Applicative Implementation of D-Stream Clustering Algorithm for the Real-Time Data of Telecom Sector

Aneequa Sundus, Muhammad Hammad Ali, Waleed Qaiser, Zeeshan Ahmed and Zahid Halim

Department of Computer Science, National University of Computer and Emerging Sciences
Islamabad, Pakistan
aneequasundus@gmail.com,{hammad070198,waleed_qaiser}@yahoo.com,zeeshan_malix@hotmail.com,
zahid.halim@nu.edu.pk

***Abstract*— In the recent past, telecommunication industry has gone through tremendous growth. It has resulted in huge production of data on daily basis for the telecom companies. Now it is a challenge for the telecom operators to manage huge amount of data and then use this data for decision and policy making processes. The data generated in telecom industry is in the form of data stream as it is continuously being generated. So we need such a clustering algorithm which can perform well with streams or continuous data. At the same time we also need to detect outliers and erroneous data. Clusters are not necessarily of globular shape as we don't have prior knowledge of number of clusters and their shape. To address all the above mentioned problems we have used D-Stream clustering algorithm in our implementation to get desired results. This paper, discusses the algorithm and implementation of D-Stream algorithm along with its experimental results on synthetically generated telecommunication data.**

Keywords- Data Streams; Clustering Telecom data; evolving clusters

I. INTRODUCTION

Due to the tremendous growth of telecom sector, a potential for decision support systems has been generated. We need to develop such kind of decision support systems which can handle large amount of real time data. Data clustering is an important technique of data mining and is used to find hidden patterns in large amount of data. Traditional data clustering techniques work well with static data but they are not much efficient while working with data streams. We define the stream as a sequence of records stamped and ordered by time [1]. In data stream we don't have large chunks of data rather data is being continuously generated with time. So we need an algorithm which can handle large amount of continuous data along with its time stamp and can perform real time analysis on it. It should also be able to detect outliers and noisy data [1] [6]. It should perform clustering of evolving data streams in a single scan of data.

The clustering and analysis of telecom data streams can be used for many useful applications like customer profiling, detecting customer churn rate, planning marketing strategies, association group mining etc. [5] [11]. In all of the above applications, data is in the form of data streams and many studies are going on for handling such a large amount of data in a single scan.

Density based clustering algorithm is thought to be efficient in handling all of these potential problems of data stream and it is one of the best choices for handling the telecom data. D-Stream algorithm can detect outliers, can handle large amount of data in a single scan and it can generate clusters of arbitrary shapes [1]. It attaches a decay factor to each incoming record and thus it gives more importance to new records and removes the outdated records which are not required in the present context. It is space efficient as it keeps on removing the outdated sporadic grids. So we have used this algorithm for our implementation for the data of telecommunication sector.

II. PREVIOUS WORK

In the previous data clustering algorithms such as in single-phase model based clustering the data streams were considered as the continuous big chunks of static data[9]. These algorithms use divide and conquer concept and their basic ideas to generate the clusters were based on the K-means clustering in the finite space [1]. These algorithms contain some serious limitations. One of the serious limitations of this approach is that they put equal weights to the recent and the outdated data which is actually not suitable for the identification of changing characteristics in the data streams which is the basic essence of the real-time based data streams analysis. The concept of moving window partially solves this problem.

One of the recent techniques used for the clustering of data streams is proposed by Aggarwal et al. This approach is usually termed as two phase clustering scheme. According to this approach we have two main components which help us to generate the final clusters [1] [9]. The working of the

online component includes the processing of the raw data coming from the real-time data streams and the generation of summary statistics from this data. Similarly the working of the offline component includes the usage of this generated summary data and the generation of clusters. The basic essence of this approach is yet again is based on K-means clustering approach which has some limitations when dealing with the real-time data streams.

One of the modifications of this two-phased scheme is called as Clustream-clustering approach which is now being recently used for many clustering based applications. The basic concept of Clustream approach is based on the two-phased clustering approach with an improved offline component using the mechanism of incomplete-partitioning strategy [8]. Extensive work has been done using this approach including clustering multiple data streams, parallel data streams and distributed data streams. A number of limitations occur in this Clustream and other co-related approaches. The problem actually lies with the usage of the K-means approach in their offline component. K-means itself contains some serious drawbacks when dealing with the real-time data streams. One of the fundamental problems with K-means is its incapability to generate clusters of arbitrary shapes because K-means always tends to generate the clusters of spherical shape [1] [7].However in some applications modified versions of k-means have been used which are able to detect interwoven and non convex clusters but yet they are not fully capable to generate the clusters of any possible shape. Another problem with K-means approach is that it requires multiple scans of the data which is practically not possible in the case of real-time data streams. K-means is also not able to detect noise and outliers efficiently which actually decreases the quality and accuracy of the clustering process.

III. DESCRIPTION OF DATA

We have used the data attributes taken from a telecom company to generate data for our experiments. There were 23 attributes whose values were generated. These attributes and their format are explained in Table 1 (Appendix-I): In order to discritize the continuous attributes, we have made ranges of those attributes which we have considered for dicritization. These attributes are duration, CallInitiatedDateTime, CallConnectedDateTime, CallDisconnectedDateTime, and AutoSwitchTime. The ranges for duration and Datetime fields are provided in Table 2 (Appendix-I).

IV. DESCRIPTION OF D-STREAM ALGORITHM

For the complete understanding and learning of the D-Stream Algorithm we are using the research paper “Density Based Clustering Approach for Real-Time Stream Data” by Yixin Chen and Li Tu [1]. We completely acknowledge the work done in this paper and we are using the knowledge from this paper about the D-Stream Algorithm. The D-Stream model consists of an online component and an offline component.

In the online component of this algorithm, each record has been read one by one from the incoming data stream. Here the time stamp is the set of integers which are: 0, 1, 2, …… ,n…… . At each time stamp the online component reads one data record and places this record in multidimensional data into the corresponding density grid in the multidimensional space and updates the characteristic vector. Remember in our case data will be multidimensional which will be in the form of real-time data streams.

In the working of the offline component the offline component will adjust the clusters at every gap time step, where the value of gap time will be in integers. After the first gap time, the algorithm will generate the initial clusters. Algorithm will also periodically remove sporadic grids and it will also regulate the clusters.

Figure 1 is providing you a very brief overview of the overall “D-Stream approach” which says that in the “Online component” grids will be generated and data records will be mapped on these density grids In the offline component grids densities along with the generation of “Dense”, “Sparse”, and “Transitional” grids will take place which will in the end will return clusters of arbitrary shape and size.

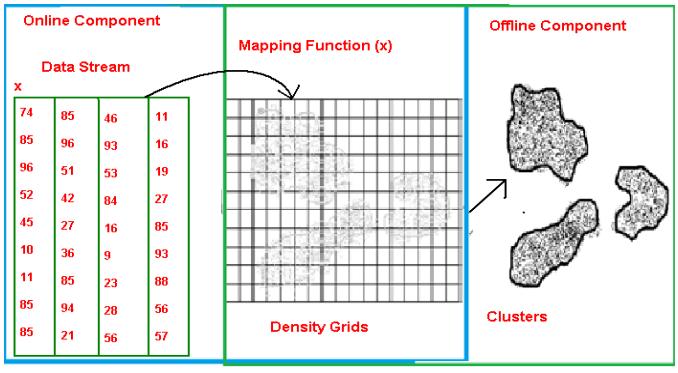

Figure 1: Basic Functionality of D-Stream Algorithm

V. IMPLEMENTATION OF ALGORITHM

In this section we will be discussing about implementation of the D-Stream algorithm. The complete algorithm of D-stream clustering is given [1]. In our implementation of this algorithm, first of all we initialize a hash table. Key of this hash table is generated by the concatenation of different attribute values. Knowledge engineer chooses those attributes from data stream which are needed to be considered in clustering. We then generate a key for hash table by concatenation of those selected attributes. We are also maintaining a characteristic vector which consists of $t_{g,}$ $t_{m,}$ D and label. $t_{g,}$ is the last time when a grid is updated, t_c is the last time when grid is removed and D is the density of the grid at the last update and label is the label of the grid which can be "dense" or "sparse".For example in a specific case, knowledge engineer chooses Country Code, Conn_type, Duration, DisconnectCauseCode and CallType for data clustering. So in this case the key for record 1 will be 92, 306, 2486, 15, 4. The selection of attribute by the knowledge engineer will be based on the requirement and the need and the selection will be made after a thorough analysis.

Record#1:
1*92****306****52****40221****2486****1/1/20105:34:21AM****1/1/20105:34:55 AM****1/1/20105:35:00AM****15****107.75.194.199****169.97.210.2821 4****9****1****30****19****4/30/20019:33:05AM****101****21****0*

A. Initial Clustering

This key is then mapped on the hash table. So we read each record one by one and map it accordingly. When one record maps on the hash table, t_c is incremented by 1. So t_c is incremented with each record. When t_c reaches to the time interval gap for first time, initial clustering function is called. Next time whenever t_c reaches the multiple of time interval gap, adjust clustering is called.

When initial clustering is called, first densities of all grids are updated as they are changed due to the effect of decay factor. At the same time, their labels are also updated. To start clustering, we label all dense grids in distinct clusters. We store these clusters in a 2D list. We then traverse the cluster's grids, and for each grid when we traverse it, we find its neighboring grids. Then for each grid g of cluster c, we will traverse all of its neighboring grids. If a neighboring grid g belongs to any other cluster, we then take the length of cluster and then decide the label of that grid. If the neighboring grid is a transitional grid, we label it as "dense" or "transitional" other-wise it will be updated as a "sparse grid". After each time interval we are continuously maintaining the characteristic vector which is used to calculate the densities of the updated grids as well assigning the label to the newly or previously updated grids.

B. Adjust Clustering

Adjust clustering is called when t_c is a multiple of time interval gap. In this process we first update the densities of all the grids. We then consider only those grids for further analysis, whose status is changed since last time clustering was called. We then check the grids and assign them new clusters, break previous clusters and change the labels of grids.

The dimensions we have used for the implementation purpose are as follows: Country code, Conn_type, Area_Code, Duration, CallInitiatedDateTime, CallConnectedDateTime, DisconnectCauseCode, OriginationTrunkID, CallType, IncomingLine, IncomingChannel, OutgoingLine, and OutgoingChannel.

For the purpose of implementation we have used Microsoft Visual Studio dot net 2010 and we have used C# language for the purpose of programming. We have also used Microsoft SQL Server 2008 for the generation and the maintenance of the database.

C. Experimentation and Results

For the experimentation purposes to check the efficiency of D-Stream algorithm, we used a PC with Core duo 2.0 GHZ with a RAM of 2GB running on Windows XP.

We have set the parameters (C_m, C_l, h and B) [1] to ensure the complete efficiency of our algorithm. The parameters are: C_m= 2, C_l=0.5 h=0.998 and B=0.3. First we have generated a data of 80k with 12 dimensions and we set the speed of data stream to be 1000 points per unit time, so the data stream will end in 80 time units. We choose 10, 30, 50, 60 and 80 as the times at which we measure the cluster quality and number of generated clusters. Cluster purity is basically the average percentage of dominant class label in each cluster. The average percentage of the dominant class label is directly proportional to the purity of the cluster. Usually more number of clusters provides higher cluster purity. The results of this experiment are shown in Figure 2.

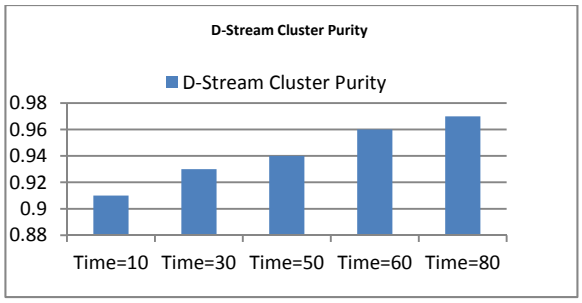

Figure 2: D-Stream Cluster Purity

The results of the experiments showing the number of clusters at time intervals 10, 30, 50, 60 and 80 are shown in Figure 3.

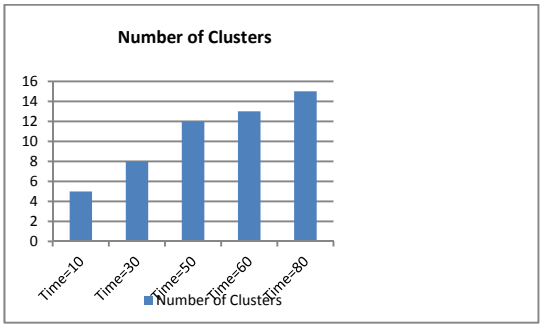

Figure 3: D-Stream Number of Clusters

For the time comparison we have used a data of 10k with 12 dimensions and noted the computational time which comes out to be around 45 seconds. This result shows that if we increase the dimensions of the data the computational time will also increase. The results of this experiment showing the increase in the computational time with the increase in the number of dimensions are shown in Figure 4.

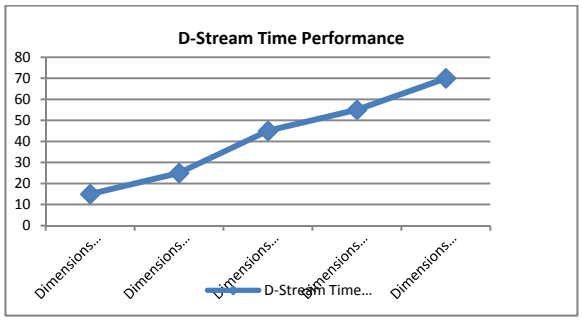

Figure 4: D-Stream Time Performance

VI. CONCLUSION

The paper thoroughly discussed the implementation of "D-Stream Approach" for the real time data streams. We find that the D-Stream clustering algorithm provides very good results while working with streams. D-Stream algorithm consists of an online and an offline component which computes the density of each grid and generates clusters based on D-Stream algorithm. We have also found that by using density decaying factor we can efficiently detect and remove outliers. Unlike previous clustering algorithms D-Stream algorithm can detect clusters of arbitrary shape and it can also detect the outliers of arbitrary shape. Moreover the experimentation results on the telecommunication data validate the required results in terms of computation time, accuracy and efficiency. This approach can be used for the huge amount of stream data generated by telecommunication companies and it can be used for the applications like detecting network intrusion from the streams, immediately detecting the unusual behavior of customers (Fraud-detection), analyzing the usage of the telecommunication network by the customers and providing immediate, targeted and customers oriented packages and schemes and detection of association groups of the customers can also be found out. So in short we have a lot of telecom based applications for "D-Stream Approach" and we propose this scheme due its accuracy and efficiency for the telecom sector streams data.

REFERENCES

[1] R Yixin Chen and Li Tu, "Density-Based Clustering for Real-Time Stream Data," *in Proc. 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, 2007.

[2] Madjid Khalilian and Norwati Mustapha, "Data Stream Clustering: Challenges and Issues," *in Proc. International Multi Conference of Engineers and Computer Scientists 2010 Vol I, IMECS 2010, March 17-19*, Hong Kong, 2010.

[3] Charu C. Aggarwal and Philip S. Yu, "Framework for Clustering Uncertain Data Streams," in Proc. ICDE '08 Proceedings of the 2008 IEEE 24th International Conference on Data Engineering, Washington, DC, USA 2010.

[4] Garry Weiss, "Data mining in telecommunications,": Kluwer Academic Publishers, 2004.

[5] Yu-Teng-Chang, "Applying Data Mining to Telecom Churn Management," presented at International Journal of Reviews in Computing, 2009.

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

[7] Renxia WAN and Lixin WANG, "Clustering over Evolving Data Stream with Mixed Attributes," presented at Journal of Computational Information System, 2010.

[8] Li Wan, Wee Keong Ng, Xuan Hong Dang, Philips. Yu and Kuan Zhang, "Density-Based Clustering of Data Streams at Multiple Resolutions,"
presented at ACM Transactions on Knowledge discovery from Data (TKDD), 2009.

[9] Charu C. Aggarwal, Jiawei Han, Jianyong Wang and Philip S. Yu, "A Framework for Clustering Evolving Data Streams," in Proc. the 29th international conference on Very large data bases - Volume 29, 2003.
[10] Dorina Kabakchieva, "Business Intelligence Applications and Data Mining Methods in Telecommunications: A Literature Review," 2009.
[11] S.M.H. Jansen, "Customer Segmentation and Customer Profiling for a Mobile Telecommunications Company Based on Usage Behavior," 2007.
[12] S. Guha, N. Mishra, R. Motwani, "Clustering data stream," in Proc. FOCS, 2000.
[13] P. Domingos and G. Hulten, "Mining high-speed data streams," in Proc. KDD, 2000.
[14] Xin Sun Yu (Cathy) Jiao, " pGrid: Parallel Grid-Based Data Stream Clustering with MapReduce," presented at Applied Software Engineering Research Group, 2009.
[15] M. Klusch, S. Lodi, and G. Moro, "Distributed Clustering Based on Sampling Local Density Estimates," *in Proc.Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

Appendix-I

Sequence Number	Attribute Name	Data Format/ Range	Explanation
1	CallSequenceNo	0- 4,294,967,295	It is the unique identification number which is assigned to every new call. When it ranges a certain thresh hold, it again starts from 0.
2	Country Code	001-0092	It is the country code of dialed number.
3	Conn_type	0300-0346	It is the code for telecom company of the dialed number.
4	Area Code	021-0543	It is the city code of dialed number.
5	CalledNo	1000-9999	It is the unique identification of dialed number.
6	Duration	0-3600	It is the duration of call. If call is not connected then this duration is 0.
7	CallInitiatedDateTime	yyyymmddhhmmss	It is the time and date when the call was initiated.
8	CallConnectedDateTime	yyyymmddhhmmss	It is the time at which call was connected to the dialed number and communication started.
9	CallDisconnectedDateTime	yyyymmddhhmmss	It the call end time.
10	DisconnectCauseCode	0-20	If call is not connected, it is the code for cause of disconnection. This field will be 0 if call was connected.
11	LocalIPAddress	xxx.xxx.xxx.xxx.	This the IP address of local server.
12	RemoteIPAddress	xxx.xxx.xxx.xxx.	This is IP address of remote side server.
13	OriginationTrunkID	0-9	This field tells the code for Origination Trunk which initiated the call. This code is usually configured with an account code.
14	CallType	1-4	This is the code for type of call e.g. voice call, fax etc.
15	CallNumberType	1 or 9	It is the code for called number numbering plan e.g. private or public
16	IncomingLine	1 or 2	This is the field which identifies the line of incoming call.
17	IncomingChannel	0-31	This is the field which identifies the channel of incoming call.
18	OutgoingLine	1 or 2	This is the field which identifies the line of outgoing call.
19	OutgoingChannel	0-31	This is the field which identifies the channel of outgoing call.
20	AutoSwitchTime	yyyymmddhhmmss	This is the time when auto switch occurred.
21	AutoSwitchDuration	01-99	These are the seconds taken in auto switching.
22	BadQualityIPEvents	1-9	This is number of bad quality events if they occurred. This number overall gives the quality of call.
23	AutoSwitchFlag	0-1	This field identifies whether call was terminated to auto switch agent.

TABLE 2: DATA DESCRIPTION

Duration Ranges	Datetime Ranges
<60	12 AM to 7AM
61-300	7 AM to 12 PM
301-1800	12 PM to 3PM
1800-3600	3PM to 7PM
3601-4000	7 PM to 12 PM

TABLE 2: RANGE DISTRIBUTION