Clustering of High Dimensional Data Streams by Implementing HPStream Method

C. Kondaiah
Department of Computer Science, JNTUA, Anantapur, Andhra Pradesh, India

ABSTRACT

Clustering is an important task in mining evolving with data streams because of data streams produces the continuous and potentially unbounded sequential of data points [1].Such streams collecting the data from the different devices. However, naturally, streaming data is high-dimensional data [1]. High dimensional data streams are frequently very large and it may include outliers .Therefore such streaming data is an significance issue in data mining process. High-dimensional data is actually very difficult in classification, clustering and similarity search. Recently, DBSTREAM, single-scan, subspace methods are used for projected clusters over the high-dimensional data sets. These methods are difficult to generalize to high dimensional data streams because of the huge volume of data generated the automatically by simple transactions ofday-to-day life. In this paper implemented a high-dimensional data streams clustering technique, known as HPStream. This technique consists offade clustering structure and projected primarily based clustering. It is continuously updatable and it's accurate scalable on both the number of dimensions and quantity of the data streams, and it offers the better high-quality clusters as compare with the preceding records movement techniques.

Keywords : DataStream, High Dimensional Data, Clustering.

I. INTRODUCTION

Data streams haveget more importance in the recent years because of forward in the hardware technology. Because of these advances had been made easy to store and record numerous transactions, In the recent years, many companies are amassing an enormous quantity of data, typically generated continuously as a sequence of events and coming from distinct locations. Telephone call logs, Bank card transactional, sensor network data, network event logs are just some of examples of data streams. The presence of data streams in a number of sensible domains has generated a variety of research in this place [8, 10,]. One of the crucial problems recently in the data stream domain is clustering [7]. The clustering problem is particularly interesting for the data streaming area due to its applications to data summarization and outlier detection.

The clustering trouble is defining as follows: for a given set of data points, partition them into one or greater agencies of similar data point, where in the notation of similarity is defined with the help of distance feature. There had been a lot of research work staunch to scalable cluster analysis in current years [2, 6]. In the data stream area, the clustering challenge calls for a technique which can continuously determine the dominant clusters in the information without being dominatedmeans of the preceding historical data stream.

The high-dimensional case affords a special undertaking to clustering algorithms even in the traditional area of static data sets. This is due to the sparsity of the data within the high-dimensional case. In the high-dimensional area, all pairs of points tend to be nearly equidistant from each other. As an end result, it's far frequently unrealistic to define distance-based clusters in a meaningful way. Some latest works on high-dimensional data make use of strategies for projected clustering that can determine clusters for a selected subset of dimensions [3, 6]. In those techniques, the definitions of the clusters are such that every cluster is specific to a subset group of dimensions. This reduces the sparsity problem in the high-dimensional area to some extent. Even although a cluster might not be meaningfully described on all the dimensions due to the sparsity ofdata, a few subset of the dimensions can usually be located on whichspecific subset of points form better quality and extensive clusters. Of course, those subset of dimensions may also vary over the one clusters to some other cluster. Such type of clusters are calledprojected clusters [2].

The idea of the projected cluster normally defined as follows. Let assumed that **k** is number of clusters to be discovered. In addition, algorithm will take as input dimensions l of the subspace where as each cluster is pronounced. The output of the set of rules

- ✓ A (k + 1)-way partition {C_1 . . .C_k, O} of the data, such that the factors in each partition element except the final form of a cluster, while the points within the final partition element are the outliers, which through definitiondon't cluster well.
- ✓ A likely different set $\boldsymbol{\varepsilon}_i$ of dimensions for every cluster C_i, 1≤ i ≤k, such that the points in $C_i$cluster well in the subspace defined by those vectors.(The vectors for the outlier set O can be assumed to be the empty set.) For every cluster C_i, the cardinality of a corresponding set $\boldsymbol{\varepsilon}_i$is identical to the user-defined parameter l.

Inside the context of a data stream, the problem of locating projected clusters turns into even more tough. This is because of the additional problem of locating the applicable set of dimensions for every cluster makes the problem considerably more computationally intensive in data stream area. While the problem of clustering has these days been studied within the data stream environment [3, 11], these methods are for the case of complete dimensional clustering. In this paper, work on the significantly harder problem of clustering high-dimensional data circulate by means of exploring projected clustering strategies. Current projected clustering strategies including those discussed in [2] can't be easily generalized to the data stream problem because they usually require a multiple of passes over the data. As a further, the algorithms in [2] are too computationally intensive to use for the data stream problem. Further, data streams fastly evolve over time [4, 5] because of which it is important to design techniques which might be designed to effectively adjust with the progression of the stream.

II. BACKGROUND

Density based cluster algorithms are appropriate to data mining in the applications. These strategies use a local criterion and define clusters because the regions within in the data space of higher density as compared to the areas of noise points or margined points. The data points can be distributed by absolutely in these regions of high density and may contain clusters of arbitrary size and shape. A normal way to discover the areas of high density is to become aware of grid cells of high densities by partitioning each dimension area into non-overlapping partitions or grids.

The earliest density based clustering technic is DBSCAN [4]. It is totally based on the technic of density region. A point is known as "core object", if inside a given radius (ε), the neighborhood of this point carries a minimal threshold range (MinPts) of object. A core object is a starting point of a cluster and as a consequence can build a cluster around it. Density based clustering algorithms are used the DBSCAN notation, can locate clusters of absolutely size and shape. Fig. 1(a) [2] indicates clusters constructed with density belief with no. Of objects > 10 and Fig 1(b) does now not construct the cluster.

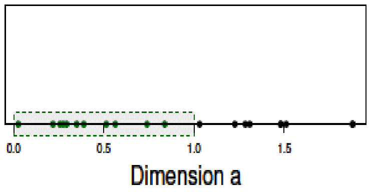
Density based approaches are normally and popularly used to find out clusters in high dimensional data. These approaches search for the probable subspaces of high densities and then the clusters hidden in those subspaces. It is defined a dense subspace, if it incorporates many data points are in threshold region in a given radius. SUBCLU [7] algorithm is the primary subspace clustering extension to DBSCAN to clustering high dimensional stream data, by using the DBSCAN.

Cluster partition evolving on stream data are often computed primarily based on time periods. The clustering a data stream problem consider in the window version, where as the weights of each data point decrease exponentially with the time t via a fad characteristic f(t) = 2-ƛ.t [3] where, ƛ > 0. The exponentially fading characteristic is broadly used in temporal applications in which it's desirable to regularly discount the history of past behavior. The higher the value of ƛ, the decrease importance of the past data in comparison to more latest data. And the overall weight of the data stream is a constant W =

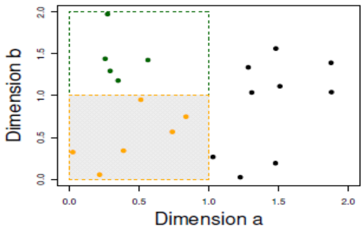
$v(\sum_{t=0}^{t=t_c} 2^{-\lambda t}) = \frac{v}{1-2^{-\lambda}}$, , where tc (tc →∞) is the current time, and v denotes the rate of movement, i.e., the quantity of points arrived in one unit time.

2.1. High-dimensional Data

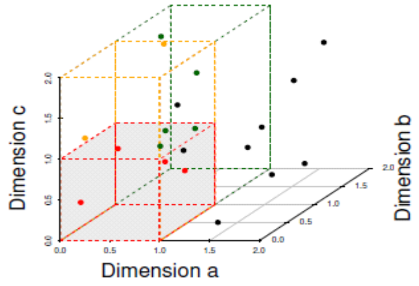
The intersection region turns into an intersection volume when the dimensions higher than two. Most of the clustering algorithms face problem with the high dimensional data, it is the curse of dimensionality. As the number of dimensions in a data stream will increase, distance measures become increasingly more meaningless. Additional dimensions distribute the points till, in very high dimensions, they're almost equidistant from each different. Figure 2 [3] illustrates how extra dimensions spread out the points in a sample dataset. The dataset includes 20 points randomly located between 0 and 2 in each of 3 dimensions. Figure 2(a) [3] suggests the data projected onto one axis. The points are near together with approximately half of them in a one unit sized bin. Figure 2(b) [3] shows the same data stretched into the second dimension. By adding some other dimension. The points spread out along every other axis, pulling them similarly aside. Now only about a quarter of the points fall into a unit sized bin. In Figure 2(c) [3] added a third dimension which spreads the data further aside. A single unit sized bin now holds only one eighth of the points



(a)



(b)



(c)

Figure 2: The *curse of dimensionality*

First data in one dimension is quite tightly packed. Add a dimension stretches the points across that dimension, pushing them similarly aside. Additional dimensions spreads the data even similarly make the high dimensional data extremely less dense.[3]

2.2 Application:

High-dimensional clustering is especially efective in domains where one can expect to find relationship across a variety of perspectives. Where Some areas High dimensional clustering has great potential are information integration system, text-mining, and bioinformatics, image recognition.

- **Information Integration Systems**: Query optimization becomes a complex problem since the data is not centralized. The decentralization of data poses a difficult challenge for information integration systems, mainly in the determination of the best subset of sources to use for a given user query. An exhaustive search on all the sources would be a naive and a costly. Application of subspace clustering in the context of query optimization for an information integration system developed here at ASU, Bibfinder[7].
- **Web Text Mining:** A fundamental problem with organizing web sources is that web pages are not machine readable, meaning their contents only convey semantic meaning to a human user. In addition, semantic heterogeneity is a major challenge. That is when a keyword in one domain holds a different meaning in another domain making information sharing and interoperability between heterogeneous systems difficult. In order to automate the process, subspace clustering will helps to learn concepts.
- **Image recognition:** Suppose you have "**n**" images, every image with a resolution of "**m**" pixels by "**l**"

pixels. Here define every pixel with in the image is one variable thus that the n images store in m x l dimensional space. From there a training set of images is used to recognize new faces its solution[13]. By using the high dimensional data clustering methods to represents the training/new images with lower dimensions its depends on the applications and images.

III. HPSTREAM METHOD

Micro-cluster-based data streams clustering algorithms uses the density inside every micro-cluster (MC) as some form of weight (e.g., the quantity of points assigned to the MC). For re-clustering, uses only the distances among the MCs and their weights are used. In this, MCs which might be closer to each other cluster combine as a single cluster based on the MC centers and their weights. This is even proper if a density-based algorithm like DBSTREAM [4] is used for re-clustering. The density in the region between MCs is not available since it isn't retained at some stage in the online stage.

This paper implements a high-dimensional projected stream clustering method by means of continuous refinement of the set of projected dimensions and data points all through the progression of the stream this is called as HPStream, since it describes the High-dimensional Projected Stream clustering method. The updating of the set of dimensions related to each cluster is carried out in such a way that the points and dimensions related to each cluster can efficaciously evolve through the time. In order to obtain this goal, using the condensed representation of the statistics of the points in the clusters. These condensed representations are selected in the sort of manner that they can be update effectively in a fast data stream. At the same time, a sufficient amount of information is stored in order that essential measures about the cluster in a given projection can be quickly computed. The fading cluster structure is also capable of performing the updates in this such a way that previous data is temporally discounted. This guarantees that during an evolving data stream, the beyond history is progressively discounted from the computation. HPStream introduces the technic of projected clustering to data streams and fading cluster structure.

Algorithms:

Algorithm for clustering High Dimensional Data Streams

```
Algorithm 1: HPStream (Data Stream Point: X, Cluster Structures: FCS,
Dimensionality Vector Sets: BS, Dimensionality: l);
begin
 {Assume that FCS includes the relevant cluster structures denoted by FCS = {FC($C_1$, t) . . . FC(Cr; t) . . . } }
 {Assume that BS includes the related cluster dimensions which are denoted by BS = {B(C1) . . . B(Cr) . . . }
 Receive the next data point X at current time t from stream DS;
 BS =ComputeDimensions (FCS, l, X);
for r = 1 to |FCS| do
 s = FindLimitingRadius (FC(C index, t ), B(C index));
if r(index) > s
then set index = |FCS| + 1 and add new fading cluster structure CjFCSj+1 with a solitary data point to FCS;
else add X to FC(C index, t);
 Remove those clusters from FCS which have zero dimensions assigned to them;
if |FCS| > k
then delete the least recently added cluster in FCS;
end;
```

Algorithm for Computing The Projected Dimensions

```
Algorithm 2: ComputeDimensions(Faded Cluster Structures: FCS, NumberofDimensions: l, Incoming Point: X);
begin
        Create |FCS| (tentative) fading cluster structures by adding X to each of the existing clusters;
        Compute the |FCS| * d radii of every of the |FCS| (tentative) clusters along every d dimensions;
        Pick the |FCS * l| dimensions with the least radii;
        Create a bit vector B(Cr) for each cluster Cr reflecting its projected dimensions;
end;
```

3.1 The Fading Cluster Structure:

Introduce the concept of a fading data structure which is able to regulate for the latest of the clusters in a flexible manner. The data streams includes a group of multi-dimensional data points X_1 …. X_k arrive at time stamp T_1….T_k. Every data point Xi is having a multi-dimensional data record it contains d dimensions, indecated by way of $X_i = (x^1_i \ldots x^d_i)$. It is assumed that every data point had a weight defined through a function f(t) to the time t.

The fade clustering structure, a data structure that is designed to find key statistical characteristics of the clusters generated for the duration of the course of the data stream. The purpose of the fading cluster structure is to capture a enough number of the underlying statistics in order that it is possible to compute key traits of the underlying clusters**.**

3.2 The High Dimensional Projected Clustering:

In this discussion, individual clusters are maintained in an online fashion. High-dimensional clustering makes use of an iterative technique which continuously determines new cluster structures at the same time as re-defining the set of dimensions included in each cluster [Algorithm 2].

First, run a normalization process due to the different dimensions having different length of values. This is because the clustering set of rules wishes to select the dimensions that are specific to every cluster via evaluating the radius alongside exceptional dimensions. Different dimensions may additionally refer to the different scales of reference including age, salary or different attributes which have more difference in ranges and variances. Therefore, it isn't feasible to compare the dimensions in a significant way using the original data. In order as a way to estimate different dimensions meaningfully, carry out a normalization processing. The aim is to equalise the standard deviation along each dimension.

IV. SCALABILITY RESULTS

Here present and analyze results on clustering quality (accuracy) and the efficiency of the comparing algorithms. Clustering purity is taken as the indecation for clustering quality.

Accuracy comparison: evaluated the clustering quality of the HPStream algorithm in comparison with the DBSTREAM, CluStream algorithm using real data set, forest Covertype.

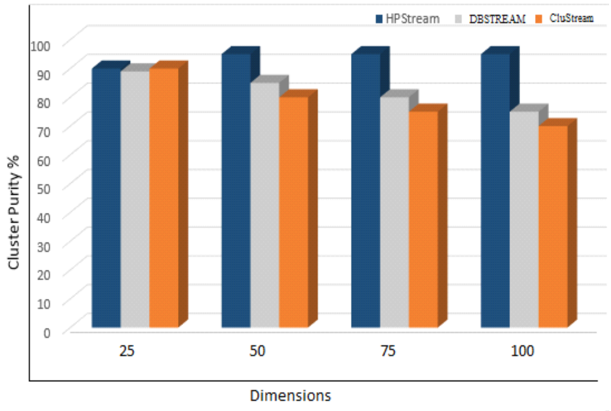


Figure 3: Quality comparison (Forest CoverType data)

An average projected dimensionality l = 75, in experiments used a series of different l's, i.e., (25, 50, 75, 100), to test clustering quality in the Figure 3 show the result. As in the Fig 3.overall l = 75 can lead to the best cluster purity, and a too small l at 25 or a too large l at 100 generate very poor clustering quality. In addition, the cluster purity for l = 50 or l = 75 is very similar to that for l = 70, which suggests as long as a value choose for l in the range from 50 to 75, HPStream gave a very superb clustering quality.

The above results in Fig.3 about the sensitivity of the mean projected dimensionality l display that as long as l value not too deviated from the proper median projected dimensionality, HPStream gives a high clustering quality. HPStream continually generated comparable clustering solutions if the l value in the range from 50 to 75.

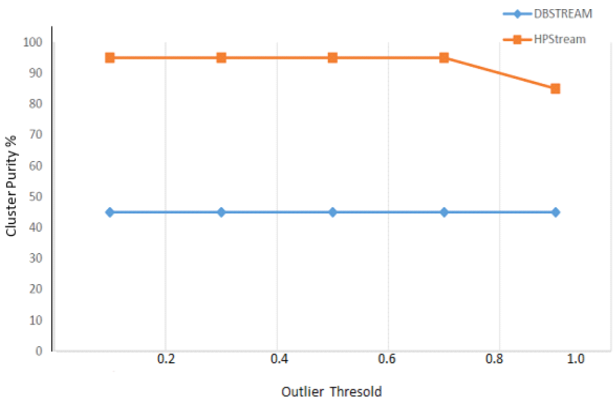


Figure 4: Clustering quality vs. outlier threshold

Sensitivity Analysis: An vital parameter of HPStream is a decay factor. It controls the importance of historic data to current clusters. In pervious experiments set it to 0.25, which is amoderate setting. However, the quality of HPStream continues to be higher than that of DBSTREAM. It can been seen that if the threshold value from 0.125 to 1, the clustering quality is quite precise and strong, and usually above 95%. Another important parameter is the outlier threshold. Figure 4 suggests the clustering quality of HPStream while threshold value is varying from 0.2 to 1. If threshold value range between 0.2 and 0.6, the clustering quality is very good. However, if it's far set to a particularly high value like 1, the quality deteriorates greatly. Because a number of points corresponding to potential clusters are pruned, the qquality is decreased. HPStream gives the higher clusters than DBSTREAM.

V. CONCLUSION

In early years, the management and processing of High-Dimensional data streams has becomes a subject of dynamic research in several fields of computer science consisting of, e.g., database system, and dat mining. Lot of research work has been carried in this area to increase an advantageous clustering algorithm for High-dimensional information streams. High Dimensional data streams are frequently generate the more amount data and contain outliers. HPStream implemented by combining a fade clustering structure and projection based cluster method to work effective with high dimensional records streams.

VI. REFERENCES

[1]. Sunita Jahirabadkar, Parag Kulkarni., "Clustering for High Dimensional Data: Density based Subspace Clustering Algorithms", International Journal of Computer Applications (0975 - 8887) Volume 63- No.20, February 2013.

[2]. Michael Hahsler, Matthew Bola˜nos., "Clustering Data Streams Based on Shared Density Between Micro-Clusters", IEEE Transactions On Knowledge And Data Engineering - Preprint, Accepted 1/17/2016.

[3]. Lance Parsons, Ehtesham Haque, Huan Liu., "Subspace Clustering for High Dimensional Data: A Review", ACM SIGKDD Explorations Newsletter - Special issue on learning from imbalanced data sets: Volume 6 Issue 1, June 2004.

[4]. Chairukwattana R., Kangkachit T., Rakthanmanon T., Waiyamai K., "Evolution-Based Clustering of High Dimensional Data Streams with Dimension Projection", Knowledge and Systems Engineering. Advances in Intelligent Systems and Computing, vol 245. Springer, 2014.

[5]. Feng Cao, Martin Ester, Weining Qian, Aoying Zhou., "Density-Based Clustering over an Evolving Data Stream with Noise", SIAM International Conference on Data Mining,2006.

[6]. Levent Ertoz, Michael Steinbach, Vipin Kumar., "A New Shared Nearest Neighbor Clustering Algorithm and its Applications"., Workshop on Clustering High Dimensional Data and its Applications at 2nd SIAM International Conference on Data Mining,(2002)

[7]. S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan, "Clustering data streams," in Proc. ACM Symp. Found. Comput. Sci., 12-14 Nov. 2000, pp. 359-366.

[8]. C. Aggarwal, Data Streams: Models and Algorithms, (series Advances in Database Systems). New York, NY, USA: Springer-Verlag, 2007.

[9]. J. Gama, Knowledge Discovery from Data Streams, 1st Ed. London, U.K.: Chapman & Hall, 2010.

[10]. Y. Chen and L. Tu, "Density-based clustering for real-time stream data," in Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2007, pp. 133-142.

[11]. L. Wan, W. K. Ng, X. H. Dang, P. S. Yu, and K. Zhang, "Density-based clustering of data streams at multiple resolutions," ACM Trans. Knowl. Discovery from Data, vol. 3, no. 3, pp. 1-28, 2009.

[12]. Amineh Amini, Teh Ying Wah., "Density Micro-Clustering Algorithms on Data Streams: A Review", Proceedings of the international multiconference of Engineers and scientists 2011, vol 1, IMESC, March 16-18-2011, Hong Kong

[13]. http://en.wikipedia.org/wiki/Eigenface.