IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 6, Ver. VII (Nov – Dec. 2014), PP 01-07
www.iosrjournals.org

Enabling Lazy Learning for Uncertain Data Streams

Suresh.M[1], Dr. MHM. Krishna Prasad[2]
[1]*Dept. of Computer Science and Engineering,*
[2]*Associate Professor & Head Dept. of Information Technology, JNTUK UCEV, Vizianagaram, A.P.-535003, India*

Abstract: *Lazy learning concept is performing the k-nearest neighbor algorithm, Is used to classification and similarly to clustering of k-nearest neighbor algorithm both are based on Euclidean distance based algorithm. Lazy learning is more advantages for complex and dynamic learning on data streams. In this lazy learning process is consumes the high memory and low prediction Efficiency .this process is less support to the data stream applications. Lazy learning stores the trained data and the inductive process is different until a query is appears, In the data stream applications, the data records flow is continuously in huge volume of data and the prediction of class labels are need to be made in the timely manner. In this paper provide the systematic solution to overcome the memory and efficiency. In this paper proposed a indexing techniques it is dynamically maintained the historical or outdated data stream records. In this paper proposed the tree structure i.e. Novel lazy tree simply called Lazy tree or L-tree.it is the height balanced tree or performing the tree traversing techniques to maintain the trained data. These are help to reduce the memory consumption and prediction it also reduces the time complexity. L-tree is continuously absorb the newly coming stream records and discarded the historical. They are dynamically changes occurred in data streams efficiency for prediction. They are experiments on the real world data streams and uncertain data streams. In this paper experiment on the uncertain data streams .Our experimented uncertain data streams and real world data streams are obtained from UCI Repository.*

Key Words: *data streams,clustering,Exemplar , Lazy Learning.*

I. Introduction

Data stream classification is presently increasing the more attention from the data mining. In this we use the clustering techniques, these are playing important role in real world applications. I.e. Data stream clustering plays an important role in spam detection, real-time intrusion [12] detection, and malicious Web page detection. In these real world applications, data stream flow continuously and the ultimate goal is to efficiency for prediction. In this for each incoming Data stream record in a particular time manner. some data stream clustering models[11], such as partition cluster algorithms and another type is Incremental clustering.etc here we perform only the lazy learning and similarly Eager learning is , the training data are materialistically compiled into a summarizing the hypothesis model and then it is completely Discarded. Examples for eager learning types are included such as neural networks, decision trees, and naive Bayes classifiers. Eager learning [8] methods consumes the low memory and more predicting efficiency for answering the queries for data stream applications.

In this Lazy learning [5] have the lazy learners are k-nearest Neighbor (kNN) classifier. It is the non-parametric and instance-based learning methods, here the trained data stream is simply stored in memory and the inductive process is different until a query is given. Lazy learning methods incurred none or low computational costs through training data but much higher costs to answering the queries. Compare to the lazy learning and Eager learning is consume the less memory and efficiency is more. Similarly in lazy learning but it efficiency is more for huge data streams also in case of eager learning less efficiency for huge data streams. Lazy learning is also performs the greater storage chucks, not scale well to the large datasets. In Data stream applications, The data streams come continuously in huge volumes, making it unrealistic to store the training records. In accumulation, stream applications are the time critical. where as class prediction required to be made in a well-timed manner. Lazy learning method is reduced in meeting. These requirements are not have been considered for data stream classification . In this paper Proposed the novel Lazy-tree (L-tree) indexing. A technique that is dynamically maintains the compact high level summaries of outdated stream records. L-tree is extended from M-tree [4].When constructing the L-tree.the exemplar structure is maintain the clusters .these are the maintain the L-tree.the root node contains the distances from the leaf nodes and it contain the exemplar address. It performs the reduce memory consumption. An exemplar is a sphere of certain size generated clusters. The exemplars are organized by height-balanced L-tree it is help to reduce the sub-linear predicting time. L-trees are performs the three key operations. They are searching, deletion, insertion. Searching is performs traverses the L-tree to retrieve the k-nearest exemplars. The insertion operation adds a new stream

record that is nothing but a query value into some exemplars in the L-tree. The deletion performs to removes outdated exemplars from the L-tree.

In this paper is organizing the following contributions:

- Section 2 is organized the analytically investigate lazy learning on data streams.
- Section 3 is organized the related work in the clustering process and Exemplar structure.
- Section 4 is organized the lazy learning, and the L-tree structure to organize exemplars and time manner, and reduce the memory consumption.
- Section 5 is contributing the experiments on real-world data streams and uncertain data streams.

II. Problem Description

In the Enabling lazy learning [3] on data streams, in that mainly focus on the predicting time and reducing memory consumption. Let us consider a data stream or uncertain dataset "S". It consists huge amount of records like { $S_1,S_2,S_3,\cdots\cdots S_n$ }, and the time stamps are {$T_1,T_2,...,....,T_n$ },where "n" is the depends on the no of clusters given by the user or automatically generate the fixed no of clusters. And class labels are the { $C_1,C_2,...,....,C_i$ } and the query value or incoming record is "X". Here to avoiding the loss of generality of data streams consider the K-NN as the lazy learning algorithm. It is one of the clustering algorithms. In data stream environment, estimating these two probabilities is very challenging because of the predicting time and memory consumption. In data streams, it is impractical to maintain all the (n − 1) records { $S_1,S_2,S_3,\ldots\ldots S_{n-1}$ } for Estimation. Thus, a memory-efficient algorithm needs to be designed for this purpose required a linear scan of all the (n − 1) records, corresponding to an O(n) time complexity, Linear scan is not acceptable for data stream applications. so, an Efficient search algorithm is needed to reduce the predicting and time complexity O(log(n)).But in this paper searching algorithm is Linear Scan algorithm taken the time complexity O(N).

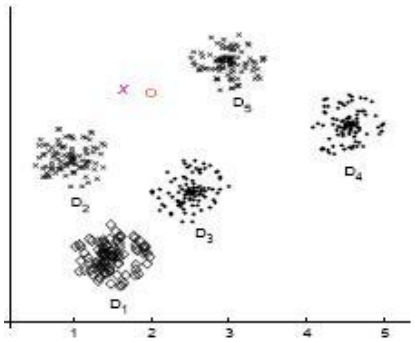

Figure.1: An Illustration of Example1

Example 1: Consider a uncertain data stream S having three classes { C_1,C_2, C_3,C_4 }, and stream records are { $S_1,S_2,S_3,\cdots\cdots,S_n$ }, For simplicity consider data stream has n=500 records, let us assume that these "n" records are distributed uniformly in "m" clusters. Let us take the clusters "m=5"then {D_1, D2 $\cdots$, D_5 }are the clusters S belongs Di where (1 <= i<= 5)and share the same class label, where class C1 is denoted by the symbol "·", class C2 is denoted by "×", and class C3 is denoted by "".The incoming record or query value "X" is given. It is shown red circle in Fig.1.sible. If the original k-NN method is chosen as the solution, then maintain the entire 500 data stream records for prediction and these are very demanding for memory consumption. Moreover, even if memory consumption is not an issue, then straight forward approach for comparing query value or incoming record "X" would take 500 comparisons, which is inefficient in terms of predicting time**.**

III. Related Work

A cluster is a group of data objects which are similar characteristics to each other within cluster and dissimilar characteristics to other data objects belonging to other cluster [1].Data clustering is a young scientific discipline under vigorous development. There are large number of research papers scattered in many conference proceedings and periodicals, mostly in the fields of data mining, statistics, machine learning, spatial database, biology, marketing, and so on, with different emphases and different techniques. Owing to large amounts of data which is collected from different data resources, cluster analysis has recently become highly active topic in data mining research.

Exemplar Structure:

Instead of maintaining raw data stream records, we perform the cluster them for that clustering we use the k-means algorithm. These clusters are the exemplars to reduce the memory consumption. An exemplar is a sphere generalizing one or multiple records. Exemplars are different and more complex in that they summarize labeled data.

Definition : An exemplar M for a set of data stream records { $S_1, S_2, S_3, \cdots \cdots S_n$ }, arriving at time stamps {$T_1, T_2, ..., ..., T_n$ }, it is a dimensional vector.

$$M = (X, R, C, N, T)$$

where X is a d-dimensional vector that representing the center or centroid, R is the sample variance of all records in data stream, and also it is radius of exemplar set M,N is the total number of records in M, and T represents the time stamps when M was last updated.

Exemplars have several merits for lazy learning on data streams:

• Exemplars are help to summarize huge volumes of data stream into compact structures which are well fit into memory.

• Exemplars are representing the historical data.

• Exemplars can be easily updated.

If a new query or incoming record "X" absorbed into M then the exemplar radius and center can be conveniently updated. The class label C can be updated using Equation.

$C = [nP(C_1|M)/(n + 1).....(nP(C_p|M) + 1)/(n+1)\ldots , \quad nP(C_i|M)/(n + 1)]$

C_p : it is the query class label

K-Means clustering algorithm:

K-means is one of the simplest partitioned clustering algorithm that solve the well known clustering problem. The process follows easy way to separation a given dataStream into a definite number of clusters fixed priori. The main proposal is to define k-centroids, one for each cluster. The next step is to take data object belonging to a given dataset and assign it to the nearest centroid. When no new data object is pending, the first step is completed and an early grouping is done. At this point we need to re-calculated k new centroids as centers of the clusters formed from the first step. After we have these k new centroids, a new binding has to be done between the same data stream points and the nearest new centroid. This method is repeated for all the data objects. The Final Outcome of this loop we may observe that the k centroids modify their location step by step until no more changes are done.

Algorithm 1: Create and maintain exemplars.

Input : initialize clusters m, data stream S, radius threshold g, maximum exemplar threshold N.

Output: A set of exemplars E.

```
Read the data stream S and convert  records into vector ;
  //Perform the clustering algorithm k-means
Em=K-Means(S, m);
   //The Exemplar process starts;
e =Search (E, x);  // e is exemplar  number
While X! = S do
if distance(e, X) > g Then
  Enew = Create Exemplar (E, X);
If |E| == N then    // reach size threshold
   E= Insert (e, X);
Else
  E =update (e, X);  // update rules
Output E;
```

The above Algorithm Explains the procedure of construct and update the data set E of Exemplars on data stream S .Initially read the data from data stream S, and performing the k-means algorithm and generate the "m " clusters, and exemplar set E. and giving the query stream record or incoming data stream record X, if it is not in the data stream then it performs the updating the exemplar in that process get the Exemplar number and distance to its nearest Exemplar retrieved from E. If the distance between e and x is larger than the given threshold g, a new Exemplar E_{new} will be created and inserted into E. Otherwise, query X will be absorbed into e using the updating rules. Example1 is to illustrate the procedure of abbreviation the 500 data stream records given in Figure.1.

Example 2: The 500 stream records can be summarized into five exemplars $\{M_1,M_2,...M_5\}$ as shown in Figure.2.

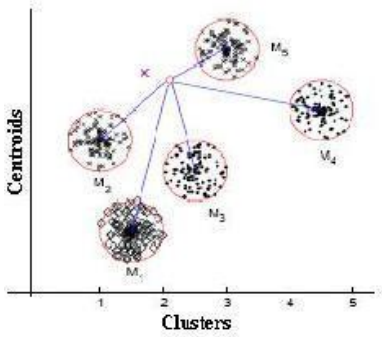

Figure2: An Illustration of Example2

The Exemplars consumes the memory space.

IV. Lazy Learning

L-Tree Structure:

L-Trees are extending from M-Trees. M-Trees [4] index objects in metric spaces such as video, image, voice, numerical data, and text. L-trees index exemplars that are spherical spatial objects. This is different from other spatial indexing structures such as R*-Trees and R-Trees that indexing rectangular spatial objects. An L-Tree Structure mainly consists of two different types of nodes: rooting nodes and leaf nodes. The root node can be considered as a special routing node that has no parent. A leaf node contains a batch of exemplars represented in the form of, (pointer, distance), pointer is referenced the memory location of an Exemplar and distance indicates the distance between the exemplar and its root node or parent node. The L-Tree structure contains the following entries in the form of, (center, radius, leaf node, distance), leaf node is a pointer that references its child node, and distance denotes the distance of the entry to its root node or parent node.

L-Trees have the following properties:

• Routing node: A routing node contains the number of entries it is the root.

• Root node: The root node is contains the two entries it is centroid and distance from the leaf node.

• Leaf node: A Leaf node is also similar to the root node process, and all leaf nodes are at the same level. The leaf nodes are maintained the exemplars and its distances from the leaf nodes. The tree structure performs the search and Insert Operations.

Search:

In this search algorithm we used the linear scan search algorithm, search operation is invokes the Query is present in the data stream and calculate the distance from Exemplar, and similarly we also take the different search algorithms such as binary search, branch and bound search algorithms.etc. Search algorithm is first traverses the L-tree to find its k nearest exemplars in leaf nodes. Then it also calculates the class label for Query is retrieved from exemplars. The exemplars in a Height balanced L-Tree can significantly reduce the search time cost from O(N) to O(log(N)), Here "m" is the total number of exemplars in the L-tree. Search method is further improved by using the branch-and-bound technique.

Example3: Let us assume the exemplar e and find the distance from the exemplar to query or incoming record is X. it is denoted by the d (e , x).

In tree structure perform breath-first search. The main purpose of the algorithm is to retrieve the cluster results using of comparison by making full use of the bound b. Consider the incoming record or Query x and The search algorithm is initially pushes the entries e1 and e2 in query x. Then, it calculates the distance d(x, e1) and d(x, e2) in this if the smaller distance value of e2 is small then it perform the tree traversing along e2. Next, similarly this process continuously compares to Query X with leaf node entries. Compared to the linear scan that requires "m" comparisons.

Insertion:

Inserting operations are taken as to absorb the new data stream records into L-Tree, So that they can quickly adapt to new patterns in data streams. Algorithm2 list gives the procedure of the inserting operation. The query record or incoming record X, and search algorithm is invokes to find its Nearest Leaf node O. When the query X is insert in the retrieved Leaf node O. and it is performs the search results are taken and inserted in

that leaf node. Similarly in this Node Splitting is the most critical step in the inserting operation. M-trees, the basic principle in splitting[10] the node is the split the Leaf nodes. In that the minimized spatial expansion is consider.

Algorithm 2: Insertion Algorithm
Input: Query record X,L-tree T, Exemplars M.
Output: Updated L-tree T^1.

```
O=SearchLeafnode(X, T); // O is the root nodes
if d(e,X)< e.g then   //e.g is exemplar threshold radius
    e=x ;    //insert the query in that exemplar
    T1=adjustTree(T, e);
else
     Enew=CreateEntry(X);
if O.Entries() < m then
    O=Enew ;
    T1=adjustTree(T,O);
else
     Rootnode o=Split(O,Enew) ;
     T1=adjustTree(T,O) ;
Output T1;
```

The above Inserting algorithm is reading the Query X and it is search in the tree and retrieve the node information and then calculating the distances from nodes contain the exemplar centers to Query .if the smaller distance is taken and pass to the right leaf node otherwise it pass through the left leaf node, similarly this process is continuously performs until they reach the threshold value. This process is shown in Figure.3.

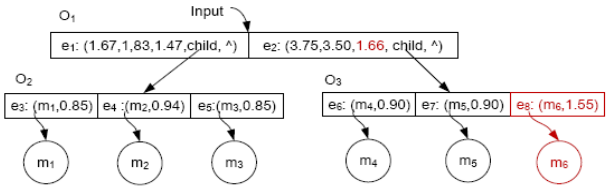

Figure3. Illustration of the inserted Query x after L-tree structure.

V. Experimental Works

All the experiments were done on 2.50GHz Intel Core i3 machine with 4GB main memory running Window 7 operating system. We implemented the GUI Tool using Java. Below experiment results shows that Enabling Lazy Learning for Uncertain data streams using L-Tree indexing technique algorithm Tree traverse algorithm gives less consume the memory usage and prediction time manner.

Results for before Inserting the Query:

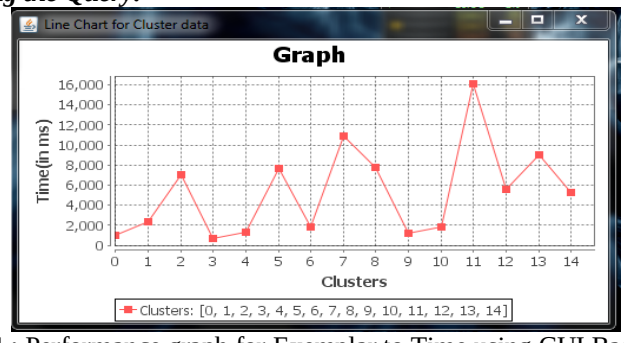

Figure 4 : Performance graph for Exemplar to Time using GUI Based Tool**.**

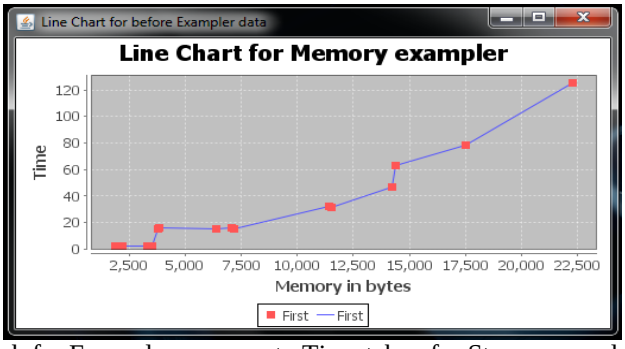

Figure 5: Performance graph for Exemplar memory to Time taken for Stream records using GUI Based Tool**.**

Results for After Inserting the Query:

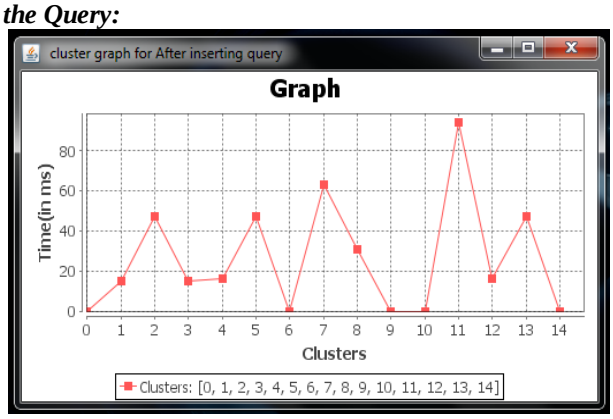

Figure 6 : Performance graph for Exemplars to Time After inserting Query.

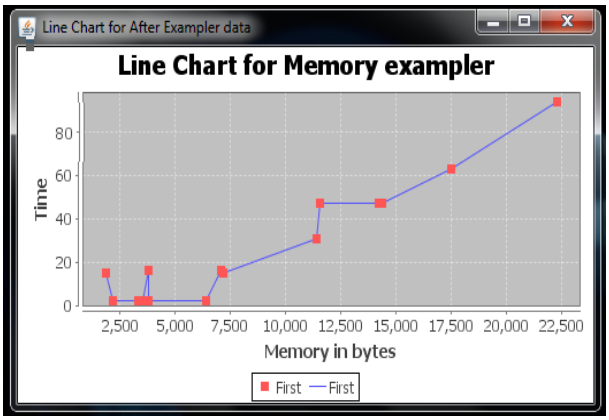

Figure 7: Performance graph for Exemplar memory to Time taken after inserting the Query in data Stream records.

The above graph results are getting by experimented the GUI Tool on Soya bean-Large dataset [15] that contains a set of board configurations possible at the end of game.It includes 668 instances and 36 attributes, each stream record corresponding to the Soya bean data.
The different real world data sets, synthetic data sets are obtained from data stream mining [14] repository and UCI data stream [13]repository.

VI. Conclusion And Future Work

In this paper presents the k-means algorithm for generating clusters and maintains the clusters in tree structure. And reduce the memory and time cost of performance. In this paper is concentrate on the uncertain data streams. Here x-means algorithm is also used instead of k-means algorithm. X-means[2] is the extension of k-means algorithm.

Future work will analyze that we focus on different tree algorithms and searching algorithms. Similarly using the different clustering algorithms[11] or classification techniques[6] are used to get different results. These tests may take efficiency and reduce the time and get the Exemplar process performance and compare its

performance to various clustering algorithms on various types of data. Some prominent topics for future work. Like complex class-aware summarization [9] techniques are used to generate the Exemplars. Another one is the Distance computation for Nearest Neighbours [7] and different weighting schemes are also is investigated.

References

[1] A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice-Hall, 1988
[2] D. Pelleg and A. Moore. X-means: Extending k-meanswith efficient estimation of the number of clusters.ICML Conf., pages 727-734, 2000
[3] Peng Zhang,Xingquan Zhu,Byron.Enabling Fast Lazy Learning for Data Streams,Dec 2011
[4] P. Ciaccia, M. Patella, and P. Zezula. M-tree Efficient access method for similarity search in metric spaces. In Proc. of VLDB 1997.
[5] D. Aha. Lazy learning. Artif. Intell. 1997.
[6] B. Dasarathy. Nearest neighbor (nn) norms: Nn pattern classification techniques.IEEE Computer Society Press, 1991.
[7] L.Clarkson,Kenneth.Nearest-Neighbor Searching and Metric Space Dimensions,oct 2005
[8] P. Domingos, and G. Hulten, P. Zezula. Mining high-speed data streams. In Proc. of KDD 2000.
[9] B. Gao and M. Ester. Turning clusters into patterns: Rectangle-based discriminative data description. In Proc. of ICDM 2006.
[10] Lars Bergstrom, Mike Rainey, John Reppy, Adam Shaw; Lazy Tree Splitting.
[11] I. K.Ravichandra Rao. Data Mining and Clustering Techniques.DRTC 2003
[12] Paul Dokas, Levent Ertoz, Vipin Kumar, Aleksandar Lazarevic, Jaideep Srivastava, Pang-Nig Tan,Data Mining for Network Intrusion Detection.USA
[13] A. Asuncion and D. Newman. UCI Machine Learning Repository. Irvine, CA,2007.
[14] X. Zhu. Stream data mining repository.Available online: http://cse.fau.edu/xqzhu/stream.html, 2010.
[15] http://archive.ics.uci.edu/ml/datasets/Soyabean-large.num