Evolution-Based Clustering of High Dimensional Data Streams with Dimension Projection

Rattanapong Chairukwattana
Department of Computer Engineering
Kasetsart University
Bangkok, Thailand
Email: g521455024@ku.ac.th

Thanapat Kangkachit
Department of Computer Engineering
Kasetsart University
Bangkok, Thailand
Email: g5185041@ku.ac.th

Thanawin Rakthanmanon
Department of Computer Engineering
Kasetsart University
Bangkok, Thailand
Email: {thanawin.r, fengknw}@ku.ac.th

Abstract—SE-Stream is an evolution-based stream clustering method that supports high dimensional data streams. SE-Stream is able to monitor and detect change in the clustering structure during the progression of data streams. In this paper, we improve performance of SE-Stream by reducing its execution time and increasing its cluster quality. SE-Stream reduces complexity of stream processing by determining appropriated subset of dimensions of each active cluster to express cluster specific characteristics during the progression of data streams. With elimination of redundant operations, SE-Stream is improved both in terms of cluster quality and execution time. Experimental results on two real-world datasets show that SE-Stream outperforms its previous version and yields comparable execution time. Compared with HPStream, a state of the art algorithm for projected clustering of high dimensional data streams, SE-Stream outperforms in terms of cluster quality and yields comparable execution time.

Keywords—High dimensional data streams, Projected clustering on active clusters, Evolution-based clustering

I. INTRODUCTION

Streams clustering is one of active data mining research topic. Streams clustering is able to monitor and detect change in the clustering structure during the progression of data streams. While using limited resources, streams clustering processes data in a single pass and summarizes it in real-time [1]–[3]. Unfortunately, performance of existing streams clustering methods drops when face with high dimensional (hundreds or thousands of dimensions) data streams. High dimensional data leads to more complexity in the clustering methods. To deal with high-dimensional data, [1], [4], [5] proposed project clustering techniques to determine appropriated subset of dimensions for each cluster. However, almost of them are difficult to generalize to handle data streams because their clustering process cannot be done in a single pass. HP-Stream [6] is considered as a state of the art algorithm for projected clustering of high dimensional data streams. For each cluster, HPStream performs a continuous refinement of the set of appropriate dimensions during the progression of the streams. Due to its adaptability to the nature of real datasets, HPStream is able to generate clustering with quality where each cluster is specific to a particular group of dimensions. HDDStream [7] is a density-based projected clustering technique for high dimensional data streams that works in both online (micro clusters) and offline (macro clusters with projected clustering) modes. Compared with HPStream, HDDStream does not require the number of clusters. Its number of clusters is variably adjusted over time, and its clusters can be of arbitrary shape.

In [8], SE-Stream is proposed by extending E-Stream [9] to support high dimensional data streams. Dimension projection is performed for both active and inactive clusters. Inactive clusters are still unable to express cluster characteristics due to their limited amount of members. In addition, some clustering operations such as merging and splitting need to be enhanced. Objective of this paper is to further improve SE-Streams by reducing its execution time and increasing its cluster quality. To achieve this goal, projection clustering is performed only over active clusters. SE-stream determines appropriated subset of dimensions of each active cluster to express its specific characteristics during the progression of data streams. With elimination of redundant operations, experimental results on two real-world datasets show that SE-Stream improves both cluster quality and execution time of its previous version and E-Stream. Compared with HPStream, SE-Stream gives better quality and comparable time performance.

To summarize, the contribution of this paper is as follows:

1) SE-Stream is improved by reducing its execution time and increasing its cluster quality.
2) SE-Stream performs dimension projection on only active clusters to extract their specific subset of dimensions.

The remaining of the paper is organized as follows. Section 2 presents evolution-based stream clustering technique algorithm. Section 3 presents our stream clustering algorithm called SE-Stream and its extension. Section 4 compares performance of SE-Stream with E-Stream and HPStream on two real-world data sets i.e. Network Intrusion and Forest Cover Type. Section 5 concludes the paper.

II. EVOLUTION-BASED STREAM CLUSTERING WITH DIMENSION PROJECTION

In this section, notations and definitions related to the evolution-based stream clustering with dimension projection are given. Assume that data streams consists of a set of multidimensional records $X_1 \ldots X_k$ arriving at time stamps $T_1 \ldots T_k$. Each data point $X_i$ is a multidimensional record containing $d$ dimensions, denoted as $X_i = (x_i^1 \ldots x_i^d)$. 
A. Cluster Representation using Fading Cluster Structure with Histogram

A Fading Cluster structure has been proposed in [6] to use as cluster representation instead of storing all the data points in a cluster. [9] introduced a Fading Cluster Structure with Histogram (FCH). For each cluster dimension, an $\alpha$-bin histogram is used to detect change of the clustering structure. In this paper, FCH is extended to support dimension projection and is defined as $FCH = (FC1(t), FC2(t), W(t), BS(t), H(t))$. Following is the description of FCH.

Let $N$ be the total number of data points of such cluster, $T_i$ be the time when data point $x_i$ is retrieved, and $t$ be the current time. The fading weight of data point $x_i$ is defined as $f(t-T_i)$ where $f(t) = 2^{-\alpha t}$ and $\alpha$ is the user-defined decay rate.

$FC1(t)$ is a vector of weighted summation of each dimension at time $t$. The $j$-th dimension is

$$FC1^j(t) = \sum_{i=1}^{N} f(t-T_i) \cdot (x_{i}^{j}). \quad (1)$$

$FC2(t)$ is a vector of weighted sum of square of each dimension at time $t$. The $j$-th dimension is

$$FC2^j(t) = \sum_{i=1}^{N} f(t-T_i) \cdot (x_{i}^{j})^2. \quad (2)$$

$W(t)$ is a sum of all weights of data points in the cluster at time $t$, i.e.,

$$W(t) = \sum_{i=1}^{N} f(t-T_i). \quad (3)$$

$BS(t)$ is a bit vector of projected dimensions at time $t$. For the $j$-th dimension is

$$BS^j(t) = \begin{cases} 
1 & \text{if dimension } j \text{ is within a set of relevant cluster dimensions} \\
0 & \text{Otherwise}.
\end{cases} \quad (4)$$

Note that number of projected dimensions are not the same for each cluster (see section II-B for more detail).

$H(t)$ is a $\alpha$-bin histogram of data values with $\alpha$ equal width intervals. For the $l$-th bin histogram of $j$-th dimension at time $t$, the elements of $H^j$ are

$$H^j_l(t) = \sum_{i=1}^{N} f(t-T_i) \cdot (x_{i}^{j}) \cdot (y_{i,t}^{j}) \quad (5)$$

where

$$y_{i,t}^{j} = \begin{cases} 
1 & \text{if } l \cdot \alpha + \min(x^{j}) \leq x_{i}^{j} \leq (l+1) \cdot \alpha + \min(x^{j}); \\
\frac{\max(x^{j}) - \min(x^{j})}{l} \cdot \alpha & \text{Otherwise}.
\end{cases} \quad \alpha$$

Clusters can be categorized into 2 types: active and inactive cluster regarding to their weight. Active cluster is a cluster having its weight greater than user-specified threshold $active\_cluster\_weight$. Only active clusters can assemble incoming data point located nearby. Meanwhile, such inactive clusters with lower weight than $active\_cluster\_weight$, can be merged together with other inactive clusters or active clusters to produce an active cluster.

B. Dimensions Projection

Clustering of high dimensional data streams is of high complexity in the clustering method. In addition, clusters quality is unsatisfied due to sparsity of data. To overcome these difficulties, only a subset of dimensions relevant to each cluster are selected. To perform clustering, the data streams are projected to these relevant dimensions instead of using all dimensions. Here, we propose the method to select relevant dimensions especially from the active clusters. The reason is that active clusters contain sufficient number of members (data points) that are able to express the characteristics of each cluster. In case of inactive clusters, all dimensions are used due to unclear characteristic expressed by the insufficient number of members. After performing selection method, the characteristics of each active cluster are maintained by preserving the similar set of projected dimensions during the progression of data streams. Indeed, an active cluster may evolve rapidly after assembling an inactive cluster. In this case, the whole processes of selection method are restarted as the same as in case of detecting new active cluster.

An example as shown in Fig. 1 is demonstrated how our proposed dimensions projection method works to determine appropriate subset of dimensions for all active clusters. Let $l$ be number of average projected dimensions for each cluster and $l$ is set to 1. Suppose that at timestamps $t$, the output clustering contains 3 active clusters of data streams with 3 dimensions. First, we compute all radii of dimensions in each cluster by considering its $FCH$. Then, all the dimensions are ranked based on their radi in ascending order. Lastly, the top $|FCH| \cdot l$ ranks of dimensions (as presented with grey color) are selected as projected dimensions i.e. dimension #1 and #2 of cluster #2 and dimension #2 of cluster #3. Notice that the number of projected dimensions and dimensions for each cluster may differ. At Fig. 2, corresponded bit vector of all the clusters is shown.
to determine specific subset of dimensions for each cluster is extended from E-Stream. Projected clustering technique to determine specific subset of dimensions for each cluster is explained. Then, we present SE-Stream algorithm composing of a set of sub-algorithms. Finally, its time-complexity is discussed and analysed.

A. Overview of SE-Stream Algorithm

This section describes SE-Stream which is an extension of E-Stream to support high-dimensional data streams. Table I contains all the notations that are used in SE-Stream algorithm. SE-Stream main algorithm is given in Fig. 3.

C. Distance Functions

To deal with projected dimensions, distance functions are modified to take into account only the relevant cluster dimensions. Suppose that $BS(t)$ is a bit vector represented cluster dimensions at timestamp $t$. The distances functions can be defined as follows.

Cluster-Point distance is a distance from a data point to a center of active cluster. For each dimension, the distance is normalized by the radius (standard deviation) of the cluster data. The function is used to find the closet active cluster for an incoming data point. The Cluster-Point distance function $dist(C, X_i)$ of cluster $C$ and incoming data point $X_i$ at timestamp $t$ can be formulated as:

$$dist(C, X_i) = \frac{1}{n} \sum_{j \in BS(t)} \frac{|center_i^j - x_i^j|}{radius_j^C}$$  \hspace{1cm} (7)

where $n$ is the number of projected dimensions of cluster $C$ in the bit vector $BS(t)$.

Cluster-Cluster distance is a distance between two cluster centers. It is used to determine whether pair of clusters that can be merged together. The cluster-cluster distance function $dist(C_a, C_b)$ of cluster $C_a$ and $C_b$ at timestamp $t$ can be formulated as:

$$dist(C_a, C_b) = \frac{1}{n} \sum_{j \in bs(t)} |center_i^j_{C_a} - center_i^j_{C_b}|$$  \hspace{1cm} (8)

where $n$ is the total number of projected dimensions and $bs(t)$ represents the bit vector of $n$ projected dimensions. Notice that the number of projected dimensions and dimensions of two clusters may differ and are based on their cluster type. If both clusters are active, $bs(t)$ is the union set of the projected dimensions of these two clusters. If both clusters are inactive, $bs(t)$ is the whole set of dimensions. Otherwise, $bs(t)$ is the set of projected dimensions of active cluster.

III. SE-STREAM ALGORITHM

In this section, we start by giving an overview of SE-Stream which extended from E-Stream. Projected clustering technique to determine specific subset of dimensions for each cluster is explained. Then, we present SE-Stream algorithm composing of a set of sub-algorithms. Finally, its time-complexity is discussed and analysed.

SE-Stream is evolution-based algorithm that supports the monitoring and the change detection of clustering structure that can evolve over time. It is designed for high dimensional data stream. Various types of clustering structure evolution are supported which are appearance, disappearance, self-evolution, merge and split. In line 1, the algorithm starts by retrieving a new data point. In line 2, it fades all clusters and deletes those have weight less than $\epsilon$. In line 3, it splits a cluster when behavior inside the cluster is obviously separated. In line 4, it checks for overlapping clusters and merges them. In line 5, when the number of cluster count exceeds the limit, it checks the closest pair of clusters and merges them until the number of cluster count does not exceed limit maximum number of clusters. In line 6, it checks all clusters whether their statuses are active. In line 7-8, the algorithm checks for new active clusters or in case of active cluster turns in to inactive cluster. If found, the set of projected dimensions of all active clusters are recomputed by ProjectDimension procedure (refer to figure 4). Notice that, a new active cluster is created in the following case: merge between two active clusters, merge of active and
Fig. 4: Details of each sub-algorithm extended from E-Stream or added to SE-stream.

inactive clusters or merge between to inactive clusters. In line 9, it finds the closest cluster to the incoming data point. In line 10-13, if the distance between closest cluster and incoming point is less than \( \text{radius factor} \) then the point is assigned to the cluster, otherwise it creates isolated data point. The flow of control returns to the top of algorithm and waits for a new data point. Following is the explanation of each sub-algorithm that has been extended from E-stream or added to SE-stream. Details of each sub-algorithm is given in Fig. 4.

**FadingAll** SE-Stream performs fading of all clusters and deletes clusters with weight less than \( \varepsilon \) (an input parameter).

**CheckSplit** SE-Stream finds the split point in the projected dimensions. If a splitting point is found in any cluster, it is split and dimensions are computed and stored with index pairs of split in \( S \).

**MergeOverlapCluster** SE-Stream finds the pairs of active clusters that are overlapped. For each pair of active clusters, cluster-cluster distance is calculated. If the distance is less than the \( \text{merge threshold} \) and the merged pair is not already in \( S \) then the two clusters are merged. Notice that the merging operation is performed only on active clusters with the new incoming point because of their slowly self-revolution change.

**LimitMaximumCluster** SE-Stream checks weather the total number of clusters reaches its maximum \( \text{maximum cluster} \). If it exceeds the maximum, then the closest pair of clusters is merged until the number of the remaining clusters is less than or equal to the \( \text{maximum cluster} \). In case of merging between two inactive clusters: if it results in a new active cluster, ProjectDimension method for dimension selection is performed after receiving a new data point.

**FindClosestCluster** SE-Stream calculates cluster-point distance. Then, determine the closest active cluster to contain an incoming data point.

In the distance calculation step, SE-Stream uses 2 types of distance: cluster-cluster distance (\( \text{CCDistance} \)) and cluster-point distance (\( \text{CP Distance} \)). To support projected clustering, these two distances have been extended from E-stream as shown in Fig. 4. Refer to section II-C (after equation 8) for more detail of the cluster-cluster distance (CCDistance). Notice that, in the MergeOverlapCluster and LimitMaximumCluster steps, projected dimensions are stored in a bit vector data structure. SE-Stream uses bitwise OR operation for merging the bit vectors of clusters. The result is new projected dimensions of the cluster that have been merged.

**B. Time-complexity of SE-Stream VS E-Stream**

Evolution-based clustering methods are able to detect change of clustering structure evolution. As result, the clustering output is of high quality in terms of both F-measure and purity. However, with high-dimensional data streams, these clustering structure evolution detection operations, increases execution time of the algorithm. This section discuss the time-complexity of SE-Stream which is an evolution-based clustering method. Fig. 3 shows time-complexity of the different steps of SE-Stream compared with E-Stream.

In case of E-Stream, MergeOverlapCluster and LimitMaximumCluster procedures consume the most execution time. This is because distance between each pair of all the existing clusters need to be calculated, at least big-O \( (K^2) \) is required. Both of them perform distance calculation of every pair of active clusters to find the closest one. Notice also that its time-complexity depends on the number of dimensions for almost
of the clustering operations. When face with high dimensional data, the run-time of E-Stream is extremely increased.

SE-Stream reduces time-complexity by performing dimension projection on active clusters and thus decreases times for distance calculation. With its dimension projection, the number of dimension \( l \) is less than the total number of dimensions \( d \). Instead of using all the dimensions, SE-Stream uses only selected dimensions for all of its clustering structure evolution detection operations then its run-time complexity is greatly reduced. For example, run-time complexity of MergeOverlap-Cluster is reduced from \( O(K^2d) \) in E-Stream to \( O(K^2l) \) in SE-Stream. From \( l \) value less than \( d \) value, it means SE-Stream use time less than E-Stream. For distance calculation, its time complexity is reduced from \( O(K^2d) \) to \( O(K^2l) \) is more than \( O(Kd) \).

IV. EVALUATION AND EXPERIMENTAL RESULTS

In this section, the performance of SE-Stream are evaluated thorough various experiments. The clustering performance is measured by both execution time and cluster quality. To compare with our SE-Stream algorithm, both E-Stream and HPStream algorithms have been implemented in C++. All the experiments were conducted on a 2.6 GHz Intel®Core i5 computer with 2 GB memory and executed on Windows 7.

In the experiments, the two standard UCI datasets are used as benchmark dataset. There are Network Intrusion dataset and Forest Covertype dataset. Only numerical attributes are selected. The Network Intrusion dataset contains 494020 records with 34 attributes and 22 classes. The Forest Covertype dataset contains 581012 records with 10 attributes and 7 classes. For the parameters setting, SE-Stream follows the same values as adopted in E-Stream i.e. stream speed, horizon, maximum number of clusters \( K \), decay-rate \( \lambda \), remove threshold \( \epsilon \) are set to 100, 2, 25, 0.1 and 0.1 respectively. For SE-Stream and HPStream, the number of projected dimensions \( l \) can be varied from 100% to 1%. Here only performance of SE-Stream with 40% is reported due to the dramatically decrease of performance when \( l \) is less than 40%.

A. Cluster Quality

The cluster quality is evaluated based on average f-measure and purity in every 50000 incoming data points. Fig. 5 shows performance of SE-Stream, E-Stream and HPStream in term of cluster quality on the two benchmark datasets. On high dimensional data streams (i.e. Network Intrusion dataset), all the three algorithms produce equivalent purity. In terms of average f-measure, it is clearly seen that SE-Stream outperforms its previous version, E-Stream and HPStream for all the streams progressions. However, during the first 100000 incoming data points, HPStream gives the best f-measure because it performs an initial off-line process to find the initial clusters while SE-Stream and E-Stream perform clustering from scratch. Fig. 5 compares performance of the three algorithms on the Forest Cover Type dataset. The results show that SE-Stream yields comparable cluster quality with E-Stream. Although SE-Stream outperforms HPStream in term of f-measure, its purity is considered equal.

In summary, we can say that SE-Stream produces better cluster quality than E-Stream and HPStream. The reason is that SE-Stream offers a higher f-measure value with equal purity value when compared with the other techniques.

B. Execution Time

Fig. 6 shows performance of SE-Stream, E-Stream and HPStream in term of execution time. The execution time of SE-Stream is significantly lower than its previous version and E-Stream. Compared to HPStream, SE-Stream gives comparable execution time. The gap between SE-Stream and HPStream is due to the additional mechanism to detect cluster evolution with time complexity \( O(K^2d) \).

V. CONCLUSION AND FUTURE WORK

In this paper, SE-Stream algorithm is enhanced for effectively clustering over high dimensional data streams. Inherited from E-Stream, the ability to monitor and detect change of clustering structure is still preserved. SE-Stream improves the dimension projection technique of its previous version by taking into account only active clusters during the streams progression. This is because active clusters contain sufficient number of members and thus they can be described by a small number of projected dimensions. In addition, several redundant operations for determining cluster merge or split are eliminated. As result, complexity of SE-Stream is decreased as well as its computation time.

Although SE-Stream aims to extract the best set of selected dimensions for each cluster, there is no guarantee that those dimensions are not redundant. Thus, our future work is to propose a mechanism to seek for the dimensions able to discriminate such cluster. This will certainly improve quality of SE-stream to perform clustering on high-dimensional data streams more efficiently.

REFERENCES

Fig. 5: Performance comparison in terms of average purity and F-measure on Network Intrusion and Forest Cover Type datasets


