Monali Patil, Vidya Chitre, Dipti Patil / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 4, July-August 2012, pp.144-148

Improved K-mean clustering with Mobile Agent

Monali Patil, Vidya Chitre, Dipti Patil
Department of Information Technology, K. C. COE, Thane
Department of Computer Engineering, Bharati Vidyapidth, Navi Mumbai
Department of Information Technology, PIIT, New Panvel
Mumbai University
Maharashtra
India

Abstract— The data stream model has recently attracted attention for its applicability to numerous types of data, including telephone records, Web documents, and click streams. For analysis of such data, the ability to process the data in a single pass, or a small number of passes, while using little memory, is crucial. D-Stream algorithm is an extended grid-based clustering algorithm for different dimensional data streams. One of the disadvantages of D-Stream algorithm is the large number of grids, especially for high-dimensional data. So we use the mobile agent to deal with the different dimensional data. This method can reduce the number of the grids and the number of the update times. So using mobile agent technique with data stream algorithm we make the data stream clustering more simplified and flexible.

Keywords— **D-Stream, Grid based clustering**

I. INTRODUCTION

Since 1991 Weiser proposed pervasive computing , pervasive computing has been paid more attention to. Pervasive application emphasizes on "real-time" reactions because of the special requirement of the environment. With the development of the distributed system and wireless communication, streams of data are able to be obtained from high-bandwidth and data streams mining has become one of the hot points in the pervasive application. Clustering can provide the real-time analyzing results in the pervasive environment. There are many mature methods can be extended to cluster the data streams. Guha et.al gave an extended k-means clustering method to deal with data streams in 2000. In 2003 Babcock et.al improved Guha's method using the data structure called exponential histogram. O'Challagham et.al proposed STREAM algorithm, which improved k-means using LSEARCH and they got the more qualitative results. CluStream algorithm was given by Aggarwal et.al, which extended the clustering feature concept of BIRCH algorithm.

Clustering is the process of grouping the data into classes or clusters so that objects within a cluster have high similarity but are very dissimilar to objects in other clusters. A good clustering algorithm should be able to identity clusters irrespective of their shapes. D-Stream, is a framework for clustering stream data using a density-based approach. The algorithm uses an online component which maps each input data record into a grid and an offline component which computes the grid density and clusters the grids based on the density. The algorithm adopts a density decaying technique to
capture the dynamic changes of a data stream [4]. It exploits the intricate relationships between the decay factor, data density and cluster structure. An extended DBSCAN algorithm was proposed by Cao F et.al called DenStream algorithm in 2006 , which also used two-phase scheme the online component and the offline component. D-Stream algorithm was an extended Grid-based method proposed by Chen et.al in 2007, which also used two-phase scheme and emphasized the detection of the isolated points.

II. THE CONCEPTS AND DEFINITIONS

Let A = (A1,A2, ..., Ad) be a bounded, all ordered domain set, then we will denote a d-dimensional numerical space by S = A1 × A2 × ... × Ad, and one dimension of S be represented Ad . Data stream is defined as a series of chronological order to reach set of data points, it will be denoted by DS = (X1, X2 ,…, Xn, ...). It is assumed that the data stream consists of a set of multi-dimensional records X1,X2,…,Xn, …arriving at time stamps t1,t2, ... ,tn, ... Each Xi is a multi-dimensional record containing d dimensions which are denoted by *Xi* = (xi1 , xi2 ,…, xid). the Xi = (xi1, xi2, ..., xid) represents a data point (xi1 _ A1, xi2 _ A2, ..., xid _ Ad) [4].

A. *DEFINITION: 1. THE GRID*

The each dimension of the numerical space S is divided evenly into ξ intervals, each interval is the dimension on a right-open interval of paragraph [li ,hi], length will be denoted by ln, where i is followed by numbered 1, 2, ..., ξ. The numerical space S is divided into ξd disjoint grid, each grid g is a d-dimensional hypercube in the S , can be

expressed as (v1 , v2 ,…, vd) , where vi is the ith ($1 \le i \le d$) dimensional interval on the corresponding number in grid g.

B. *DEFINITION: 2 DATA DENSITY, ATTENUATION FACTOR*

The data arriving at time tc be denoted by p, its density D(p,t) is a change in the amount of time t, is defined as:
$D(p,t) = \lambda^{t-tc}$, where λ ($0<\lambda<1$) is a constant, called the attenuation factor.

Definition: 3 Grid density

Set up the grid g at time t contains the data items are p1, p2,…,pN , their time stamp (referring to the arrival time), respectively, t1, t2, …, tn , thus at time t, the density of grid is defined as the sum of the data density of all data items in the grid g at the time t , we will denote the density of grid by

$D(g,t) = \sum_{i=1}^{n} \lambda^{t-ti}$

Definition: 4 Dense grid

In the t moment, if $D(g,t) \ge c/N(1-\lambda) = Dm$ grid g , then claimed the grid as a dense grid , where C is $C \ge 1$,and is a constant (in this algorithm is set up C = 3).

Definition 5. Eigenvector of grid

The vector is defined as a six-tuple (Fx ,Qx , P, density ,class , tg), Where: wherein Fx and Qx each correspond to a vector of d entries. The definition of each of these entries is as follows:

For each dimension, the sum of the squares of the data values is maintained in Qx. Thus, Qx contains d values. The p-th entry of Qx is equal to

$$\sum_{j \in grid} (x_{ij}^{\,p})^2$$
g

For each dimension, the sum of the data values is maintained in Fx. Thus, Fx contains d values. The p-th entry of Fx is equal to

$$\sum_{j \in grid} (x_{ij}^{\,p})$$
g

Where P is a d-dimensional vector and the grid coordinates, it will be denoted by the corresponding dimension's interval numbers in grid. Where density is density values of the grid at time tg , class is the grid cluster labeling belongs to cluster ,tg is a time grid data points recently.

Definition 6. Density threshold function

Let tc be the current time, and tg be the time of the last increased data in grid g. Then the density threshold function is defined as:

$D_{min}(tg, tc) = (1-\lambda^{tc-tg+1})/N(1-\lambda)$

where N is the total number of the grid space.

Its property is as follows:

(1) If $t1 \le t2 \le t3$ then Dmin(t1 ,t3)= Dmin(t1 ,t2) λ^{t3-t2}+ Dmin(t2 +1,t3);
(2) Dmin(t1 ,t1)=1/N;
(3) $\lim_{t->\infty}$ (t1 ,t)= □ □*N(1-λ)*□
(4) Let t1<t2, then Dmin(t1 ,tc)= Dmin(t2 ,tc).

III. D-STREAM ALGORITHM

D-Stream is a new framework for clustering real-time stream data. There are two components: the online component and the offline component. The online component reads and maps each input data record into a grid, and the offline component computes the density of each grid and clusters the grids using a density-based algorithm in every fixed interval of time (gap). Actually, in the offline component there is the general grid-based clustering algorithm. In the online component the algorithm adopts a density decaying technique to capture the dynamic changes of a data stream and exploits the intricate relationships between the decay factor, data density and cluster structure.

Once the densities of grids are updated in the gap, the clustering procedure is similar to the general grid-based clustering. This algorithm can find arbitrary shape clusters by a systematic visit of the neighbors and it is not necessary to specify the number of cluster like k-means algorithm. However, there is a higher requirement that if the dimension is high, the number of the grids must be very large.

For a data stream, at each time step, the online component of D-Stream continuously reads a new data record, place the multi-dimensional data into a corresponding discretized density grid in the multi-dimensional space, and update the characteristic vector of the density grid (Lines 5-8 in figure 1). The offline component dynamically adjusts the clusters every gap time steps, where gap is an integer parameter. After the first gap, the algorithm generates the initial cluster (Lines 9-11). Then, the algorithm periodically removes sporadic grids and regulates the clusters (Lines 12-15).

```
1.procedure D-Stream
2. tc = 0;
3. initialize an empty hash table grid list;
4. while data stream is active do
5. read record x = (x1, x2, · · · , xd);
6. determine the density grid g that contains x;
7. if(g not in grid list) insert g to grid list;
8. update the characteristic vector of g;
9. if tc == gap then
10. call initial clustering(grid list);
11. end if
12. if tc mod gap == 0 then
13. detect and remove sporadic grids from grid list;
14. call adjust clustering(grid list);
15. end if
16. tc = tc + 1;
17. end while
```

18. end procedure.

The algorithm adopts a density decaying technique to capture the dynamic changes of a data stream. Exploiting the intricate relationships between the decay factor, data density and cluster structure, the algorithm can efficiently and effectively generate and adjust the clusters in real time. We assume that the input data has d dimensions, and each input data record is defined within the space,
S= S1*S2*……*Sd
where Si is the definition space for the ith dimension. In D-Stream, we partition the d−dimensional space S into density grids. Suppose for each dimension, its space Si, i = 1, · · · , d is divided into pi partitions as

Si= Si,1U Si,2 U…… U Si,pi,

then the data space S is partitioned into $N = \pi^{d}_{i=1}$ Pi density grids. For a density grid g that is composed of S1,j1 *S2,j2 …….. * Sd,jd , ji = 1 …….. pi, we denote it as g = (j1,j2, …… , jd).

A data record x = (x1, x2, · · · , xd) can be mapped to a density grid g(x) as follows:
g(x) = (j1,j2,….., jd) where xi € Si,ji

For each data record x, we assign it a density coefficient which decreases with as x ages. In fact, if x arrives at time tc, we define its time stamp T(x) = tc, and its density coefficient D(x, t) at time t is

$$D(x,t) = \lambda^{t-T(x)} = \lambda^{t-tc} \quad ,$$

where λ€ (0, 1) is a constant called the decay factor. Grid Density for a grid g, at a given time t, let E(g, t) be the set of data records that are map tog at or before time t, its density D(g, t) is defined as the sum of the density coefficients of all data records that mapped to g. Namely, the density of g at t is:

$$D(g,t) = \sum_{x \in E(g,t)} D(x,t)$$

The density of any grid is constantly changing. However, we have found that it is unnecessary to update the density values of all data records and grids at every time step. Instead, it is possible to update the density of a grid only when a new data record is mapped to that grid [3]. For each grid, the time when it receives the last data record should be recorded so that the density of the grid can be updated according to the following result when a new data record arrives at the grid.

IV. MOBILE AGENT TECHNIQUES

An agent is a program, which can act independently and autonomously. A mobile agent is an agent, which can move in various networks under their own control, migrating from one host to another host and interacting with other agents and resources on each. When a mobile agent's task is done, it can return to its home or accept other arrangement. The structures of the mobile agent are different for the different system. However, there are two parts generally speaking, which are MA (Mobile Agent) and MAE (Mobile Agent Environment). MAE realizes the migration of the mobile agents among the hosts using agent transfer protocol and distributes the executing environment and service interface to them [1]. It also controls the security, communication, basic service and so on. MA is above the MAE and it can move into another MAE. MA can communicate with other MA using agent communication language.

V. K-MEAN ALGORITHM

I/p: D= {t1,t2,t3,……..,tn} // set of elements.
k // no. of desired clusters
o/p: K // set of clusters

1) Assign initial values for means m1, m2, m3, ……, mk;
2) Repeat
   a. Assign each item ti to the cluster which has the closest mean;
   b. Calculate new mean for each cluster;

Until convergence criteria is met;

VI. THE NEW FRAMEWORK BASED ON MOBILE AGENT

We propose a new framework based on the mobile agent to gather and cluster the data. The Fig.2 shows that the structure of the framework based on mobile agent. There are three sorts of the mobile agents: the main agent, the subagent and the result agent. The main agent is in the online component and it gathers the data in every gap. It distributes the each dimension data to the right subagent. If we have d-dimensional data, we need about d subagents. However, during clustering the data, if the data is not changed in a certain dimension, the main agent will distribute another dimension data to this subagent. This method can save the space.

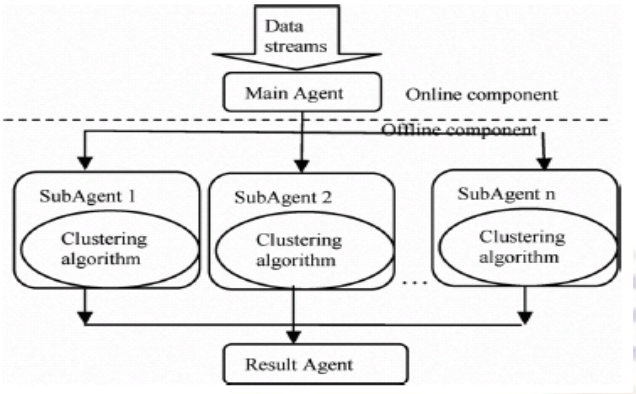


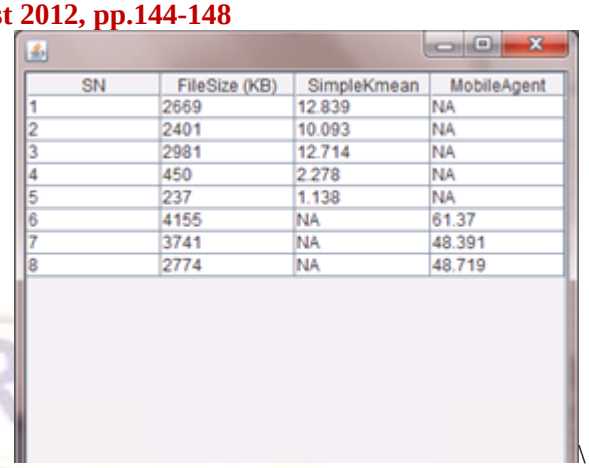
Fig. 1 The structure of the data streams clustering based on mobile agent

Every subagent contains a clustering algorithm model. When a subagent receives the data from the main agent, it detects whether there is change of the data. If there is no change in this dimension, it will send the message to the result agent to use the last time result. Then send the message to the main agent, the main agent will distribute another dimension data to it. If there is change in this dimension at this gap, the subagent creates new grids to cluster the data. This method avoids creating large number grids every time. All subagents execute clustering task parallel.

VII. PROCESS OF THE CLUSTERING ALGORITHM

The initial clustering part is not changed and we improve the clustering part and add the programming of the mobile agent including the main agent, the subagent and the result agent. The clustering algorithm is described as follow:
Step l: Main agent gets the data in a gap time.
Step2: Main agent distributes the data into the subagents. Each subagent gets one dimension data. If the data is d-dimension, we need d subagents.
Step3: The subagent detects the data. If there is no change compared with the last gap, go to step5. Or else go to step4.
Step4: The subagent creates new grids for the new data.
Step5: The subagent sends the result to the result agent and sends the message to the main agent to require the new task.
Step6: All the results of the subagent are collected in the result agent. The result agent computes these results and gets the final result. Go to step 1.

VIII. RESULTS



SN	FileSize (KB)	SimpleKmean	MobileAgent
1	2669	12.839	NA
2	2401	10.093	NA
3	2981	12.714	NA
4	450	2.278	NA
5	237	1.138	NA
6	4155	NA	61.37
7	3741	NA	48.391
8	2774	NA	48.719

Fig. 2 Statistical Analysis of k-mean and mobile agent

For statistical analysis we have consider different input files for k-mean and mobile agent algorithm. For k-mean algorithm input datasets are of only 2attributes, and for mobile agent project we have consider dataset up to 28 attributes and for handling dynamic input entry EXT folder is made in which different files save datasets, and that files are renamed according to time interval.

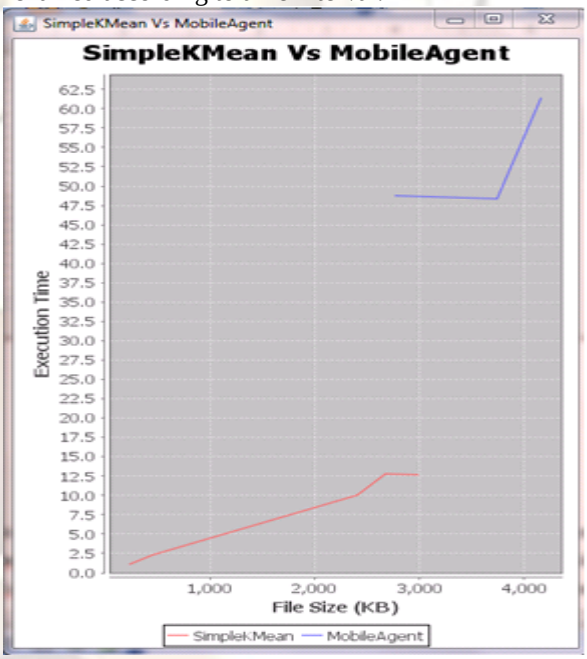


Fig. 3 Graphical results of k-mean and mobile agent

IX. CONCLUSION

The data stream model is relevant to new classes of applications involving massive data sets, such as web click stream analysis and multimedia data analysis. To handle such massive input data in this paper we suggested Density

stream algorithm with mobile agent technique and clustering is carried out by k-mean. We improve D-Stream algorithm using mobile agent in form of main agent, sub agent and result agent, so computing techniques to cluster data in the offline and online component become parallel and fast.

REFERENCES

[1] Zhang Hongyan, Liu Xiyu, " A Data Streams Clustering Algorithm Using DNA Computing Techniques Based on Mobile Agent", IEEE, 2000, Science and Technology Project of Shandong Education Bureau.

[2] Sudipto Guha, Nina Mishrat, Rajeev Motwani, Liadan O'Callaghan, "Clustering Data Streams", Department of Computer Science, Stanford University, 2000.

[3] `Yixin Chen, Li Tu, "Density-Based Clustering for Real-Time Stream Data**",** Washington University in St. Louis St. Louis, USA**.**

[4] WeiserMark The Computer for the Twenty-first Century [J]. Scientific American, 1991,265 (3): 94-100.

[5] Luo Ke ,Wang Lin, " Data Streams Clustering Algorithm Based on Grid and Particle Swarm Optimization", 2009 International Forum on computer Science-Technology and Applications.