Pre-Clustering Algorithm for Anomaly Detection and Clustering that uses variable size buckets

Manish Sharma, Dr. Durga Toshniwal
Electronics and Computer Engineering, I.I.T Roorkee
Indian Institute of Technology Roorkee
Roorkee, India
manish09.iitroorkee@gmail.com , durgatoshniwal@gmail.com

Abstract—Clustering is known as grouping of data based on their similarities. This paper introduces an algorithm of k means for clustering of data streams and detection of outliers. The introduced technique for detection of outliers is based on distance as well as on time on which they arrive in the cluster. This paper also takes into account the selection of k centers and variable size of buckets with the help of which space can be effectively utilized during clustering. Most traditional algorithms make clustering a very difficult problem by reducing their quality for a better efficiency. This paper indicates that with a small increase in time you can efficiently cluster the data without much loss of quality of data.

Keywords-clustering; boolean data; categorial data; k means; anomaly detection;

I. INTRODUCTION

Clustering of data stream refers to the process of grouping the data based on similarities or splitting the data based on dissimilarities. Anomaly refers to unwanted data points in a clustered data set which are least similar to other data points. Clustering with anomaly detection is a difficult task. In recent years, many applications are generated in the form of streams such as network flow, sensor data and web click streams. Analysis and mining of such data is increasingly becoming a hot issue. As a basic technique of data mining, clustering analysis in data flow environments has been attracted attention from academic and industry. Clustering algorithm for data stream has the following requirements: 1) the number of natural clusters without assumptions; 2) the ability to discover clusters of arbitrary shape; 3) the ability to deal with outliers.

Response to the above the requirement, this paper proposes an algorithm. The data allowed to enter are clustered into groups based on Euclidean distance from cluster centers and then the clustered groups when finally filled and reached to their maximum point limit then they are allowed to go into the bucket. The clusters go into the bucket from which they have least Euclidean distance. The data is checked for the outliers based on time stamp associated with each point and distance from the centers when each bucket gets filled. The filled bucket can get more space if other buckets are not filled. This allows the buckets to change their size. Therefore this method also deals with variable capacity of buckets.

The rest of the paper is organized as follows. Section 2 briefly describes the related work which has been done in previous years, how this work is ineffective or what are the drawbacks of their proposed methods due to which this method is proposed. Section 3 briefly describes the advantages of this algorithm over other algorithms and the method proposed to get a brief overview of the algorithm that how it works, what are its key features and section 4 gives a detailed description of the proposed method. Application of this method can cluster as well as detect the outliers with variable size buckets. The proposed method has many cases which are actually handling the conditions which arrive in while dealing with data and many of the conditions arrive very frequently. This method is described in a lot which will help to understand it effectively.

Section 5 provides a detail of experiments and results which includes the testing of algorithm on the different data sets and the results which came as output. This section provides a detailed analysis of the results. Section 6 provides the future work which could be done or not yet done in this algorithm. Section 6 provides the future work which could be done or not yet done in this algorithm. Section 7 gives the final conclusion derived from the detailed analysis and testing of the proposed method.

II. RELATED WORK

K means technique is a common technique to analyze data. It is known that the problem is NP-hard and this many approximation algorithms have been developed. Traditionally, research has been focused to outlier detection in the data and quality of the clusters. Recently there has been growing interest in reducing problems to k means problems. There are many other k means algorithms which didn't take into account the selection of k centers.

Previous work related to clustering give more emphasize on the time efficiency and quality of cluster is poor. Ankit and Amit [1] presented a nice way to cluster data but didn't account the quality of cluster. Mathew and Samir [2] proposed method to detect outliers but accounted the time efficiency of algorithm more than quality of clusters. Dhariwal, Bhatia and Bansal [20] proposed a distance based approach for outlier detection which works well in dynamic data streams but didn't accounted the time of arrival of data as a approach to delete the outliers. This algorithm introduces a new aspect of variable size of buckets. Proposed algorithm is a simple technique of efficient clustering.

III. PROPOSED ALGORITHM

A. *Motivation of the work*

This work is motivated by the observation of shortcoming in the proposed methods. Some methods deals with outlier data well but didn't account the quality of cluster whereas some deal with the clustering of data in well manner but didn't account the outlier data. In many algorithms the proposed method didn't explain the issue of selection of k value this algorithm also takes into this account. In some of the algorithms a problem arises that sometimes some buckets gets filled while other has a lot space. It should use that space for the filled bucket. This introduces the work of variable size buckets.

B. *Proposed method*

Method is proposed to cluster the data by means of pre cluster algorithm. The algorithm includes some k pre clusters, initially the k points which arrive appointed as cluster center then the distance between the k clusters is calculated. Minimum distance between the clusters is diameter of each and every cluster. As a point arrives which doesn't lie in any cluster then the two clusters having minimum distance are merged and new point is appointed as cluster center.

The filled cluster which is transferred is checked for outlier detection and finally outlier free data is transferred as shown in fig. 1 to the buckets nearest to the cluster and the distance is measured using (1).

$$D(C_B,C_C) = \sum(X_{Ci} — X_{Bi})^2 \quad (1)$$

Where i=1 to p.
p- Number of dimensions.
C_B- Center of bucket.
C_C- Center of the cluster.
X_{Ci}- Coordinate of i_{th} dimension of center of cluster C in meters.
X_{Bi}- Coordinate of i_{th} dimension of center of Bucket B in meters.
D- Distance of cluster from bucket in meters.

When the bucket gets filled then it is allocated some space from another bucket which is not filled till now.

But when all the buckets get filled the points are deleted to empty some space in bucket.

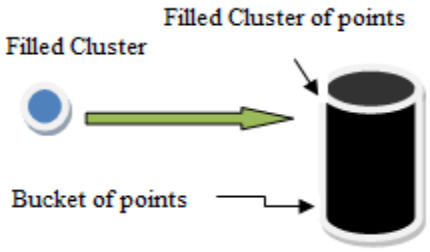

Figure 1. Filled Cluster moving in a bucket.

IV. METHOD

Data stream contains a large amount of data. N points from the data stream are allowed to enter. Then the k points are selected as k centers based on the following algorithm.

N- Number of points allowed to enter, this should be very large than the value of k.

k- Number of buckets we use this can vary and is decided on the basis of available memory.

- For each element or point from the n points calculate its distance from all other points and sum them.
- The point for which the sum is minimum is appointed as a cluster center.
- The steps 1 and 2 are repeated till k centers are selected but don't include the selected centers for step 1 and 2.

```
For i=1 to n
        For j=1 to n
                sum[i]=sum[i]+distance between i, j point.

For k=1 to n
        The k minimum values are appointed as cluster
        centers.
```

From huge amount of data first k points are allowed. The k points should be distinct; if not then the same points are considered in the same cluster. Now as a data point comes it is assigned to a particular cluster.

Each cluster can have maximum of z points. After that cluster is considered as filled and no more points can come. The radius of each cluster is min(r)/2.

min(r) - minimum distance between any two clusters from the k clusters in meters.

k- Total number of pre clusters.

z- Maximum number of points in a cluster.

Each point is associated with a time stamp say T1, T2…….etc.

There are some key points:

- A time stamp is associated with each cluster center as well as a point.
- As a new center is appointed or a center is modified then the time stamp associated with the center is also modified.
- If there are k clusters then buckets are also k only
- Each bucket have a capacity of storing k*z number of points.

A. *Analysis of Different cases*

1) *Case(1)*

New point arrives and doesn't lie in any cluster. Then the new point is appointed as a cluster center and the two clusters having minimum distance are merged with a new center C which is defined as

$$C= (C_1+C_2)/2 \quad (2)$$

and radius is again obtained as previously obtained that is min(r)/2. Now if some points of merged cluster doesn't lie in the cluster due to new radius then the points are considered as new points and case(1) is repeated for them and the point which has arrived first according to time stamp is considered first for repetition of case(1).

2) Case(2)

If a cluster doesn't get any point in its cluster as other clusters gets filled and go to bucket, then this point is considered as outlier and deleted. The others k-1 clusters should get filled from initially being emptied.

3) Case(3)

When a cluster gets filled and goes into a bucket but a next point arrives which lay into the assigned radius of a cluster then the point will go into that cluster and the new cluster is not appointed for that point. But if newly arrived point doesn't lie into any of the k-1 clusters then the points appointed as a new cluster center and the radius of each cluster is calculated as discussed initially.

4) Case(4)

When a cluster gets filled and go into a bucket and a new point arrives and case(3) occurs and every time like this when a new point arrives case(3) occurs and like this all clusters gets filled and go into buckets. Now the new distinct k points which arrive are appointed as cluster centers.

5) Case(5)

When a cluster gets filled and is ready to go into the bucket then a check is performed on the cluster for outlier detection. There is an algorithm to detect outliers in the cluster.

Here we calculate d_p and t_p and will be divided with D_{total} and T_{total} respectively, so that it becomes normalized.

d_p and t_p stored for each point is evaluated and the points are deleted according to the following algorithm.

d_{pi}- distance of point i from center of cluster in meters.
t_{pi}- time stamp associated with point i in seconds.

- D_{total} is calculated which is equal to

$$D_{total}=\sum d_{pi} \quad (3)$$

Where i=1 to n.
D_{total}- sum of d_p of all points in meters.
n- Total points in the cluster.

- T_{total} is calculated which is equal to

$$T_{total}=\sum t_{pi} \quad (4)$$

Where i=1 to n.
T_{total}- sum of t_{pi} of all points in seconds.
n- total points in the cluster.

- The point having maximum value of outlier and much different from other points is considered as an anomaly and deleted.

$$outlier=d_p/D_{total}+t_p/T_{total} \quad (5)$$

The points which have outlier value greater than $D_{total}/(n*d_v)+T_{total}/(n*T_v)$are considered as anomaly and deleted.

d_v- 1 meter
T_v- 1 second
These values are used just to make the equation dimensionless.

These steps are executed iteratively till all the outlier points are deleted. Then an outlier free cluster will proceed.

6) Case(6)

When a cluster goes through the case(5) successfully and didn't recognize as cluster then the cluster is ready to go into a bucket. The bucket is assigned to the cluster by using (1). Then the center of bucket needs to be modified due to the new point and the new center of bucket is calculated using (6).

$$C= (C_1*N_1+C_2*N_2)/(N_1+N_2) \quad (6)$$

C- New center of bucket after arriving of new cluster.
C_1- center of bucket before coming of cluster.
N_1- number of points in bucket before coming of the cluster.
N_2- points in the filled cluster ready to go into a bucket.
C_2- center of filled cluster arrived to go into a bucket.

This method is a weighted method to calculate the center of bucket after accepting a cluster.

7) Case(7)

When a cluster successfully clears case(5) means not detected as a outlier or anomaly point and then a bucket is also allotted to the cluster through the case(6) then the cluster comes into a bucket.

But suppose the cluster finds that the bucket filled then the filled bucket will vary its size and take some space from some other bucket which is not filled yet. So in this way size of filled bucket gets increased and now it can accommodate the cluster and the size of another bucket gets decreased. In this way the size of buckets vary and considered as variable buckets.

8) Case(8)

When a cluster goes through case(7) and no bucket has enough space to allocate to the bucket in which the cluster will go then the points which lie outside to $r^{3/4}$ of all the filled buckets are deleted. In this way some memory is freed from the bucket and then again checks whether the bucket have sufficient space or not, if there is sufficient space avaliable then the case(6) and case(7) are repeated and if not then

case(8) is repeated till the cluster didn't get enough space in the bucket.

r- Radius of bucket in meters.

9) Case(9)

When a bucket doesn't get any cluster or entry and lags by the k times average the time taken to fill a bucket then the whole bucket is considered as containing outlier data and is being emptied.

B. Calculation of z

$$k*z+k*k*z=N \quad (7)$$

N-maximum number of point storage capacity, the total memory of storage.

Solving the (7) we get

$$z=N/(k+k^2) \quad (8)$$

If z is not an integer then the least integer greater than N/ (k+k2) is used.

With the help of (7) or (8) we can calculate the values of k and z that are used in this whole algorithm.

V. RESULTS

A. Data sets and parameter settings

Many test data sets are used to test this algorithm such as iris data set [5] and KDD 1999 data set of intrusion detection. This algorithm is tested for small sets and low dimensional data sets.

Furthermore all the tests are carried out for 30 runs and 30 iterations per run. For all the tests an average error percentage and outlier density is calculated and shown in experimental results.

B. Experimental Results

First we will discuss about the error percentage shown in fig. 2. This graph is plotted for the KDD cup 1999 data set of intrusion detection.

A table (Table I.) is drawn on the basis of error percentage and algorithms to compare the data set of intrusion detection of KDD cup 1999. The algorithm 1 and 2 are some other algorithms which are used to compare the quality of clustering. The names of those algorithms are not used directly here.

TABLE I. ERROR PERCENTAGE

Algorithm	Error Percentage
Pre clustering with variable size buckets.	0.03
Algorithm 1	0.045
Algorithm 2	0.06

This intrusion detection in a system is a task to detect network intrusions and protect computer systems from unauthorized users, including perhaps insiders for the purpose of security.

The intrusions are detected various techniques and anomaly detection is from one of them. So we have applied our algorithm of anomaly detection and tests whether the algorithm is correct or not and efficiently detects the outliers or not.

According to the statistics shown in the Table I, the proposed method gives a quality of 99.97 percent whereas the other algorithms as shown in fig. 2 gives a error percentage of 0.045 and 0.06.

The fig. 2 compares the results of our algorithm and other algorithms which show that our algorithm is better as it has a low error percentage which makes it an improved algorithm.

The experimental results for clustering of data streams on iris data set shows only the algorithms ability to cluster the data set and not tests its ability to detect the outliers.

The experimental results about the algorithm show that the proposed algorithm clusters the data more efficiently and detect the anomaly efficiently. This algorithm compromises with the time efficiency but improves the quality. This doesn't have a better time complexity than other algorithms.

We discuss a series of assessments results of the improved algorithm in this section. This algorithm uses Microsoft visual C++ programming. The code can be written in JAVA also.

Y-axis – error percentage

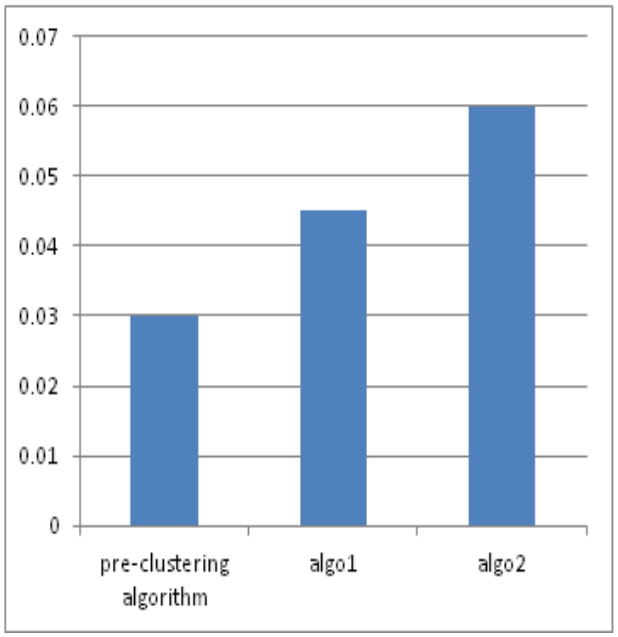

Figure 2. Error Percentage of different algorithms on kdd cup 1999 data set of intrusion detection.

VI. FUTURE WORK

The proposed algorithm has worked on mostly numerical data sets and for low dimensional data sets. In future the work can be extended for the other type of data such as Boolean data and categorical data. This algorithm deals with outliers considering time effect and distance effect in combined, but in future we can also use only time effect for anomaly detection. As we move towards the high dimensionality the quality of clustering algorithm gets affected. The work can be done on this part also. The technique of variable size of bucket is used at a preliminary level; it could be extended much more so that it can be used efficiently.

VII. CONCLUSIONS

The experiments and results of the above proposed algorithm show that the above considers quality over time efficiency. The algorithm performs a three time check on entry of each cluster firstly in case(2) when it checks the cluster for its last entry, second time in case(5) when the check is performed just before the entry in bucket, third time in last case where it deletes the points that doesn't come in $r^{3/4}$ range of radius of bucket.

This algorithm deals with a newly introduced efficiently technique of variable size buckets. With the help of such buckets the essential points can maintained in the bucket for longer time.

ACKNOWLEDGMENT

I would like to thank my professor to guide me so well while doing this project and to help me in framing this paper. A special thanks to others who helped in designing the paper by giving their valuable comments and suggestions which helped me a lot in improving this technique.

REFERENCES

[1] Ankit aggarwal, amit deshpande and Ravi Kannan, "Adaptive sampling for k-means clustering", Microsoft research India, I. Dinur et al. (Eds.): APPROX and RANDOM 2009, LNCS 5687, pp. 15–28, 2009., Springer-Verlag Berlin Heidelberg 2009.

[2] Richard Matthew McCutchen and Samir Khuller, "Streaming algorithms for k-center clustering with outliers and with anonymity", A. Goel et al. (Eds.): APPROX and RANDOM 2008, LNCS 5171, pp. 165–178, 2008, Springer-Verlag Berlin Heidelberg 2008.

[3] R.A. Fisher Iris data set, 1936. Available from: http://archive.ics.uci.edu/ml/datasets/Iris.

[4] Data minig concepts and techniques- Jiawei Han and Micheline kamber.

[5] KDD cup 1999 dataset of network intrusion detection in systems. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[6] Aloise , D. Deshpande, A. Hansen, P. Popat, "NP-hardness of Euclidean sum-of-squares clustering". Machine Learning 75(2), 245-248(2009).

[7] Moses Charikar, Liadan O'Callaghan, Rina Panigraphy, "Better Streaming Algorithms for Clustering problems" , In Proc. Of STOC'03, June 9-11, 2003, San diego, California, USA, 2003 ACM.

[8] Ramaswamy S., Rastogi R., Kyuseok, "Efficient Algorithms for Mining Outliers from Large Data Sets",Proc. ACM SIDMOD Int.Conf. on Management of Data, 2000

[9] M.M. Breunig, H.P.Kriegel, R.T. Ng and J.Sander, "Identifying Density-Based Local Outliers" ACM SIGMOD 2000

[10] Manzoor Elahi, Kun Li , Wasif Nisar, Xinjie Lv ,Hongan Wang ,"Efficient Clustering-Based Outlier Detection Algorithm for Dynamic Data Stream".In Proc. Of the Fifth International Conference on Fuzzy Systems and Knowledge Discovery(FSKD.2008).

[11] Angiulli, F. and Fassetti, "Detecting Distance-based Outliers in Streams of Data". In Proc. of the Sixteenth ACM Conf. on information and Knowledge Management (Lisbon, Portugal, November 2007).CIKM '07.

[12] Charu C. Aggarwal, Philip S. Yu, "Outlier Detection for High Dimensional Data", Proc. of the 2001 ACM SIGMOD int. conf. on Management of data, p.37-46, May 21-24, 2001, Santa Barbara, California, United states

[13] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data". In Data Mining for Security Applications, 2002.

[14] Knorr, E. M., Ng, R.T, "Algorithms for Mining Distance-Based Outliers in Large Datasets", Proc. 24th VLDB, 1998

[15] A. Arning, R. Agrawal, P. Raghavan, "A Linear Method for Deviation Detection in Large Databases". In: Proc of KDD'96, 1996: 164 -169.

[16] S.Harkins, H.He, G. J.Willams, R.A. Baster, "Outlier Detection Using Replicator Neural Networks". In: Proc of DaWaK'02, 2002: 170-180.

[17] Dragoljub Pokrajac ,Aleksandar Lazarevic ,Longin Jan Latecki , "Incremental Local Outlier Detection for Data Streams". IEEE Symposium on Computational Intelligence and Data Mining (CIDM), April 2007.

Article in conference proceedings:

[18] T. Zhang, R. Ramakrishnan, M. Livny. BIRCH: "An Efficient Data Clustering method for very large Databases", ACM SIGMOD Conference 1996.

Article in a journal:

[19] K. Jain and Vijay Vazirani, "Approximation Algorithms for Metric facilityLocation and k-median problems using the Primal-Dual Schema and Langrangian Relaxation." . Journal of the ACM 2001.

[20] Parneeta dhaliwal, MPS Bhatia and Priti Bansal, "A Cluster-based approach for Outlier Detection in Dynamic Data Streams (KORM: k-median OutlieR Miner)", JOURNAL OF COMPUTING, VOLUME 2, ISSUE 2, FEBRUARY 2010, ISSN 2151-9617