ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 2, February 2014

Dynamic Learing Algorothm For Massive Data Streams

Ms.Manjusha Reddy[1], Prof. Rajesh Bharati [2]

ME Student, Dept. of Computer Engineering, D.Y.P.I.E.T, Pimpri, Pune, India[1]

Professor , Dept. of Computer Engineering, D.Y.P.I.E.T. Pimpri, Pune. India[2]

ABSTRACT: Many organizations having huge databases; the databases grow without limit at a rate of several million records per day. Mining these continuous data stream brings unique opportunities. VFDT use constant memory and constant time to builds decision trees per example. Using off the –shelf hardware, VFDT can to generate examples. The Hoeffding bounds output nearly similar to conventional learner. In this paper we introduce an effective algorithm for mining decision trees from massive data streams, based on the ultra-fast VFDT decision tree learner. Anthor algorithm defined as CVFDT,stays current while making the most of old data by growing an alternating sub tree whenever an old one becomes questionable ,and replacing the old with the new when the new becomes more accurate. CVFDT learns a model which is similar in accuracy to the one that would be learned by reapplying VFDT to a moving window of examples every time a new examples arrives, but with O(1)complexity per example ,as opposed to O(w),where w is the size of the window.

KEYWORDS: Decision tree learner, additive Cheroff bound, incremental learning, massive data streams, subsampling,,concept drifts.

I. INTRODUCTION

The algorithm VFDT makes the assumption that training data is random sample drawn from a stationary distribution. Today large databases and data streams available for mining . They exits over month or years ,and underlying processes generating them changes during this time ,sometimes radically .For example ,a new product or promotion, a hackers attack, a holiday, changing weather conditions, changing economic conditions, or a poorly calibrated sensor could all lead to violations of this assumption. For classification systems, which attempt to learn a discrete functions given examples of its inputs and outputs, this problem takes the form of changes in the target function over time, and is known as concept drift. Traditional systems assume that all data was generated by a single concept. Traditional systems learn incorrect models when they erroneously assume that the underlying concept is stationary if it is drifting.

One common approach to learning from timing-changing data is repeatedly apply a traditional learner to a sliding window of w examples arrive they are inserted into the beginning of the window, a corresponding rapid number of examples is removed from the end of the window, and the learner is reapplied. As long as w is small relative to the rate of a model reflecting the current concept generating the data .If the window is too small , this may result in insufficient examples to satisfactorily learn the concept. Further ,the computational cost of reapplying a learner may be prohibitively high, if example arrive at a rapid rate and the concept changes quickly.

To meet these challenges we propose the CVFDT system ,which is capable of learning decision trees from high speed ,time changing data streams .CVFDT works by efficiently keeping a decision tree up-to-date with a window of examples. In a particular ,it is a able to learn a nearly equivalent model to VFDT would learn if repeatedly reapplied to a window of examples ,but in O(1) time instead of O(w)time per new example.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 2, February 2014

In the next section we discuss the basics of VFDT system ,and in the following section we introduce the CVFDT system.

II. LITERATURE REVIEW

A. THE VFDT SYSTEM

The classification problem is defined as follows .A set of N training examples of the form (x ,y) is given ,where y is the discrete class label and x is a vector of d attributes ,each of which may be symbolic or numeric .The gThe goal is to produce from these examples a model y=f(x) which will be predict the classes y of future examples x with high accuracy . For examples ,x could be a description of a client ‘s recent purchases ,and y the decision to send that customer a catalog or not; or x could be a record of a cellular telephone call ,and y the decision whether it is fraudulent or not .One of the most effective and widely used classification method is decision learning .Learners of this type induce model in the form of decision trees ,where each node contains a test on an attribute ,each branch from a node corresponds to a possible outcome of the test ,and each leaf contains a class prediction .The label y=DT(x) for an example is x obtained by passing the example down from the root to a leaf, testing the appropriate attribute at each node and following the branch corresponding to the attributes value in the example. A decision tree is learned by recursively replacing leaves by test nodes, starting at the root. The attribute to test at a node is choosing the best one according to some heuristic measure.

We presented the VFDT (Very Fast Decision Tree)system ,which is able to learn from abundant data within practical time and memory constraints .Thus ,only the first examples to arrive on the data stream need to be used to choose the spilt attribute at the root; subsequent ones are passed through the induced portion of the tree until they reach a leaf are used to choose a spilt attribute there, and so recursively. To determine the number of examples needed for each decision, VFDT uses a statistical result known as Hoeffding bounds or additive Chernoff bounds. After n independent observations of a real-valued random variable r with confidence 1-d ,the true mean of r is at least r̄ -€ ,where r̄ is the observed mean of the samples and

$$\varepsilon = \sqrt{\frac{R^2 ln(1/\delta)}{2N}}.$$

Let G (X_i) be the heuristic that generated the observations. Let $G(X_i)$ be the heuristic measure used to choose test attributes.(we use information gain).After seeing n samples at a leaf ,let X_a be the attribute with the best heuristic measure and X_b be the attribute with the second best. Let $\Delta G=G(X_a) - G(X_b)$ be a new random variable, the difference

between the observed heuristic values. Applying the Hoeffding bound to ΔG ,we see that if $\Delta G > €$,we can confidently say that the difference between $G(X_a)$ and $G(X_b)$ is larger than zero ,and select X_a the split attribute. The count $n_{i,jk}$ the sufficient statistics needed to compute most heuristic measures; if other quantities are required ,they can be similarly maintained. When sufficient statistics fill the available memory, VFDT reduces its memory requirements by temporarily deactivating learning in the least promising nodes; these nodes can be reactivated later if they begin to look more promising than currently active nodes .VFDT employs a tie mechanism which precludes it from spending inordinate time deciding between attributes whose practical difference is negligible.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 2, February 2014

Table 1: VFDT ALGORITHM

Input:

S: sequence of examples

X : attributes

G(); Evaluation function(gain, gain ratio, gain index)

d : accuracy parameter

tau : user provide the breaking value

Algorithm

1. For each example in s
2. Retrieve $G(X_a)$ and $G(X_b)$ // two highest G(X)
3. If $((G(X_a)-G(X_b) > €)$ or $(G(X_a)-G(X_b)>=€ =tau)$
4. Spilt on X_a
5. Recurse to next node

break

B. THE CVFDT SYSTEM

CVFDT (Concept –adapting Very Fast Decision Tree learner) is an extension to VFDT which maintains VFDT's speed and accuracy to change in the example –generating process. Like other System with this capability , CVFDT works by keeping its model consistent with a sliding window of examples. However, it very does not need to learn a new model time from scratch every time a new example arrives ;instead, it updates the sufficient statistics at its nodes by incrementing the example the new counts corresponding to the new example arrives ;decrementing the counts corresponding to the oldest e xample in the window. This will statistically have no effect if the underlying concept is stationary. If the concept is changing ,some splits that previously passed the Hoeffding test will no longer do so ,because an alternatively attribute now has higher gain .In this case CVFDT begins to grow an alternative sub tree with the new best attribute at its root. When this alternative sub tree becomes more accurate on new data than the old one, the old sub tree is replaced by the new one.

- Increment count with new example
- Decrement old example
- Sliding window
- Nodes assigned monotonically increasing IDs
- Grows alternate more accurate => replace old
- O(w) better runtime than VFDT-window

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 2, February 2014

Table2: The CVFDT Algorithm

5) Alternate trees for each node in HT start as empty.

6) Process examples from the stream indefinitely. For each

-pass (x,y) down to a set of leaves using HT and all alternate trees of the nodes (x,y)passes through.

* Assign ID of leaf to the example.

* Add (x,y) ID to the sliding window of examples.

* If (window size > w) Remove and forget the effect of the oldest examples,if the sliding window overflows.

Condition: nodeID <=exampleID

*CVFDTGrow—increment count for each node

*checkSplitVality

-- If examples seen since last checking of alternate trees.

-- If (G(Xb)-Gs(Xc)> € or G(Xb)-G(Xc)>=tau/2 and € < tau then spilt on Xb)

--Xb: second best splitting attr.

--Xc: Third Best spilting attr.

-- Tau/2: to avoid formation of excessive alt trees

8) Return HT.

Tables 2 contain a pseudo-code outline of the CVFDT algorithm. CVFDT does some initializations, and then processes examples from the stream S indefinitely. As each example (x,y) is incorporated into the current model .CVFDT periodically scans HT all alternate trees looking for internal nodes whose sufficient statistics indicate that some new attribute would make a better test than the chosen spilt attribute. An alternate sub tree is started at each such node.

Table 2 contains pseudo-code for tree-growing portion of the CVFDT system. It is similar to the Hoeffding Tree Algorithm, but CVFDT monitors the validity of its old decisions by maintaining sufficient statistics at every node in HT .Forgetting an old example is slightly complicated by the fact that HT may have grown or changed since the examples was initially incorporated .Therefore, nodes are assigned a unique, monotonically increasing ID as they are created. When an example is added to W, the maximum ID of the leaves it reaches in HT and all alternate trees is recorded with it. An examples are forgotten by decrementing the counts in the sufficient statistics of every node the example reaches in HT whose ID is ≤ the sorted ID.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 2, February 2014

CVFDT periodically scans the internal nodes of HT looking for ones where the chosen spilt attribute would no longer be selected ;that is ,where $G(X_a) - G(X_b) \leq €$ and $€ > t$. When it finds such a node a CVFDT known that it either initially made a mistake splitting on X_a or that something about the process generating examples has changed.

III.EXPERIMENTAL RESULTS

We are applying VFDT to mining the stream of web pages.

Table 3: Data seta used in the Web Experiments.

Data set	Train Exs	Test Exs	%Classo
30 sec	10,850K	1,944K	46.0
1 min	7,921K	1,419K	45,4
15 min	2,555K	454K	58.1

Table 4. Results of Web Experiments.

Ds	C4.5 error	VFerror	C4.5 error	MT Ksec	VFtime Ksec
30 sec	37.70%	36.95%	60	5	247
1 mn	37.30%	36.67%	71	6	160
5mn	33.59%	33.23%	61	15	72

IV. FURURE SCOPE

We plan to apply CVFDT to more real-world problems; its ability to adjust to concept changes should allow it to perform very well on a broad range of tasks. CVFDT may be a useful tool for identifying
anomalous situations. Currently CVFDT discards sub trees that are out-of-date, but some concepts change periodically and these sub trees may become useful again – identifying these situations and taking advantage of them is another area for further study. Other areas for study include: comparisons with related systems; continuous attributeS.

V. CONCLUSION

VFDT is a high-performance data mining system based on Hoeffding trees. Empirical studies show its effectiveness in taking advantage of massive numbers of examples.
This application also uses CVFDT, a decision-tree induction system capable of learning accurate models from the most demanding high-speed, concept-drifting data streams. CVFDT is able to maintain a decision-tree up-to-date with a window of examples by using a small, constant amount of time for each new example that arrives.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 2, February 2014

The resulting accuracy is similar to what would be obtained by reapplying a conventional learner to the entire window every time a new example arrives. Empirical studies show that CVFDT is effectively able to keep its model up-to-date with a massive data stream even in the face of large and frequent concept shifts.

VI . ACKNOWLEDEMENT

This is a great pleasure and immense satisfaction to express my deepest sense of gratitude and thanks to everyone who has directly or indirectly helped me . I express my my gratitude towards my project guide Prof.Rajesh Bharati ,for his suggestion and constant guildline during paper presention.

It is a privilege for me to thank my P.G. Cordinator Prof. Jyoti Rao for her constant support.

REFERENCES

[1]. Geoff Hutten, Pedro Domingo's, "Mining High-Speed Data Streams", ACM.
[2].G. Hulten, L. Spencer, and P. Domingo's. Mining time-changing data streams. In *KDD'01*, pages 97–106. ACM, 2001.
[3]. .Peipei Li, Xindong Wu, Xuegang Hu, "Learning from Concept Drifting Data Streams with Unlabeled Data".
[4]. G. Holmes, R. Kirkby, and B. Pfahringer. Moa: Massive online analysis. http://sourceforge.net/projects/moa-datastream, 2007.
[5]. J. Gama, P. Medas, and P. Rodrigues. "Learning Decision Trees from Dynamic Data Streams.", IEEE 2005.
[6] .Chunquan Liang, Yang Zhang, Qun Song, "Decision Tree for Dynamic and Uncertain Data Streams".
[7] .L. Breiman, J. H. Friedman, R. A. Olshen, and C. J.Stone. Classi_cation and Regression Trees.Wadsworth, Belmont, CA, 1984.