluster is represented by a set of centroids that are one by one assessed using various parameters. Centroid updates continuously depending on the arrival of a new data object in the streaming input. The next step, clustering all fit points, is for the centroid to coincide with the cluster. This is based on the attribute of the data-set Opposite to K-means clustering, the method which we are proposing a cluster can have multiple center points, one typical point per cluster.[3] The major contribution of this research is:- A Multi-level stream clustering algorithm capable of processing large multidimensional data-sets.

Literature Review

Researchers have made several attempts to improve the efficiency and effectiveness of data stream clustering shown in Table 1. In this,[4] the author proposes a density-based clustering algorithm for IoT streams. This method is fast and very useful for real-time IoT applications. In the Experimental results, authors find that this method can provide high-quality results for real data sets and synthetic data sets in a short computation time. In this[5] paper, author's have developed an efficient and effective client method called CluStream for large-scale clustering. They are evolving data streams. This method has obvious advantages. Discourse about the latest technologies to combine Flow as a process that changes over time rather than looking at the entire stream at once. The stream CluS model provides a wide range of capabilities for describing clusters of data streams over different time periods in an evolving environment. In [6] author's proposed a hybrid K-Means algorithm that joint the steps of reduction

Corresponding Author: Ankit Kumar Dubey, Department of Computer Science, St. Aloysiu s College (Autonomous), Jabalpur, MP, India, e-mail: ankitkdubey@yahoo.com

How to cite this article: Dubey, A.K., Gupta, R., & Mishra, S. (2022). Multi-level K-Means Density-based Flow Clustering Algorithm for Data Stream Clustering. *SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology,* 14(1), 125-130.

Source of support: Nil

Conflict of interest: None

of dimensionality with PCA, a new approach for initializing cluster centers, and the steps for assigning data points to the appropriate clusters. Use The suggested algorithm divides the specified data-set into k clusters, reducing the total clustering errors for all available clusters as much as possible while keeping the distance between the clusters as large as possible. Authors of,[7] proposed UMicro's algorithm for clustering undefined data streams. Undefined data streams can be present in many real-world applications due to inaccurate write mechanisms. Data flow uncertainties significantly impact the clustering process, as different attributes behave differently relative to each other and affect distance calculations. This paper[8] gives an overview of the data A stream clustering algorithm applied on a stream of big data and large data-sets. The documentation shows a deep comparative analysis of all the methods reviewed and an overview of progression and progress Dataflow clustering algorithms for large data-sets. The article is also reviewed a proposed and recently implemented algorithm. This article[9] has summarized a simple set of conditions for a data stream clustering algorithm. An important requirement that algorithm designers often ignore is the need for clear isolation of outliers in the data stream. This is because a sufficient number of outliers may indicate that the clustering model needs to be modified. Authors provide analytical tools to effectively track these changes. This[10] study presents a new discovery method based on a scalable framework for identifying relevant ones in a community on the web. We propose a multi-level clustering (MCT) method that uses a textual information structure to identify a local community called a microcosm. Experimental evaluations of reference models and data-sets show the effectiveness of the approach. This research contributes to a new dimension for identifying cohesive communities on social media. This approach provides better understanding and clarity to explain how low-level communities develop and operate on Twitter.

The Clustering Methods and Algorithm's

K-means Clustering

Clustering algorithms generally fall into two different categories: partitions and hierarchies. The partition clustering algorithm divides a data-set into non-overlapping groups. k-means algorithm fall into this category.[10] It's an unsupervised learning algorithm. This algorithm categorizes the data set items into k groups or clusters of similarity. To calculate the similarity, we can use different methods like hamming distance, Euclidean distance, Manhattan distance, etc.

Table 1: Summary of a Literature Review

Stream Type	*Clustering method*	*Reference*
IoT Stream	Density-based clustering	[4]
Large Scale	CluStream	[5]
High dimension	Hybrid K-Means	[6]
Undefined data streams	UMicro's algorithm	[7]
Communities on social media	Multi-level clustering	[10]

Algorithm 1: K-means

```
Input: K number of clusters, D list of data points
step 1: select K number of random data points as initial centroids.
step 2: Repeat till cluster centers stabilize:
a. Allocate each point in D to the nearest kth centroids.
b. Compute centroid for the cluster using all points in the cluster.
```

Multi-level Kmeans_clustering Algorithm

This clustering technique creates diverse views for each and every cluster. The algorithm is divided mainly in two stages. We call this the Subset Generator. The first step in sub-clustering phase clusters x injects data into y sub-clusters (k <x <y) similar to the algorithm_ means. In the next step, a few of these centers of gravity are clubbed together into one large group. This is called the recovery stage. Each cluster is constructed from the same set of input using different parameters. The multi-level K-Means algorithm is a combination of the popular greedy K-means algorithm and the multi-level paradigm of clustering problems. The layered paradigm is a simple problem-solving technique with a coarse recursive structure that easily solves simple and minor problems. The layered paradigm consists of four stages: broad outline, initial decision, forecasting, and correction. The broad outline phase aims to combine the volatile data related to the problem to form a cluster.[1] Clusters are used recursively, each cluster representing the original problem. It builds a hierarchy of problems with less freedom. This phase continues till the minimum problem size reaches the specified reduction limit. Then the solution to the problem is generated at the unrefined level is projected in reverse order to each intermediate level. "The solution at each child level is improved before moving to the parent level. A common feature that characterizes multi-level algorithms is that the solution to one of the critical problems is a legitimate solution to the original graph".

Algorithm 2: The Multilevel K-Means Algorithm

```
Input: Problem $x_0$
Output: Solution $y_{final}$ ($x_0$)
Begin
lvl:=0;
/* Proceed with Coarsening */
while not stop do
$d_{lvl+1}$ := lvl+1;
lvl := lvl+1;
$QC_{start}$ ($d_{lvl}$) = initial_clustering (d_level);
/*Proceed with Uncoarsening and Refinement */
```

```
9. while(lvl > 0)do
10.$Qy_{start}$ ($d_{lvl-1}$) ≔ Extend ($Qy_{final}$ ($d_{lvl}$));
11.$Qy_{final}$ ($d_{lvl-1}$) ≔ K - means ($Qy_{start}$ ($d_{lvl-1}$));
12.lvl ≔ lvl-1;
                                end
```

The first phase of multi-level clustering is the pruning phase, where is the set of data objects to a cluster. The next level of coarser is constructed from with two different algorithms,. The random coarsening scheme is the first-level algorithm. In random order, data objects are accessed. If the data object is not yet mapped, a mismatched data object is randomly selected consisting of the two data objects and the; a new data object is randomly selected. A new set of data object attributes is calculated by averaging each attribute from and the corresponding attribute from. Just copy the data object to the next level without merging. The second algorithm is coarsening algorithm, based on the concept of distance. This algorithm uses a measure of the strength of the connection between data objects. However, instead of adding the object with a random object, the data object is combined with so that (2) is minimized. Use the newly created data object to identify a new small problem and repeat the clipping process recursively until the size of the problem reaches the desired threshold. (1, 2, 3, 4, 5,6 line).

If we see line 7, initialization is easy and consists of using a random procedure to generate an initial clustering of problems (). All individual clusters in the population are assigned random labels from a set of cluster labels. Due to the improved quality of clustering at the level, it is necessary to expand the parent level of . In line 10, If the is assigned the cluster , The combined pair of data objects that it represents, is also assigned the cluster label Line 11, The clustering found at level is minimal; with respect to m, predicted clustering may not be optimal. Predictive clustering is fine, and the K-mean converges faster and clusters better over several iterations. Premature convergence occurs if the cluster does not change over the number of iterations. The k-means are expected to converge at each level if all clusters do not change within five consecutive iterations.

Merge and Split Algorithm

This section presents a highly accurate stream clustering algorithm and is preferable for an extensive genre of applications when processing an unlimited amount of data. There are three main modules: subset generator, microcluster generator, and microcluster splitting and merging. A data preprocessor is an online component that preprocesses streaming data from raw data—generated as a data stream by the previous component. The k-means algorithm with the divide and conquer technique was applied to create microclusters.[11]

- *Subset generator.* This module is very important for calculating an object length in the data window and using the K-means algorithm to create a one-dimensional vector for clustering. The clustering result will divide the data into several subsets, depending on their size and level.

  $$M(O) = \sqrt{\sum_{z=0}^{x} O_z^2}, \qquad (1)$$

  Determines the object's length, Number of sizes and their feature values. Vector mapping is used to compress the data in signal processing to divide data into sub-clusters. "Subsets are modeled as probability density functions represented by prototype" vectors (centroids). The easy and fine version of vector mapping randomly selects a data point vector from a particular data-set. It then regulates the respective centers and amendments the centroid of the *mapped* vector based on this new object. This vector moves to the present insertion point and continues this process for the entire data-set.
- *Micro-clusters generator.* To improve the accuracy of the grouping process, we need to maintain a high level of detail in the data structure. Adopt the compressed data point cluster maintenance process to reach this exact target. Such groups are called microclusters. Clustering is called batch processing instead of incremental online processing (taking samples from a single stream). Following this strategy, microclusters are created and stored in the microcluster repository for further analysis.
- *Split and merge.* Due to compactness and separability criteria, frequent splitting and merging were designed to detect conceptual drift. You can combine the two clusters into one cluster if the means value of the two clusters is closest to each other for their compactness and separability. You can also split the microcluster, if there are too many elements in the microcluster and the compactness decreases in the meantime. However, the user must define parameters such as split and merge thresholds. However, you can use those recorded as frequent splits and merges to detect conceptual anomalies.

Algorithm 3: Merge and Split Algorithm

```
Input: Micro clusters list($M_1$)
Output: Micro clusters (M)
1. for x = 1 to k, $M_1$
2. if 2–$M_1$ means of clusters Are approximately
   close, merge & make one cluster
3. If end;
4. Divide the microcluster when there are too
   many elements in the microcluster and the
   compactness decreases in the meantime.
                        end;
```

Streaming sub-clustering and Proposed MKDCSTREAM Algorithm

Figure 1 shows, our sub-clustering module is working on this streaming vector mapping method. The input data is randomly ordered. When a fresh data point is displayed, the center which is closest to the fresh data shifts gradually in that direction. It holds this center point updated and moved.

Algorithm 4: Streaming sub clustering

```
Input: streaming input (s) in dimension (m), a
       current set of centroids(x)
Output: X is the new centroids set
1. Take a new input s
2. Calculation of each center point Distance
   x ∈ X and the news
3. find a center point nearest cluster, $x_m$.
4. shift $x_m$ nearest to s
5. Return the updated X
```

Algorithm 4 shows the stream sub clustering algorithm. The input s is the m-dimensional point of the streaming data, and X is the current set of centroids for the they-cluster. This point will be attached to the nearest one. It is a cluster, and the center point of that cluster is updated with the new using the following formula:

$$x_{m_{t+1}} - \mu . s_t = (1-\mu) . x_{m_t} \tag{2}$$

In the above formula, is the learning rate is a latest input time t, is the center for the cluster , and is an updated center point. Our method takes an unlimited stream, so you may either supply random points to the sub-clustering unit or can use a pre-computed module from the software. A subset of data may use its first part K-means for data indexing. The stream sub clustering modules are completely parallel because they are autonomous. The pruning module combines the centroids of the subclusters to find the matching cluster ID for every data point.

The steps of the MKDCSTREAM algorithm are as follows.

Algorithm 5: MKDCSTREAM Algorithm

```
Input: h dimension data stream
Output: the latest set of centroids with Cluster
Id
**Step 1**
Use theMultilevel K-Means algorithm for
Coarsening and Refinement.
**Step 2**
For merging the microcluster use merge and sort
algorithm.
**Step 3**
For the stream cluster use the stream sub
clustering algorithm
                    End.
```

The sub-clustering module uses different parameters; each cluster is constructed from the same set of input sets. Each process is completely independent, which makes it highly extensible. Our streaming method minimizes the complexity of computation and resources and parallelizes these independent operations.

Centre points
Data points
(a)At Time t0
New Centre points
New data points
(b) At Time t1

Figure 1: This process updates and moves this center point as new data is displayed.

Experimental Activities and Result Discussion

All performance tests were performed on a 2.40GHz Intel® Core™ i3-3110M CPU. Windows 8 (64-bit) operating system with 3.25GB of RAM. The implementation was done in Pycharm 2020 3.5 for MKDCSTREAM encoding.

We test our approach using several different sets of data parameters. Table 2 shows the data set that we have used for experimental purposes. The data-set which we have used here is data-sets of 2D and 3D dimensions –moons and 3D clouds, from UCI Machine Learning Repository (spam base and census 1990).[12] We compared algorithm MKDCSTREA with three frequently used clustering algorithms for processing data streams, namely with Mini Batch K-mean and Birch algorithm.

Accuracy

We review and compare the results of clustering the sample data-set with other clustering algorithms: MiniBatchKmean and Birch. Clustering results of 2D (Two Dimensional) synthetic data are presented in Table 3. These methods group data points around one center point per cluster, so the in-moon data set methods cannot find the true cluster. MKDCSTREAM is suitable for making cluster of the data-sets. Select this algorithm for the mitigation process to properly group the data-sets.

Units compare clustering costs

Average distance between each centroid in the cluster and data points. The cost is estimated by the value of the objective function, similar to the k-means algorithm.

$$\text{argmin}(k) = \sum_{x=1}^{c} \sum_{k \in K_i} ||k - a_i||Y \tag{3}$$

Table 2: Multidimensional DataSets for Test[12]

Data_set	*Dimension (d)*	*Size of the data*	*Clusters (k)*
moons	2	1500	2
3D clouds	2	16384	128
Spambase	57	4601	10
census 1990	68	2458285	10

Table 3: Data clustering result of 2D Synthetic

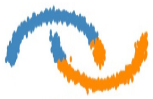
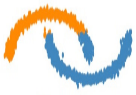
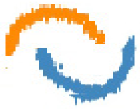
Data Set	*Mini Batch K-mean*	*Birch*	*MKDCSTREAM*
moons			

Table 4: Cost result comparison

Data Set	*Mini Batch K-mean*	*Birch*	*MKDCSTREAM*
3D clouds	158.95	157.18	152.12
Spambase	96.96	111.92	102.19
census 1990	35.36	35.47	35.31

In the above function, k are n input data, $\{K_1, K_2, \ldots K_c\}$ present c, which denotes clusters, and every one of them is presented using one center point. is A_i, a minimum cost method, cost value. Our cost result comparison is with MiniBatchKmean and Birch in Table 4.

Conclusions

For a multi-level paradigm, the strength to improve the merging behavior of the K-Means algorithm is expected to cover all cases. Although the reason for this merging behavior, which is detected in the multi-level paradigm, is not clear, it can be concluded. As mentioned before, in a layered paradigm, one solution to a gross problem must provoke a legitimate solution to the original problem. In order to obtain the final solution to the problem, after initialization at any step, the latest solution of the problem can be used to all levels of the task. In our case, we are violating this requirement. The derived level element of each object are calculated at the base level by calculating the average of the elements at two different objects. We designed a stream clustering algorithm for multi-level streaming high-dimensional data sets. Our clustering algorithm can handle an unlimited amount of large streaming data. It presents clustering results comparable to available clustering algorithms. The planned method evaluates subclusters based on large amounts of data and applies task-specific clustering algorithms to these sub-clusters.

References

[1] Bouhmala, N., Viken, A., & Lonnum, J. B. (2016). A multi-level K-Means algorithm for the clustering problem. 2016 IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA). doi:10.1109/icccbda.2016.7529544

[2] Ng, H.P., Ong, S.H., Foong, K.W.C., Goh, P.S., Nowinski, W.L.: Medical Image Segmentation Using K-Means Clustering and Improved Watershed Algorithm. 2006 IEEE Southwest Symposium on Image Analysis and Interpretation.

[3] Lee, D., Althoff, A., Richmond, D., Kastner, R.: A streaming clustering approach using a heterogeneous system for big data analysis. 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). (2017).

[4] Amini, A., Saboohi, H., Ying Wah, T., Herawan, T.: A Fast Density-Based Clustering Algorithm for Real-Time Internet of Things Stream. The Scientific World Journal. 2014, 1–11 (2014).

[5] Aggarwal, C.C., Yu, P.S., Han, J., Wang, J.: A Framework for Clustering Evolving Data Streams. Proceedings 2003 VLDB Conference. 81–92 (2003).

[6] Dash, B., Mishra, D., Rath, A., Acharya, M.: A hybridized K-means clustering approach for high dimensional data-set. International Journal of Engineering, Science and Technology. 2, (2010).

[7] Aggarwal, C.C., Yu, P.S.: A Framework for Clustering Uncertain Data Streams. 2008 IEEE 24th International Conference on Data Engineering. (2008).

[8] Dubey, A.K., Gupta, R., Mishra, S.: Data Stream Clustering for Big Data Sets: A comparative analysis. IOP Conference Series: Materials Science and Engineering. 1099, 012030 (2021).

[9] Barbará, D.: Requirements for clustering data streams. ACM SIGKDD Explorations Newsletter. 3, 23–27 (2002).

[10] Inuwa-Dutse, I., Liptrott, M., Korkontzelos, I.: A multi-level clustering technique for community detection. Neurocomputing. 441, 64–78 (2021).

[11] Ahmad, A., Dey, L.: A k-mean clustering algorithm for mixed numeric and categorical data. Data & Knowledge Engineering. 63, 503–527 (2007).

[12] Khalilian, M., Mustapha, N., Sulaiman, N.: Data stream clustering by divide and conquer approach based on vector model. Journal of Big Data. 3, (2016).

[13] Lichman, M.: UCI machine learning repository, http://archive.ics.uci.ed/ml.