An Efficient Indexing Structure for Ensemble Classification of Data Streams Using Forest-Tree Mechanism

Ms.Priyanka Narayan Kamble, Prof. Sonali A. Patil, Ms. Swati S. Joshi

Abstract— Ensemble learning is used for data stream classification, as it facing problem to large size of stream data and concept drifting. Direct output of an extensive number of base classifiers in the ensemble amid expectation keeping group gaining from being viable for some true time critical data stream applications, e.g. Web traffic. In this data streams usually come at a speed of GBPS, and it is important to order every stream record in a timely manner. That’s why we propose a novel E-tree indexing structure to sort out all bases in an ensemble for fast prediction. Heavy transaction loads at any servers eventually causes the break of the same, so many different steps are been taken to avoid this congestion of data streams. Most of the data stream techniques are based on the size of the data, this techniques are handling streams by maintaining a priority queue. These Methods of handling Data streams are bit old and they create chaos in the network instead of smooth handling. So as a solution for this Ensemble Learning gives a proper perception about data streaming using E-tree and R-tree Models, which successfully incorporates the streaming technique based on the past learning history.

Index Terms— Ensemble learning, Linear Clustering, Entropy Evaluation, E-Tree Formation

I. INTRODUCTION

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 malicious website monitoring. In the applications, data arrive continuously in a stream fashion, timely predictions in identifying malicious records are of essential importance. Compared to traditional classification, data stream classification is facing two extra challenges: large/increasing data volumes and drifting/evolving concepts. To address these challenges, many ensemble-based models have been proposed recently, including weighted classifier ensembles, incremental classifier ensembles, classifier and cluster ensembles to name a few. While these models vary from one to another, they share striking similarity in their design: using divide-and-conquer techniques to handle large volumes of stream data with concept drifting.

Specifically, these ensemble models partition continuous stream data into small data chunks, build one or multiple light-weight base classifier(s) from each chunk, and combine base classifiers in different ways for prediction. Such an ensemble learning design enjoys a number of advantages such as scaling well, adapting quickly to new concepts, low variance errors, and ease of parallelization. As a result, ensemble has become one of the most popular techniques in data stream classification. To date, existing works on ensemble learning in data streams mainly focus on building accurate ensemble models. Prediction efficiency has not been concerned mainly because prediction typically takes linear time, which is sufficient for general applications with undemanding prediction efficiency. Existing works only consider combining a small number of base classifiers, e.g., no more than 30. However, there are increasingly more real-world applications where stream data arrive intensively in large volumes. In addition, the hidden patterns underneath data streams may change continuously, which requires a large number of base classifiers to capture various patterns and form a quality ensemble. Such applications call for fast sub-linear prediction solutions. In online webpage stream monitoring, ensemble learning can be used to identify malicious pages from normal pages, both arriving continuously, in real time. We deployed a detection system on a Linux machine with 3 GHz cpu and 2 GB memory. Each day, a batch of base classifiers is trained using decision trees with all base classifiers being combined to classify pages in the next day. In our experiments, in total 120 days of stream data were used.

II. PROBLEM DEFINITION AND SCOPE

• PROBLEM DEFINITION

Most of the time huge data cannot be classified linearly due to its distribution factor in the irregular form. Many methodologies handle the data based on its occurrence rather than its distribution factor. So handling these data in the form of tree structure can enhance the process of the classification more efficiently.

Manuscript received Oct, 2017.

Priyanka Narayan Kamble, Pursuing M.E. in Computer Engineering at Bhavarabai Savant Institute of Technology & Research Wagholi, Pune, India. 9028591714

Prof. Sonali A. Patil, Assistant Professor Computer Engineering Dept, Bhavarabai Savant Institute of Technology & Research Wagholi, Pune-412207, Wagholi, Pune India.

Swati S Joshi, Pursuing M.E. in Computer Engineering at Bhivarabai Savant Institute of Technology & Research Wagholi, Pune India.
• **SCOPE**
  1. Easy user interface
  2. System is fueled to classify larger dataset with respect to the available resources
  3. Easy to install and needs minimum software support
  4. Step by step process Detail
  5. Accurate classification label generation

### III. SYSTEM ARCHITECTURE

![Figure 1: System Overview of web traffic data streaming classification system](image)

The steps that are depicted in the system overview of figure 1 can be elaborated using the following steps.

**Step 1: Preprocessing** - This is the initial step of the proposed methodology where system is given input of web traffic data log files that is collected from the URL http://recsys.yoochoose.net/challenge.html. Here this data set consists of some attributes like Session ID, Item ID, Product ID and date time. Here in this step of the model system reads the whole data set in the form of buffer string then it stores the each line of the buffer string in a list. Then this list is subjected to select some needed attributes like session id, Price id and item id. Once these attributes are selected then they are stored in a double dimensional list to process further.

**Step 2: Linear Clustering** – Here in this step all the attributes that are stored in the two dimensional array is subjected to cluster linearly. By doing this distribution of the attributes can be verified using entropy evaluation process of the next step. This linear clustering can be depicted using the following algorithm of 1.

**ALGORITHM 1: LINEAR CLUSTERING**

```plaintext
// Input : Set A = { S,P,I }
// output : Set C = { {S_i}, {p_j}, {I_k} }
Step 0: Start
Step 1: Get Set A
Step 2: Create a List T, Set Count = 0
Step 3: For i = 0 to size of A
        Step 4: count++
        Step 5: Add Ai to T
        Step 6: IF Count = 10
        Step 7: Add T to C
        Step 8: Set Count = 0
        Step 9: Empty T
Step 10: END FOR
Step 11: return C
Step 12: END
```

**Step 3: Entropy Evaluation** – This is the process of evaluating the distribution factor of each attributes that eventually helps to create the high end Clusters that thereby helps us to get proper classification labels. This process begins with the process of identifying all the unique attributes and then to search these attributes for the presence of clusters.

Then by using Shannon information gain theory entropy of the attributes are calculated using the equation 1.

\[
IG(E) = - \left( \frac{X}{T} \right) \log \left( \frac{X}{T} \right) - \left( \frac{Y}{T} \right) \log \left( \frac{Y}{T} \right)
\]

Where

- \( X \) = Number of the clusters where attribute is present
- \( Y \) = Number of the clusters where attribute is not present
- \( T \) = Total number of clusters

\( IG(E) \) = Entropy of the given attribute using Information Gain theory.

Here the Information gain value that eventually represents the distribution factor of the attributes. This
entropy value is varied from 0 to 1. If any attribute is having entropy nearer to 1 means that is been distributed more in the web traffic data. And if this entropy value is nearer to 0 means the respective attribute is very least distributed over the web traffic data.

**Step 4: E-Tree Formation and indexing** – This is the step where an improved Ensemble tree is created using the distribution factor of the attributes that are estimated in the prior step. Here the first attribute is fixed as the root node and all the attributes are fixed on their locations based on the scale of the entropy value. And the attributes which are having less entropy value that of the root node is assigned to the left child. If attributes are having higher entropy value that of root node then they are assigned to the right child.

If the attributes are having the entropy as of any existed node in the created ensemble Tree then it is accumulated in the same cluster of the node based on the index of the tree nodes. And this process of creation of E-tree is depicted in algorithm 2.

**Algorithm 2: E-tree**

//input : Attribute Set A={ A, E}

Where

A is Attribute value
E is Entropy value

// output : E-tree E_T

Step 0: Start
Step 1: Create an empty tree as T
Step 2: FOR i=0 to size of A
Step 3: IF i==0
Step 4: Create the Root Node for first Attribute R_a
Step 5: END IF
Step 6: ELSE
Step 7: get A and E
Step 8: Compare E with the instance root node R_a
Step 9: IF (E support < R_a)
Step 10: IF E ∈ T
Step 11: Then add to the A_index node
Step 12: ELSE
Step 13: Add node as left child in T
Step 14: ELSE

Step 15: IF E ∈ T
Step 16: Then add to the A_index node
Step 17: ELSE
Step 18: Add node as right child in T
Step 19: End FOR
Step 20: return T
Step 21: Stop

**Step 5: Clustering and Classification Label formation** - This is the step where created E Tree is traversed in pre ordered manner to fetch all the similar node entities which in turn gives rise to list of elements. Then these elements are been in stored separate list to create the clusters of web traffic data along with the entropy values.

Finally these entropy values are gathered and the respective attribute names are collected with their unique inherited values to get the classification labels through generated E-tree.

IV. SNAPSHOTs
I. CONCLUSION AND FUTURE ENHANCEMENT

This proposed system of classification based on the entropy for E-tree creation yields more than that of mentioned two methodologies like OC SVM & SVDD. Proposed system can be enhanced to work on heterogeneous datasets. Proposed system can be designed to handle huge datasets using cloud computing paradigm. Proposed system can be enhanced to work as a ready made API.

V. ACKNOWLEDGMENT

The authors would like to thank the researchers as well as publishers for making their resources available. We are very much thankful to our Principal Dr. T.K. Nagraj & HOD Prof. G.M. Bhandari. We are also thankful to our PG Coordinator Prof. A.C. Lomate & Project Guide Prof. Sonali A. Patil. We are thankful to the authorities of Savitribai Phule University of Pune, for their constant guidelines and support. Finally, we would like to extend a heartfelt gratitude to friends and family members.

REFERENCES


Priyanka Narayan Kamble, Pursuing M.E. in Computer Engineering at Bhavarakab Sawant Institute of Technology & Research Wagholi, Pune-421207

Prof. Sonali A. Patil, Assistant Professor Computer Engineering Dept. Bhavarakab Sawant Institute of Technology & Research Wagholi, Pune-412207, Wagholi, Pune-412207

Swati S Joshi, Pursuing M.E. in Computer Engineering at Bhavarakab Sawant Institute of Technology & Research Wagholi, Pune-421207