Moderated VFDT in Stream Mining Using Adaptive Tie Threshold and Incremental Pruning

Conference Paper · August 2011

DOI: 10.1007/978-3-642-23544-3_36 · Source: DBLP

2 authors, including:

Simon Fong
University of Macau
573 PUBLICATIONS **2,449** CITATIONS

SEE PROFILE

Moderated VFDT in Stream Mining Using Adaptive Tie Threshold and Incremental Pruning

Hang Yang and Simon Fong

Department of Science and Technology, University of Macau,
Av. Padre Tomás Pereira Taipa, Macau, China
henry.yh@gmail.com, ccfong@umac.mo

Abstract. Very Fast Decision Tree (VFDT) is one of the most popular decision tree algorithms in data stream mining. The tree building process is based on the principle of the Hoeffding bound to decide on splitting nodes with sufficient data statistics at the leaf. The original version of VFDT requires a user-defined tie threshold by which a split will be forced to break to control the tree size. It is an open problem that the tree size grows tremendously with noise as continuous data stream in and the classifier's accuracy drops. In this paper, we propose a Moderated VFDT (M-VFDT), which uses an adaptive tie threshold for node splitting control by incremental computing. The tree building process is as fast as that of the original VFDT. The accuracy of M-VFDT improves significantly even under the presence of noise in the data stream. To solve the explosion of tree size, which is still an inherent problem in VFDT, we propose two lightweight pre-pruning mechanisms for stream mining (post-pruning is not appropriate here because of the streaming operation). Experiments are conducted to verify the merits of our new methods. M-VFDT with a pruning mechanism shows a better performance than the original VFDT at all times. Our contribution is a new model that can efficiently achieve a compact decision tree and good accuracy as an optimal balance in data stream mining.

Keywords: Data Stream Mining, Hoeffding Bound, Incremental Pruning.

1 Introduction

Since the early 2000s, a new generation of data mining called data stream mining (DSM) has received much research attention. DSM requires only one pass on infinite streaming data and the decision model is dynamically trained, while the incoming new data streams are being received in run-time [1]. Very Fast Decision Tree (VFDT) is a well-known decision tree algorithm for DSM [2]. Its underlying principle is a dynamic decision tree building process that uses a Hoeffding bound (HB) to determine the conversion of a tree leaf to a tree node by accumulating sufficient statistics from the new samples. Although VFDT is able to progressively construct a decision tree from the unbounded data stream, VFDT suffers from tree size explosion and the deterioration of prediction accuracy when the data streams are impaired by

A. Cuzzocrea and U. Dayal (Eds.): DaWaK 2011, LNCS 6862, pp. 471–483, 2011.

noise. Such imperfect data often exists in real life, probably because of unreliable communication hardware or temporary data loss due to network traffic fluctuation. Although VFDT and its variants have been extensively studied, many models assume a perfect data stream and have sub-optimal performance under imperfect data streams. In this paper, we devise a new version of VFDT called Moderated VFDT (M-VFDT) that can provide sustainable prediction accuracy and regulate the growth of decision tree size to a reasonable extent, even in the presence of noise. This is achieved by revising the decision tree building process – in particular, the conditional check of whether a leaf should be split as a new tree node is modified. The new checking condition is made adaptive to the distribution of the incoming data samples, which in turn influences the value of the HB that is a key factor in the decision tree construction. It is adaptive in the sense that no human intervention is required during the data stream mining; we let the incoming data decide on how precisely (or how frequently) the tree node splitting should be done, hence the depth of the decision tree. Improved accuracy is achieved by an adaptive tie threshold rather than a user-defined tie threshold, but the tree size is still as big as in VFDT. To solve this problem, incremental pruning methods are proposed to complement the adaptive tie threshold mechanism for controlling the tree size as well as maintaining the accuracy. The result is an optimally compact decision tree that has good prediction accuracy, by M-VFDT. This work is significant because the proposed algorithms (adaptive tie threshold and pruning) are both lightweight and adaptive and this makes M-VFDT favorable in a data stream mining environment.

This paper is structured as follows. Section 2 introduces a research framework that summarizes the background of VFDT, the effect of the tie threshold in tree building and the impact of noise in data stream mining. Section 3 presents details of our proposed model M-VFDT that consists of the adaptive tie threshold and incremental pruning mechanisms. Experimental validation is carried out in the following Section. Both synthetic and real-world stream datasets are used to thoroughly test the performance of M-VFDT compared to VFDT. The experimental results demonstrate that M-VFDT performs better than the original VFDT at all times. The conclusion of this study is given in Section 5.

2 Research Background

2.1 Very Fast Decision Tree (VFDT)

The VFDT system [2] constructs a decision tree by using constant memory and constant time per sample. It is a pioneering predictive technique that utilizes the Hoeffding bound (HB). The tree is built by recursively replacing leaves with decision nodes. Sufficient statistics of attribute values are stored in each leaf. Heuristic evaluation function is used to determine split attributes converting from leaves to nodes. Nodes contain the split attributes and leaves contain only the class labels. The leaf represents a class that the sample labels. When a sample enters, it traverses the

tree from root to leaf, evaluating the relevant attribute at every single node. After the sample reaches a leaf the existing statistics are updated. At this time, the system evaluates each possible condition based on attribute values: if the statistics are sufficient to support the one test over the other, a leaf is converted to a decision node. The decision node contains the number of possible values for the chosen attribute of the installed split test. The main elements of VFDT include: firstly, a tree initializing process that only has a single leaf at the beginning; secondly, a tree growing process that contains a splitting check using the heuristic evaluation function *G(.)* and the HB. VFDT uses information gain as *G(.)*. A flowchart that represents the operation of the VFDT algorithm is shown in Figure 1.

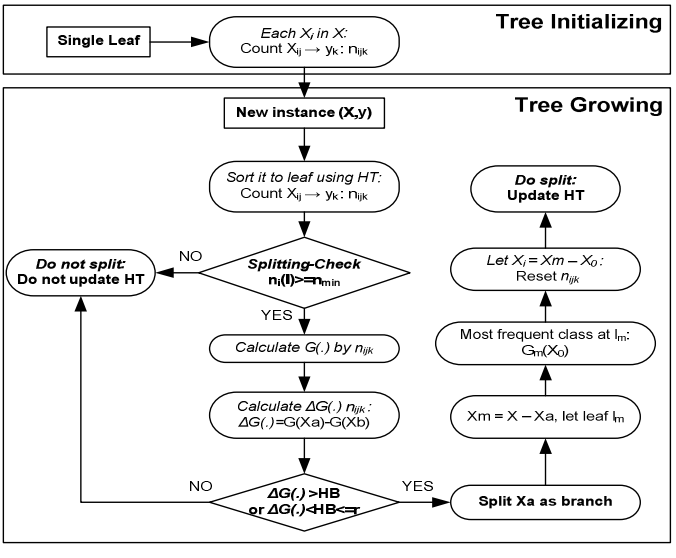

Fig. 1. A workflow representing the VFDT algorithm tree building process

The HB in Equation (1) is used by the necessary number of samples (sample#) to ensure control over error in attribute splitting distribution selection. For *n* independent observations of a real-valued random variable *r* whose range is *R*, the HB illustrates that with confidence level 1-δ, the true mean of *r* is at least $\bar{r} - \varepsilon$, where $\bar{r}$ is the observed mean of samples. For a probability the range *R* is 1, and for an information gain the range *R* is $\log_2$Class#.

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

VFDT makes use of the HB to choose a split attribute as a decision node. Let x_a be the attribute with the highest *G(.)*, x_b be the attribute with the second highest *G(.)*. $\Delta\overline{G} = \overline{G}(x_a) - \overline{G}(x_b)$ is the difference between the two top quality attributes. If

$\Delta\overline{G} > \varepsilon$ with N samples observed in leaf, while the HB states with probability $1\text{-}\delta$ that x_a is the attribute with highest value in $G(.)$, then the leaf is converted into a decision node which splits on x_a. However, in some cases the highest and the second highest $G(.)$ do not differ greatly, and so the process gets stuck in a tie condition. Resolving the tie in detail may slow down the VFDT operation. A user pre-defined threshold τ is thus used as an additional splitting condition so that if ΔG is below τ, a split will be enforced and it quickly breaks the tie.

2.2 Effects of Tie Breaking in Hoeffding Trees

When two candidates of nodes competing to become a splitting node are equally good (having almost the same value of information gain), it may take a long time and intensive computation to decide between them. This situation not only drains significant amounts of computational resources, but the tie-breaking result at the end might not always contribute substantially to the overall accuracy of the decision tree model. To alleviate this, the research team [3], introduced a tie breaking parameter τ. This tie threshold is added as an additional splitting condition in VFDT, so that whenever the HB becomes so small that the difference between the best and the second best splitting attributions is not obvious, τ comes in as a quick deceive parameter to resolve the tie. Using less than τ as a comparing condition, with the value of τ arbitrarily chosen and fixed throughout the operation, the candidate node is chosen to be split on the current best attribute, regardless of how close the second best candidate splitting attribute might be. The percentage of the condition being broken is related to the complexity of the problem. It is said that an excessive invocation of tie-breaking significantly reduces VFDT performance on complex and noise data [6], even with the additional condition by the parameter τ.

Their proposed solution [6] to overcome this detrimental effect is an improved tie - breaking mechanism, which not only considers the best and the second best splitting candidates in terms of heuristic function, but also uses the worst candidate. At the same time, an extra parameter is imported, α, which determines how many times smaller the gap should be before it is considered as a tie. The attribute splitting condition becomes: when $\alpha \times (G(X_a) - G(X_b)) < (G(X_b) - G(X_c))$, the attribution X_a shall be split as a node, instead of the original one shown in Figure 1. Obviously, this approach uses two extra parameters, α and X_c, which bring extra computation to the original algorithm. In this paper, we propose an alternative design of a tie threshold parameter that is adaptive and is calculated directly from the mean of the HB, which is found to be proportionally related to the input stream samples.

2.3 Detrimental Effect of Noise in Data Stream

Noise data is considered a type of irrelevant or meaningless data that does not typically reflect the main trends but makes the identification of these trends more difficult. Non-informative variables may be potentially random noise in the data stream. It is an idealized but useful model, in which such noise variables present no information-bearing pattern of regular variation. However, data stream mining cannot eliminate those non-informative candidates in preprocessing before starting

classification mining, because concept drift may bring the non-informative noise variables into informative candidates. Our experiment on VFDT reenacts this phenomenon in Figure 2. Evidently, the inclusion of noise data reduces the accuracy of VFDT as well as increasing tree size. This consequence is undesirable in decision tree classification. There has been an attempt to reduce the effect of noise by using supplementary classifiers for predicting missing values in real-time and minimizing noise in the important attributes [7]. Such methods still demand extra resources in computation.

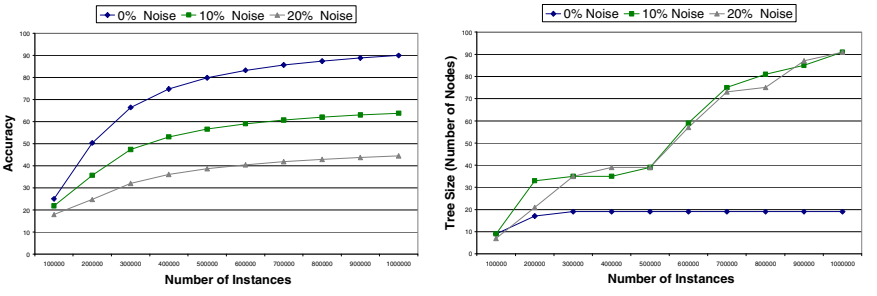

Fig. 2. This experiment demonstrates the detrimental effect of noise data in data stream. Experimental dataset is synthetic one-million samples LED dataset, which contains 24 nominal attributes and one million sample records. VFDT settings – split confidence $\delta = 10^{-7}$, tie threshold $\tau = 0.05$ (small value for smaller tree size), grace period $n_{min} = 200$; the split criterion is information gain.

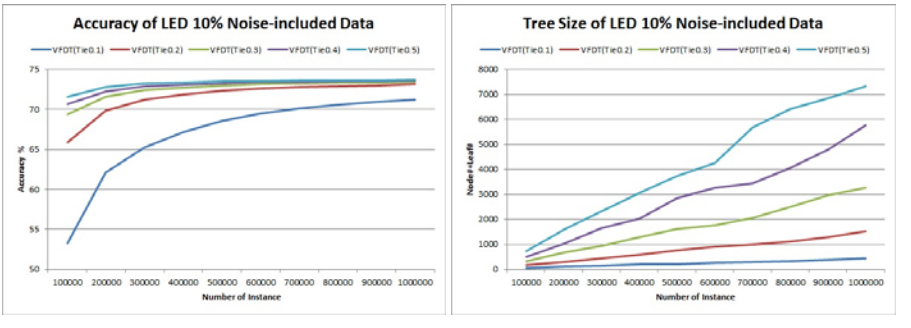

Fig. 3. Influence of tie threshold τ to VFDT. The setup is same as that in Figure 2, except for the selection of τ. The value of τ varies from 0.1 to 0.5.

The experimental results in Figure 3 show the influence of different values of τ to VFDT accuracy and tree size. The high value of τ gives rise to loose (relaxed) attribute splitting conditions, whereby the tree size becomes large. As the tree size grows, more rules are generated and the classification conditions become refined, and a better VFDT accuracy is, therefore, obtained. However, τ is a user predefined value. We are unable to know in advance which value of τ is the best, until all the combinations are tried by means of trial and error. To the best of the authors'

knowledge, no in-depth study has yet been conducted on how to find an optimum solution amongst a suitable value of τ, tree size and accuracy in VFDTs. This problem reduces the applicability of VFDT to real-time applications.

3 Moderated Very Fast Decision Tree (M-VFDT)

A new model called Moderated VFDT (M-VFDT) is proposed. It embraces data with noise with two additional techniques called adaptive tie threshold and incremental pruning to control tree size and improve accuracy.

3.1 Observation of Hoeffding Bound Fluctuation

The new technique, namely the Adaptive Tie Threshold, is based on the observation of Hoeffding bound fluctuation. The Hoeffding bound (inequality), or Chernoff bound to use its alternative name, is widely known as an important probabilistic bound for achieving good accuracy of a decision tree in stream mining. In particular, the HB) is used in deciding the attribute on which to split. A splitting attribute appears when tree structure update conditions meet and the corresponding HB is computed according to Equation 1. In terms of the accumulated HB values, the mean and the variance are recorded respectively. Under the noise data stream, it is found that HB values and variances fluctuate within a range of maximum and minimum values. The fluctuation intensifies with the increase of noise. As shown in the group of sub-graphs in Figure 4, the HB values and variances are spread out in groups along the y-axis. Under a noise-free environment, the contrasting HB values and variances differ very little (Fig. 4a).

This phenomenon strongly implies that a steady HB is desirable even though it receives heavy noise data in the construction of a decision tree. In other words, if we can keep a tight hold of the HB fluctuation, the resulting decision tree could be relieved from the ill effects of data noise, at least to certain extent. The mathematical property of HB is defined as a conservative function and has been used classically in Hoeffding tree induction for many years. (HB formulation is simple and works well in stream mining; it depends on the desired confidence and the number of observations.) We were inspired to modify the node splitting function, based on the mean of HB, instead of modifying the HB formulation. Holding on to the mean of HB is equivalent to avoiding the fluctuation of HB values, thereby reducing the noise effects. Table 1 shows the HB changing with different noise percentages. Clearly, a noise-free data stream produces the lowest HB mean and variance during the attribute splitting process. The distributions of the changing HB are represented in Figure 4 in the different settings of noise levels.

Table 1. HB values varying in VFDT (tie0.05) attribute splitting in LED dataset

Noise %	Min. HB	Max. HB	HB Mean	HB Variance
0	0.049713	0.666833	0.084249	0.003667
5	0.049862	0.666833	0.102919	0.005114
10	0.049861	0.666833	0.101125	0.004882
15	0.04986	0.666833	0.108844	0.006011
20	0.049872	0.666833	0.103495	0.005086

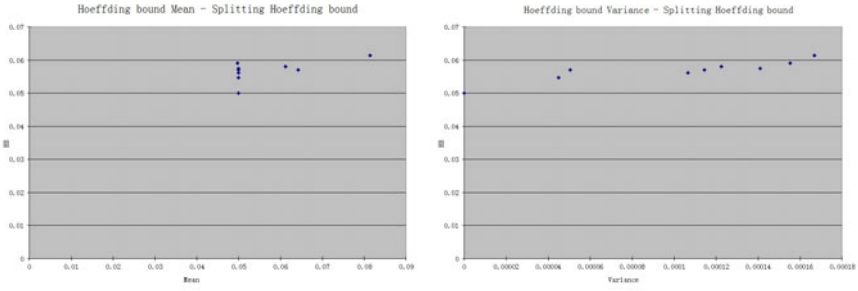

Fig. 4.a. HB distribution in LED dataset NP=0

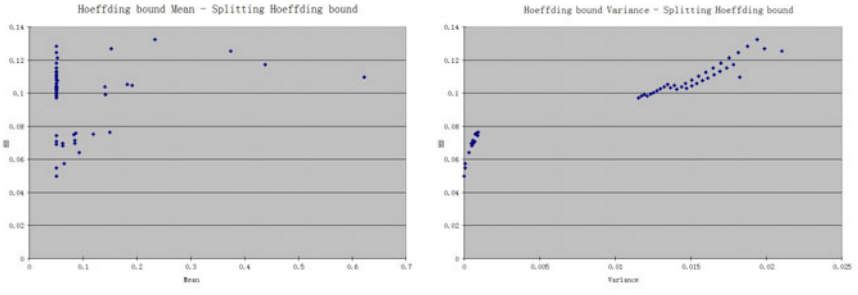

Fig. 4.b. HB distribution in LED dataset NP=5

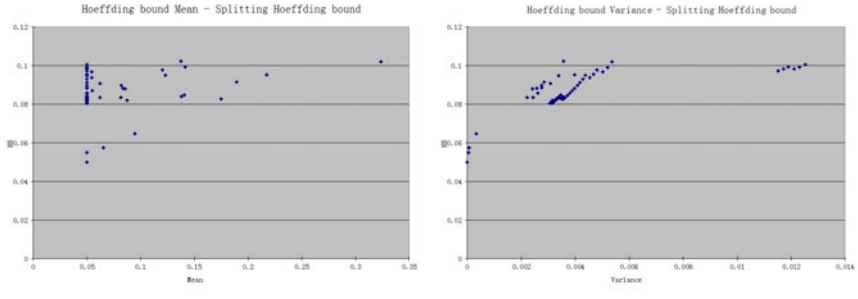

Fig. 4.c. HB distribution in LED dataset NP=10

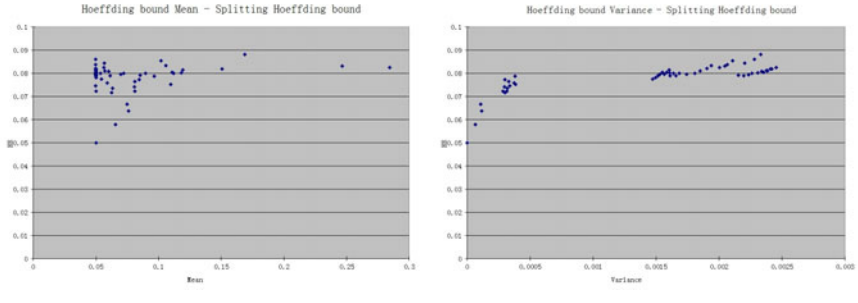

Fig. 4.d. HB distribution in LED dataset NP=15

Fig. 4. Distribution charts of different noise-included datasets in VFDT (tie=0.05), comparing the Hoeffding bound to mean and variance

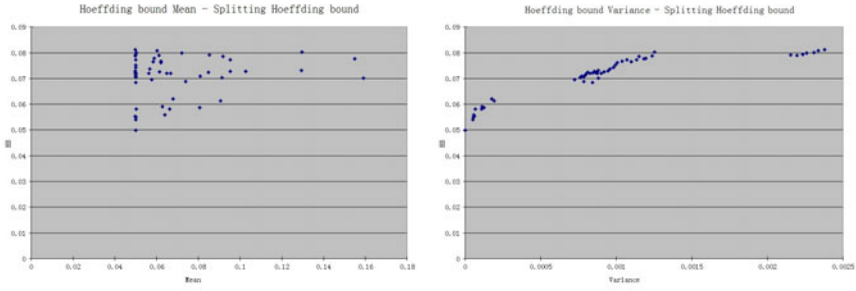

Fig 4.e. HB distribution in LED dataset NP=20

Fig. 4. (*continued*)

3.2 Adaptive Splitting Tie Threshold

As noted in Section 3.1, HB fluctuation intensifies with the increase of noise, which has a detrimental effect on VFDT accuracy. To solve this problem, we modify the attribute splitting process by using a dynamic tie threshold τ that restricts the attribute splitting as a decision node. Traditionally, τ is a user pre-configured parameter with a default value in VFDT. We are not able to know which value of τ is the best until all possibilities in an experiment are tried by brute force. Longitudinal testing on different values in advance is certainly not favorable in real-time applications. Instead, we assign an adaptive tie threshold, equal to the dynamic mean of HB as the splitting tie threshold, which controls the node splitting during the tree building process. Tie breaking that occurs near the HB mean can effectively narrow the variance distribution. The HB mean is calculated dynamically whenever new data arrives and HB is updated. It consumes few extra resources as HB would have to be computed in any case, as shown in Equation 2. When a new splitting method is implemented, τ is updated corresponding to the Hoeffding bound mean value.

$$\tau_k = \frac{\sum_{i=1}^{k} HB_i}{k} \Rightarrow \tau_{k+1} = \frac{(\sum_{i=1}^{k} HB_i) + HB_{k+1}}{k+1} \tag{2}$$

The new τ is updated when HB is computed each time with the incoming data. With this new method in place, M-VFDT has a dynamic τ whose value is no longer fixed by a single default number but adapts to the arrival instances and HB means. The M-VFDT operation with an adaptive τ is presented in Figure 5. The tree initializing process is the same as the original VFDT shown in Figure 1. The main modification is in the tree building process as follows:

- *Count(l)*: sufficient count of splitting-check of examples seen at leaf *l*
- *HBMean(l)*: dynamic mean of HB in splitting of examples seen at leaf *l*
- *HBSum(l)*: incremental statistic sum of HB in splitting of examples seen at leaf *l*
- *Prune*: the pruning mechanism. Default value is Null, which means un-pruning

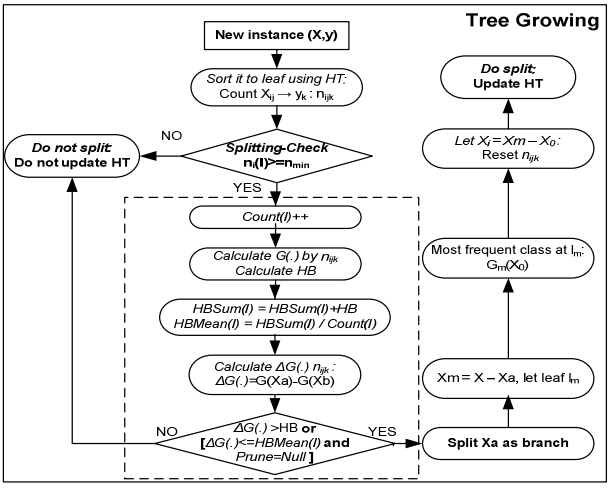

Fig. 5. M-VFDT algorithm. The adaptive tie threshold is computed in the dotted box.

3.3 Pruning Mechanisms

Pruning is an important part of decision tree learning that reduces the tree size by removing sections of the tree which provide little power to classify instances. It helps to reduce the complexity of a tree model. The pruned sections are often the fallout of noisy or erroneous data. By using an adaptive tie threshold for controlling the splitting condition, the accuracy of VFDT has significantly improved, as shown in Section 4. However, high accuracy comes with a large increase in tree size. To rectify the tree size problem, we propose two pruning approaches, the strict pruning mechanism and the loose pruning mechanism, each of which reflects a strong and weak pruning strength, respectively.

VFDT collects sufficient statistics to compute the number of instance counts by filtering the new instance to a leaf by the current tree model. When the splitting condition is satisfied, the attribute splitting approach simply proceeds at various points of the tree construction algorithm, without re-scanning the data stream.

In our pruning mechanisms, a leverage variable is used to identify the observations which have a great effect on the outcome of fitting tree models. The leverage point is set when a new instance is entering into a leaf, according to the current tree building model. It is not absolutely true that all unseen instances can fall into the leaves of the current tree. Suppose the count of unseen instances number falling into an existing leaf is called a *PseudoCount*, and the count of unseen instances number not falling into a current leaf is *RealCount*. We therefore let *Leverage* be the difference of *RealCount* minus *PseudoCount*. *Leverage* can be a negative, zero or positive number. It is calculated by: *Leverage* = *RealCount* – *PseudoCount*.

Mechanism 1: Strict Pruning. This pruning mechanism works with dynamic mean splitting condition and it is incremental in nature. It keeps a very small tree size by sacrificing some accuracy. The pruning condition is simply: $\underline{Leverage} \leq 0$.

Strict pruning only considers the horizontal comparison of attribute splitting with respect to the current leverage point. It imposes a strict breaking criterion so that the pseudo-count falls beyond the true count during the whole tree building process. In additional to horizontal comparison, we suggest a vertical comparison of current Leverage and the last Leverage estimated at the previous cycle of splitting process. The difference of current Leverage and last Leverage is defined as *DeltaLeverage*, where *DeltaLeverage* = *Leverage* – *LastLeverage*.

Mechanism 2: Loose Pruning. This pruning mechanism encompasses strict pruning, adding an optional splitting condition of *DeltaLeveage*. The pruning condition is: $\underline{Leverage \leq 0}$ OR $\underline{DeltaLeverage \leq 0}$.

The reasons why these two pruning mechanisms are chosen:

(1) The tree building process of VFDT uses sufficient statistics that count the number of instances filtered to a leaf on the current Hoeffding tree (HT). The splitting is based on these counts. The filtering process can easily compute the number of real counts and pseudo-counts without much extra effort. In strict pruning, if the real count becomes smaller than the pseudo-count, it means the performance of the current HT is not so adaptive to the new arrival instances (noise induced tree branches start building up). It is a strict condition that it shall be pruned during tree building, if the *Leverage* is lower than zero.

(2) In the loose pruning mechanism, *DeltaLeverage* is used as a pre-pruning condition that compares the current Leverage with its previous one. It examines the trend of the built tree's performance in the nearest two splitting processes. If the current Leverage is smaller than its previous one, it means the current HT's performance is declining. In this case, the tree should be pruned by an extra splitting node condition where *i* is the current step of the HT building process.

4 Experiments

In this section, a variety of large data streams, with nominal and numeric attributes, are used to stress test our proposed model. The M-VFDT with adaptive tie threshold and pruning shows consistently better performance than the VFDT that uses a fixed default tie threshold. The datasets are both synthetic from a stream generator and obtained from live data of real world applications. The characteristics of the experimental datasets are given in Table 2. With the same experiment settings as those in Figure 3, we estimate the best tie threshold value to be used for VFDT as a base comparison to our model by trying different tie threshold values from 0.1 to 0.9.

Table 2. Characteristics of the experimental datasets

Dataset	Description	Type	Attr.#	Class#	Ins#	Best Tie
LedNP10	LED display [4]	Nominal	24	10	1.0×10^6	0.7
LedNP20	LED display [4]	Nominal	24	10	1.0×10^6	0.4
Wave	Waveform [4]	Numeric	22	3	1.0×10^6	0.3
Connect-4	UCI Data [5]	Nominal	42	3	67,557	0.6
Nursery	UCI Data [5]	Nominal	9	5	12,960	0.3

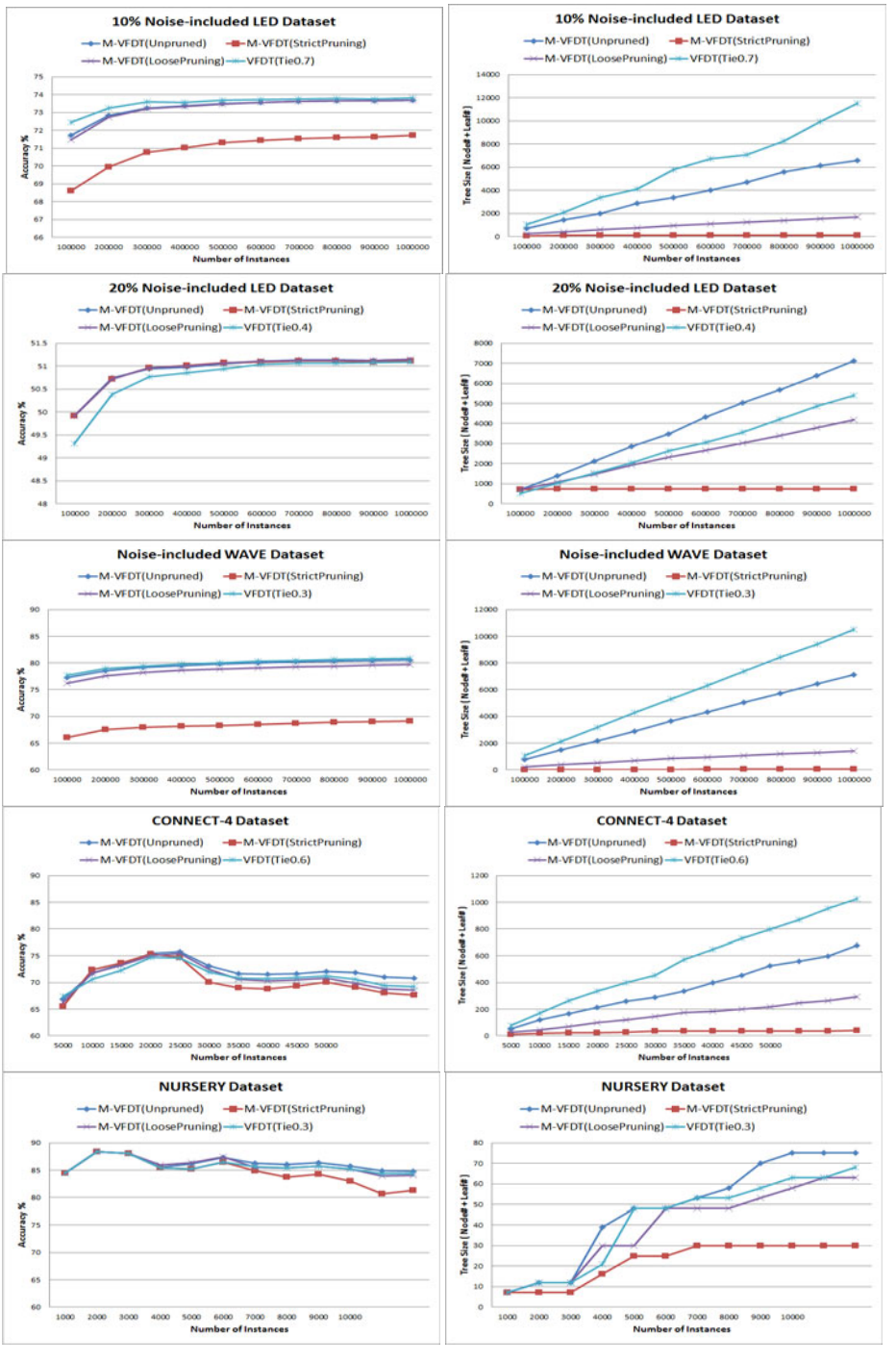

Fig. 6. Accuracy and tree size comparison of M-VFDT and VFDT in different datasets

The results in Figure 6 validate the accuracy and the tree size comparison of these two pruning mechanisms on the same experimental dataset. It compares to the original VFDT with a stationary tie threshold whose value is found to be the best by the brute-force method earlier on. In general, we observe that M-VFDT with strict pruning keeps the tree size smallest, but the accuracy is worse than that of others. The loose pruning method for M-VFDT yields reasonable accuracy that is on par with VFDT, but its tree size is more compact than that of VFDT, although it is still larger than the tree by strict pruning. Both strict and loose pruning methods in M-VFDT make compromises between accuracy and tree size. This can be explained by the fact that a small tree classifies instances coarsely because of the relatively small paths of tree branches, and so accuracy is adversely affected. A classifier that can perform precisely in classification usually requires a bushy tree with many conditional rules.

Our current research focus is on how to choose an optimal between the strict and loose pruning. Strict pruning has its merits in achieving a small tree size, from which concise classification rules can be derived and, generally, they are easily readable. In contrast, the loose pruning mechanism, although resulting in a bigger tree size, achieves much higher accuracy than that of the strict pruning mechanism. A significant contribution of M-VFDT is that loose pruning achieves an almost similar level of accuracy as the VFDT with the best chosen but stationary value of tie-breaking, and M-VFDT results in a much smaller tree size.

5 Conclusion and Future Work

We have proposed an improved decision tree algorithm for data stream mining, which is called Moderated VFDT or simply M-VFDT. M-VFDT embraces an adaptive tie threshold for deciding on splitting nodes whose value is calculated dynamically from the mean of the Hoeffding bound. Tie threshold is an important parameter as advocated by [6] in speeding up the construction of the decision tree in stream mining, however it was assumed to be a stationary default value in most other research works in studying VFDT. With the adaptive tie threshold, the accuracy of M-VFDT has greatly improved in comparison to the original VFDT. The performance improvement by M-VFDT is shown to be more apparent when the data streams are infested by noise. This work is important because noise in input data is already known to cause very adverse effects both on the accuracy degradation and tree size explosion. In addition, we proposed two pre-pruning mechanisms for M-VFDT to reduce the tree size and make the classification model compact. Two types of pruning, namely strict and loose pruning are proposed; they are both incremental in nature (as we know that, in stream mining, post-pruning may not be favorable because the tree is continually being updated as data streams in). All of these extra mechanisms, including incremental pruning and adaptive tie threshold, are lightweight in computation. This makes M-VFDT suitable to a stream mining environment, where speed, accuracy and tree size, as it relates to memory constraint, are of concern.

In the future, we intend to adopt M-VFDT as a core enabling model in different case studies of real-time stream mining applications. We also want to extend the concepts of adaptive tie threshold and incremental pruning to other variants of VFDT.

Furthermore, it will be interesting to find a formula for automatically estimating an optimal balance between tree size and accuracy by using some optimization theories.

References

1. Maron, O., Moore, A.W. Hoeffding races: Accelerating Model Selection Search for Classification and Function Approximation. In: NIPS, pp. 59–66 (1993)
2. Domingos, P., Hulten, G.: Mining high-speed data streams. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 71–80 (2000)
3. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD 2001, pp. 97–106. ACM, New York (2001)
4. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth International Group, Belmont (1984)
5. Frank, A., Asuncion, A.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA (2010), http://archive.ics.uci.edu/ml
6. Geoffrey, H., Richard, K., Bernhard, P.: Tie Breaking in Hoeffding trees. In: Gama, J., Aguilar-Ruiz, J.S. (eds.) Proceeding Workshop W6: Second International Workshop on Knowledge Discovery in Data Streams, pp. 107–116 (2005)
7. Yang, H., Fong, S.: Aerial Root Classifiers for Predicting Missing Values in Data Stream Decision Tree Classification. In: 2011 SIAM International Conference on Data Mining (SDM 2011), Mesa, Arizona, USA, April 28-30 (2011) (accepted for Publication)