ENHANCED E-TREE FOR MINING HIGH DIMENSIONAL DATA

BY

S. SALAM * M. ROJA ** T.V. RAO ***

* Associate Professor, Department of Computer Science and Engineering, Sree Vidyanikethan Engineering College, Tirupati, India.
** PG Scholar, Department of Computer Science and Engineering, Sree Vidyanikethan Engineering College, Tirupati, India.
*** Professor, Department of Computer Science and Engineering, Sree Vidyanikethan Engineering College, Tirupati, India.

ABSTRACT

Data Stream classification is one of the critical tasks in data mining. At the point when DataStream touches the base at a pace of GB/sec, we need to recognize spam, web observing and capacity. It is a troublesome operation and falls flat in the existing System. Actualizing two Algorithms namely, E-tree Algorithm (Ensemble-tree) and Avaricious Algorithm and Executing E-tree algorithm, the authors have maintained a strategic distance from the existing issues. Ensemble tree (E-tree) takes care of extensive volumes of stream data and drifting. E-tree, Classifies and groups the Data Stream and stores the data effectively. Furthermore, foresee web checking and spam identification precisely. Controlling the web movement, the authors have actualized the greedy algorithm.

Keywords: E-Tree (Ensemble Tree), Data Stream, Web Monitoring.

INTRODUCTION

Data Mining

Data Mining is the procedure of extricating hidden, interesting and valuable examples from vast data sets. It incorporates a few issues like complex nature, huge data size that can't be unraveled by ordinary strategies or methodologies. Consequently, the procedures of transformative algorithms were utilized to take care of the single objective issues. In any case, numerous genuine issues have different clashing execution measures or destinations. To tackle these various clashes, a few multi objective developmental algorithms have been proposed for the essential data mining task prediction.

There are numerous application spaces, where clients make and share data; news blogs, scientific networks, social networking groups, or disaster management networks. Current information sharing tools, like content management software (e.g., Microsoft Share-Point), permit clients to share data and Annotation on (label) them in a specially appointed way. Also, Google Base permit clients to characterize the characteristics for their items or look over predefined formats. This explanation procedure can encourage ensuing data revelation. Numerous explanation frameworks permit just "untyped" keyword annotation: in case, a client may clarify a climate report utilizing a tag, for example, "Storm Category 3."

Annotation systems that make use of quality worth sets are for the most part more expressive, as they can contain more data than untyped approaches. In such settings, the above data can be entered. A recent line of work towards using more expressive queries that leverage such annotations, is the "pay-as-you-go" querying strategy in Data spaces. In Data spaces, clients give data integration indications at inquiry time. The suspicion in such frameworks is that the data sources as of now contain organized data and the issue is to coordinate the query traits with the source properties.

Numerous frameworks, however, don't have the fundamental "attribute-value" annotation that would make a "pay-as you-go" querying feasible. Explanations that utilize "characteristic worth" sets oblige clients to be more principled in their Annotation endeavors. Clients need to know the hidden composition and field sorts to utilize; they need to be likewise known when to utilize each of these fields. With the schema that regularly have tens or even many accessible fields to fill, this undertaking gets to be difficult and cumbersome.

This results in data entry users ignoring such annotation capabilities. Even if the system allows users to arbitrarily annotate the data with such attribute-value pairs, the users are often unwilling to perform this task: The task not only requires considerable effort, but it also has unclear usefulness for subsequent searches in the future: who is going to use an arbitrary, undefined in a common schema, an attribute type for future searches. But even when using a predetermined schema, when there are many potential fields that can be used, and these fields are going to be useful for searching the database in the future.

Data Stream Classification speaks to a standout amongst the most critical task in datastream mining [5],[6], which is prominently utilized as a part of real-time detection, spam filtering and malicious website monitoring. Compared to traditional classification, Contrasted with conventional characterization, Data Stream arrangement is confronting two additional difficulties, they are substantial/expanding information volumes and float/advancing ideas. In the existing framework, they utilized divide and conquer techniques to handle expansive volumes of stream information [2] with the concept drifting. It comes up short in some focuses. So in the proposed method, by making use of E-tree exactness and time effectiveness has been developed, utilizing two algorithms E-tree and greedy algorithm. Greedy calculation is done for controlling the web traffic.

1. Literature Survey

1.1 B-Tree

B-tree is a stable tree in which all the records are held in the tree's leaves and the leaves are sequentially connected. It is a stable tree and in its each loop, it shows a disc page where there are ingressions. B-tree is predicated on ranking one dimensional data on its leaf nodes. Thus it is not efficient to store involute data which has a spatial location.

A B-tree of order m is a tree which satisfies the following properties:

- Every node has atmost m children.
- Every non-leaf node (except root) has at least [m/2] children.
- The root has at least two children if it is not a leaf node.
- A non-leaf node with k children contains K-1 keys.
- All leaves appear in the same level.

1.2 R-Tree

A R-tree is a index structure for n-dimensional spatial items comparable to a B-tree. It is a height balanced tree with records in the leaf nodes each containing a n-dimensional rectangle and a pointer to a data object having the rectangle as a bouncing box. Larger amount of nodes contain comparative entries which are connected to lower nodes. The nodes relate to disk pages if the structure is disk resident, and the tree is composed so that few nodes will be verified when a spatial search is performed [9]. The index is fully dynamic, where insertions and deletions will be intermixed with searches and no occasional recognition is required. A spatial database comprises of a group of records containing spatial objects, and every record has an interesting identifier, which can be used to recover it. The authors have estimated each spatial object by a bounding rectangle, i.e. a collection of intervals, one along the dimension:

$$I = (I_0, I_1, .. I_{n-1}) \quad (1)$$

where n represents the number of dimensions and Ii is the closed limit interval of [a, b] describing the extent of the object along dimension i. And again Ii may have two or more endpoints, demonstrating that the object extends outcome indefinitely. Leaf nodes in the tree contain index record entries of the structure (I, tuple-identifier), where tuple-identifier points to a tuple in the database and I is a n-dimensional rectangle containing the spatial object it refers to. Non-leaf nodes contain entries of the structure (I, Child-pointer), where child pointer is the location of another node in the tree and I covers all rectangles in the lower node entries. In other words, I spatially contains all data objects indexed in the subtree rooted at I's entry. Let M be the maximum number of entries that will fit in one node and let $m < M/2$ be a parameter indicating the base number of entries in a node [9]. A R-tree fulfills the accompanying properties:

- Every leaf node contains between m and M index

records unless it is the root.

- For every index record (I, tuple-identifier) in a leaf node, I is the smallest rectangle that spatially contains the n-dimensional data object represented by the demonstrated tuple.
- Every non-leaf node has amongst m and M children unless it is the root.
- For every entry (1 , child-pointer) in a non-leaf node, I is the smallest rectangle that spatially contains the rectangles in the child node.
- The root node has at least two kids unless it is a leaf.
- All leaves exist in the same level.

1.3 R Trees*

R* Trees are a variation of R-trees used for indexing spatial data. R* trees have remotely higher development cost than standard R-trees, as the information may should be reinserted; however the subsequent tree will generally have a superior question execution [7]. Like the standard R-tree, it can store both point and spatial information.

- The R* trees use the same algorithm as the standard R-tree for insertion and deletion operations.
- When embedding, the R*-tree uses a mixed procedure. For leaf nodes, cover is minimized, while for inward hubs, broadening and region are minimized.
- When part, the R*-trees uses a topological split that separates a split node predicated on edge, then minimizes cover.
- In advisement to an improved split system, the R*-trees also attempts to shun parts by reinserting objects and subtrees into the tree, propelled by the idea of B-trees [8].

The insertion system to the R*-trees is with O(M log M) more involute than the direct split methodology O(M) of the R-tree, yet less perplexing than the quadratic split procedure O(M2) for a page size of M questions and has a little effect on the aggregate many-sided nature. The aggregate addition complexity is still commensurable to the R-tree. Reinsertions influence at most one branch of the tree and consequently O(log n) reinsertions, commensurable to playing out a split on a traditional R-tree. So in general, the involution of the R*-trees is equipollent to that of a standard R-tree.

The following are the some of the issues that will occur while performing operations on spatiotemporal database indexing data structures [1].

1.3.1 Robust Ensemble Learning for Mining Noisy Data Streams

- Stream-mining problem is for seen using a statistical estimation framework, and discriminative model, for fast mining of noisy data streams.
- It supports vector machine algorithm [3], [4] has been proposed for continuous learning.
- It is not an ideal choice for fast learning on streaming data.

1.3.2 An Adaptive Engine for Stream Query Processing

- To generate a straightforward initial query plan, which then adapts automatically to a better initial plan, and continues to adapt as conditions change.
- Reducing run-time memory requirements for continuous query processing.
- It addresses the problem of balancing query execution against profiling and optimizing.

1.3.3 Indexing Boolean Expressions

- The goal is to rapidly find the set of Boolean expressions that evaluate to true for a given assignment of values to attributes.
- Boolean expression evaluation on the indexed objects supports the complex query expressions.
- It doesn't scale to a large number of attributes due to the inherent limitations of high dimensional indexing.

1.3.4 Enabling Fast Lazy Learning for Data Streams

- Real-time analysis of these data streams is becoming a key area of data mining research.
- It is easier to reproduce and there is a little cost of storage and transmission.
- Fewer in number than batch methods and only a concept of huge number of examples.

2. Existing System

The existing works think just about combining as a little number of base classifiers. It misses a few datastreams to records, since Data streams lands at pace GB/sec. So it doesn't deal with that high speed [10], [11]. In existing strategy, node splitting in R-trees, and choice standards contain discrete attributes that can't be changed during the area enlargement. Existing spatial indexing strategies are intended for traditional spatial information. A classifier-ensemble based on active learning framework that selectively labels instances from data streams to build a classifier ensemble. Minimum-variance principle and the ideal weighting module are then combined to manufacture a dynamic learning structure for data streams.

3. Problem Formulation

It may not classify every stream records in an auspicious way, as it considers just a few number of base classifiers. Decision rule may contain discrete attributes that can't be changed during the area enlargement. It takes more opportunity for order. There is scalability problem in the existing framework.

4. Proposed System

It decreases the time cost without expanding the error rate. E-tree is stretched out of R-tree. The Proposed E-tree indexing structure for sub straight time, many-sided quality for fast stream records. It effortlessly adjusts with new idea, low variance and ease of parallelization. Greedy algorithm is utilized to control web movement. The proposed minimum-variance is based on dynamic learning method. The proposed strategy is basically not sensitive to the noisy data and small chunk sizes.

4.1 Advantages

- Takes care of a large volume of data stream.
- To avoid the adaptability issue.
- It adapts rapidly to new ideas,
- Achieve low variance errors and ease of parallelization.
- By using Greedy algorithm, the web traffic is controlled.

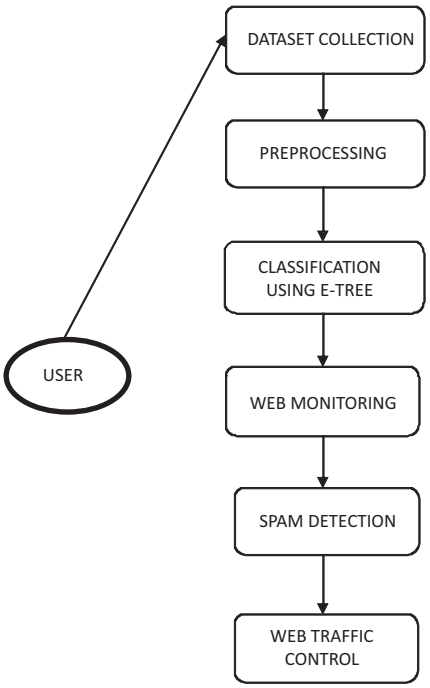

Figure 1. Data Flow Diagram

5. Data Flow Diagram

5.1 Data Load and Preprocessing

Most ordinarily, a dataset compares to the contents of a database table, or a single statistical data matrix, where each section of the table holds to a specific variable, and every line relates to a given individual from the dataset. The dataset records values for each of the variables, for example, height and weight of an object, for every individual from the dataset.

Data preparation and filtering steps can take significant measure of handling time. It incorporates normalization, transformation, feature extraction, selection, etc., analyzing the data that has not been precisely screened for such issues can create misleading results. Therefore, the representation and nature of data is more important before running an analysis. Figure 1 shows the Data Flow Diagram.

5.2 Data Stream Utilizing E-tree

In this module, the authors have computed datastream and these outfit models allot consistent data streams into small data chuncks, build one or more light weight base classifiers from each data chunck, and join base classifiers in various courses for prediction. E-tree structure

that composes base classifiers in a height adjusted tree structure to accomplish logarithmic time complexity for prediction. E-tree performs operations, like Search: traverse an E-tree to order an approaching stream record Insertion: Integrate new classifiers into an E-tree. Which leads to an effective result.

5.3 Spam Detection

User tend to be much less bothered by spam slipping through filters into their mail box than having desired email blocked. Trying to balance false negatives versus false positives is critical for a successful anti-spam system. Some systems let individual users have some control over this balance by setting "spam score" limits, etc. Most techniques have both kinds of serious errors, to varying degrees. In these modules we detect spam message and good message separately.

Inorder to adjust false negatives versus false positives is a basic for a fruitful against spam framework. A few frameworks let singular clients have some control over this equalization by setting "spam score" limits, and so on. Most systems have both sorts of genuine blunders, to fluctuating degrees. In these modules spam message and great message are recognized independently.

5.4 Web Monitoring

Website monitoring is the procedure of testing and checking whether end-clients can interface with the site anticipated. Website monitoring is frequently utilized by organizations to guarantee website uptime, execution, and usefulness as expected. Monitoring accumulates

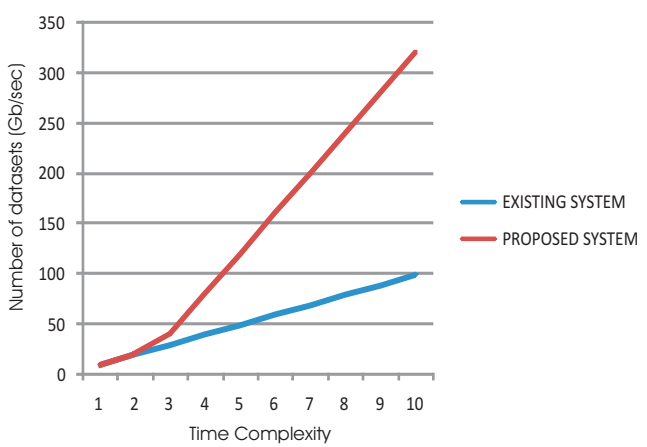

Figure 2. Results

broad information on website execution, such as, load times, server response times, page element performance that is regularly examined and used to further optimize the website performance.

5.5 Web Traffic Control

Web traffic control is the procedure of managing, controlling or reducing the network traffic, especially Internet bandwidth. It is utilized by system network administrators, to reduce congestion, latency and packet loss. This is a part of bandwidth management. To utilize these tools successfully, it is important to measure the web traffic to decide the reasons for network congestion and attack those issues particularly.

6. Experimental Results

Initially, a data stream and a dataset is given as an input to the E-tree. We can see whether the data has been loaded or not. It starts the preprocessing procedure for the dataset provided. When the preprocessing procedure is completed, it will show the difference between the raw data and the preprocessed data. And then it will show the properties of the data stream that has been uploaded. Classification of data can be done according to the structure of the ensemble tree. In order to calculate the web traffic for the dataset, it divides the dataset into small chuncks and a class label is assigned to it. Then a confusion matrix is generated and greedy algorithm is applied to that matrix. The results which are generated after applying the rules present in the greedy algorithm, identifies the spam in the dataset and a mail is sent.

In Figure 2 graph, X-axis represents the difference between the time complexity of the existing system and the proposed system, whereas Y-axis represents the number of datasets that appears at Gb/sec.

Existing system simply concentrates on converting data streams to data records.

Conclusion

The authors have implemented the E-tree algorithm that can efficiently classify the Data Stream. E-tree considers an expansive volume of stream information and drifting. It can also predict the malicious URL and spam identification. E-tree indexing structure for sub linear time

complexity is for classifying high speed stream records. They have derived an ideal weighting strategy to assign weight values for the base classifiers, such that they can frame an ensemble with minimum error rate.

References

[1]. ChuanZhou, (2015). "E-Tree: An Efficient Indexing Structure for Ensemble Models on Data streams". *IEEE Transactions on Knowledge and Data Engineering,* Vol.27, No.2.

[2]. J. Gao, R. Sebastiao, and P. Rodrigues, (2009). "Issues in Evaluation of Stream Learning Algorithms". In *KDD 2009.* pp.329-338.

[3]. H. Yu, L. Ko, K. Y, S. Hwang, and W. Han, (2011). "Exact Indexing for Support Vector Machines". In *SIGMOD 2011.* pp. 709-720.

[4]. Z. Lu, X. Wu, X. Zhu, and J. Bongard, (2010). "Ensemble Pruning Via Individual Contribution Ordering". In *KDD 2010.* pp. 871-880.

[5]. A. Machanavajjhala, E. Vee, M. Garofalakis, and J. Shanmugasundaram, (2008). "Scalable Ranked Publish Subscribe". In *VLDB 2008.* Vol. 1, No. 1, pp. 451-462.

[6]. Y. Zhang, S. Burer, and W. Street, (2007). "Ensemble Pruning Via Semi Definite Programming". *Journal of Machine Learning Research,* Vol. 7, pp. 1315-1338.

[7]. C. Domeniconi and D. Gunopulos, (2011). "Incremental Support Vector Machine Construction". In *ICDM 2011,* pp. 589-592.

[8]. Y. Tao and D. Papadias, (2014). "Performance Analysis of R*-trees with Arbitrary Node Extents". *IEEE Transactions on Knowledge and Data Engineering,* Vol. 16, No. 6, pp. 653-668.

[9]. A. Guttman, (1984). "R-Trees: A Dynamic Index Structure for Spatial Searching". *Proc. ACM SIGMOD,* pp. 47-57.

[10]. P. Domingos and G. Hulten, (2000). "Mining High-Speed Data Streams". *Proc. Sixth ACM SIGKDD Int'l Conf. Knowledge Discovery and DataMining (KDD),* pp. 71-80.

[11]. C. Domeniconi and D. Gunopulos, (2001). "Incremental Support Vector Machine Construction", *Proc. IEEE Int'l Conf. Data Mining (ICDM).*

ABOUT THE AUTHORS

Shaik Salam is currently working as an Associate Professor in the Department of Computer Science and Engineering at Sree Vidyanikethan Educational Institutions, Tirupati, India. He received his Bachelor's Degree from Marathwada University, and then he received his Master's Degree from Sathyabhama University, Chennai. He is also pursuing his Ph.D Degree at Aacharya Nagarjuna University, Guntur. His research interests include Spatial Mining, Web Mining, and Programming Languages.

M. Roja received her Bachelor's Degree from Yogananda Institute of Technology & Science, Tirupathi. She is currently pursuing her Postgraduation in the Department of Computer Science and Engineering, at Sree Vidyanikethan Educational Institutions, Tirupathi. Her research areas include Data Mining, and Programming Languages.

Dr. Venkat Tiruveedhula received his Bachelor's Degree from June 1977 with First Class from A.U. College of Engineering, Andhra Visakapatnam, University, India, and M.E., (Computer Science) in September, 1979 with First Class from P.S.G. College of Technology, Coimbatore, University of Madras, India. He obtained his Ph.D., (Computer Engineering) in December, 1992 with 3.76 G.P.A. in the Department of Electrical and Computer Engineering, from Wayne State University, Detroit, U.S.A. He is currently working as an Associate Professor in the Department of Computer Science and Engineering at Sree Vidyanikethan Educational Institutes, India.