ISSN 2229-5518

A New Indexing Data Structure for Classifying Stream Data

D. Jeevan Kumar
M.Tech Student, Department of CSE
KMM Institute of Science and Technology
Tirupati, India
Email-jeevanturt@gmail.com

C. Sudarsana Reddy
Assistant Professor, Department of CSE
KMM Institute of Science and Technology,
Tirupati, India
Email-cheruku1sudarsana2reddy3@gmail.com

Abstract:

Continuous data is common in areas. Note that very large stream data mining is very important and it becomes one of the most important data mining. For state-of-the-art stream data mining algorithms require in many cases exactly one and only one pass in order to process full training data sets. For managing very large stream data powerful and more sophisticated and state-of-the-art indexing data techniques are needed to process, search, and query the huge stream data. Ensemble ETREE technique is one of the best multidimensional indexing data structure for storing classification rules of ensemble classifiers. Ensemble data indexing tree structure reduces time complexity of classifying a new stream data record from linear time complexity to logarithmic time complexity. A new indexing technique is proposed for tree node splitting in the case of ensemble indexing tree construction. New node splitting method is convenient, easy to use, understand and process all the required results during the time of ensemble tree construction process for the important operations of tree insertion, modification and deletion. Here main goal of stream data mining algorithms is to design and develop algorithms only for processing data in pass only.

1. INTRODUCTION

Stream Data Mining is defined as the process of extracting knowledge data from continuous, potentially very fast groups of data records. Actually the data stream consists of an ordered sequence collection of instances. Main requirement of applications of data stream mining is performed only once by using limited computing resources and very large storage capabilities. Some examples of data streams are:

1. Traffic stream data
2. Satellite stream data
3. Computer network traffic data details
4. Bank ATM transactions details
5. Electronic-Card transactions
6. Telephone data
7. War simulation data
8. Weather simulation data
9. Web search engine data
10. Very large net-work flow of data
11. Internet browsing details
12. Electronic-commerce data
13. Internet document searching
    Very large business data transactions

Data stream mining can be considered a subfield of data mining, machine learning, and knowledge discovery. In very large stream data mining, many data mining algorithms need many passes to process, compute and mine training data sets. But many real applications require only one pass to mine very large volumes of data streams. Effective, correct, productive, and very lengthy stream data processing is fundamentally a new, most popular, and latest stream data mining research area. The well known information system called **data stream** is a state-of-the-art information system proposed and used for the purpose of monitoring a group of potentially very large data streams.

In many applications of stream data the fundamental and most important goal is to find the class label of new instance in the continuous or dynamic data stream given some knowledge about the class membership of earlier instances in the dynamic data stream. Machine learning techniques can be used to learn this prediction task from labeled examples in an automated way. In many applications, especially operating within non-stationary environments, the distribution underlying the instances or the rules underlying their labeling may change over time, i.e. the goal of the prediction, the class to be predicted or the target value to be predicted, may change over time continuously. These changes in data details are referred to as **concept drift.**

Data streaming is the transfer of data at a steady and very high-speed data transfer rates that are potentially sufficient to support such applications as high-definition television or the continuous backup copying to a storage medium of the data flow within a computer. In many real life applications more sophisticated and also state-of-the-art multidimensional indexing techniques are needed to search, process, query the large streams of data. Hashing technique indices and B^+ Trees are index structures that can handle only one dimensional data. In order to manage multidimensional data one must use special indexing data techniques.

Spatial data is categorized as one type of multidimensional data. In order to handle spatial data, a lot of possible and helpful indexing tree data structures are:

1. T-trees, K-d Trees,
2. Quad trees, Oct Trees
3. R*Trees, RTREEs, R+ Trees, and many other variants of R-tree related trees
4. M-Trees, and
5. Ensemble TREEs and so on.

Data stream classification represents one of the most important tasks in data stream mining which has been popularly used in real-time intrusion detection, spam filtering, and nasty website monitoring [1]. Stream data mining is a more challenging task in many real life cases. There exist many of the difficulties by means of stream data querying but frequently requires less "precision", e.g., no join, grouping, sorting patterns are hidden and more general than querying it may require exploratory analysis [2]. There exist many algorithms for constructing indexing decision trees structures among the most useful, well known and widely used of all machine learning methods. Among decision tree algorithms J. Ross Quinlan's ID3 and its successor C4.5 are most likely the most popular and most renowned in the literature of machine learning community [3]. Geographic Information Systems provide solutions to a wide panel of problems [4]

2. RELATED WORK

Some of the indexing tree data structure names are summarized here as- K-d Trees are two dimensional indexing structures and K-d-B tree is a multidimensional indexing data structure and it is an extension to the K-d Tree. Also note that quad tree are two dimensional indexing structures but an Oct-tree is an example for 3-dimensional indexing structure. RTREE and many of its variants are very useful and powerful indexing structures for indexing many real objects such as points, circles, line segments, rectangles, squares and polygons. An RTREE indexing structure is a balanced tree structure with the indexed objects stored in leaf nodes. Each tree node is directly associated with one bounding rectangle box. The bounding box of a terminated node is the smallest rectangle parallel to the coordinate axes that contain the bounding boxes of its child nodes. Each leaf node stores the already indexed objects and each internal node stores the bounding boxes of the child nodes along with the pointers to the child nodes.

Most important operations of RTREE are:
1) Search, 2) Insert, and 3) Deletion

1. Search:

Overlap may occur between or among the bounding boxes associated with sibling nodes. For searching a specific point all the child nodes of a bounding box containing the point must be searched. For searching and querying multiple paths may have to be searched.

2. Insert a node into dynamic ensemble tree

In order to insert a new object into the ensemble tree, one must find the leaf node that has the sufficient place for entering new record. Note that finding an exact leaf node for insertion is costly in terms of comparisons. Also note that sometimes many indexing tree data structure traversals are required. If the rectangle bounding box of the new object and the rectangle bounding box of any children of ensemble ETREE are same then that common bounding rectangle box will be selected automatically as a heuristic. If the selected leaf node of the index tree for insertion is full then the leaf node must be split into two different nodes and the corresponding bounding rectangle are adjusted accordingly.

Stream data length is potentially very large:

Internet data, video conference data, satellite data, telecommunication data, network data, bank transaction data and so on are some of the examples for stream data. Very Large volumes of data streams are generated in Internet traffic, communication networks, retail industry, current data, grid data, remote sensors, and electric power grids, weather data and in scientific and engineering applications. It is very difficult to store large volumes of stream data and it is also very difficult to process the stream data more than once. One must need accurate, timely, efficient, fast and effective state-of- the-art indexing techniques to develop single-scan, on-line, multi level, multidimensional stream processing techniques. Stream data sizes are increasing from terabytes to Peta bytes rapidly. Many stream data analysis techniques are invented. Different types of stream data mining applications are:

1. Spatial and multidimensional analysis of stream data
2. Large stream data cube based modeling techniques
3. Large stream data frequent pattern mining
4. Classification of stream data
5. Clustering of stream data

For accurate, efficient, effective and fast processing of stream data new data structures, analysis tools, processing techniques, models, tools and algorithms are needed. There must be tradeoff between accuracy and storage capacity because infinite storage is not available. We need efficient, robust, scalable, effective algorithms in terms of space and time complexity. The best time complexity for stream data processing is logarithmic. That is O(log n).

Random sampling is a technique where a set of samples are taken for mining instead of taking entire stream data. Reservoir sampling technique can be used to select an unbiased random sample of S elements without replacement. Here S number of sample sets is selected for mining. Sliding window model is another technique useful for stream data mining. This method uses only the recent data. Each time the number of tuples equal to the size of the window is processed, next time a new window is processed. Sliding window model is particularly useful in applications such as stocks, sensors, video conference, cell phone stream data etc.

Data streams are potentially infinite and it is impossible to process each element more than once. There may exist multiple parallel data streams in a stream data management system. We can also pass queries to stream data. Data stream query may be either one-time or continuous.

Data streams are dynamic, infinite, fast, continuous, multi-dimensional, combinations of different data patterns and different distributions of complexities. In order to find critical changes, unusual patterns and interesting patterns, multidimensional stream data analysis on aggregate measures required.

Sometimes there is a need to apply frequent-pattern mining on data streams. All the previous data mining algorithms must be modified in order to handle stream data. Stream data management is really a challenging task. Lossy counting algorithm is an efficient algorithm to find frequent items.

Main characteristic of stream data is that they are time-varying. Most important point that must be considered in the case of stream data is that concept drift. In the stream data, data distributions change continuously. The change in the data distributions causes changes in the target classification model. In other words classification model is directly affects to the changes in data distributions in the stream data. Different types of proposed methods for stream data classification are:

1. Hoeffding tree algorithm
2. Very fast decision tree (VEDT)
3. Concept adapting very fast decision tree (CVFDT)
4. Classifier ensemble approach

Ensemble classification approach is particularly suitable for stream data classification. An ensemble is a group of classifiers such as C4.5, ID3, Naïve Bayes, Bayesian belief networks etc. Nowadays stream data sizes are very large. Sometimes, it is not possible to process each record more than one stream data mining techniques must be fast, effective, efficient and accurate. When the stream data size is very large computational cost increases exponentially. The main characteristic of any stream data mining algorithm is that it must process the complete data in one pass only. Large volumes of data stream processing DM algorithms must be enriched with many latest trends and techniques. Most important stream data mining techniques are categorized as follows:

1. Stream data classification
2. Stream data clustering
3. Frequent stream data pattern mining
4. Indexing data streams
5. Dimensionality reduction in stream data
6. Stream data mining in distributed environment
7. Stream data mining in sensor networks
8. Join operations in stream data
9. Change detections in stream data
10. Data stream cube analysis

Indexing data streams

Data stream indexing is very useful for certain operations such as aggregate queries, transaction data analysis, whether simulation data analysis, war simulation data analysis, etc. Indexing usage is inevitable in analysis, computing, processing, querying and various other mining techniques of large data streams. Indexing usage is inevitable for many data-centric applications such as trend analysis, network traffic management, web-click streams, finding location of a specific pervasive device, and intrusion detection and so on. Indexing usage is both time and space proficient and also provides a high quality of answers to user queries. Indexing a dynamic stream data is very hard. A dynamic indexing structure provides high query efficiency. A good index is for all time associated with parameters such as accuracy, speed, space and time.

Indexing data structures that are used for stream data mining are RTREE, R+ tree, R*-tree. Corresponding to new stream data records, the new features must be computed and inserted into the dynamic index structure and at the similar time old features must be deleted from the index organization spatial data is indexed using R*-tree indexing structure. There is a need of efficient indexing architecture for both time and space to extract features of stream data and then insert these indexing features for improving query performance. Also there is a need of low price maintenance measures for the index structure. Indexing technique definitely decreases streams record processing time and minimizes the space required for incremental computation.

Stream data classification is one of the majority significant data mining techniques. Ensemble learning is one of the state-of-the-art data stream classification technique. Ensemble stream data learning method manages large volumes of stream data and controls concept drifting. All the existing data stream management methods mainly concentrates only on constructing accurate ensemble models from the stream data. Existing stream data learning techniques can be useful to only a narrow set of real world applications because forecast cost of stream data learning is very high. In many real time applications data streams arrive at a speed of Giga Bytes (GB) or even Tera Bytes (TB) of time per second but classification or clustering job have to be completed within a restricted time unit. Must indexing structures are planned to speed up the stream data classification and also each indexing structure have its own advantages and disadvantages.

Accessible stream data classification techniques use a linear scan technique to process all the base classifiers. This linear scan technique is used to process all the base classifiers. That is, linear scan is expensive for many actual time applications. Hence, a new and low cost ensemble stream data classifications technique is desirable for many modern real time applications that are based on ensemble learning.

A new indexing data structure called ETREE is proposed to reduce the time complexity from linear to log. ETREEs are automatically and dynamically updatable in order to reflect stream data changes and pattern changes. Early ensemble streams mainly concentrated on construction only accurate ensemble stream data models. The prediction cost of ensemble learning increases linearly as the number of classifiers in the ensemble learning increases. This linear time difficulty is not suitable for numerous real life applications.

Classification is one of the most essential data mining techniques. Ensemble learning is one of the most important data stream classification techniques in dynamic data stream mining. Dynamic stream data mining has many real time applications such as ATM dealings management, finding incorrect documents in website organization, Spam filtering etc. Problems with stream data are:

1. Quantity of stream data is potentially very high
2. Changes in the features and patterns of stream data occurs so frequently

Many ensemble based stream data models have been proposed in the literature of stream data management. Ensemble classification, ensemble clustering, ensemble association, ensemble incremental classification and clustering, ensemble fuzzy classification and clustering, are proposed based ensemble management modelling techniques. Divide-and-conquer is the most popular ensemble stream data management technique to handle large volumes of stream data with concept drifting. In ensemble stream data classification stream data is divided into a predefined number of partitions and a new classifier is constructed for each partition and then these base classifiers are combined in different ways for predicting the class label of a new incoming stream record.

Advantages of ensemble learning design are

1. Ability to handle big volumes of stream data
2. High scalability in supervision huge streams of data
3. Capability to handle changes in patterns and trends that occur with dynamism
4. Increased accuracy of classification and clustering
5. Reduced **errors** in classification
6. Simple to parallelize stream data processing and forecast
7. Reduced search times for obtaining accurate results in stream data processing

All the previous ensemble learning stream data modeling techniques are mostly determined in constructing correct ensemble models and much significance is not given to competence in stream data processing. previous methods are sufficient when total number of base classifiers are less than 30 and forecast efficiency is not significant. Previous ensemble learning models are not suitable for many real world applications, particularly to model dynamic changes in trends and patterns in flow data. Time complexity of ensemble learning is linear, that is O(n), which is not suitable for many genuine time applications. Many state-of-the-art ensemble learning models have sub-linear time complexity.

Ensemble approaches are stream data association tools mainly cooperative methods for management concept drift. Ensemble approach is an up to date dynamic stream data record classification approach. It rejects least accurate classifiers and updates all the left over classifiers. It is incrementally updatable. Ensemble classification results are more precise than every single classifier approach.

Features of Stream Data

Stream data is dynamic and main distinctiveness of stream data are:

1. Stream data length or size is potentially infinite
2. Alteration in the stream data occur instantaneously
3. Very quick response times are mandatory
4. Very complicated to store completely and also very complex to process stream data
5. Number of scans of the same data is not possible
6. Data distributions within the flow data alteration with dynamism and speedy
7. Combined actions of stream data changes dynamically
8. Multi dimensional learning and modeling is needed
9. Sometimes manifold modeling techniques are desirable
10. High-tech structures, tools, methods indexing technique and other processing ideas are required.

3. PROBLEM DEFINITION

Every tuple is represented by n attributes and (n + 1)th attribute is the class label attribute given at the end of row. To make the task easier only two class problems are considered. Imagine that ensemble consists of k number of classifiers (c_1, c_2, c_3, …. , c_k). When numerous base classifiers are united the resulting one is called ensemble classifier. Main aspire of at hand study is to decrease time complexity of flow data record prediction from linear to logarithmic. Every classifier consists of r rules and here there are k classifiers in the ensemble. Completely k. multiplied by r number of rules are stored in the multidimensional indexing tree data structure. All k x .r rules are transformed into k x r spatial objects and then these are stored in the multidimensional indexing tree structure.

Dynamic stream records are represent as S={(x_i, y_i)}, where each x_i represents a set of k attributes and y_i is a class label with two values-yes or no. For simplicity a two class problem is considered. Stream data is divided into n partitions and 'n' decision tree base classifiers c_1, c_2, c_3…….c_n are constructed all 'n' base classifiers are combined into a single ensemble classifier E. Each base classifier c_i consists of x decision rules represented by if-then clauses. Total number of decision rules are n*x. All these n*x. decision rules are represented in spatial database as n*x spatial items. Major goal is to build best assembly model to forecast inward stream record exactly within logarithm time complexity O(log n). To reach this goal every base decision tree classifier C_i is changed into a group of spatial objects. This group of spatial objects is called spatial database. Entire ensemble form E is changed to a spatial database containing all spatial objects. By means of this idea given single problem is changed to classifying each incoming dynamic stream record r by systematically searching over the spatial database (SD).

4. ALGORITHM

ALGORITHM : Inserting a new node in the ETREEREF

INPUT OF ALGORITHM :
1. ETREEREF T
2. Classifier C consisting of set of rules
3. min , minimum number of elements in the node
4. MAX, maximum number of elements in the node

OUTPUT OF ALGORITHM : modified ETREEREF T^{prime}

1. PTR←T .tree.root
2. foreach of the decision rule R∈C do
3. L ← Call searchLeaf(R, PTR)
4. If(L.size < min) then
5. $L \leftarrow L \cup R$
6. T^{prime} ← UpdateParentnode (L)
7. Else
8. {P_L, P_R} ← ***ModifiedsplitNode(L, R)***
9. T^{prime} ← adjustTree(T, P_L, P_R)
10. end-of-if
11. end-of-for
12. insert the selected classifier in the table structure
13. Output the tree, T^{prime} , after inserting the new classifier, C of rules

Explanation of Modified ETREE insertion operation

Node split occur during the insertion of dynamic stream data records into the ensemble tree indexing structure. Planned node splitting way reduces the time complexity of node split task from O(n^2) to O(n log n). New method is easy and simple to split the nodes. Always minimum and maximum node sizes are specified clearly at the beginning.

Advantages new indexing tree

1. Splitting process of node in the ETREE is very fast.
2. It is simple, easy, effective and efficient heuristic technique for node splitting.
3. Entries in the node are sorted first and then divided into two parts called left node and right node.
4. Proposed method is reasonably very good with asymptotic time complexity.

Whenever new trends and patterns are found during dynamic stream data classification, the ETREE insertion algorithm insert new base classifiers into the ensemble model and this model is automatically converted into the spatial space data object and inserted into the ETREE indexing structure. Whenever a new classifier, C, is created, a new entry associated with the newly created classifier is inserted into the E-table structure and similarly all the decision rules, R, belonging to the newly created classifier are inserted into the ETREE structure one after the other, and linked all the rules jointly by the respective pointer entries.

Inserting decision rules of a new classifier into ETREE is very like to the insertion operation of RTREE indexing structure starting from the root hunt proceeds downwards in order to find a leaf node that covers the rules of the new classifier. Once a leaf node is found, it is checked to locate whether space for insertion is accessible or not. If the leaf node contains less than the highest number of entries, MAX, then new set of rules are inserted into the leaf node. Parent node is updated suitably. After searching, if it is found that there is no space for new insertion, then the current leaf node is split into two nodes and then new set of rules are inserted. Node splitting in ETREE is a difficult step.

A decision classifier C4.5 is used to produce decision rules from the lively stream data. All decision rules are "hard" decision rules only. That is decision rules are not fuzzy. Different measures that are used for online query assessment are:

1. Time cost- computational cost of ETREEs is much lesser than the computational cost of traditional ensemble models because ETREEs are height balanced indexing tree structures that are used to index all classifiers in the ensemble.

2. Memory cost- Through ETREEs consumes larger memory during stream data record classification, the memory volume is in affordable range only.

3. Accuracy- prediction accuracy of ETREEs is high and it is equal to the correctness of original ensemble models.

ETREE learning by ensemble technique

Architecture of ensemble models on dynamic data streams using ETREE indexing structure is shown in the fig 4. Training module maintains and controls all the insertion and deletion operations of ETREEs. Training module is in charge to monitor ETREE operations for constant updating of ETREE. Prediction module is accountable to predict the class label of newly arriving dynamic stream record x by using synchronized copy of ETREE received from training module. ETREE search operation is applied to make online predictions. Main assumption is that stream is coming with un-labeled data. Initially the buffer in the training data module is filled with incoming stream records. Records in the buffer are labeled by human experts or intelligent labeling machines. This labeling process is very time-consuming, precious and time consuming and only a small fraction of received records at regular intervals regularly in order to offer uniform labeling. As soon as buffer is full, mechanically a new classifier is constructed and then it is inserted in the ensemble. New classifiers of ETREE are constructed regularly. Once the highest capacity of the ETREE is reached automatically old or out-of-date or un-useful classifiers are deleted from the ETREE by executing ETREE deletion operation. Updated and latest ETREE will be synchronized and ETREE copy is passed to the prediction module.

5. EXPERIMENTAL DETAILS

At the beginning only a set of decision tree classifier models are created. Every classifier represents a group of decision rules that can be represented as a set of if-then rules. These rules are very easy to interpret. All these decision rules are transformed into spatial database objects by using a suitable procedure. Spatial data objects are stored in the ensemble tree indexing data structure. Main operations of ensemble tree data structure are – insert, search, deletion and classification operations. Previously these trees are used only for searching. In this paper, ensemble tree data structures are used for classifying the newly arriving dynamic stream record.

RTREE is the best indexing data structure for indexing spatial database objects. In the case of RTREE Rectangles are used to represent rules of decision tree classifiers. A group of decision tree classifiers models are generated and then these classifiers models are united for ensemble classification. Ensemble classification is mainly useful for dynamic flow data management.

A mapping function is defined to change all classifier rules into equivalent spatial database objects. When the numbers of classifiers in stream data classification are very small then normal indexing structure is sufficient. When the set of classifiers is very big we have to use a new state of the art indexing structure for stream data organization. Ensemble tree indexing structure is one such state-of-the-art dynamic indexing tree data structure.

Ensemble tree indexing structure is an extension of the RTREE like structure. Balanced search trees B-trees are considered as only one dimensional indexing data structure. They are suitable for indexing spatial database objects. B-trees are extended for constructing state-of-the-art indexing data structures. R-tree variants are actually extensions of original B-trees. R-trees are specific for some applications in the real time situations.

TABLE 1 Input data to Multi Dimensional Indexing Tree Structure

X1	Y1	X2	Y2	Label
30	100	20	120	1
30	100	40	130	1
50	85	60	125	1
50	20	90	140	1
25	65	75	95	1
70	60	100	130	1
65	90	95	120	1
15	20	30	40	1
35	25	55	60	1
110	40	130	80	1
110	10	120	70	1
115	25	125	45	1
80	70	100	80	1
40	20	50	30	1
50	70	60	80	1
60	100	90	130	1
60	70	90	100	1

TABLE-2 Output

[25.0,65.0,100.0,130.0, 15.0,20.0,90.0,140.0, 50.0,10.0,130.0,130.0]

[25.0,65.0,75.0,130.0, 50.0,70.0,90.0,100.0, 65.0,70.0,100.0,120.0]

[25.0,65.0,75.0,95.0,1.0, 30.0,100.0,20.0,120.0,1.0, 30.0,100.0,40.0,130.0,1.0]

[50.0,70.0,60.0,80.0,1.0, 60.0,70.0,90.0,100.0,1.0]

[65.0,90.0,95.0,120.0,1.0, 80.0,70.0,100.0,80.0,1.0]

[15.0,20.0,55.0,60.0, 50.0,20.0,90.0,140.0]

[15.0,20.0,30.0,40.0,1.0, 35.0,25.0,55.0,60.0,1.0]

[50.0,20.0,90.0,140.0,1.0, 60.0,100.0,90.0,130.0,1.0]

[50.0,10.0,120.0,130.0, 110.0,25.0,130.0,80.0]

[50.0,85.0,60.0,125.0,1.0, 70.0,60.0,100.0,130.0,1.0, 110.0,10.0,120.0,70.0,1.0]

[110.0,40.0,130.0,80.0,1.0, 115.0,25.0,125.0,45.0,1.0]

FIG-2 Output stored in the multi dimensional indexing structure

6. CONCLUSIONS

The main motivation for employing the data stream model is that it allows processing of data that is many times bigger than accessible working memory. Ensemble classification is frequent and more accepted in stream data classification. Ensemble tree indexing data structure is constructed for ensemble classification. Node splitting occurs through ensemble tree construction. A new heuristic method is established for node splitting. In this new technique, time complexity of node splitting is O(n log n) and new technique is very painless to apply. First, values at the at hand node are ordered and then node split is applied. In future, there is a possibility to find other node splitting methods with linear time complexity.

REFERENCES

[1] Peng Zhang, Chuan Zhou, Peng Wang, Byron J. Gao, Xingquan Zhu, and Li Guo ETREE: An Efficient Indexing Structurefor Ensemble Models on Data Streams "IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 27, NO. 2, FEBRUARY 2015"

[2] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, third ed. Morgan Kaufmann, 2011.

[3] J. Quinlan, C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.

[4] Guillaume Noël, Sylvie Servigne, Robert Laurini Liris Real-Time Spatiotemporal Data Indexing Structure

[5] Albert Bifet and Richard Kirkby August 2009 DATA STREAM MINING A Practical Approach