Adaptive Tree-Based Incremental Learning for Real-Time Data Stream Classification

Vaghashia Pratixa Pravinbhai

Head of Department

Computer Engineering,

N. G. Patel Polytechnic,ISroli

Abstract : Knowledge Discovery in Databases (KDD) is an iterative and multi step process that aims at extracting previously unknown and hidden patterns from a huge volume of databases. Data mining is a stage of the entire KDD process that involves applying a particular data mining algorithm to extract an interesting knowledge. Traditional classification processes are not suitable for stream data or continuous or online data and created a need to upgrade traditional classification algorithms. It has inspired data mining community to apply incremental approach in the traditional algorithms. The novel tree based classification algorithm; NID3 uses CAIR criterion for attribute selection and CAIM as online discretization during model preparation. Efficient storage structure is proposed to store the incremental classification model tree of NID3. The proposed classification algorithm deals with numeric as well as multi valued categorical data and assures the incremental classification model to be exact replica of the traditional classification model tree at any instance of time without losing the classification accuracy.

I. INTRODUCTION

Classification consists of predicting a certain outcome based on a given input. Classification has been applied in areas such as retail target marketing, medical diagnosis, weather prediction, credit approval, customer segmentation, and fraud detection to predict certain output based on historical data. In order to predict the outcome, the algorithm processes training set containing a set of attributes and the respective outcome, usually called goal or prediction attribute. By discovering relationships between the attributes algorithm try to predict the outcome. Unknown data set called prediction set is given to algorithm which contains the same set of attributes, except for the prediction attribute which is not yet known. The algorithm analyses the input and produces a prediction. The prediction accuracy defines how good the algorithm is.

Many classification models have been developed like Bayesian classification, neural networks, genetic models, and decision trees [1]. Among these models, decision trees are particularly suited for data mining because compared to the neural network or the Bayesian classifier, the decision tree is easily interpreted by human beings, and can be constructed relatively fast [2] . One of the main drawbacks with the traditional tree based model is that when new data having some new characteristic added then we need to change the classification model. However, constructing a classifier for large growing dataset from scratch would be extremely wasteful and computationally prohibitive. So to classify growing data researcher are motivated to propose technique which build and update model as new data arrive at different point in time in contrast to the traditional tree induction algorithm. This is known as an incremental classification approach.

The Novel incremental tree based classification algorithm; NID3, proposed by R. Mehta et.al [3] uses CAIR statics [4] in decision making for the attribute selection in the classification tree and CAIM; the efficient discretization criterion as online discretization during the classification model preparation. Efficient storage structure is proposed to store the model incrementally along with the storage for the intermediate statistical decision making processes. The resultant tree is an assured replica of the traditional classification tree at any instance of time with competent classification accuracy.

The paper is organized as follows. Section 2 covers related literature survey followed by proposed storage structure for NID3 algorithm in section 3. The NID3 algorithm is simulated with a small datasets; described in section 4.

II Related Work

Incremental learning algorithms gets sample record by record as input. ID3 is top down approach to create decision tree. For non-incremental classification tasks, ID3 is often a good choice for training a decision tree and testing [5]. However, for incremental tasks, to accept instances incrementally, without recreating decision tree every time it is strongly desired to update existing tree.

ID4[2] is extension of ID3 to support incremental classification.ID4 uses E-score as a attribute selection.ID4 accepts a training instance incrementally. To update tree information needed to compute the E-score for the possible test attribute is kept at each node. If the current test attribute does not have the lowest E-score, then it is replaced by the non-test attribute with the lowest E-score. Whenever a non-test attribute replaces the test attribute at a node it discards all sub trees below the node. ID4 creates same decision tree as ID3. ID4 could not handle attribute with continuous value and it can handle only binary valued attribute. ID5R [6] can guarantee to generate the same decision tree as ID3 for a given training set. Unlike ID4, this algorithm can effectively apply the non-incremental method from ID3 to incremental tasks. ID5R maintains sufficient information to calculate E-score for an attribute at a node as well, so that it can replace the test attribute to the one with the lowest E-score. However, instead of discarding the sub trees below the original test attribute, ID5R restructures the tree by making the desired test attribute at the root. This process, named pull-up, it is a tree manipulation that preserves consistency with the existing instances, and brings the implied attribute to the root node of the tree or sub tree. Like ID4 it cannot handle numeric attribute and can handle binary

valued attribute only. ITI[5] is extension of ID5R to overcome its limitation like handling numeric attribute and more than two valued attribute. Attribute having more than two value is encoded in binary variable e.g "income={low,medium,high}" is converted to "income=low",“income=medium" and "income=high".But this will increases number of attribute which in turn increases process complexity as time complexity of algorithm is depend on number of attribute .ITI uses gain ratio as attribute selection.

Domingos and Hulten have also proposed very fast machine learning algorithm based on Hoeffding bound [7]. This technique is based on assumption that probability to choose splitting attribute from small dataset is the same that would be chosen using infinite examples. Model generated by this technique is different from the conventional model. CVFDT [8] It is extension to VFDT and retains speed and accuracy of VFDT and it makes an alternative sub tree, whenever an old one becomes obsolete it replaces old tree with new tree. In this way it makes decision up to date. It maintains statistical information at every node. VFDT and CVFDT is created based on probability, which does not guarantee same tree as that of the conventional model. Major drawback of ID4, ID5, ID5R and ITI is that it could not handle attribute with value more than two value and to overcome this limitation, NID3; a novel tree based incremental classification was proposed. Following section describes the NID3 algorithm briefly.

III. NID3:Novel Tree based Incremental Classification

ID3 is a simple decision tree learning algorithm developed by Ross Quinlan (1983). The basic idea of ID3 algorithm is to construct the decision tree by employing a top-down, greedy search through the given sets to test each attribute at every tree node. In order to select the most informative attribute for classification in the classification tree, ID3 uses information gain. But Study suggest that there exists a problem with this method, as it is often biased to select attributes with more taken values, however, which are not necessarily the best attributes. So to construct decision tree, novel tree based classification [3] algorithm is used. The This algorithm uses CAIR as attribute selection rather than information gain which in turn increases classification accuracy .Figure 1 depicts the step wise procedure of NID3 algorithm.

NID3 algorithm discretize continuous attribute by using CAIM. Then it calculates CAIR value of every attribute and will select attribute having higher CAIR value as splitting attribute and make it as current node which divides dataset in to subset. Then recursively apply algorithm to each subset until all instances of subset belongs to same class.

To make classification model created using NID3 learn incrementally it is required to store classification model along with statistical information at every node. Following section describes storage structure to store classification model.

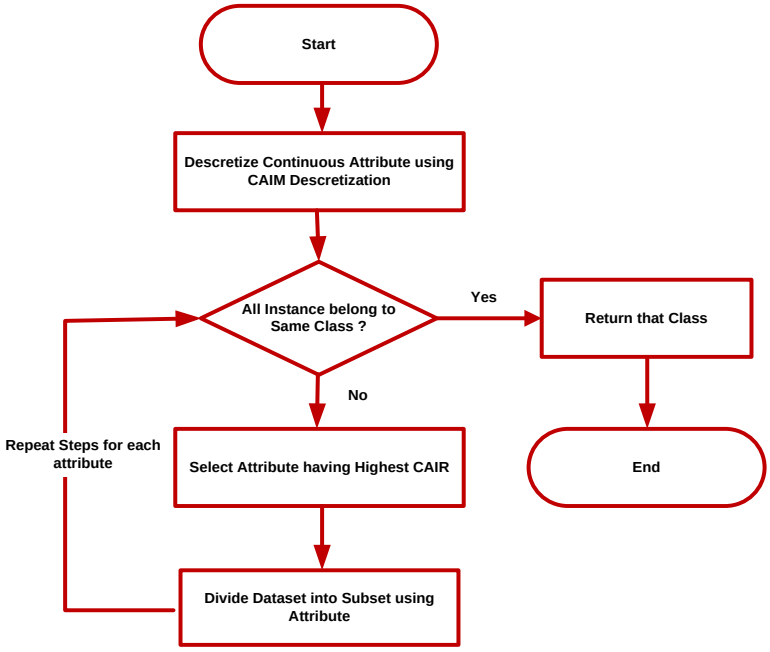

Figure1.NID3 Algorithm

4. Storage Structure

To retain previously learned information classification model is stored in the database. Every node of tree is stored in database along with all information required to update tree .At every node CAIR value and quanta matrix of all attribute at every node is stored so that on arrival of the new record quanta matrix at every node is updated and CAIR value of every attribute is recalculated at every node and tree is updated if drift of decisive information is found. Classification model is stored in database with help of four tables.

i)Tree

In tree table basic information of every node of classification model is stored. Every node of tree has given unique identification number. Along with unique identification other information like level, child of node and whether current node is leaf node or not is also stored in table.

Tree= (Nodeno, Node_ Name, Branch_value, level, IsLeaf, child)

ii) Attribute

In attribute table unique identification number of every attribute is stored.
Attribute=(attribute_id,attribute_name,interval)

iii) cairval

In cairval table CAIR value of every attribute of all node is stored. This table plays crucial role in tree updation.when new record is added CAIR value of every attribute with newly added record is recalculated and checked whether splitting attribute will changes or not, if splitting attribute change, tree will be restructured.
cairval=(node_no,cair,attribute_id)

iv) Quanta matrix

quanta matrix table is used to store quanta matrix of every attribute of every node of tree. Quanta matrix is used to calculate CAIR value. When tree is updated quanta matrix of every attribute is updated depending upon attribute value and CAIR value is recalculated from updated quanta matrix.
Quanta_matrix=(Node no, Attribute_id, Rowid)

4.1 Illustration of storage structure by an example

Proposed algorithm and storage structure is tested for the dataset. The tree created for the data is depicted in figure 2. Basic information of every node is stored in tree table. Table 1 shows view of tree table. Number of record in tree table will be equal to number of nodes in tree.

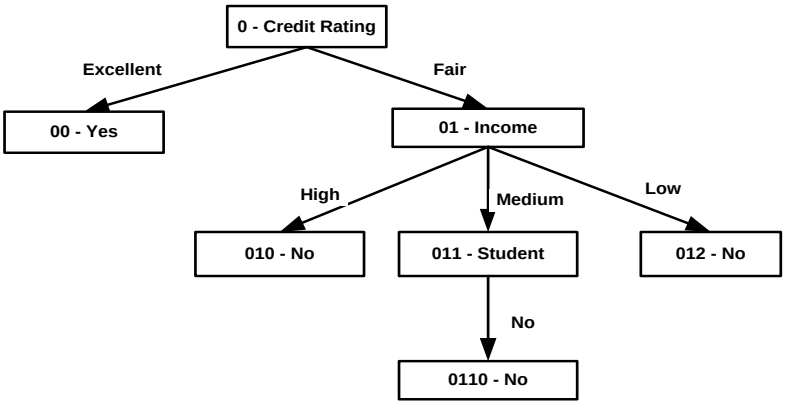

Figure 2.Classification Model

Table 1.Tree table

Node_no	Node_name	Branchval	level	isleaf	child
011	student	No	2	false	0110
0110	no	Null	3	true	null
01	income	high,medium,low	1	false	010,011,012
00	yes	Null	1	true	null
010	no	Null	2	true	null
012	no	Null	2	true	null
0	credit_rating	excellent,fair	0	false	00,01

Other statistical information of every node like CAIR, quanta matrix and attribute identification numbers will store in table cair, quanta matrix and attribute table respectively Along with efficient storage structure for classification model efficient tree updation method is also required to update tree when there is drift in decisive information following section describes tree updation method

5. Tree Updation Method

Updating tree involves including training instance in to tree by passing it to proper branch until it reach to proper leaf node. Figure 3 shows method followed to update existing classification model.

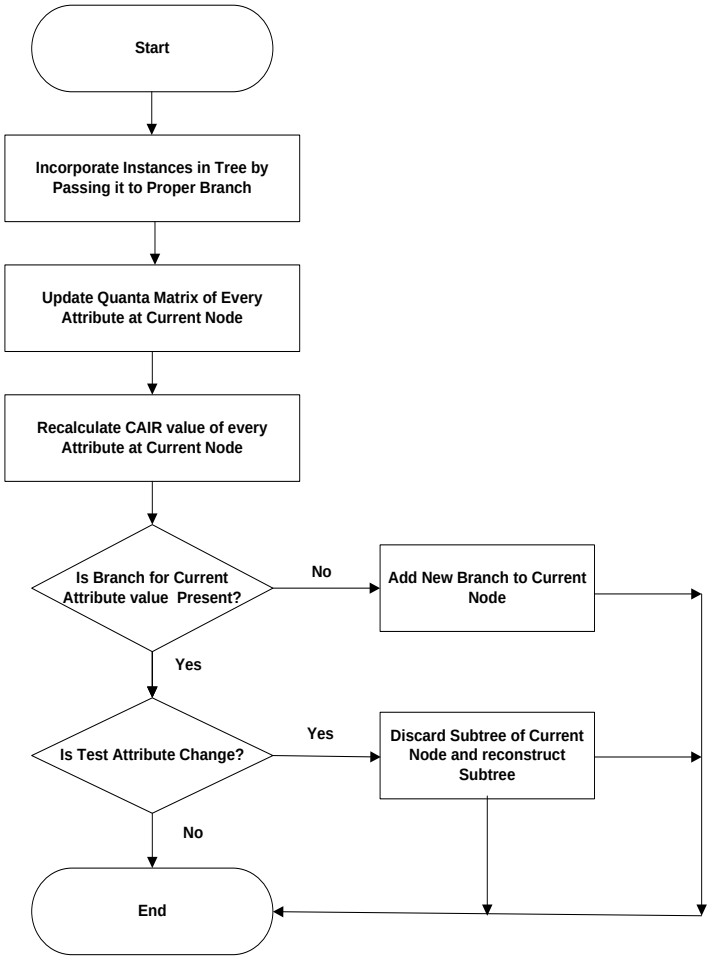

Figure 3.Tree updation method

Tree updation method is heart of incremental classification. Data instances in current window are incorporated in tree by passing it to proper branch. Whenever an example is included, the branches of the tree are followed as far as possible according to the values in the dataset. In path quanta matrix for every attribute at every node is updated and CAIR value is recalculated. If test attribute is changes at node then sub tree of current node is discarded and restructured, if test attribute does not change but branch of tree is not exists corresponding to value of attribute then new branch is added in tree.

Tree updation method guarantee that updated tree will be similar to tree created from the scratch by using dataset used to form existing tree and new dataset in current window.

6. Conclusion

Proposed incremental classification creates the classification tree; incrementally updated for the each instance of the stream data. The efficient tree storage structure ensures the latest statistical information of the tree which is updated for each input instance. The proposed method deals with numeric attribute by performing online discretization with help of CAIM discretization. The CAIR criterion based classification tree makes the classification efficient and the tree created at any stage of time maintains the classification accuracy. The support for the multivalued categorical attribute due to multi branch tree structure makes the proposed algorithm superior than the other algorithms of the same family.

7. References

[1] Carson Kai-Sang Leung, Quamrul I. Khan, Tariqul Hoque, CanTree: A Tree Structure for Efficient Incremental Mining of Frequent Patterns, Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05),2005.

[2] D. Kalles and T. Morris, Efficient Incremental Induction of Decision Trees,Machine Learning, 24, pp 231-241,1999.

[3] M.R.Lad, R.G.Mehta, D.P.Rana,Novel tree based calssification, International Journal of engineering science and Advace technology,volume2,Issue-3,pp 581-586,May- June 2012.

[4] L. Kurgan and K.J. Cios, CAlM Discretization Algorithm, IEEE Transactions of Knowledge and Data Engineering, Vo1.16, No.2, February 2004

[5] P.E.Utgoff,Improved Algorithm for Incremental Induction of Decision tree. In Proceedings of the Eleventh International Conference on Machine Learning,1994

[6] P.E.Utgoff,N.C.Berkman,and J.A.Clouse.Decision Tree Induction Based on Efficient Tree Restructuring.In Machine Learning,1997

[7] Domingos P, Hulten G. Mining high-speed data streams. In Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.New York,ACM Press,2000.

[8] Hulten G, Spencer L, Domingos P. Mining time changing data streams, In Proceedings of the 7thACM SIGKDD International Conference on Knowledge Discovery and Data mining. New York, ACM Press, 2001.

[9] P. E. Utgoff, Incremental Induction of Decision Tress, Machine learning, 4(2), pp161-186.1989.

[10] Johannes Gehrke,Raghurama Krishan,Vnkatesh Ganti,RainForest-A Framework for Fast Decision Tree construction of Large Datasets, Data Mining and Knowledge Discovery, 4, 127–162, 2000.

[11]Ahmed Sultan Al-Hegami, Classical and Incremental Classification in Data Mining Process, IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.12, December 2007.

[12]Prerna Gupta,Amit Thakkar,Amit Ganatra, Comprehensive study on techniques of Incremental learning with decision trees for streamed data, International Journal of Engineering and Advanced Technology, Volume-1, Issue-3, February 2012.

[13]Hitul Patel, Prof. Mehul Barot , Prof. Vinitkumar Gupta,Efficient Tree Based Structure for Mining Frequent Pattern from Transactional Databases, International Journal of Computational Engineering Research,Vol, 03,Issue, 6