IOSR Journal of Computer Engineering (IOSR-JCE)
ISSN: 2278-0661, ISBN: 2278-8727, PP: 05-09
www.iosrjournals.org

A Semi-Random Multiple Decision Tree Algorithm for mining a Data Streams

Prof. U.S.Dodmise [1], Dr. S.S. Apte[2], Prof. S.J.Salunkhe[3]

[1] *(Department of CSE, BMIT Solapur, Maharashtra, India)*
[2]*(Department of CSE, WIT Solapur, Maharashtra, India)*
[3] *(Department of IT, JJMCOE, Jaysingpur, Maharashtra, India)*

ABSTRACT: To *implement Data mining Software using SRDT model that improves performance in time, space, accuracy and anti-noise capability in comparison with other models like VFDT, VFDTc for classifying data streams. The model we have created is used for classification purpose that gives the accuracy above 90% while classifying the streaming data. Generally, data mining is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both .This is core a model. We can use this model in various applications like Online Shares Trading, Human Resource Management System, Phone Call Records, Project Management System etc*

I. Introduction

To implement Data mining Software using SRDT model that improves performance for classifying data streams in time, space, accuracy and anti-noise capability in comparison with other models like RDT, VFDT, VFDTc .Mining with streaming data is a hot topic in Data mining. Previously we were using the Decision Tree Algorithms Such as VFDT, VFDTc, SRDT (Semi-Random Decision Tree).VFDT deals with Discrete Attributes. VFDTc deals with numerical attributes When Dealing with Data Streams with high attributes dimensions or the high-rate noise, these algorithms become unsuitable. SRDT (Semi-Random Decision Tree) algorithm improves performance in time, space, accuracy & anti noise capability.The SRDT (Semi Random Decision Tree) uses the inequality of Hoeffding bounds to choose the minimum number of split examples, a heuristic method to compute the information gain for obtaining the split threshold of numerical attributes and Naive Bayes classifier to estimate the class labels of tree leaves in-addition with the increasing error rate of classification at leaves for testing data or due to space limitation the created decision tree adjust their height dynamically, and the updating mechanism adopt to the characteristics of streaming data and deals with the classification of data streams efficiently With the rapid development and broad applications of Information technologies, streaming data have become universal, such as supermarket transactions, internet search requests, telephone call records, data from satellites and astronomy etc. Because streaming data are continuous, high volume and open-ended, traditional mining algorithms cannot mine databases from these data environments in real times, which lead to loss of useful information. Also, storing the entire data from such streaming environment difficult, hence efficient mining of data streams become a popular research topic in data mining. An important technique in machine learning, decision trees are used extensively in data mining. They are able to produce human-readable descriptions of trends in the underlying relationships of a dataset and can be used for classification and prediction tasks. SRDT works with both discrete and numerical attributes. SRDT also has strong anti-noise capability. In SRDT we use the heuristic method which deals with the numerical attributes to improve the classification accuracy of decision tree .It is beneficial to reduce whole scale of decision tree.Mining with streaming data is a hot topic in Data mining. Previously we were using the Decision Tree Algorithms Such as VFDT, VFDTc, RDT (Random Decision Tree).VFDT deals with Discrete Attributes. VFDTc deals with numerical attributes When Dealing with Data Streams with high attributes dimensions or the high-rate noise, these algorithms become unsuitable.

II. VFDT and VFDTc

The VFDT algorithm based on decision trees designed and after that its recent successor VFDTc are developed. These are two successful classification algorithms for mining data streams.
VFDT was published in 2000 and can manage thousands of examples with similar performance to a batch decision tree with enough examples. In VFDT, a decision tree recursively replacing leaves with decision nodes. Each leaf stores sufficient statistics about attribute values, which are used to evaluate a merit of split – tests in a heuristic evaluation function.

VFDT and VFDTc maintain a set of sufficient statistics at each decision node and install a split-test at that node.
VFDT only works with descrete attributes, VFDTc extends it to process numerical attribute and applies a Naïve Bayes classifier in tree leaves. Both algorithm reinforce the any-time characteristics, but are weak in the antinoise capabilities and have a high memory cost. They are not suitable for data streams with high dimension and noisy data.

2.1 RDT:
RDT were first proposed in 1997,to generate feature subsets from all features randomly and make use of heuristic information entropy in traditional decision tree learning to select split node. Random Decision Tree model is an ensemble of decision trees. The decision tree construction process in RDT chooses the split attributes of node and the split threshold of numerical attribute randomly, which is unrelated to the training database. RDT only needs to scan the training database once, which has the characteristics of a lower space cost and a strong anti-noise capability, but needs to know the value ranges of numerical attributes in advance in order to process them in a simple way that does not have to fit with the given datastreams.

In RDT the databases have numerical attributes that need to be processed in this model, the value ranges of the numerical attributes must be know in advance. Their split thresholds are determined randomly, which reduces the accuracy, especially if the rate of the numerical attribute dimensions is high and the value ranges of the numerical attributes are large. **Semi Random Decision Tree Algorithm**
1) Create the framework of semi-random decision tree
a. Generate the root of a single tree

b . Choose an available attribute Aj from A as the split attribute at the current node

If Aj is a discrete attribute

Generate m+1 child nodes (where m is the number of different values in Aj)
Else
Generate two child branches < and >= a cut point value of Aj;
For each child node

Go to Step b) until the height of tree = h

2) Update the node counts and determine the split threshold of nodes with the training data For each training data record:

a. Sort training record into the leaf of SRDT.
b. If the current node with Aj is a numerical attribute and the count of examples of this node>= nmin
Compute the value of ϵ by Hoeffding bounds inequality.

Set value of Aj with the highest gain value as the split point.Classify the testing data for each testing data record

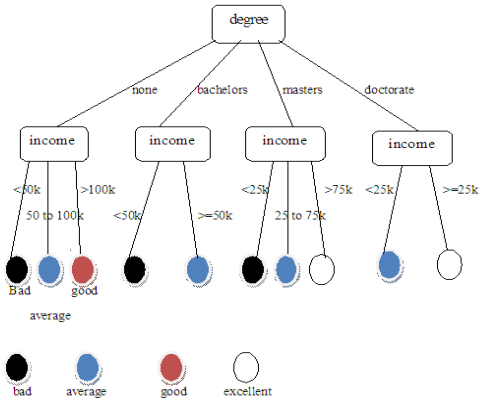
b. Travel the tree from the root to leaves and increase the counts of class labels at each passed node
c. Classify the class label of this testing data record by the judging function of class labels.

2.2 Decision Tree
3) Decision tree is a classifier in the form of tree structure ,where each node is either a leaf node or a decision node. It classifies the example starting from the root of the and moving up to the leaf node.
2.3 Decision –Tree Classifiers

Decision-tree classifiers use a tree; Each leaf node has an associated class, an each internal node has a predicate associated with it.

Fig. shows an example of a decision tree. To classify a new instance, we start at the root and traverse the tree to reach a leaf; It an internal node we evaluate the predicate(or function)on the data instance, to find which child to go to. The process continues till we reach a leaf node. For a example, if the degree level of a person is mastres, and the person's income is 40k,starting from the root we follow the edge labeled "masters," and

from there the edge labeled "25k to 75k,"to reach leaf. The class at the leaf is "good," so we predict that the credit risk of that person is good.



III.Building Decision-Tree Classifiers

Building the decision tree from the given set of training instances we use a greedy algorithm, which works recursively,starting at the root and building the tree downword. There is only one node, the root,and all training instances are associated with that node.

At each node, if all ,or "almost all" training instances associated with the node belong to the same class, then a node becomes a leaf node associated with that class. Otherwise a partitioning attribute and partitioning conditions must be selected to create child nodes. The data associated with each child node is the set of training instances that satisfies the partitioning condition for that child node. In the above example, the attribute degree is chosen,and four children ,one for each value of degree,are created. The conditions for the four children nodes are degree=none,degree=bachelors, degree=masters, and degree=doctorate, respectively. The data associated with each child consist of training instances satisfying the condition associated with that child. The data associated with each node consist of training instances with degree attribute being masters and the income attribute being in each of these ranges, respectively.

IV.Entropy

The concept is used to quantify information is called entropy. Entropy is used to measure the amount of uncertainty or surprise or randomness in a set of data. When all data in a set belong to a single class, there is no uncertainty. In this case entropy is zero. The objective of decision tree classification is to iteratively partition the given data set into subsets where all elements in each final subset belong to the same class.

The formal definition of entropy is shown below. The value for entropy is between 0 and 1 and reaches a maximum when the probabilities are all the same.

Definition: Given probabilities $p_1,p_2,\ldots,p_s$ where $\sum_{i=1}^{s} pi = 1$, entropy is defined as

$H(p_1,p_2,\ldots,p_s)=\sum_{i=1}^{s}(pi \log (1/pi))$

Where $p_1,p_2,\ldots,p_s$ probability value.The function plog(1/p) is the expected information based on probability of an event.

4.1 Information Gain

The information gain is equal to the total entropy for an attribute if for each of the attribute values a unique classification can be made for the result attribute.

Definition:-

IG(Ex,a)=H(Ex) -□|{x∈ Ex|value(x,a)=v}.H({x ∈Ex|value(x,a)=v})|∈values(a)|Ex|

where,

Ex - set of training example a – attribute
Attr – set of all attribute
H – entropy

4.2Hoeffding bound
Consider a real-valued random variable r whose range is R (e.g., for a probability the range is one, and for an information gain the range is log c, where c is the number of classes). Suppose we have made n independent observations of this variable, and computed their mean r. The Hoeffding bound states that, with probability 1 -∂ , the true mean of the variable is at least r - ε , where

$$\square = \sqrt{R2\ln(1/\delta)/2n}$$

The Hoeffding bound has the very attractive property that it is independent of the probability distribution generating the observations. The price of this generality is that the bound is more conservative than distribution-dependent ones (i.e. it will take more observations to reach the same δ and ε).
For implementing the SRDT algorithm we use different user defined methods. We use the methods like calEntropy, getGain, calGain, BuildTree, CreateChild, appendChild and so other.All these methods are defined in different classes.

Entropy class
First we calculate the number of records and classes present in the training file using start method. After that calEntropy method is used for calculating the entropy of each class found in the training file. For this purpose we read the input file trainingfile which contains number of records and classes ..

Information Gain New class
In this class first use the getGain method. In this method we separate the attributes and classes.Calculate the total number of nodes in each line and total number of records in the training file. After that call the start method of entropy. After getting the total number of records and number of nodes we call the calGain method for calculating the information gain.In this method we find out the total number of positive and negative records. Then calculate the gain of positive and negative records. Finally calculate the information gain of each attribute.

After getting the information gain of each attribute we can arrange the records in descending order .For this purpose we use the temp class.
Tree Node class
When we get the attributes with descending order we choose the first record as the split point. Using that split point we can build the decision tree. First attribute is chosen as the parent node. After that child node will be created. When child node is created other nodes will be appended. While appending the nodes each time condition will be checked whether the attribute is less than or greater than equal to the split point. From that attribute will be appended correct place. In this way all nodes will be appended up to the end of file.

Main Algorithm class
This class implements the srdt algorithm. Implementing the srdt algorithm involves building decision tree by using training file. Once decision tree is build the records in the testing file is parsed on the tree. After that classification and testing is performed.

In this class BuildTree function build decision tree. DFS function traverse the decision tree according to depth first search and calculates the classtokens(name of no of classes), classcounts(no of a particular class). After that maximus function used for classification purpose. Testing is performed by using Testing function. Once we build decision tree and if the accuracy is less than 90% then for calculating new split point hbound function is used. After building the decision tree if the accuracy is less than 90% and hvalue(value for split point) is less than threshold value then getFrequentIndex function is used. In that function the index of maximum record passed in that node is stored. Similarly getBestIndex function stores the best index.Finally the accuracy is shown to the user.

First GUI form is created. When we select training file then path of that training file will be shown in the text area similar to that when we select training file then path of that testing file will be shown in the text area. When we select testing file before training file then it gives error. After selecting both the input files when

we select Run SRDT Algorithm then it gives accuracy.

V.Conclusion

SRDT makes use of the information entropy, combines the Hoeffding bounds to choose the cut-points of numerical attributes and introduces Naïve Bayes classifiers from VFDTc .Our experiments have shown that SRDT has improved performance in time, space and the anti-noise capability as compared with the state-of-art algorithm VFDTc.

VI.Future Enhancement

- SRDT only classifies the given input data.
- We cannot classify the streaming data using SRDT algorithm, for classifying such streaming data some changes in SRDT algorithm made.
- The new algorithm is the SRMDT (Semi Random Multiple Decision Tree).
- The future work is implementation of SRMDT algorithm. In SRMDT algorithm we can classify the streaming files using multiple decision trees at a time.

References

[1.] Xue-Gang Hu, Pei-Pei Li, Xin-Dong Wu, and Gong-Qing Wu. "A Semi-random Decision-Tree for Mining data streams". JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY. Sept. 2007
[2.] Pedro Domingos & Geoff Hulten."Mining High Speed Data Stream".
[3.] http://dms.irb.hr/tutorial/tut_dtrees.php.
[4.] http://ftp.utcluj.ro/pub/users/nedevschr/Av/5_EvaluareClassification/entropy.pdf
[5.] http://en.wikipedia.org/wiki/Entropy_(information_theory).
[6.] Software Engineering A Practitioner's Approach- Rojar S. Pressman.
[7.] The Unified Modeling Language User Guide – Grady Booch, James Rumbaugh.

Books

[1.] Margarent H. Dunham and S. Sridhar."Data Mining Introductory and Advanced Topics".
[2.] Pearson Education, Page no-71-95.
[3.] Henry F. Korth and S. Sudarshan."Database System and concepts".
[4.] McGRAW-HILL INTERNATIONAL EDITION (Fifth edition),Page no.-742-750.
[5.] Herbert Schildt . "The complete Reference Java 2".

Article in Journals

[1.] Xue-Gang Hu, Pei-Pei Li, Xin-Dong Wu etc .A semi-random multiple decision-tree algorithm for mining data streams[J] Journal of Computer Science and Technology, 2007,V22(5): 711-724

World Wide Websites

[2.] http://dms.irb.hr/tutorial/tut_dtrees.php.
[3.] http://ftp.utcluj.ro/pub/users/nedevschr/Av/5_EvaluareClassification/entropy.pdf
[4.] http://en.wikipedia.org/wiki/Entropy_(information_theory).