ISSN:0975-9646

A Review on Tree Based Incremental Classification

Ponkiya Parita, Purnima Singh

Faculty of IT Department, Parul Institute of Engineering and Technology/GTU,India

Abstract -Data mining has been identical as the technology that offers the possibilities of discovering the hidden knowledge from this large amount of data. Classification is an important well-known problem in the field of data mining, and has remained an extensive research topic within several research communities. When data stream is continuous or online traditional classification will fail so it's required to develop classification model which deals with online data. Contiguous growth of data makes previously constructed classification tree outdated. So this problem can be solved by building classification tree incrementally. Incremental decision tree method allows existing tree to be updated as when new data instances arrives without re processing old data instances. Many decision tree methods like C4.5 construct tree using whole dataset.

I. INTRODUCTION

Today's explosion of Big data companies need more advance methods for leveraging their data-methods they don't relay on trivial knowledge, personal experience or best guesses. Big data have large volume i.e. current volume of data is unprecedented with these massive data sets, it's close to impossible to figure out what to query. No of queries exponentially explodes with the no. of data elements. [2]

Added to volume is velocity of the data; that data is growing up faster and faster. A company encounters a continuous stream of real-time data, social media updates, customer feedback, sales figures, financial data, supply chain data etc. there is simply not enough time to manually query the data.

Data mining has been identical as the technology that offers the possibilities of discovering the hidden knowledge from this large amount of data. When data stream is continuous or online traditional classification will fail so it's required to develop classification model which deals with online data. [3]

There are two types of classification in Data mining Supervised Classification and Unsupervised Classification. In Unsupervised classification [12], no class or target variable is defined. Instead, the data mining algorithm searches for patterns and structure among all the variables. The most unsupervised data mining classification is clustering.

Supervised classification [11] in data mining is the process of assigning Instances in a collection to target classes. The goal of classification is to accurately predict the target class for each instance if dataset. For example, classification model could be used to identify loan applicant as low, medium, or high credit risks. The most supervised data mining classification is Decision tree classification.

There are many supervised classification techniques are available like Decision tree based methods, Rule-based methods, Memory based reasoning, Neural Networks, Naïve Bayes and Bayesian Belief Networks, Support Vector Machines.

Decision tree are particularly suited for data mining for following reasons [8].

Compared to the neural network or Bayesian classifier, the decision tree is easily interpreted and comprehended by human beings and can be constructed relatively fast.

While training neural networks takes a long time and thousands of iteration, including decision tree is efficient and is thus suitable for large datasets. Also decision tree generation algorithms do not require additional information, e.g prior knowledge of domains or distribution on the data and classes, other than that already contained in the dataset.

II. DECISION TREE CLASSIFICATION

A decision tree is flow chart like tree structure in which internal node denotes on an attribute, branch represents out-come of the test leaf node represents class labels or class distribution. This solves a problem known as supervised classification because the dependent attribute and the number of classes (values) that it may have are given. [8]

Classification trees is a classifier in the form of a tree structure (as shown in fig 1.1) where each node is either a leaf node, indicating the value of the target attribute or class of the examples, or a decision node, specifying some test to be carried out on a single attribute-value, with one branch and sub tree for each possible outcome of the test. A classification trees can be used to classify an example by starting at the root of the tree and moving through it until a leaf node is reached, which provides the classification of the instance. [7]

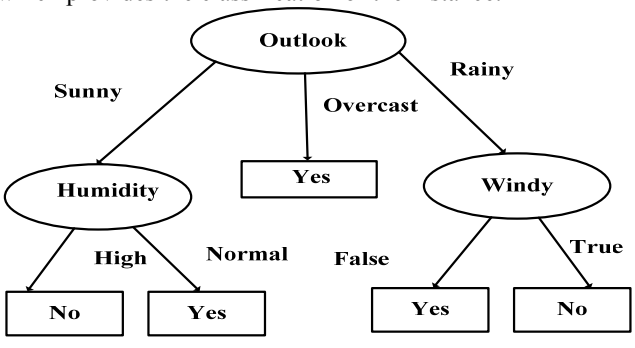

Figure 1.1 Simple Decision Tree

Table 1.1 Simple Classification Example

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

III. INCREMENTAL CLASSIFICATION

One of the main drawbacks with the classical tree induction algorithms in figure 2.1 is that they do not consider the time in which the data arrived. This draw back motivated researcher to develop method which update decision tree classification model as new data arrives instead of rerunning algorithm from scratch, that results incremental classification.[10]

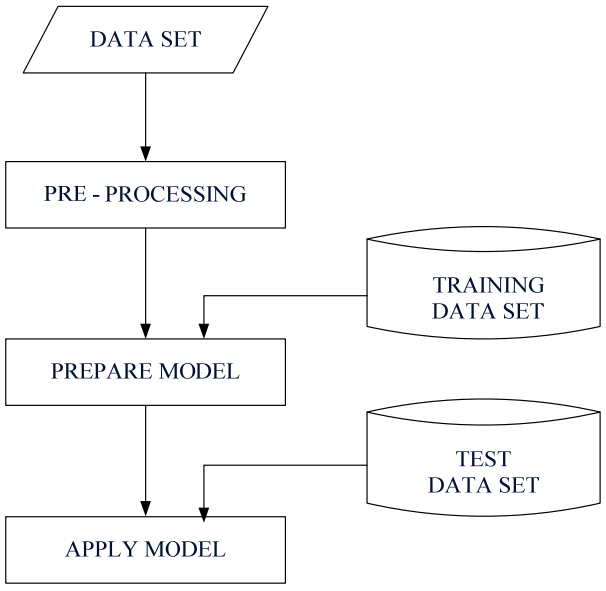

Figure 2.1 Traditional classification model

The incremental classification makes extracting useful information (KDD) process useful and capable. Incremental algorithms shown in figure 2.2 build and filter the model as new information appear at different points in time, in contrast to the traditional tree building algorithms where they build model batch approach. Incremental classification is mostly used method that provides effective and efficient accuracy. Gaining such data is often dull, time-consuming, and costly. New classification model is build from scratch by combining historical data and current data. This approach results in loss of all previously gathered intelligent information. Combining historical data and new Data isn't always feasible option. If historical data are discarded, lost or unavailable. The solution of this problem is incremental classification can be defined as process of extracting new information (patterns) without losing prior of any domain knowledge from future dataset that are available later. [7]

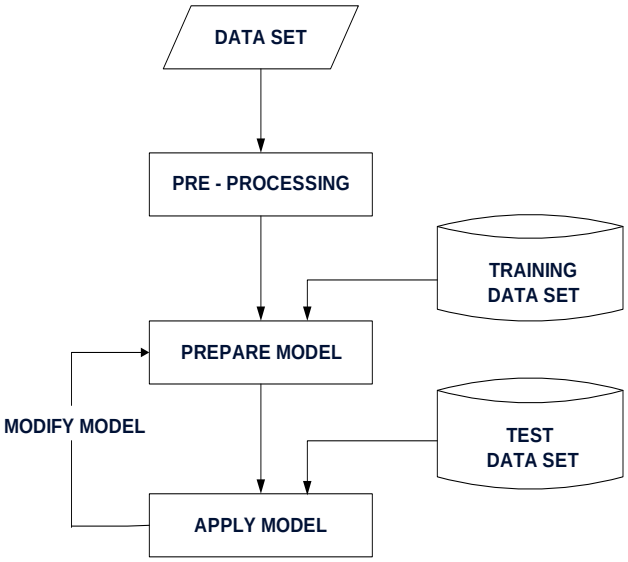

Figure 2.2 Incremental classification models.

Detail of Existing Incremental Classification Methods

Incremental learning algorithms get sample per sample as input. It gives a rough introduction of some typically incremental versions of ID3 and VFDT. ID3 is top down approach to create decision tree. VFDT is a decision tree learning system based on Hoeffding trees. The most of them are divided in two families: ID3 Family and VFDT family. The methods are described below.

1 ID3 FAMILY [10]

For non-incremental classification tasks, ID3 is often a good choice for training a decision tree and testing. However, for incremental tasks, to accept instances incrementally, without creating a new decision tree at each time is strongly desired to update existing tree . In this section, we will first review ID4 algorithm, which is designed to learn decision trees incrementally. Then a new incremental tree-construction algorithm called ID5R, which builds the same decision tree as ID3 from a given training set, is presented. Finally, a powerful successor of ID5R called ITI will be introduced.[10]

a. ID4 Algorithm

ID4[9] is extension of ID3 to support incremental classification.ID4 uses E-score as a attribute selection. ID4 accepts a training instance and then updates the decision tree. In this procedure, 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.Limitation of ID4 is that, it does not support classification of numeric attribute and it can handle only binary valued attribute. [10]

b. ID5R Algorithm

ID5R[12] 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, without the unacceptable expense of generating a new tree after new training instances come. 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, making the desired test attribute at the root. this process, named pull-up, 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 more than two valued attribute.

c. ITI Algorithm [9]

ITI is extension of ID5R to overcome its limitation like handling numeric attribute and more than two valued attribute. In ITI each symbolic attribute is encoded in binary variable e.g. "Outlook= {sunny, rain, windy}" is converted to "outlook=sunny", outlook=rain" and "Outlook=windy". But this will increases number of attribute. which in turn increases process complexity as running time of algorithm is depend on number of attribute. Section, ITI encodes numeric ones using a threshold in the manner similar to C4.5 (i.e., for two adjacent points, choosing corresponding attribute value < midpoint as the split).ITI uses gain ratio as attribute selection. At each decision node, ITI sustains a list of possible tests that could be used as the test at the node. For each binary test based on a categorical attribute, a table of frequency counts for each class and value combination is maintained. For each numeric attribute, a sorted list of the value seen, tagged with its class, is maintained with the attribute, and along with its best threshold.

2 HOEFFDING BOUND-BASED TECHNIQUES.

Domingo's and Hulten have proposed a generic strategy for scaling up machine learning algorithms termed very fast machine learning (VFML).[7] This approach based on deciding an upper bound for the learner's accuracy loss as a function in the number of facts (example) records in each step of the algorithm. Hoeffding bound has been the key for the growth of the VFML techniques. Hence, this group of methods as Hoeffding bound-based techniques. Consider a random variable r whose range is R.

Suppose we have n self-determining observations of this variable, and calculate their mean. The hoeffding bound states that, with probability 1-δ, the true mean of the variable is at least $\bar{r}$-ε, where

$$\epsilon = \sqrt{\frac{R^2 \ln\left(1/\delta\right)}{2n}}$$

a. Very Fast Decision Tree (VFDT) Algorithm

VFDT [7] algorithm uses basic fundamental of a decision-tree learning which is based on the Hoeffding tree algorithm. In VFDT algorithm, threshold value is specified by user. Best statistics attribute split, if the value of split attribute is less than a user-specified threshold. Because it's decided identical attribute is useful or not. So simply compute G and use split to find best attribute periodically. After comparing statistical analysis there may be chance to drop less useful leaves when needed as well as it rescan old data when time available. To keep counts for all leave nodes memory is required, this have a higher priority than VFDT's memory use. If j is the number of attributes, n is the maximum number of values per attribute, and c is the number of classes, VFDT requires O (jnc) memory to store the counts at each stage of leaf nodes. If l is the number of leaves in the tree, the total memory required is O (ljnc). This is neither dependent on how many number of examples we have seen nor training data set size. Pruning play key role in Hoeffding as well as in VFDT algorithm. Pruning is classified into three categories:

1. No-pruning: - In no-pruning VFDT filters tree for an indefinite period
2. Pre-pruning:- In pre-pruning VFDT consider one more factor "τ". Others
3. Post pruning:- VFDT supports all three kind of pruning option. "τ", and when the difference between the best split candidate all others is less than G (.) it stops rising any leaf and G (.) < "τ". When leaves in the decision tree is pre-pruned at certain point than VFDT algorithm terminates. VFDT is work on incremental approach, whenever a new example come it merged with old data. So usable model is available after applying this algorithm on few examples. [11]

b. Concept Adapting Very Fast Decision Tree (CVFDT) Algorithm

VFDT [7] is included in knowledge data discovery assume training data is a sample drawn from stationary distribution. Data stream not considered this assumption because of concept drift. So to continuously change data stream is our only goal. CVFDT [7] is an extended version of VFDT which provides same speed and accuracy advantages but if any changes occur in example generating process provide the ability to detect and respond. Various systems with this capability (Widmer and Kubat, 1996, Ganti et al., 2000), CVFDT uses sliding window of various dataset to keep its model consistent. In Most of systems, it needs to learn a new model from scratch after arrival of new data. Whenever new

data arrives, CVFDT incrementing counts for new data and decrements counts for oldest data in the window. The concept is stationary than there is no statically effect. If the concept is changing, however, some splits examples that will no longer appear best because new data provides more gain than previous one. Whenever this thing occurs, CVFDT create alternative sub-tree to find best attribute at root. Each time new best tree replaces old sub tree and it is more accurate on new data.

CVFDT continuously monitors the validity of its old decisions, by maintaining more than sufficient statistics at every node in Decision tree. As decision classification tree grows it's complex to forgetting instances (example) due to fact that decision tree have altered since the instances was originally integrated. Nodes have assigned unique, monotonically increasing ID as they are created to avoid forgetting instances from a node that has never seen it. The maximum ID of the leaves it reaches in Decision tree and all alternate trees is recorded with it, after addition of an instances to W. An example's effects are forgotten if the example whose ID is less than or equal to stored ID by decrementing the counts in the sufficient statistics. CVFDT Grow: In CVFDT Grow, for each node reached by the example in Hoeffding Tree Increment the corresponding statistics at the node. If sufficient examples seen at the leaf in HT which the example reaches, Choose the attribute that has the highest average value of the attribute evaluation measure (information gain or gini index). If the best attribute is not the "null" attribute, create a node for each possible value of this attribute.[11] Forget Old Example: Maintain the sufficient statistics at every node in Hoeffding tree to monitor the validity of its previous decisions. VFDT only maintain such statistics at leaves. It will assign unique increasing ID as they are created. After each node reached by the old example with node ID no larger than the max leave ID the example reaches, Decrement the corresponding statistics at the node and For each alternate tree Talt of the node, forget (Talt, example, maxID).[11]

CONCLUSION

The aim of this paper is to provide comprehensive survey about various incremental classification algorithms. There are many incremental classification algorithms are available and survey is performed on different incremental classification algorithms in this paper. All classifiers are different from each other because they have different method of tree building and pruning and also have different splitting criteria. So this difference leads to the disparity of decision tree based classification.

REFERENCES

[1] M.R Lad, R.G Mehta, D.P Rana, "Novel Tree Based Classification", IJESAT, 2012.

[2] T. Ryan Hoens · Robi Polikar · Nitesh V. Chawla, " Learning from streaming data with concept drift and imbalance: an overview" , Springer, 13 January 2012

[3] Mohamed Medhat Gaber, " Advance in data stream mining ", Springer, 2012

[4] Tusharkumar Trambadiya, Praveen Bhanodia, " A Comparative study of Stream Data mining Algorithms" , International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 3, September 2012

[5] Ryan Elwell, Robi Polikar, "Incremental Learning of Concept Drift in Nonstationary Environments" , IEEE VOL. 22, NO. 10, OCTOBER 2011

[6] Sheng Chen • Haibo He, " Towards incremental learning of nonstationary imbalanced data stream: a multiple selectively recursive approach" , Springer 2010

[7] 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

[8] Hankil Yoon, Khaled Alsabti, Sanjay Ranka, "Tree based classification for large datasets",1999

[9] P. E. Utgoff, " an improved Algorithm for Incremental Induction of Decision Tress",. Springer 1994.

[10] P. E. Utgoff, " Incremental Induction of Decision Tress ", Machine learning, 4(2), pp 161-186. Springer 1989.

[11]http://docs.oracle.com/cd/B28359_01/datamine.111/b28129/classify.htm #DMCON004

[12] http://dmdingwang.blogspot.in/2007/05/supervised-versus-unsupervised-methods.html