IASJ المجلات الاكاديمية العراقية

Tikrit Journal of Pure Science

ISSN: 1813 – 1662 (Print) --- E-ISSN: 2415 – 1726 (Online)

Journal Homepage: http://tjps.tu.edu.iq/index.php/j

DED: Drift Principle in Educational Evolved Data

Ammar Thaher Yaseen Al Abd Alazeez

Computer Science Department , Computer Science and Mathmatics College, University of Mosul, Mosul , Iraq

https://doi.org/10.25130/tjps.v26i2.128

ARTICLE INFO.

Article history:

-Received: 10 / 12 / 2020

-Accepted: 2 / 2 / 2021

-Available online: / / 2021

Keywords: Big Data; Data Stream Clustering Algorithms; Clustering Educational Data

Corresponding Author:

Name: Ammar Thaher Yaseen

E-mail:

ammarthaher@uomosul.edu.iq

Tel: 07732826477

ABSTRACT

Clustering data streams is one of the prominent tasks of discovering hidden patterns in data streams. It refers to the process of clustering newly arrived data into continuously and dynamically changing segmentation patterns. This article presents a stream mining algorithm to cluster the data stream with focusing on its evolution and concept drift. Even though concept drift is expected to be present in data streams, explicit drift detection is rarely done in stream clustering algorithms. Concept drift is caused by the changes in data distribution over time. Relationship between concept drift and the occurrence of physical events has been studied by applying the algorithm on the education data stream. Viber education data streams produced by Viber Groups in our Computer Science Department are used to conduct this study. The results show that our proposed algorithm superiority existing ones in purity, entropy, and sum of square error measurements. Experiments led to the conclusion that the concept drift accompanied by a change in the number of clusters and outliers indicates a significant education event. This kind of online monitoring and its results can be utilized in education systems in various ways, such as present the capabilities of participants.

Introduction

With the spread of the COVID 19 pandemic, schools, universities, and educational organizations have become heavily dependent on platforms, systems, and ready-made programs that help lecturers, students, departments, and colleges to complete academic preparations from a distance. This has formed a large group of a lot of data which summarized in having a large problem in terms of having a smooth and flexible operation of managing such data. This issue would make the data simultaneously available, and anyone may rapidly have it. In addition, this expedites the analysis of this data to enhance educational performance and encourage basic research on learning. In the educational part, data analysis can have main impact on all employees from lecturers to students and even leads in educational management. A lot of educational organisations collect a large amount of data every day in the form of attendance and participation activities, students' problems, teacher's evolutions, and details about their social and economic state. Therefore, such "Big Data" gathering and analysis follows allow educational organisation to start give more special education. Viber Groups are one of the familiars' applications that might be used to facilitate educational process.

Recently, data stream mining has become a tremendous measure of consideration [1]. An data stream is an arrangement of constantly arriving data which forces a solitary pass limitation where arbitrary access to the data isn't practical [2]. The speed of data appearance, just as the speed of data handling, may fluctuate from application to application. For certain applications, the appearance and handling of data can be acted in a disconnected group examination design, others require constant and continuous investigations; here and there it requires prompt activity upon the preparing of approaching data streams, for example, dynamic administration of data centres [3]. Data stream mining can be characterized as the way toward finding hidden patterns inside a huge volume of unbounded data streams. In such cases, mining methods must notice and acknowledge existence limitations and must have the option to find right hidden designs inside the requirement limits. Data stream clustering methods mean to find clustering structures (clusters) basic the stream data as per similitudes between their highlights [4]. Data arriving in streams regularly con-

tain anomalies, which may have equivalent significance as clusters. Along these lines, it is attractive for data stream clustering methods to have the option to identify the outliers just as the groups [5].

A significant test in data streams investigation is concept drift, where the example encoded in the stream changes after some time. Concept drift exists in all real life issues, for example, occasional climate changes, stock market downturns, rallies, computer network traffic, remote sensor data, telephone discussions, web-based media, promoting data, web pages, power utilization meters, online opinion investigation, intrusion and fraud detection, etc. Concept drift are ordinarily unexpected and erratic. There are various types of concept drifts, for example slow drift, where past and new examples encoded in the stream overlay for a brief timeframe, unexpected drift , where the example in the stream shifts suddenly, the new patterns appear in a split second and the old example vanishes simultaneously. At that point there are incremental or evolving streams, where the example doesn't remain stable and changes continually, and, re-happening drifts, where recently observed and ceased designs re-happen. Obviously, there are likewise mixes of these sorts of streams. A data mining model ought to consistently respond the current concept and hence needs to adjust rapidly [6].

These changes cause that predictive models trained over data streams become ultimately obsolete, not adjusting appropriately to the new distribution (concept). The unpredictability of conquering this issue, and its pervasiveness over numerous real situations, make concept drift recognition and adaptation recognized difficulties in data streams [7].

This paper investigates extensively the existing literature in the field of data stream clustering and identifies the essential processing units underpinning various existing algorithms. The paper then proposes a strategy to find concept drift in educational data streams toward providing a response to evolving environments in near-real time. A concept drift is the change of the distribution of the feature vector in relation to its class overtime. Practically, our research results can benefit a range of real-time big data applications such as sensor network monitoring, social media data analysis, etc.

Related Work

Numerous real-world data mining applications need to manage unlabelled streaming data. They are unlabelled on the grounds that the sheer volume of the stream makes it illogical to name a huge data. The data streams can evolve after some time and these progressions are called concept drifts. Concept drift have various qualities, which can be utilized to classify them into various sorts. An assortment of strategies have been given to the subject of concept drift recognition with unlabelled data, yet these methodologies frequently are generally appropriate for just a subset of the concept drift types [8]. Scientists have attempted to deal with concept drift through various methodologies after some time, which incorporates sliding window versus batch processing, fixed length window versus variable length window, fixed stream speed versus variable stream speed, named versus unlabelled or halfway named data points, overlooking the handling consequences of a window versus keeping a background marked by the drift proportion of every window, utilizing a drift ready instrument versus keeping up drift logs to be utilized later, etc. The entire concept drift location measure has advanced after some time by settling on such decisions and upgrading the cycle steadily. Scientists have proposed numerous concept drift discovery techniques as of late, which have taken care of this issue in various manners for various circumstances [9].

A fundamental execution of online process mining concentrate on Cognitive Computing was proposed by Barbon *et al*. The authors suggested a system to discover concept drift in business data streams toward giving a reaction to advancing conditions in near-real time. A concept drift is the difference in the distribution of the component vector corresponding to its group over time. The outcomes were promising, yet they didn't consider data streaming execution measurements while tending to the irregular conduct. A conventional meaning of concept drift was presented by Kelly *et al.* [10]. Here, a concept at time moment *t* is characterized as a group of probabilities of the classes and class-conditionals:

$CD = \left(P(c_1), P(x^t|c_1)\right); (P(c_2), P(x^t|c_2)); \dots; (P(c_m), P(x^t|c_m)$ …(1). By checking changes in this arrangement of probabilities *CD*, an data stream model can recognize and adjust as indicated by idea drifts [10]. The main characteristics of the data streams can be found in [11]. The dynamic idea of data streams may be found in [12]. In addition, the computational approaches can be found in [13] [4].

1.1 Evolution and Concept Drift in Data Streams Clustering

Cluster evolution includes five principle types: group appearance (Birth), group vanishing (Death), group parting (Split), group combining (Merge), and group perseverance (Survival). The Birth happens when another group is shaped by recently arrive data points. The Death happens when data points are not, at this point related with a specific cluster, making that cluster ultimately vanish. The Merge happens when data points from two groups cover essentially, showing a consolidation of two groups. The Split happens when data points inside a cluster begin to separate because of approaching new data points that include enraptured two places inside the limit of the first cluster. Likewise, cluster(s) that have not been fundamentally influenced by advancement is viewed as Survival groups [14]. Figure 1 represents different instances of group advancement over a period$(t, t+1, t+2)$.

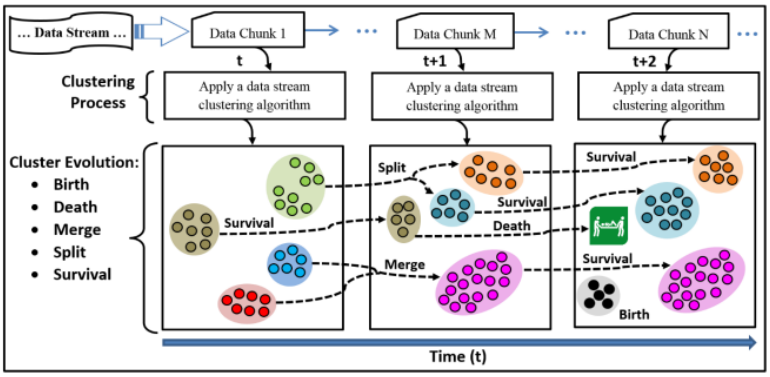

Fig. 1: Cluster evolution scenarios ([14]**)**

Concept drift implies an adjustment in the profile of a current concept. In data stream clustering, concept drift is a normal marvel in the accompanying sense. The statistic model, for example one or a combination of distributions, producing the instances of each stream may change after some time. All in all, concept drift is reflected by truth that the clustering structure accumulated with the past data in the past time (t), is not, at this point substantial in the present status under perception at time $(t+1)$ or $(t+2)$ that may speak to new relations of proximity between data in the streams (see Figure 2). In this manner, the learning specialist must forget previous tense data, and grasp the changes. The cluster development history explained before reflects the concept drift over the long time. Creative methodologies are expected to think about this chance of progress and data stream clustering strategies must have the option to manage concept drift issue [15].

The focal issue of concept drift in data stream clustering can be considered as in a faded window model in which the weight of every data object diminishes dramatically with time *t* through a fading function *F*: $F(t) = 2^{-\lambda.t}$ …(2), where $0 < \lambda < 1$ characterizes the rate of decay of the weight over time and $t = (t_c - t_o)$, where t_c indicate the current time and t_o is the make/change time of the data point [16]. The dramatically fading function is broadly utilized in transient applications where it is attractive to slowly limit the historical backdrop of past conduct [17]. The higher the estimation of λ is, the lower significance of the recorded data contrasted with later data, for example λ controls the significance of the verifiable data (see [16] for more detail of the faded function). Thusly, most of the data stream clustering methods (for example DenStream) are embraced and embedded with the fading function to adapt to inconsistent of concept drift over the long time. In this paper, we follow a similar course of adaptation of the fading function to reflect and examine the progressions of the data streams over approaching data chunks. The requirement of data stream clustering algorithms can be found in [18] and the promises and challenges of data stream clustering can be found in [19] [13].

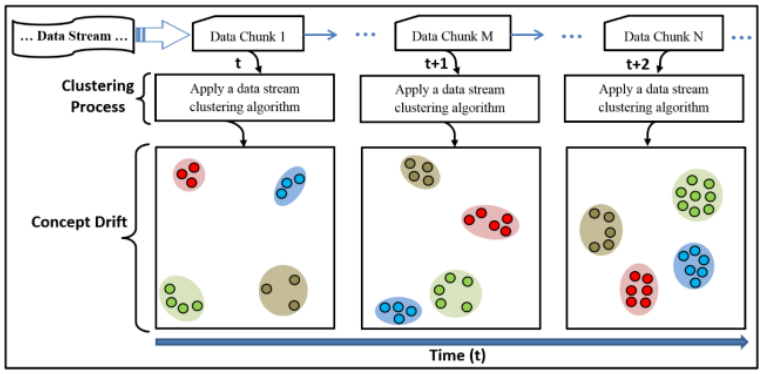

Fig. 2: Illustrative example of concept drift

1.2 Existing EINCKM and EDDS Algorithms for Clustering Data Streams

• EINCKM Algorithm

EINCKM (enhanced incremental K-Means) [20] is an incremental strategy for clustering model data streams. It relies upon a conventional system for data stream clustering that includes three primary measures [20][21] Build Clusters (BC), Merge, and Prune. Build Clusters incorporates the clustering method that used to discover the groups from input data chunk, Merge (stage 2) is utilized to coordinate the new and existing arrangement of clusters, and Prune (stage 3) is to recognize outliers and check the fading cycle. The technique utilizes a heuristic way to deal with foresee the *K* (number of clusterings), a radius figuring to join overlapped clusterings and a variance way to deal with distinguish the outliers. The strategy is adaptable and prepared for additional

upgrade. Nonetheless, the strategy created to introduce curved shape groups. As such, it doesn't distinguish right groups on the non-convex shapes.

• **EDDS Algorithm**

EDDS (enhanced density data stream) [21] is incremental based strategy for clustering data streams. It seeks after a comparable system for data stream clustering that incorporates three standard steps; Build Clusters (BC), Merge, and Prune. The technique distinguishes clusterings and outliers. It changed the ordinary DBSCAN (density-based spatial clustering of applications with noise) technique to gather each clustering with respect to a great deal of surface-core data records. The strategy executes the density reachable thought of DBSCAN as its consolidating procedure and prunes within cores using a heuristic action. The strategy similarly removes the old clusters and outliers depending upon a fading function. In any case, this method has high execution time contrasting to EINCKM. Moreover, it recognizes curved and non-raised shape clusters.

Proposed DED Algorithm

The method is intended to identify new clusters in an approaching data chunk, combine new groups and existing outliers to the presently existing groups, produce altered groups and anomalies prepared for the following round, and installing concept drift. Figure 3 shows the pseudo-code of the DED method. The data sources are data chunks of size *M*, a pool of outliers, the base number of data points per group, and a group of existing clusters outline. Each cluster outline is a tuple (N, LS, LSS, μ, R), where N is various data points, LS is the straight amount of the data points, LSS is the amount of squared data points, μ is the centroid, R is the radius. The outputs are K clusters and outliers. The method recall for each new data chunk arrives.

DED Algorithm:

Inputs:

CH : Data chunk of size M. Initially *CH = {} //Just in the beginning is empty, i.e. first round*

CF: Set of clusters summary (N, LS, LSS, μ, R); // Previous clustering summary of *T* clusters. Initially, *CF = {} // CF is cluster feature*

Po: Pool of outliers; //Previous Pool of *W* outliers. Initially, *Po = {}.*

$MinPts$: Minimum number of data points per cluster.

Outputs:

$CF`$: Modified CF;

$Po`$: Modified Po;

Algorithm Steps:

1. $CH = CH \cup Po$;
2. $cf = EINCKM(CH, K, Ini)$ *or* $cf = EDDS\ (CH, Eps, MinPts)$;// *cf* is a structure contains cf.member as cluster members & cf.summary as cluster summary. In this step we identify all possible clusters that might be in the new incoming chunks
3. Calculate cluster summary for S_i//i.e. *N*, *LS*, *LSS*.
4. $CF = Merge(CF, cf)$; //Merge overlapping clusters. If we call EINCKM we depend on overlapped clusters' radiuses & if we call EDDS we will depend on density-reachable
5. $\langle CF, Po\rangle = Prune(CF, cf, MinPts)$; //Filter outliers

Fig. 3: Pseudo-code of the proposed algorithm

The *Build Clusters* step in our method comprises of two functions EINCKM and EDDS. The *Merge* step in the overall system is executed by a solitary *Merge* work. The pruning step of the overall system is executed by a solitary *Prune* work as appeared in Figure 4. The function utilizes the variance of each cluster and *MinPts* number to separate the anomalies from the clusters delivered by the *Merge* process. The function filters the data points in every last group and contrasts their distance to the centroid and the radius of that cluster. In the event that the distance is more noteworthy than the radius, the data point is considered as an outlier.

Prune Function:

$\langle CF, Po\rangle = Prune\ (CF, cf, MinPts, \lambda)$

$For\ i = 1\ to\ size(cf)$

{ $For\ j = 1\ to\ size(S_i)$ // cf.member

{ $D = dist(p(j), \mu_i)$; // p is a data point in S_i

$If\ D > R_i\ \ then\ \ Po = Po \cup p(j);$ }

$If\ size\ (S_i) < MinPts\ \ then\ \ Po = Po \cup S_i;$

Update cluster summary for S_i }

$For\ i = 1\ to\ size(CF)$ // Check the aged core points

$If\ (t_c - t_o) = F(t)\ \ then\ \ CF(i) = \{\};$

$For\ i = 1\ to\ size(Po)$ // Check the aged outliers

$If\ (t_c - t_o) = F(t)\ \ then\ \ Po(i) = \{\};$

Fig. 4: Pseudo-code for the prune function

Implementation DED Algorithm in Clustering Data Streams

1.3 Application dataset

Numerous real-world applications can be found like weather, fashion, and consumer habits etc. where the concepts recur when the corresponding contexts repeat. Concept recurrence is a common scenario in many real-world applications. When concepts are related to hidden contexts, re-appearance of contexts leads to concept recurrence. The changes in season, economic conditions, fashion etc. are examples of contexts that drive related concepts in real-world streams. In this research we will depend on educational data streams as case study and focuses on Viber group in our Computer Science Department. Viber stream data of our department is used for this experiment. Data collected at 6 months interval contains nine parameters; Event_ No., Event_Release, Event_Type, Start_Date, Start_Time, End_Date, End_Time, Attach, and Response. The stream starts 1/1/2020 and finishes in 1/7/2020. The off-line clustering is performed on the initial 50 samples to create the first model. The following Table 1 includes data about features that summarizes from the topics and response messages that have been made.

Table 1: Sample of Viber Group data stream in Computer Science Department

Event No.	Event_Release	Event_Type	Start_Date	Start_Time	End_Date	End_Time	Attach	Response
1	Head	Congratulation	1/1/2020	11:19am	1/1/2020	12:00pm	1	31
2	Participant	Intimation	1/1/2020	11:45am	1/1/2020	2:20pm	0	6
3	Participant	Intimation	1/1/2020	3:34pm	1/1/2020	7:08pm	1	15
…	…	…	…	…	…	…	…	…
228	Participant	Intimation	30/6/2020	11:41pm	30/6/2020	11:49pm	0	17

1.4 Embedding the Concept Drift into the Prototype-based Clustering Algorithm

To accommodate concept drift in the prototype_based methods (for instance EINCKM method), data is maintained about when each cluster was made and when it was last altered. In the Prune process, a fading function is presented utilizing the equation (2) like different works detailed in [16]. With the fading function, the accompanying advances are taken to deal with concept drift issue with regards to a lifetime L (a user-defined threshold, for example L=4 min.) and a fading boundary λ (we set λ to 0.2 in our tests):

1. Makes and keeps up two timestamp tables:

A. CTS (group timestamps) containing sections [centroid, timestamp] for each group.

B. OTS (outlier timestamps) containing sections [outlier, timestamp] for every anomaly.

2. For every emphasis of preparing new approaching data chunk do:

A. Figures the estimation of the fading function F utilizing the equation (2);

B. Diminishes the timestamp by F sum for CTS and OTS;

C. Sets the timestamp for new centroids and exceptions to a limit L;

D. For combined groups, refreshes all influenced centroids by setting the timestamp to L;

E. Erases all centroids and anomalies which have lapsed (for example L=0).

1.5 Embedding the Concept Drift into the Density-based Clustering Algorithm

To accommodate concept drift in the density_based method (for instance EDDS method), data is maintained about when each cluster was made and when it was last adjusted. In the *Prune* process, a fading function is presented utilizing the equation (2). To deal with concept drift issue, like the methodology taken in Section 4.2, in the *Prune* step, we present a fading function that:

1. Sets a fading boundary λ to 0.2 in our investigations.

2. Makes and keeps up two tables for timestamp checking:

A. CTS containing passages [surface-centers, timestamp] for each group.

B. OTS containing passages [outlier, timestamp] for every anomaly.

3. For every emphasis of preparing new approaching data chunk do:

A. Figures the estimation of the fading function F utilizing the equation (2);

B. Decreases the timestamp by F sum for CTS and OTS;

C. Sets the timestamp for new surface-centers and anomalies to L (for example L=4);

D. For consolidated groups, set the timestamp to L for all influenced surface-centers;

E. Erases every surface-center and outliers which have terminated (for example L=0)

1.6 EVALUATION oF DED METHOD

Various ways to deal with assess clustering method execution exist in the writing [22]. To assess the accuracy of the proposed method, we have chosen to utilize the assessment by the reference technique, for example to discover if the method can restore the known clusters in a given ground truth. To assess accuracy, we utilized three usually utilized evaluators: Purity (was utilized in [23]), Entropy in [24], and the Sum of Square Error (SSE) in [4]. The general formula of SSE described by the equation, $SSE = \sum_{i=1}^{K} \sum_{x_j \in C_i} |x_j - c_i|^2$ ... (3. Purity of particular cluster C_i of size k_i, is identified to be $P(C_i) = \frac{1}{k_i} max_j(k_i^j)$, where k_i^j is the number of data points of the j^{th} class that were assigned to the i^{th} output

cluster. The final purity of the output clusters is calculated as a weighted sum of every cluster purity, $Purity = \sum_{i=1}^{n} \frac{k_i}{k} P(C_i) \dots (4)$, where n is the number of classes and k is the total number of data points in the ground truth. The entropy of given specific cluster C_i of size k_i is identified by $E(C_i) = -\frac{1}{\log p} \sum_{i=1}^{p} \frac{k_i^j}{k_i} \log \frac{k_i^j}{k_i}$, where p is the number of groups in the ground truth, k_i^j is the number of data points of the j^{th} class that are assigned to the i^{th} output cluster. The entropy of the whole clusters is then calculated as the sum of every cluster entropy weighted by cluster size;

$Entropy = \sum_{i=1}^{n} \frac{k_i}{k} E(C_i) \dots (5)$.

MATLAB was utilized to develop EINCKM, EDDS, and DED methods and the test structure. We split a given dataset into two sections: the dynamic arriving data chunks of a specific size and the underlying dataset before the appearance of the primary powerful data chunk. We arbitrarily chose an underlying assortment of 50 data points as the underlying dataset and the leftover 178 were haphazardly chosen as data points in the dynamic chunks. Our proposed method doesn't treat the underlying dataset and later arrived chunks in an unexpected way, and consequently an unfilled arrangement of existing groups and a vacant arrangement of exceptions were expected as the data sources when the main chunk is prepared. The thought behind the arbitrary choice of the data points is to explore the conduct of the methods when there is no control on the arrangement of data points, for example we didn't choose explicit data points from explicit clusterings in the first datasets. No supposition that was made that the underlying data chunk represent to the whole data domain. To limit the impact of arbitrary choice of data points, the investigations were repeated multiple times, and the mean is calculated.

- ***Purity***

Figure 5 shows the subtleties of assessment results between the known clusters and the result clusters from EINCKM, EDDS, and DED techniques independently. DED has the greatest purity. This is considering the way that it keeps all the agent data centres and rigid method. EDDS in like way has a decent flawlessness by pruning and sparing the surface-centres data objects, in any case this system excuses non-raised shape groups. EINCKM has a reasonable variance moreover and insults a great deal of data centre records which may not affect the last clusters.

- ***Entropy***

As appeared in Figure 6, DED has the base entropy. EDDS has progressively raised proportion of entropy. EINCKM has the strangest proportion of entropy among the three systems. The outcomes exhibit that the pruning inside focus data centres impacts group exactness. However, this outcome should be inspected along with the Purity assessment results to have a sensible view on clustering exactness.

- ***Sum of Square Error (SSE)***

As appeared in Figure 7, DED has the most insignificant SSE, followed by EDDS which in this way is followed by EINCKM, again displaying the expense of pruning inside focus data centres. Of course, EDDS and EINCKM still have the most really low SSE score; exhibiting that blending incorrectly data points into found clusters does in like way impact group quality. It should be referred to that SSE may not be the ideal evaluator for nature of non-convex shape clusters.

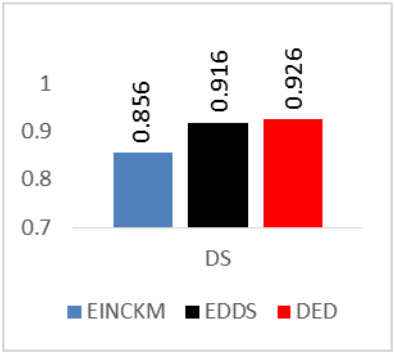

Fig. 5: The purity measurement

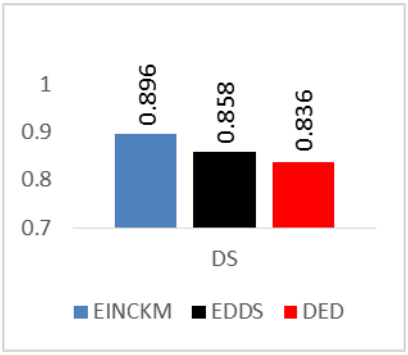

Fig. 6: The entropy measurement

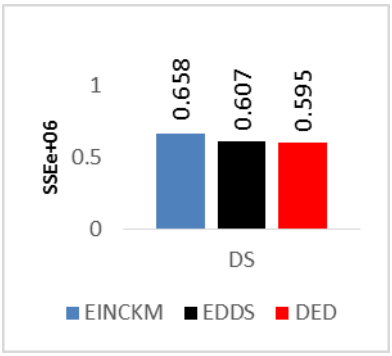

Fig. 7: The SSE measurement

Efficiency Evaluation

- ***Execution Time***

Execution time is the degree of the extent of time in seconds that proceed for the method in finishing the clustering task. Concerning utilization time, the DED method has the base execution time searched after by EINCKM, by then EDDS (see Figure 8). Ensuing to getting these yield results we could uphold that DED method is quicker than EINCKM and EDDS.

Adaptation Prototype-based and Density-based Methods to the Concept Drift

Concept drift has been perceived as one significant issue in data stream clustering and classification [13]. In clustering, concept drift refers to the developmental changes to cluster models over time. Static data clustering just has one concept: the global model of clusters while data stream clustering may have various concepts that develop over time. We recognized concept drift at two levels: the variation level and change checking level.

At the variation level, our method, by following the incremental methodology of data stream clustering, consistently refines the current model of clusters considering the recently arrived data chunk, and consequently consistently adjusts to the progressions reflected by new clusters added into the model or change to the current groups through merging.

At the change checking level, the proposed method itself doesn't keep a set of experiences trail of the changed cluster models. Truth be told, execution of the method may utilize variable parameters to keep a solitary duplicate of the new model of clusters, overwriting the past model.

Accordingly, when a cluster(s) is refreshed with new data point(s), the current group is given a higher weight than more established ones. Figure 9 shows an illustration of concept drift in the DED method utilizing a model dataset. In time *t* there are 2 little output clusters representing to the original clusters. Nonetheless, the snapshot around then doesn't give exact centroids areas and radii. In time *t* + 1 the snapshot of the output clusters shows that the outcome groups are nearer to the ground truth after the advancement of approaching data streams. At long last, in time *t* + 2 the preview represents to that the output clusters are more precise contrasting and the ground truth.

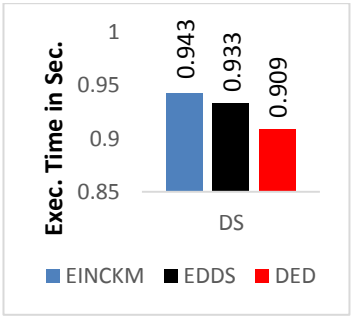

Fig. 8: The efficiency measurement

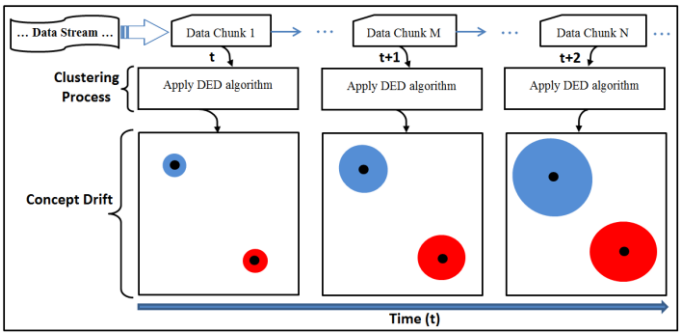

Fig. 9: Illustrative example of concept drift in prototype-based algorithms. Dark dots represent a centroid and around shaded circle represent a radius

Conclusion

In this paper we study how the clusterings produced by different data stream clustering methods change, relative to the ground truth, as quantitatively different types of concept drift are encountered. This paper makes two contributions to the literature. First, we propose a method for generating real-valued data streams with precise quantitative concept drift. Second, we conduct an experimental study to provide quantitative analyses of data stream clustering methods performance with real-valued data streams and show how to apply this knowledge to real world data streams. Future work will focus on updating the method. Because of the method is flexible, those updating thoughts can deals with the major functions of the method. Firstly, we will investigate the topology computation to present more sophisticated version of the embeding concept drift step. Secondly, we will investigate hybridizing different fading functions, like neural network-based, swarm-based, and genetic-based functions to test the modularity of DED method.

Reference

[1] Kumar, D. (2016). Big data Clustering for Smart City Applications. Ph.D. thesis, The University Of Melbourne, Department of Electrical and Electronic Engineering: 122 pp.

[2] Guha, S., Mishra, N., Motwani, R. & O'Callaghan, L. (2000). Clustering Data Streams. *0-7695-0850-2/00 $10.00 0 2000 IEEE* 359–366.

[3] Marcos, D. A., Rodrigo, N. C., Silvia, B., Marco, A. S. N. & Rajkumar, B. (2014). Big Data Computing and Clouds: Trends and Future Directions. *J. Parallel Distrib. Comput.* 1–44 (2014).

[4] Aggarwal, C., Han, J., Wang, J. & Yu, P. (2003). A Framework for Clustering Evolving Data Streams. *Proc. 29th VLDB Conf. Ger.*

[5] Isaksson, C. (2016). New Outlier Detection Techniques For Data Streams. Ph.D. thesis, Southern Methodist University, Bobby B. Lyle School of Engineering: 154.

[6] Stahl, F., Badii, A., Oldenburg, M. & Theodorstahldfkide, F. Building Adaptive Data

Mining Models on Streaming Data in Real-Time (2020). *Comput. Intell.* 3, 12.
[7]. Lobo, J. L., Del, J., Eneko, S., Albert, O. & Francisco, B. (2020). CURIE: A Cellular Automaton for Concept Drift Detection. *arXiv Prepr. arXiv* . 5, 15.
[8] Hu, H., Kantardzic, M. & S. Sethi, T. (2019). No Free Lunch Theorem for concept drift detection in streaming data classification: A review. *Br. Assoc. Adv. Sci.* 2, 16.
[9] Toor, A. A. *et al.* (2020). Mining Massive E-Health Data Streams for IoMT Enabled Healthcare Systems. *Sensors MDPI* 2, 20.
[10] Yeoh, J. M., Caraffini, F., Homapour, E., Santucci, V. & Milani, A. (2019). A Clustering System for Dynamic Data Streams Based on Metaheuristic Optimisation. *MDPI Mathmatics* 2, 1–24.
[11] Ding, S., Wu, F., Qian, J. & Jia, H. (2013). Research on data stream clustering algorithms. *Springer* Artif Inte, 593–600.
[12] Silva, J., Faria, E., Barros, R., Hruschka, E. & Carvalho, A. (2013). Data Stream Clustering : A Survey. *ACM Comput. Surv.* 1–37.
[13] Nguyen, H. L., Woon, Y. K. & Ng, W. K. (2015). A survey on data stream clustering and classification. *Knowl. Inf. Syst. Springer* 535–569. doi:10.1007/s1015-014-0808-1
[14] Bifet, A., Carvalho, A. & Gama, J. (2017). *BigData Stream Mining*.51(1):24-54.
[15] Sethi, T. S. & Kantardzic, M. (2017). On the reliable detection of concept drift from streaming unlabeled data. *Expert Syst. Appl.* 82, 77–99, ISBM 09574174.
[16] Udommanetanakit, K., Rakthanmanon, T. & Waiyamai, K.(2007). E-Stream: Evolution-Based Technique for Stream Clustering. *Springer-Verlag Berlin* 403, 42–55.
[17] Aggarwal, C. C., Han, J., Wang, J. & Yu, P. S. (2004). A Framework for Projected Clustering of High Dimensional Data Streams. *Proc. Thirtieth Int. Conf. Very large data bases* 30, 863.
[18] Davies, R. N. (2017). Efficient Analysis of Data Streams. M.Sc. thesis, Lancaster University, Department of Computing and Communications).
[19] Yogita & Toshniwal, D. (2012). Clustering Techniques for Streaming Data – A Survey. *3rd IEEE Int. Adv. Comput. Conf.* 951–956.
[20] Al Abd Alazeez, A., Jassim, S. & Du, H. (2017). EINCKM: An Enhanced Prototype-based Method for Clustering Evolving Data Streams in Big Data. *Proc. 6th Int. Conf. Pattern Recognit. Appl. Methods* 173–183. doi:10.5220/0006196901730183
[21] Al Abd Alazeez, A., Jassim, S. & Du, H. (2017). EDDS: An Enhanced Density-Based Method for Clustering Data Streams. *2017 46th Int. Conf. Parallel Process. Work.* 103–112 (2017). doi:10.1109/ICPPW.27
[22] Kremer, H. *et al.* (2011). An effective evaluation measure for clustering on evolving data streams. *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.-KDD '11* 868–876. doi:10.1145/2020408.2020555
[23] Cao, F., Ester, M., Qian, W. & Zhou, A. (2006). Density-based clustering over an evolving data stream with noise. *Proc. Sixth SIAM Int. Conf. Data Min.* 206, 328–339.
[24] Zhao, Y. & Karypis, G. (2001). Technical Report Criterion Functions for Document Clustering: Experiments and Analysis. *Univ. Minnesota, Dep. Comput. Sci. / Army HPC Res. Center/ Tech. Rep.* 1–30.

DED : مفهوم الانحدار في بيانات التعليم المتطورة

عمار ظاهر ياسين ال عبد العزيز
قسم علوم الحاسوب ، كلية علوم الحاسوب والرياضيات ، جامعة الموصل ، الموصل ، العراق

الملخص

عنقدة البيانات المستمرة هي واحدة من المهام المميزة لاكتشاف الانماط المخفية في البيانات المستمرة. فهي تشير الى عملية اكتشاف مجاميع جديدة في البيانات مستمرة الوصول والتي تغير من انماطها باستمرار. هذه المقالة تقدم خوارزمية تنقيب جديدة لعنقدة البيانات المستمرة ومراقبة تطورها ومفهوم الانحدار. على الرغم من ان مفهوم الانحدار هو متوقع في البيانات المستمرة، التعريف الصريح لاكتشاف هذا الانحدار نادرا ما يكون واضح في هذه الخوارزميات. مفهوم الانحدار يحدث بسبب التغييرات التي تطرا على البيانات مع مرور الوقت. العلاقة بين مفهوم الانحدار وتوالي الاحداث الحقيقية تم دراستها بواسطة تطبيق الخوارزمية المقترحة في هذا البحث على البيانات التعليمية المستمرة. البيانات التعليمية المستمرة لجروب برنامج الفايبر في قسم علوم الحاسوب تم استخدامها في هذه الدراسة. النتائج بينت ان خوارزميتنا المقترحة تتفوق على الخوارزميات الحالية الموجودة من حيث مقاييس النقاوة والعشوائية والتباين. التجارب التي تم اجرائها على هذه البيانات قادت الى استنتاج ان مفهوم الانحدار مرتبط مع التغييرات في عدد العناقيد والبيانات الشاذة والتي تشير الى احداث مهمة في عملية التعليم. هذا النوع من المراقبة الآنية ونتائجه ممكن ان يستخدم في انظمة التعليم في مختلف الطرق، منها ابراز قدرات المشاركين في هذه الجروبات.