IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 11, Issue 2, February 2022

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.569

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 11, Issue 2, February 2022 ||

| DOI:10.15680/IJIRSET.2022.1102115 |

Density-normalized Ant Colony Stream Clustering Algorithm for Dynamic Data Streams

Asha P.V.[1]

Lecturer, Department of Computer Engineering, Govt. Polytechnic College, Kothamangalam, Kerala, India

ABSTRACT: As we all know, streaming data needs to be processed without having access to all of the data. The proposed Density Normalized Ant Colony Stream Clustering (DNACSC) algorithm is a density-based clustering algorithm; here clusters are identified as high-density areas of the feature space separated by low-density areas. DNACSC identifies clusters as groups of micro-clusters. A window model is used to read a stream and rough clusters are incrementally formed during a single pass of a window. Refinement of the rough clusters is done using a method inspired by the observed sorting behavior of ants. Experimental results show that the clustering quality of DNACSC is better than leading ant clustering and stream-clustering algorithms. The performance can be enhanced by normalizing density among micro-clusters.

KEYWORDS: Data mining, Data stream, Clustering, Micro-clustering, Density-based micro-clustering.

I. INTRODUCTION

Data mining is widely used for summarizing data into relevant information. Not only data but data streams also can be mined. But the mining of data streams is not as simple as data mining. The mining of data streams has some additional problems compared to traditional data mining. A data stream is an unbounded sequence of data. In dynamic environments, the properties of this data may change over time in unexpected ways. The performance of traditional classifiers and predictive models may degrade as a stream progresses. The characteristics of the target objects may also change. Detecting this change is a challenge. This challenge is further compounded by the problem of labels. Because, in a streaming environment, newly arriving data is not only expensive but also time-consuming to label. Because of this, we go for unsupervised learning techniques, such as clustering. This can potentially be used to mitigate this labeling problem and also as a change detection mechanism.

Additional considerations are required for the clustering of data streams[2]. Generally, it is only possible to examine the data streams only once. So clustering needs to be performed quickly to prevent potential data loss and other related issues. A limited amount of memory becomes a challenge when streams can be potentially infinite. So it becomes necessary to summarize identified clusters in a meaningful way. As the stream progresses, clusters can appear, disappear or reappear cyclically. Therefore, it is difficult to know a priori how many clusters are present in the stream. Although traditional partitioning clustering techniques, such as k-means and its variants have been successfully applied to data streaming,they have the drawback of requiring k to be specified a priori. Density-based clustering, a form of hierarchical clustering, overcomes this limitation.

The initial clusters are discovered after the single-pass of the data. These clusters are refined using a special sorting method,which is used by ants. The basic idea is “pick-and drop”. The sorting method is to pick up isolated items and then drop at other locations where similar items are present. Another important feature is the refinement of initial clusters by probabilistically picking micro-clusters and dropping them in more suitable clusters. The probabilistic operations cause the dissolution of smaller clusters and their contents moved to similar, larger clusters. As a result of this, all points of a specific class is collected in one cluster instead of distributed in a number of smaller clusters.

Generally in other density-based streaming algorithms that use micro clusters, micro-clusters are defined by two parameters; the maximum radius ε and *minPoints* [9] which determines the minimum number of data points within ε for the micro-cluster to be considered dense. In Density Normalized Ant Colony Stream Clustering (DNACSC), each

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 11, Issue 2, February 2022 ||

| DOI:10.15680/IJIRSET.2022.1102115 |

point is initially treated as its own micro-cluster and so the *minPoint* parameter is effectively 1 and therefore not required. This removes the complication of defining micro-clusters as either core, potential or outlier[4] as each micro-cluster is treated equally. To further simplify we merge the concepts of density-reachable, directly density-reachable, and density-connected into one concept density-reachable, which determines if two micro-clusters are connected and should be considered part of the same cluster. DNACSC assigns points to a cluster before creating and merging micro-clusters so when a new micro cluster is directly density-reachable to any in the cluster; it is density-reachable and density-connected to all in the cluster. This simplifies the algorithm reduces the overall complexity. The main features of DNACSC can be summarized as follows.

1) The first phase forms initial clusters and summarize the cluster features.

2) Micro-clusters are defined using a single parameter ε.

3) Clusters are formed using a single concept of density.

4) Sorting operations are performed locally, through sampling from each cluster.

The rest of this paper is organized as follows. Section II presents related work in this area. The proposed DNACSC is described in Section III. Section IV presents the experimental study based on several real and synthetic data sets. Section V is result analysis. Section VI concludes this paper.

II. RELATED WORK

H. Azzag et al. [1] presented a new clustering algorithm for unsupervised learning. It is based on the self-assembling behavior of real ants. The ants progressively become attached to an existing support and then successively to other attached ants. Similarly, a tree will be built by the artificial ants that they have defined. Each ant represents one data. The way ants move and build this tree is completely based on the similarity between the data. The results obtained are compared with the k-means algorithm and by AntClass on numerical databases (either artificial, real). It shows that AntTree significantly improves the clustering process.

Charu C. Aggarwal et al. [2] presented a fundamentally different philosophy for data stream clustering which is guided by application-centered requirements. The idea is to divide the clustering process into an online component and an offline component. The online component periodically stores detailed summary statistics. The offline component is utilized by the analyst, who can use a wide variety of inputs to provide a quick understanding of the broad clusters in the data stream. The problems of efficient choice, storage, and use of this statistical data for a fast data stream is quite tricky. For this purpose, they used the concepts of a pyramidal time frame in conjunction with a micro clustering approach. Their performance experiments over several real and synthetic data sets illustrate the effectiveness, efficiency, and insights provided by their approach.

Thomas A. Runkler et al. [3] simplified the original ant system algorithm and provided a generalized ant colony optimization algorithm that can be used to solve a wide variety of discrete optimization problems. It is shown hard and fuzzy c-means can be optimized using particular extensions of this simplified ant optimization algorithm. Experiments with artificial and real datasets show that ant clustering produces better results than alternating techniques.

Martin Ester et al. [4] proposed a new approach for discovering clusters in an evolving data stream.The core-micro-cluster is introduced to outline the clusters with discretionary shape, while the potential core-micro-cluster and outlier micro-cluster structures are proposed to maintain and distinguish the potential clusters and outliers. A different pruning approach is designed based on these concepts, which assures the precision of the weights of the micro-clusters with limited memory. The experimental performance evaluation over several real and synthetic data sets demonstrates the effectiveness and efficiency of DenStream in discovering clusters of arbitrary shape in data streams.

LuningXia Jiwu Jing et al. [5] introduced the concept of clustering ensemble to avoid the difficulty of selecting a single appropriate threshold. Operating DBSCAN many times with distinct thresholds picked up from a pre-constructed interval, the final partition can be figured out via a consensus function. Experimental results show that this method can go above DBSCAN both in the validity and stability, and bypass the inefficiency caused by any improper thresholds.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 11, Issue 2, February 2022 ||

| DOI:10.15680/IJIRSET.2022.1102115 |

Philipp Kranen et al. [6] proposed a work in which they proposed a parameter-free algorithm that automatically adapts to the speed of the data stream. Instead of storing all incoming objects, a cluster feature tuple CF = (n, LS, SS) of the number n of represented objects, their linear sum LS, and their squared sum SS is maintained.Any cluster feature (CF) then represents a micro-cluster.They propose maintaining cluster features (CFs) by extending index structures from the R-tree family. Such hierarchical indexing structures provide the techniques for accurately locating the right place to insert any object from the stream into a micro-cluster. The idea is to build a hierarchy of micro-clusters at different levels of granularity.If this micro-cluster is related enough, it is amended incrementally by this object's values. Otherwise, a new micro-cluster may be formed.

RashmiDuttaBaruah and PlamenAngelov [7] developed a new online evolving clustering approach for streaming data is proposed. This approach uses cluster weight and distance before making new clusters. The dynamics of the data stream is defined by the cluster weight. Which is described in both data and time-space in such a way that it decays exponentially with time.Concepts from computational geometry is also applied to determine the neighborhood information while forming clusters. The real outliers can be effectively identified by making a distinction between core and noncore clusters. The approach not only adds rules but also removes them and thus maintains a lower complexity of models. This not only saves memory space but also reduces the estimation time. Furthermore, it enhances the speed by detecting and removing noncore clusters at the appropriate time.

NesrineMasmoudi et al. [8] presented in this paper a new bio-inspired algorithm that dynamically creates groups of data. This algorithm is established on the concept of artificial ants that move together in a complex way with simple localization rules. Each ant represents one datum in the algorithm. The moves of ants aim at creating homogeneous groups of data that evolve together in a graph environment. He also suggests an extension of this algorithm to treat data streaming. That extended algorithm has been tested on real-world data. Their algorithms yielded competitive results as compared to K-means and Ascending Hierarchical Clustering (AHC), two well-known methods. Their method is based on ants system is well known to be adaptive and efficient to dynamical situations.

Shengxiang Yang et al. [9] presented a bio-inspired approach to clustering non-stationary data streams. The proposed algorithm, Ant-Colony Stream Clustering (ACSC), is based on the concept of artificial ants. That identifies clusters as nests of micro-clusters in dense areas of the data. Micro-clusters are defined as N-dimensional spheres with a maximum radius ε. In ACSC the ε-neighborhood, crucial in density clustering is adaptive and doesn't require expert, data-dependent tuning. The sliding window model is used in this algorithm and summary statistics for each window are stored offline. From the experimental results over real and synthetic data sets, it is found that the clustering quality of ACSC is promising than leading stream-clustering algorithms. Moreover, less computation is required with fewer parameters.

ConorFahy et al. [10] presented an online, bio-inspired approach to clustering dynamic data streams. Their proposed Ant Colony Stream Clustering (ACSC) algorithm is a density-based grouping approach. It identifies clusters as high-density areas of the feature space separated by low-density areas. ACSC identifies clusters as groups of micro-clusters. The stream of data is read by a tumbling window. Initial clusters are incrementally formed during a single pass. Several points can be moved in a single operation. This speeds up the algorithm with just a minor expense to execution .The rough clusters are then refined using a method inspired by the observed sorting behavior of ants. Ants pick-up and drop items depends on the similarity with the surrounding items. Artificial ants sort clusters by picking and dropping micro-clusters based on local density and local similarity. Micro clusters summary statistics are stored offline and are used for creating final clusters. It is clear from the experimental results that the clustering quality of ACSC is versatile, robust to noise and higher than leading ant clustering algorithms. It also requires fewer parameters and less computational time.

III. PROPOSED SYSTEM

DNACSC employs the window model when dealing with data streams. DNACSC identifies clusters as a group of micro-clusters, and a micro-cluster is a set of neighboring points within a certain radius; the ε-neighborhood determines this radius. A micro-cluster is described using three components [6].

1) The number of data points the micro-cluster contains (N).

2) The linear sum (LS) of each dimension.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 11, Issue 2, February 2022 ||

| DOI:10.15680/IJIRSET.2022.1102115 |

3) The squared sum (SS) of each dimension.

LS and SS are d-dimensional arrays, where d is the number of dimensions in a point. From these, the radius r and center c of the micro-cluster [10] can be determined by

$$c = \frac{\mathrm{LS}}{\mathrm{N}}$$

$$r = \sqrt{\frac{\mathrm{SS}}{\mathrm{N}} - \left(\frac{\mathrm{LS}}{\mathrm{N}}\right)^2}$$

LS and SS have the properties of additivity and increment-ability, which allow micro-clusters to absorb new data points and to merge with other micro-clusters. A micro-cluster m can absorb point p if, after updating the LS and SS of m with p, radius $(m) \leq \varepsilon$. Similarly, two micro-clusters m_i and m_j can attempt to merge [10]into a single micro-cluster m_k as follows:

m_k =(Ni +Nj, LSi + LSj , SSi +SSj).

If radius(m_k) $\leqslant \varepsilon$, the clusters merge; otherwise, the merging operation fails.

Micro-clusters m_i and m_j are said density reachable [9] if *dist* (c_{mi},c_{mj}) $\leqslant \varepsilon$.

where$c_{m_i}$and $c_{m_j}$are the centers of micro-clusters m_i and m_j respectively.

The solution provided by DNACSC is a set of clusters containing density-reachable micro-clusters.

The proposed system is shown in Fig. 1

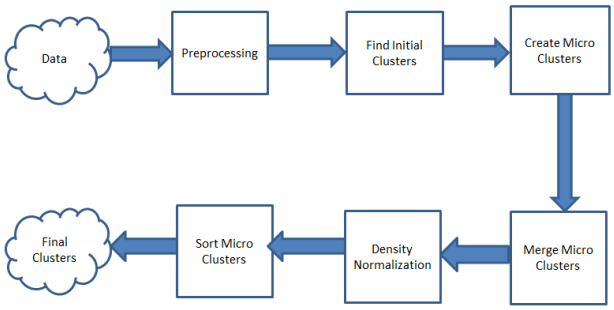


Fig 1. Proposed System

A. Find Clusters

DNACSC employs the window model when dealing with data streams. At each iteration, a fixed size non overlapping chunk of data is considered. So, at the beginning of this step, there will be *WindowSize* [9]points. In a single-pass of the window, clusters are incrementally formed. The first point seeds the first cluster. Subsequent points are assigned to an existing cluster or used to seed anew cluster. Point p's suitability with cluster c is evaluatedusing theEuclideandistance from p with a sample *nSamples*taken from c. The suitability of point p with cluster c is estimated as follows[10]:

$$\text{Suitability } (p,c) = \frac{\sum_{j=1}^{\min(nSamples,n)} dist(p,c_j)}{\min(nSamples,n)}$$

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 11, Issue 2, February 2022 ||

| DOI:10.15680/IJIRSET.2022.1102115 |

where n is the number of points that are present in the cluster. Each cluster is evaluated and the point is assigned to the most suitable one provided the suitability is equal to or below ε; if not, the point seeds a new cluster. The parameter determines the maximum radius for a micro-cluster in the subsequent step and also serves as the minimum suitability measure in this step.

As we estimate each point's suitability with each cluster, we record each suitability. This is useful information, especially when afforded just a single pass of the data. Upon joining (or establishing) a cluster, we update the similarity information between the selected cluster and its neighboring clusters. The similarity between clusters *a* and*b* is the average of each point *p* in cluster *a*'s suitability with cluster *b* [10].

Similarity $(a,b) = \frac{1}{n}\sum_{i=1}^{n} Suitability(a_i, b)$

The similarity to each neighboring cluster is a rolling average updated whenever a new point is assigned to the cluster. Although comparable, Similarity $(a,b) \neq$ Similarity (b,a).

Algorithm: Find Clusters[10]

1. **while**<Window>**do**
2. **for**<each data point >**do**
3. **if**<clusters>**then**
4. Find best cluster
5. Add point to cluster
6. Update cluster similarity
7. **else if**< No clusters ||
   No suitable cluster >**then**
8. Create new cluster
9. Add point to cluster
10. Update cluster similarity
11. **return** Clusters, Cluster Similarity

B. Merge Clusters

The previous step discovers clusters in a single-pass of the window. The clusters identified at this stage are often rough, impure and too-many. Then, micro-clusters need to be created, merged, and intercluster sorting is performed.

Initially, each d-dimensional point p in each cluster is treated as its own micro-cluster m. This micro-cluster will have a radius of 0 and a center of *p*. Formally, we have

$m.N = 1$
$m.LS_i = p_i, i=\{1,...,d\}$
$m.SS_i = p_i^2, i = \{1...d\}$

wherep_i is the i^{th} dimension of point *p*[10].
Before sorting, each micro-cluster attempts to merge with other micro-clusters in the same cluster. The merging operation is performed by comparing each micro-cluster with every other in the same cluster. If, after summing their constituent parts, the radius is less than or equal to ε, the mergingoperation is a success. Merging at this stepensures that only neighboring micro-clusters attempt to merge and avoids the unnecessary computation of comparing micro-clusters in different dense areas.

Algorithm: Merge Clusters[10]

1. Create a new empty micro-cluster *c*
2. Initialise *c*: = *a*
3. Add *b* to *c*
4. *r* := radius of *c*

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 11, Issue 2, February 2022 ||

| DOI:10.15680/IJIRSET.2022.1102115 |

5. **if**($r \leq \varepsilon$) **then** merge successful
6. Delete *a* and *b*
7. Return *c*
8. **else**
9. Delete *c*
10. Return **false**

C. Sort Clusters

During the sorting phase, n points represented by a micro-cluster can be moved in a single operation, speeding up the sorting process and reducing the number of pairwise comparisons. A micro-cluster m is chosen at random from cluster k and is iteratively compared with *nSamples* micro-clusters in the same cluster. The Euclidean distance from the center of m to each of the selected micro-clusters is calculated and if both are density-reachable, then a reachable count is incremented. Then probability of a pick-up is calculated using the reachable count.

$$P_{pick} = 1 - \frac{reachable}{nSamples}$$

If the probability of pick is greater than a threshold value, then all the micro clusters in cluster k is picked and dropped to their most similar clusters. After that the old cluster is deleted and cluster similarity is updated.

Algorithm: Sort Clusters

1. Create micro-clusters
2. Merge micro-clusters in each cluster
3. Calculate normalized density
4. **for**< each cluster **C** >**do**
5. select a random micro-cluster *m*
6. select*nSamples* from **C**
7. Calculate P_{pick}
8. **if**<$P_{pick} \geq$ Threshold >**then**
9. Set Carrying :=**true**
10. Add **C** to Dropped
11. **while**< Carrying >**do**
12. **for**< all micro-clusters in **C** > **do**
13. **for**< all Clusters >**do**
14. Calculate Similarity t_{sim}
15. Find cluster $\mathbf{C}_{max}$ with MAX(t_{sim})
16. Drop micro-cluster into C_{max}
17. Update Similarity
18. Delete cluster **C**
19. **return** Clusters

IV. EXPERIMENTAL SETUP

The performance of DNACSC is evaluated on both stationary and nonstationary datasets. DNACSC is compared with three popular ant clustering algorithms. The performance of these algorithms is taken from results already published in the literature. Similar to ACSC, DNACSC also employs micro-clusters to identify dense areas of the stream.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 11, Issue 2, February 2022 ||

| DOI:10.15680/IJIRSET.2022.1102115 |

A. Performance Metrics

DNACSC is evaluated across the following metrics:

1. *Purity.*
2. *F-measure*

Each dataset used is labeled and the ideal "correct" clustering solution is known, so performance is measured with respect to this ground truth. In each metric, a bad clustering will have a value close to 0 and an ideal clustering solution will have a value of 1. The purity metric measures how homogeneous a cluster is. A cluster is assigned to the class which appears most frequently within the cluster, the accuracy of this is evaluated by summing the instances of this class and dividing by the total number of instances in the cluster.

In the following, R represents the clustering result returned by the algorithm. R contains n clusters. In every identified cluster R_i (i ={ 1,...,n}), V_i represents the most frequently appearing class label in cluster R_i, V_{isum} is the number of instances of Vi in R_i, and V_{itotal} represents the total number of instances of Vi in the current window. From these, we define the following features for cluster R_i:

$$\text{precision}_{\boldsymbol{R_i}} = \frac{\boldsymbol{V}^{\boldsymbol{i}}_{\text{sum}}}{|R_i|}$$

$$\text{recall}_{\boldsymbol{R_i}} = \frac{\boldsymbol{V}^{\boldsymbol{i}}_{\text{sum}}}{\boldsymbol{V}^{\boldsymbol{i}}_{\text{total}}}$$

$$\text{Score}_{R_i} = 2 * \frac{\text{precision}_{\boldsymbol{R_i}} * \text{recall}_{\boldsymbol{R_i}}}{\text{precision}_{\boldsymbol{R_i}} + \text{recall}_{\boldsymbol{R_i}}}$$

1. *Purity*

The purity metric measures how homogeneous a cluster is. A cluster is assigned to the class which appears most frequently within the cluster, the accuracy of this is evaluated by summing the instances of this class and dividing by the total number of instances in the cluster.

Overall, purity (P) can now be expressed in terms of the total number of clusters identified, as follows:

$$\text{Purity (P)} = \frac{1}{n}\sum_{i=1}^{n} \text{precision}_{R_i}$$

2. *F-measure*

The F-measure (sometimes called F-score or F1-score) is the harmonic mean of the precision and recall scores obtained by the algorithm.

F-measure (F) can now be expressed in terms of the total number of clusters identified, as follows:

$$\text{F-measure (F)} = \frac{1}{n}\sum_{i=1}^{n} \text{Score}_{R_i}$$

B. Datasets

DNACSC is compared with other ant-based clustering solutions across three well known and popular nonstationary datasets; Iris, Wine, and Zoo. These datasets were taken from the UCI Machine Learning Depository

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569|

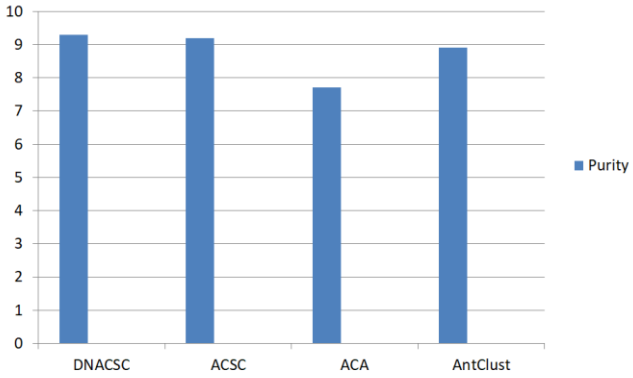
|| Volume 11, Issue 2, February 2022 ||

| DOI:10.15680/IJIRSET.2022.1102115 |

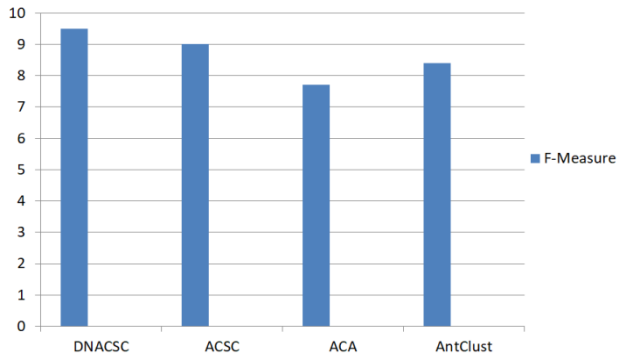
V. RESULT ANALYSIS

Implemented the new system and it has the following features: The new system is better than the existing system in terms of Purity. There is no data loss in the new system compared to the existing systems. A new technique called Density Normalization is added to the new system.

Purity Analysis



F-measure Analysis



Performance Comparison

Algorithm	Purity(%)	F-measure(%)	Data loss
DNACSC	93	95	No
ACSC	92	90	Yes
ACA	77	77	Yes
AntClust	89	84	Yes

The new system is currently implemented and installed. The implementation is based on the existing system but it shows improved performance compared to existing system. It also has an additional feature for Density Normalization.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 11, Issue 2, February 2022 ||

| DOI:10.15680/IJIRSET.2022.1102115 |

VI. CONCLUSION

A density normalized algorithm DNACSC is implemented for clustering dynamic data streams. In this, clusters are formed in a single pass of the data. The initial clusters discovered are further refined using a method inspired by the sorting behavior of ants. This sorting method is based on the classic pick-and-drop ant clustering algorithm. The probabilistic functions for picking and dropping are biased toward the dissolution of smaller clusters and incorporating their contents into similar, larger clusters. A new method of density normalization is introduced and density is normalized before the pick and drop operations. This improves the purity of clustering. Moreover, all the micro-clusters are moved to similar clusters before deleting the old cluster and any data point will not be left as outliers. So, there is no data loss. In the traditional algorithm, data points are moved individually which can take a long time. By grouping similar points into micro-clusters, a number of points can be moved in a single operation, which speeds up the algorithm. DNACSC was shown to outperform other ant-based clustering algorithms in the literature.

The ε parameter is shown to be very sensitive and greatly affects the performance of the DNACSC algorithm. It is data-dependent and requires manual tuning. So the investigation for an adaptive, local ε parameter can be extended as future work.

REFERENCES

[1] H. Azzag, N. Monmarche, M. Slimane, and G. Venturini, "AntTree: A new model for clustering with artificial ants," in Proc. IEEE Conf. Evol. Comput., vol. 4. Canberra, ACT, Australia, 2003, pp. 2642–2647

[2] Charu C. Aggarwal, T. J. Watson Resch. Ctr.Jiawei Han, Jianyong Wang, UIUC, Philip S. Yu, T. J. Watson Resch. Ctr., "A Framework for Clustering Evolving Data Streams," Proceedings of the 29th VLDB Conference, Berlin, Germany, 2003.

[3] Thomas A. Runkler Siemens AG Corporate Technology, Information, and Communications, 81730 Munich, Germany, " Ant Colony Optimization of Clustering Models," in INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 20, 1233–1251, 2005.

[4] F. Cao, M. Ester, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," in Proc. SIAM Int. Conf. Data Min., vol. 6, 2006, pp. 328–339.

[5] Luning Xia Jiwu Jing, "An Ensemble Density-based Clustering Method," Information Security State Key Laboratory, Graduate University of Chinese Academy of Science, Beijing 100049, P. R. China,2007.

[6] P. Kranen, I. Assent, C. Baldauf, and T. Seidl, "The ClusTree: Indexing micro-clusters for any time stream mining," Knowl. Inf. Syst., vol. 29, no. 2, pp. 249–272, 2011.

[7] R. D. Baruah and P. Angelov, "DEC: Dynamically evolving clustering and its application to structure identification of evolving fuzzy models," IEEE Trans. Cybern., vol. 44, no. 9, pp. 1619–1631, Sep. 2014.

[8] Nesrine Masmoudi, Hanane Azzag, Mustapha Lebbah, Cyrille Bertelle, Maher Ben Jemaa, "How to Use Ants for Data Stream Clustering," in Proc. IEEE Conf. Evol. Comput., vol. 4. Canberra, ACT, Australia2015.

[9] Conor Fahy and Shengxiang Yang, "Dynamic Stream Clustering Using Ants," Published in UKCI 2016.

[10] Conor Fahy, Shengxiang Yang, Senior Member, IEEE, and Mario Gongora, "Ant Colony Stream Clustering: A Fast Density Clustering Algorithm for Dynamic Data Streams," in IEEE TRANSACTIONS ON CYBERNETICS, VOL. 49, NO. 6, JUNE 2019.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor:
7.569

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details