Splitting with Confidence in Decision Trees with Application to Stream Mining

Rocco De Rosa
Dipartimento di Informatica
Università degli Studi di Milano

Nicolò Cesa-Bianchi
Dipartimento di Informatica
Università degli Studi di Milano

Abstract—**Decision tree classifiers are a widely used tool in data stream mining. The use of confidence intervals to estimate the gain associated with each split leads to very effective methods, like the popular Hoeffding tree algorithm. From a statistical viewpoint, the analysis of decision tree classifiers in a streaming setting requires knowing when enough new information has been collected to justify splitting a leaf. Although some of the issues in the statistical analysis of Hoeffding trees have been already clarified, a general and rigorous study of confidence intervals for splitting criteria is missing. We fill this gap by deriving accurate confidence intervals to estimate the splitting gain in decision tree learning with respect to three criteria: entropy, Gini index, and a third index proposed by Kearns and Mansour. Our confidence intervals depend in a more detailed way on the tree parameters. Experiments on real and synthetic data in a streaming setting show that our trees are indeed more accurate than trees with the same number of leaves generated by other techniques.**

Keywords: incremental decision tree learning, stream mining, split confidence.

I. Introduction

Stream mining algorithms are becoming increasingly attractive due to the large number of applications generating large-volume data streams. These include: email, chats, click data, search queries, shopping history, user browsing patterns, financial transactions, electricity consumption, traffic records, telephony data, and so on. In these domains, data are generated sequentially, and scalable predictive analysis methods must be able to process new data in a fully incremental fashion. Decision trees classifiers are one of the most widespread non-parametric classification methods. They are fast to evaluate and can naturally deal with mixed-type attributes; moreover, decision surfaces represented by small trees are fairly easy to interpret. Decision trees have been often applied to stream mining tasks —see, e.g., the survey [13]. In such settings, the tree growth is motivated by the need of fitting the information brought by the newly observed examples. Starting from the pioneering work in [28], the incremental learning of decision trees has received a lot of attention in the past 25 years. Several papers build on the idea of [21], which advocates the use of measures to evaluate the confidence in choosing a split. These works include Sequential ID3 [10], VFDT [5], NIP-H and NIP-N [14]. Sequential ID3 uses a sequential probability ratio test in order to minimize the number of examples sufficient to choose a good split. This approach guarantees that the tree learned incrementally is close to the one learned via standard batch learning. A similar yet stronger guarantee is achieved by the Hoeffding tree algorithm, which is at the core of the state-of-the-art VFDT system. Alternative approaches, such as NIP-H e NIP-N, use Gaussian approximations instead of Hoeffding bounds in order to compute confidence intervals. Several extensions of VFDT have been proposed, also taking into account non-stationary data sources —see, e.g., [8], [7], [2], [30], [24], [12], [16], [18], [9], [29], [17], [26], [6]. All these methods are based on the classical Hoeffding bound [11]: after m independent observations of a random variable taking values in a real interval of size R, with probability at least $1-\delta$ the true mean does not differ from the sample mean by more than

$$\varepsilon_{\mathrm{hof}}(m,\delta) = R\sqrt{\frac{1}{2m}\ln\frac{1}{\delta}}\ . \tag{1}$$

The problem of computing the confidence interval for the split gain estimate can be phrased as follows: we are given a set of unknown numbers G (i.e., the true gains for the available splits) and want to find the largest of them. We do that by designing a sample-based estimator $\widehat{G}$ of each G, and then use an appropriate version of the Hoeffding bound to control the probability that $\bigl|\widehat{G}-G\bigr| > \varepsilon$ for any given $\varepsilon > 0$. It is easy to see that this allows to pick the best split at any given node: assume that $\widehat{G}_F$ is the highest empirical gain (achieved by the split function F) and $\widehat{G}_{F_2}$ is the second-best (achieved by the split function F_2). If $\widehat{G}_F - \widehat{G}_{F_2} > 2\varepsilon$ then with probability at least $1-\delta$ the split function F is optimal[1] —see Fig. 1. Although all methods in the abovementioned literature use the

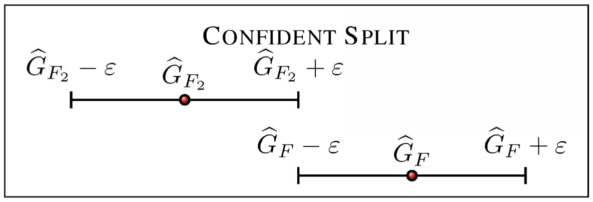
CONFIDENT SPLIT
$\widehat{G}_{F_2}-\varepsilon$
$\widehat{G}_{F_2}$
$\widehat{G}_{F_2}+\varepsilon$
$\widehat{G}_{F}-\varepsilon$
$\widehat{G}_{F}$
$\widehat{G}_{F}+\varepsilon$

Fig. 1. The condition $\widehat{G}_F - \widehat{G}_{F_2} > 2\varepsilon$ guarantees that the confidence intervals for the true gains G_F and G_{F_2} are non-overlapping.

Hoeffding bound (1) to compute the confidence intervals for the splits, we show here that the standard entropy-like criteria require a different approach.

The rest of the paper is organized as follows. Section II discusses related work. In Section III we state the basic decision tree learning concepts and introduce the notation used in the rest of the paper. In Section IV we derive the new bounds for the splitting criteria. In Section V we apply the confidence bounds to the incremental learning of a decision tree and

[1]In the original work VFDT [5] $\widehat{G}_F - \widehat{G}_{F_2} > \varepsilon$ is erroneously used.

978-1-4799-1959-8/15/\$31.00 ©2015 IEEE

derive a formal guarantee (Theorem 4) on the probability that examples in the stream are classified using suboptimal splits based on any of the three splitting criteria. These theoretical guidelines are empirically tested in Section VI, where we show that our more refined bounds deliver better splits that the splits performed by the other techniques.

II. Related work

The problem of computing the confidence interval for the splitting gain estimate has two main source of difficulties: First, splitting criteria —like entropy or Gini index— are nonlinear functions of the distribution at each node. Hence, appropriate large deviation bounds (such as the McDiarmid bound [20]) must be used. As the McDiarmid bound controls the *deviations* $\big|\widehat{G}-\mathbb{E}\widehat{G}\big|$, further work is needed to control the *bias* $\big|\mathbb{E}\widehat{G}-G\big|$. The first problem was solved (for entropy and Gini) in [25]. The authors used McDiarmid bound to derive estimates of confidence intervals for various split measures. For instance, in a problem with K classes, the bound on the confidence interval for the entropy gain criterion is

$$\varepsilon_{\mathrm{md}}(m,\delta) = C(K,m)\sqrt{\frac{1}{2m}\ln\frac{1}{\delta}} \qquad (2)$$

where $C(K,m) = 6\big(K\log_2 e + \log_2 2m\big) + 2\log_2 K$. The authors proposed to replace Hoeffding bound (1) by McDiarmid bound (2) in the VFDT algorithm and its successors. However, although this allows to control the deviations, the bias of the estimate is ignored. More recently, in [6] the same authors apply the Hoeffding bound to the entropy splitting criterion, focusing on binary trees and binary classification. They decompose the entropy gain calculation in three components, and apply the Hoeffding bound to each one of them, obtaining a confidence interval estimate for the splitting gains. However, this still ignores the bias of the estimate and, besides, the authors do not consider other types of split functions. The work [19] directly uses the classification error as splitting criterion rather than a concave approximation of it (like the entropy or the Gini index). Though this splitting criterion can be easily analyzed via the Hoeffding bound, its empirical performance is generally not very good —see Section IV for more discussion on this.

In this work, we significantly simplify the approach of [25] and extend it to a third splitting criterion. Moreover, we also solve the bias problem, controlling the deviations of $\widehat{G}$ from the real quantity of interest (i.e., G rather than $\mathbb{E}\widehat{G}$). Moreover, unlike [19] and [6], our bounds apply to the standard splitting criteria. Our analysis shows that the confidence intervals associated with the choice of a suboptimal split not only depend on the number of leaf examples m —as in bounds (1) and (2)— but also on other problem dependent parameters, as the dimension of the feature space, the depth of the leaves, and the overall number of examples seen so far by the algorithm. As revealed by the experiments in Section VI-A, this allows a more cautious and accurate splitting in complex problems. Furthermore, we point out that our technique can be easily applied to all extensions of VFDT (see Section I) yielding similar improvements, as these extensions all share the same Hoeffding-based confidence analysis as the Hoeffding tree algorithm.

III. Batch Decision Tree Learning

For simplicity we only consider binary classification problems. The goal is to find a function $f(\boldsymbol{X})$ assigning the correct category $Y = \{0,1\}$ to a new instance $\boldsymbol{X}$. We consider binary decision trees based on a class $\mathcal{F}$ of binary split functions $F : \mathbb{R}^d \to \{0,1\}$, that is, the test functions through which the feature space is partitioned[2]. Training examples $(\boldsymbol{X}_1,Y_1),(\boldsymbol{X}_2,Y_2),\ldots \in \mathbb{R}^d \times \{0,1\}$ are i.i.d. draws from a fixed but unknown probability distribution. Decision tree classifiers are typically constructed in an incremental way, starting from a tree consisting of a single node. The tree grows through a sequence of splitting operations applied to its leaves. If a split is decided for a leaf i, then the leaf is assigned some split function $F \in \mathcal{F}$ and two nodes i_0 and i_1 are added to the tree as children of the split node. Examples are recursively routed throught the tree starting from the root as follows: when an example $(\boldsymbol{X}_t,Y_t)$ reaches an internal node i with split function F, then it is routed to child i_0 if $F(\boldsymbol{X}_t) = 0$ and to child i_1 otherwise. A decision tree T induces a classifier $f_T : \mathbb{R}^d \to \{0,1\}$. The prediction $f_T(\boldsymbol{X})$ of this classifier on an instance $\boldsymbol{X}$ is computed by routing the instance $\boldsymbol{X}$ through the tree until a leaf is reached. We use $\boldsymbol{X} \to i$ to indicate that $\boldsymbol{X}$ is routed to the leaf i. Then $f_T(\boldsymbol{X})$ is set to the most frequent label $y = \arg\max_y \mathbb{P}(Y = y|\boldsymbol{X} \to i)$ among the labels of all observed examples that reach that leaf. The goal of the learning process is to control the binary classification risk $\mathbb{P}\big(f_T(\boldsymbol{X}) \neq Y\big)$ of f_T. For any leaf i, let $Y_{|i} = \mathbb{P}(Y = 1|\boldsymbol{X} \to i)$ be the random variable denoting the label of a random instance $\boldsymbol{X}$ given that $\boldsymbol{X} \to i$. Let $\mathcal{L}(T)$ be the leaves of T. The risk of f_T can then be upper bounded, with the standard bias-variance decomposition, as follows

$$\begin{aligned}&\mathbb{P}\big(f_T(\boldsymbol{X}) \neq Y\big)\\ &\le \overbrace{\sum_{i\in\mathcal{L}(T)} \mathbb{P}(Y \neq y_i^* \mid \boldsymbol{X} \to i)\mathbb{P}(\boldsymbol{X} \to i)}^{\text{bias error}}\\ &+ \overbrace{\sum_{i\in\mathcal{L}(T)} \mathbb{P}\big(f_T(\boldsymbol{X}) \neq y_i^* \mid \boldsymbol{X} \to i\big)\mathbb{P}(\boldsymbol{X} \to i)}^{\text{variance error}}\end{aligned}$$

where $y_i^* = \mathbb{I}\big\{\mathbb{P}(Y = 1 \mid \boldsymbol{X} \to i) \ge \frac{1}{2}\big\}$ is the optimal label[3] for leaf i and $\mathbb{I}\{\cdot\}$ is the indicator function of the event at argument. The *variance* terms are the easiest to control: $f_T(\boldsymbol{X})$ is determined by the most frequent label of the leaf i such that $\boldsymbol{X} \to i$. Hence, conditioned on $\boldsymbol{X} \to i$, the event $f_T(\boldsymbol{X}) = y_i^*$ holds with high probability whenever the confidence interval for the estimate of y_i^* does not cross the $\frac{1}{2}$ boundary[4]. The bias terms compute the Bayes error at each leaf. The error vanishes quickly when good splits for expanding the leaves are available. However, due to the large number of available split functions F, the confidence intervals for choosing such good splits shrink slower than the confidence interval associated with the bias error. Our Theorem 4 accurately quantifies the dependence of the split

[2]Although we only considered binary classification and binary splits, our techniques can be potentially extended to multi-class classification and general splits.

[3]The label assigned by the Bayes optimal classifier in the leaf partition.

[4]This confidence interval shrinks relatively fast, as dictated by Hoeffding bound applied to the variable $y_i^* \in \{0,1\}$.

confidence on the various problem parameters. Let $\Psi(Y)$ be a shorthand for $\min\{\mathbb{P}(Y=0), \mathbb{P}(Y=1)\}$. Every time a leaf i is split using F, the term $\Psi(Y_{|i})$ gets replaced by

$$\mathbb{P}\bigl(F=0 \mid \boldsymbol{X}\to i\bigr)\Psi\bigl(Y_{|i} \mid F=0\bigr) + \mathbb{P}\bigl(F=1 \mid \boldsymbol{X}\to i\bigr)\Psi\bigl(Y_{|i} \mid F=1\bigr)$$

corresponding to the newly added leaves (here and in what follows, F also stands for the random variable $F(\boldsymbol{X})$). The concavity of min ensures that no split of a leaf can ever make that sum bigger. Of course, we seek the split maximizing the risk decrease (or "gain"),

$$\Psi(Y_{|i}) - \mathbb{P}\bigl(F=0 \mid \boldsymbol{X}\to i\bigr)\Psi\bigl(Y_{|i} \mid F=0\bigr) - \mathbb{P}\bigl(F=1 \mid \boldsymbol{X}\to i\bigr)\Psi\bigl(Y_{|i} \mid F=1\bigr) \; .$$

In practice, splits are chosen so to approximately maximize a gain functional defined in terms of a concave and symmetric function Φ, which bounds from the above the min function Ψ (used in [19] as splitting criterion). The curvature of Φ helps when comparing different splits, as opposed to Ψ which is piecewise linear. Indeed Ψ gives nonzero gain only to splits generating leaves with disagreeing majority labels —see, e.g., [4] for a more detailed explanation. Let Z be a Bernoulli random variable with parameter p. Three gain functions used in practice are: the scaled binary entropy $H_{1/2}(Z) = -\frac{p}{2}\ln p - \frac{1-p}{2}\ln(1-p)$ used in C4.5; the Gini index $J(Z) = 2p(1-p)$ used in CART, and the function $Q(Z) = \sqrt{p(1-p)}$ introduced by Kearns and Mansour in [15] and empirically tested in [4]. Clearly, the binary classification risk can be upper bounded in terms of any upper bound Φ on the min function Ψ,

$$\mathbb{P}\bigl(f_T(\boldsymbol{X}) \neq Y\bigr) \le \sum_{i\in\mathcal{L}(T)} \Phi(Y_{|i})\mathbb{P}(\boldsymbol{X}\to i) + \sum_{i\in\mathcal{L}(T)} \mathbb{P}\bigl(f_T(\boldsymbol{X}) \neq y_i^* \mid \boldsymbol{X}\to i\bigr)\mathbb{P}(\boldsymbol{X}\to i) \; .$$

The gain for a split F at node i, written in terms of a generic entropy-like function Φ, takes the form

$$\begin{aligned} G_{i,F} &= \Phi(Y_{|i}) - \Phi\bigl(Y_{|i} \mid F\bigr) \\ &= \Phi(Y_{|i}) - \mathbb{P}\bigl(F=0 \mid \boldsymbol{X}\to i\bigr)\Phi(Y_{|i} \mid F=0) \\ &\quad - \mathbb{P}(F=1 \mid \boldsymbol{X}\to i)\Phi(Y_{|i} \mid F=1) \; . \end{aligned} \tag{3}$$

Now, in order to choose splits with a high gain (implying a significant reduction of risk), we must show that $G_{i,F}$ (for the different choices of Φ) can be reliably estimated from the training examples. In this work we focus on estimates for choosing the best split F at any given leaf i. Since the term $\Phi(Y_{|i})$ in $G_{i,F}$ is invariant with respect to this choice, we may just ignore it when estimating the gain.[5]

IV. Confidence Bound For Split Functions

In this section we compute estimates $\widehat{\Phi}_{i|F}$ of $\Phi(Y_{|i} \mid F)$ for different choices of Φ, and compute confidence intervals for these estimates. As mentioned in Section I, we actually bound the deviations of $\widehat{\Phi}_{i|F}$ from $\Phi(Y_{|i} \mid F)$, which is the real quantity of interest here. Due to the nonlinearity of Φ, this problem is generally harder than controlling the deviations of $\widehat{\Phi}_{i|F}$ from its expectation $\mathbb{E}\,\widehat{\Phi}_{i|F}$ —see, e.g., [25] for weaker results along these lines. In the rest of this section, for each node i and split F we write $p_k = \mathbb{P}(Y=1, F=k)$ and $q_k = 1-p_k$ for $k \in \{0,1\}$; moreover, we use $\widehat{p}_k$, $\widehat{q}_k$ to denote the empirical estimates of p_k, q_k.

A. *Bound for the entropy*

Let $\Phi(Z)$ be the (scaled) binary entropy $H_{1/2}(Z) = -\frac{p}{2}\ln p - \frac{1-p}{2}\ln(1-p)$ for Z Bernoulli of parameter p. In the next result, we decompose the conditional entropy as a difference between entropies of the joint and the marginal distribution. Then, we apply standard results for plug-in estimates of entropy.

Theorem 1: Pick a node i and route m i.i.d. examples $(\boldsymbol{X}_t, Y_t)$ to i. For any $F \in \mathcal{F}$, let

$$\widehat{\Phi}_{i|F} = \widehat{H}_{1/2}(Y_{|i}, F) - \widehat{H}_{1/2}(F)$$

where $\widehat{H}_{1/2}$ denotes the empirical scaled entropy (i.e., the scaled entropy of the empirical measure defined by the i.i.d. sample). Then, for all $\delta > 0$,

$$\Bigl|\widehat{\Phi}_{i|F} - H_{1/2}(Y_{|i} \mid F)\Bigr| \le \varepsilon_{\text{ent}}(m,\delta)$$

$$\text{where} \quad \varepsilon_{\text{ent}}(m,\delta) = (\ln m)\sqrt{\frac{2}{m}\ln\frac{4}{\delta}} + \frac{2}{m} \tag{4}$$

with probability at least $1-\delta$ over the random draw of the m examples.

Proof: In appendix A. ■

B. *Bound for the Gini index*

In the Bernoulli case, the Gini index takes the simple form $J(Z) = 2p(1-p)$ for Z Bernoulli of parameter p. First we observe that $J(Y_{|i} \mid F)$ is the sum of harmonic averages, then we use the McDiarmid inequality to control the variance of the plug-in estimate for these averages.

Theorem 2: Pick a node i and route m i.i.d. examples $(\boldsymbol{X}_t, Y_t)$ to i. For any $F \in \mathcal{F}$, let

$$\widehat{\Phi}_{i|F} = \text{HM}(\widehat{p}_1, \widehat{q}_1) + \text{HM}(\widehat{p}_0, \widehat{q}_0)$$

where HM denotes the harmonic mean $\text{HM}(p,q) = \frac{2pq}{p+q}$. Then, for all $\delta > 0$

$$\Bigl|\widehat{\Phi}_{i|F} - J(Y_{|i} \mid F)\Bigr| \le \varepsilon_{\text{Gini}}(m,\delta)$$

$$\text{where} \quad \varepsilon_{\text{Gini}}(m,\delta) = \sqrt{\frac{8}{m}\ln\frac{2}{\delta}} + 4\sqrt{\frac{1}{m}} \tag{5}$$

with probability at least $1-\delta$ over the random draw of the m examples.

Proof: In appendix B. ■

[5]Note that, for all functions Φ considered in this paper, the problem of estimating $\Phi(Y_{|i})$ can be solved by applying essentially the same techniques as the ones we used to estimate $\Phi(Y_{|i} \mid F)$.

C. Bound for the Kearns-Mansour index

The third entropy-like function we analyze is $Q(Z) = \sqrt{p(1-p)}$ for Z Bernoulli of parameter p. The use of this function was motivated in [15] by a theoretical analysis of decision tree learning as a boosting procedure. See also [27] for a simplified analysis and some extensions.

In this case McDiarmid inequality is not applicable and we control $Q(Y_{|i} \mid F)$ using a direct argument based on classical large deviation results.

Theorem 3: Pick a node i and route m i.i.d. examples $(\boldsymbol{X}_t, Y_t)$ to i. For any $F \in \mathcal{F}$, let

$$\widehat{\Phi}_{i|F} = \sqrt{\widehat{p}_1\widehat{q}_1} + \sqrt{\widehat{p}_0\widehat{q}_0}\ .$$

Then, for all $\delta > 0$,

$$\left|\widehat{\Phi}_{i|F} - Q(Y_{|i} \mid F)\right| \le \varepsilon_{\mathrm{KM}}(m,\delta)$$

$$\text{where} \quad \varepsilon_{\mathrm{KM}}(m,\delta) = 4\sqrt{\frac{1}{m}\ln\frac{8}{\delta}} \tag{6}$$

with probability at least $1-\delta$ over the random draw of the m examples.

Proof: In appendix C. ■

V. Confidence Tree Algorithm

A setting in which confidence intervals for splits are extremely useful is online or stream-based learning. In this setting, examples are received incrementally, and a confidence interval can be used to decide how much data should be collected at a certain leaf before a good split F can be safely identified. A well-known example of this approach are the so-called Hoeffding trees [5]. In this section, we show how our confidence interval analysis can be used to extend and refine the current approaches to stream-based decision tree learning. For $t = 1, 2, \ldots$ we assume the training example $(\boldsymbol{X}_t, Y_t)$ is received at time t. C-Tree (Algorithm 1) describes the online

Algorithm 1 C-Tree

```
Input: Threshold $\tau > 0$
 1: Build a 1-node tree $T$
 2: for $t = 1, 2, \ldots$ do
 3:     Route example $(\boldsymbol{X}_t, Y_t)$ through $T$ until a leaf $\ell_t$ is reached
 4:     if $\ell_t$ is not pure then
 5:         Let $\widehat{F} = \operatorname{argmax}_{F \in \mathcal{F}} \widehat{\Phi}_{\ell_t,F_1}$ and $\widehat{F}_2 = \operatorname{argmax}_{F \in \mathcal{F} : F \neq \widehat{F}} \widehat{\Phi}_{\ell_t,F_1}$
 6:         if $\widehat{\Phi}_{\ell_t,\widehat{F}} \le \widehat{\Phi}_{\ell_t,\widehat{F}_2} - 2\varepsilon_t$ or $\varepsilon_t \le \tau$ then
 7:             Let $F_{\ell_t} = \widehat{F}$ and expand $\ell_t$ using split $F_{\ell_t}$
 8:         end if
 9:     end if
10: end for
```

decision tree learning approach. A stream of examples is fed to the algorithm, which initially uses a 1-node decision tree. At time t, example $(\boldsymbol{X}_t, Y_t)$ is routed to a leaf ℓ_t. If the leaf is not pure (both positive and negative examples have been routed to ℓ_t), then the empirically best $\widehat{F}$ and the second-best $\widehat{F}_2$ split for ℓ_t are computed. If the difference in gain between these two splits exceeds a value ε_t, computed via the confidence interval analysis, then the leaf is split using $\widehat{F}$. The leaf is also split when ε_t goes below a "tie-break" parameter τ, indicating that the difference between the gains of $\widehat{F}$ and $\widehat{F}_2$ is so tiny that waiting for more examples in order to find out the really best split is not worthwhile. Let $\varepsilon(m,\delta)$ be the size of the confidence interval computed via Theorem 1, 2 or 3. Fix any node i and let $m_{i,t}$ be the number of examples routed to that node in the first $t-1$ time steps. Clearly, for any $F, F' \in \mathcal{F}$ with $F \neq F'$, if $\widehat{\Phi}_{i|F} \le \widehat{\Phi}_{i|F'} - 2\varepsilon(m_{i,t},\delta)$ then $G_{i,F} \ge G_{i,F'}$ with probability at least $1-\delta$. Now, since the number of possible binary splits is at most $dm_{i,t}$, if we replace δ by $\delta/(dm_{i,t})$ the union bound guarantees that a node i is split using the function maximizing the gain.

Lemma 1: Assume a leaf i is expanded at time t only if there exists a split $\widehat{F}$ such that

$$\widehat{\Phi}_{i|\widehat{F}} \le \widehat{\Phi}_{i|F} - 2\varepsilon\big(m_{i,t}, \delta/(dm_{i,t})\big)$$

for all $F \in \mathcal{F}$ such that $F \neq \widehat{F}$. Then, $\widehat{F} = \operatorname*{argmax}_{F \in \mathcal{F}} G_{i,F}$ with probability at least $1-\delta$.

The next result provides a bound on the probability that a random example is classified using a suboptimal split. A similar result was proven in [5] for Hoeffding trees.

Theorem 4: Assume C-Tree (Algorithm 1) is run with

$$\varepsilon_t = \varepsilon\left(m_{\ell_t,t}, \frac{\delta}{(h_t+1)(h_t+2)tdm_{\ell_t,t}}\right) \tag{7}$$

where $\varepsilon(m,\delta)$ is the size of the confidence interval computed via Theorem 1, 2 or 3 and h_t is depth of ℓ_t. Then the probability that a random example $\boldsymbol{X}$ is routed via a τ-suboptimal split is at most δ.

Proof: In appendix D. ■

Remark 1: Theorem 4 controls the classification of a single random example. However, choosing $\delta = \frac{1}{t}$, and applying the union bound over the time steps, guarantees that only a logarithmic number of examples in the stream are classified via suboptimal splits.

VI. Experiments

We ran experiments on synthetic datasets and popular benchmarks, comparing our C-Tree (Algorithm 1) against two baselines: H-Tree (VDFT algorithm [5]) and CorrH-Tree (the method from [6] using the classification error as splitting criterion). The bounds of [25] are not considered because of their conservativeness. In fact, these bounds generate 1-node trees in all the experiments, even when the confidence is set to a very low value.

The three methods (ours and the two baselines) share the same core, i.e., the HoeffdingTree (H-Tree) algorithm implemented in MOA[6]. In order to implement C-tree and the

[6]moa.cms.waikato.ac.nz/

baseline CorrH-Tree, we directly modified the H-Tree code in MOA. The grace period parameter[7] was set to 100. In contrast to the typical experimental settings in the literature, we did not consider the tie-break parameter because in the experiments we observed that it caused the majority of the splits. Based on Theorem 4 and Remark 1, we used the following version of our confidence bounds $\varepsilon_{\mathrm{KM}}$ and $\varepsilon_{\mathrm{Gini}}$ (the bound for $\varepsilon_{\mathrm{ent}}$ contains an extra $\ln m$ factor),

$$\widetilde{\varepsilon}_{\mathrm{KM}} = \widetilde{\varepsilon}_{\mathrm{Gini}} = c\sqrt{\frac{1}{m}\ln\bigl(m^2h^2td\bigr)} \qquad (8)$$

where the parameter c is used to control the number of splits.

In a preliminary round of experiments, we found that the Gini index delivered a performance comparable to that of entropy and Kearns-Mansour, but —on average— produced trees that were more compact for all three algorithms (ours and the two baselines). Hence, we ran all remaining experiments using the Gini index.

In all experiments we measured the *online performance*. This is the average performance (either accuracy or F-measure) when each new example in the stream is predicted using the tree trained only over the past examples in the stream ("Interleaved Test-Then-Train" validation in MOA).

A. *Controlled Experiments*

In order to empirically verify the features of our bounds we performed experiments in a controlled setting. These experiments show how the detailed form of our confidence bound, which —among other things— takes into account the number d of attributes and the structure of the tree (through the depth of the leaves to split), allows C-Tree to select splits that are generally better than the splits selected by the baselines. In particular, we generated data streams from a random decision trees with 50 leaves and observed that C-Tree dominates across the entire range of parameters and —unlike the other algorithms— achieves the best accuracy when the number of its leaves is the same as the one of the tree generating the stream. The random binary trees were generated according to Algorithm 2 with fixed class distributions in each leaf. The random binary trees are constructed through recursive random splits. More precisely, we start at the root with a budget of n leaves. Then we assign to the left and right sub-trees $\lfloor nX \rfloor$ and $n-1-\lfloor nX \rfloor$ leaves respectively, where X is uniformly distributed in the unit interval. This splitting continues with i.i.d. draws of X until the left and right sub-trees are left with one leaf each. Whenever a split is generated, we assign it a uniformly random attribute and a random threshold value. In the experiment, we generated 1000 random binary trees with $n = 50$ leaves. The random splits are performed choosing among $d = 5$ attributes. For simplicity, we only considered numerical attributes and thresholds in the $[0, 1]$ interval. A random binary tree is then used to generate a stream as follows: for each leaf of the tree, 10,000 examples are uniformly drawn from the subregion of $[0, 1]^5$ defined by the leaf, obtaining 500,000 examples. Each of these examples is given label 1 with probability 0.7 for a left leaf and with probability 0.3 for a right leaf. In Figure 2 we show online performances averaged

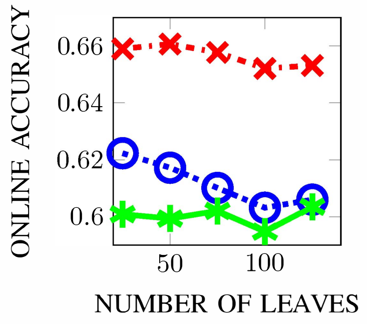

Fig. 2. Online accuracy against number of leaves achieved across a grid of 200 input parameter values on 1000 synthetic datasets ($c \in (0, 2)$ for C-Tree (cross and red line), $\delta \in (0, 1)$ for H-Tree (circle and blue line) and CorrH-Tree (star and green line).

over 1000 streams, each generated using a different random binary tree. In order to span a wide range of tree sizes, we used a grid of 200 different values for the algorithms' parameters controlling the growth of the trees. Namely, the parameter δ available in MOA implementation of H-Tree and CorrH-Tree, and the parameter c of (8) for C-Tree (in C-Tree δ is set to $\frac{1}{t}$ according to Remark 1). The plots are obtained as follows: for each dataset and algorithm we logged the running average of the online performance and the total number of leaves in the tree as the stream was being fed to the algorithm.

B. *Experiments on real-world data*

We constructed ten different streams from each dataset listed below here by taking a random permutation of the examples in it. A9A, COD-RNA and COVERTYPE are from

Dataset	Dimension	Examples	$\lvert + \rvert$	$\lvert - \rvert$
A9A*	123	48842	11687	37155
AIRLINES	7	539383	240264	299119
COD-RNA*	8	488565	162855	325710
COVERTYPE	54	581012	283301	297711
ELECTRICITY	8	45312	26075	19237

TABLE I. DATASETS USED FOR BENCHMARKING.

the LIBSVM binary classification repository[8]. AIRLINES and ELECTRICITY are from the MOA collection[9]. On the unbalanced datasets (marked with a star in Table I) we used the F-measure on the smallest class to measure performance whereas accuracy was used for the remaining datasets. The parameters δ (H-Tree and CorrH-Tree) and c (C-Tree) were individually tuned on each dataset using a grid of 200 values, hence plots show the online performance of each algorithm when it is close to be optimally tuned. Even if the datasets are not particularly large, the plots show that trees generated by our algorithm compare favourably with respect to the baselines especially in the first learning phases.

VII. Conclusions and Future Works

The goal of this work was to provide a more rigorous statistical analysis of confidence intervals for splitting leaves in decision trees. Our confidence bounds take into account all

[7]This is the parameter dictating how many new examples since the last evaluation should be routed to a leaf before revisiting the decision —see [5].

[8]www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html

[9]moa.cms.waikato.ac.nz/datasets/

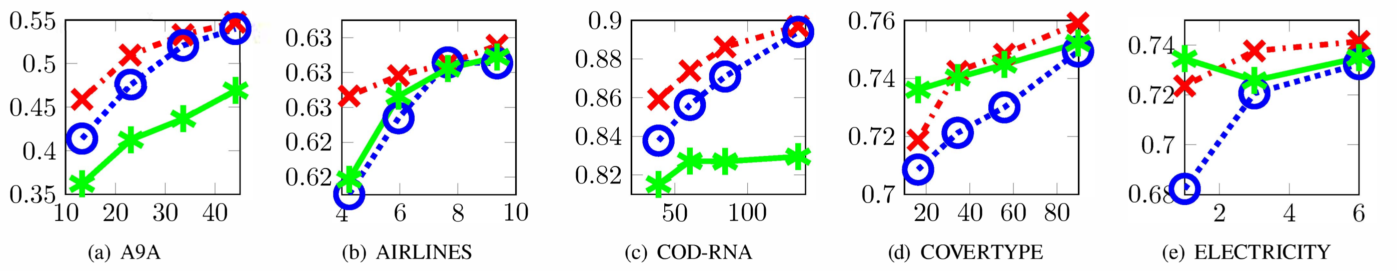

Fig. 3. Online performance (accuracy or F-measure) against number of leaves for each bin and dataset achieved by C-Tree (cross and red line), H-Tree (circle and blue line) and CorrH-Tree (star and green line).

Algorithm 2 RandCBT

```
Input: tree $T$, total number of leaves num-leaves, number
    of attributes $d$, leaf class conditional probability $q$
Output: complete binary tree $T$
    current-node $i$ = CreateNode()
    if num-leaves == 1 then
        mark $i$ as leaf
        if $i$ is a left child then
            $\mathbb{P}(Y = 1 | \boldsymbol{X} \to i) = q$
        else
            $\mathbb{P}(Y = 1 | \boldsymbol{X} \to i) = 1 - q$
        end if
    else
        $x$ = UniformSample$[0, 1]$
        left-leaves = $\max\{1, \lfloor$num-leaves $\cdot x\rfloor\}$
        right-leaves = num-leaves − left-leaves
        $i$ = RandomAttribute$(1, \dots, d)$
        $v$ = RandomValueInSubRegion$(i)$
        add split test $(i, v)$ to $i$
        l-child = RandCBT($i$, left-leaves, $d$, $q$)
        r-child = RandCBT($i$, right-leaves, $d$, $q$)
        add l-child and r-child as a descendent of $i$
    end if
    return current-node $i$
```

the relevant variables of the problem. This improved analysis is reflected in the predictive ability of the learned decision trees, as we show in the experiments. It is important to note that the proposed bounds can be easily applied to the many proposed variants of VFDT. Interesting directions for future research are the application of active learning methods to the incremental learning process (in order to save on the number of labels extracted from the stream), and the extension of our technique to drifting concept scenarios.

Appendix A
Proof Theorem 1

Let H be the standard (unscaled) entropy. Using the standard identity $H(Y_{|i} \mid F) = H(Y_{|i}, F) - H(F)$, we have $\widehat{\Phi}_{i|F} = \widehat{H}_{1/2}(Y_{|i}, F) - \widehat{H}_{1/2}(F)$. We now use part (iii) of the remark following [1, Corollary 1], we have that

$$\left|\widehat{H}_{1/2}(F) - \mathbb{E}\,\widehat{H}_{1/2}(F)\right| \le \frac{\ln m}{2}\sqrt{\frac{2}{m}\ln\frac{4}{\delta}}$$

$$\left|\widehat{H}_{1/2}(Y_{|i}, F) - \mathbb{E}\,\widehat{H}_{1/2}(Y_{|i}, F)\right| \le \frac{\ln m}{2}\sqrt{\frac{2}{m}\ln\frac{4}{\delta}}$$

simultaneously hold with probability at least $1 - \delta$. These bounds hold irrespective to the size of the sets in which $Y_{|i}$ and F take their values.

Next, we apply [23, Proposition 1], which states that

$$-\ln\left(1 + \frac{N-1}{m}\right) \le \mathbb{E}\,\widehat{H}(Z) - H(Z) \le 0$$

for any random variable Z which takes N distinct values. In our case, $N = 2$ for $Z = F$ and $N = 4$ for $Z = \big(Y_{|i}, F\big)$. Hence, using $-a \le -\ln(1 + a)$ for all a, we get

$$\left|H_{1/2}(F) - \mathbb{E}\,\widehat{H}_{1/2}(F)\right| \le \frac{1}{2m}$$

$$\left|H_{1/2}(Y_{|i}, F) - \mathbb{E}\,\widehat{H}_{1/2}(Y_{|i}, F)\right| \le \frac{3}{2m} .$$

Putting everything together gives the desired result.

Appendix B
Proof Theorem 2

Lemma 2 (McDiarmid's inequality): Let G be a real function of m independent random variables $X_1, \dots, X_m$ such that

$$\Big|G(x_1, \dots, x_i, \dots, x_m) - G(x_1, \dots, x_i', \dots, x_m)\Big| \le c \quad (9)$$

for some constant $c \in \mathbb{R}$ and for all realizations $x_1, \dots, x_i, x_i', \dots, x_m$. Then

$$\mathbb{P}\big(\big|G - \mathbb{E}\,G\big| \ge \epsilon\big) \le 2\exp\left(\frac{-2\epsilon^2}{m\,c^2}\right) .$$

If we set the right-hand side equal to δ, then

$$\big|G - \mathbb{E}\,G\big| \le c\sqrt{\frac{m}{2}\ln\frac{2}{\delta}}$$

is true with probability at least $1 - \delta$.

Note the following fact

$$\begin{aligned}
&J(Y_{|i} \mid F) \\
&\quad = \mathbb{P}(F = 1)\,2\,\frac{\mathbb{P}(Y_{|i} = 1, F = 1)}{\mathbb{P}(F = 1)}\,\frac{\mathbb{P}(Y_{|i} = 0, F = 1)}{\mathbb{P}(F = 1)} \\
&\quad + \mathbb{P}(F = 0)\,2\,\frac{\mathbb{P}(Y_{|i} = 1, F = 0)}{\mathbb{P}(F = 0)}\,\frac{\mathbb{P}(Y_{|i} = 0, F = 0)}{\mathbb{P}(F = 0)} \\
&\quad = 2\,\frac{\mathbb{P}(Y_{|i} = 1, F = 1)\,\mathbb{P}(Y_{|i} = 0, F = 1)}{\mathbb{P}(F = 1)} \\
&\quad + 2\,\frac{\mathbb{P}(Y_{|i} = 1, F = 0)\,\mathbb{P}(Y_{|i} = 0, F = 0)}{\mathbb{P}(F = 0)} \\
&\quad = \text{HM}(p_1, q_1) + \text{HM}(p_0, q_0) . \qquad (10)
\end{aligned}$$

In view of applying McDiarmid inequality, let $\widehat{p}_k = \frac{r}{m}$ and $\widehat{q}_k = \frac{s}{m}$. We can write the left-hand side of condition (9) in Lemma 2 for each term of (10) as

$$\frac{2}{m}\left|\frac{rs}{r+s} - \frac{r's'}{r'+s'}\right|$$

where $r, s = 1, \dots, m$ and r', s' may take the following forms: $(r+1, s-1)$ —when a label of an example in the current leaf is flipped, $(r+1, s)$ —when an example is moved from the sibling leaf to the current leaf, and $(r-1, s)$ —when an example is moved from the current leaf to the sibling leaf. Since the harmonic mean is symmetric in r and s, we can ignore the cases $(r-1, s+1)$, $(r, s+1)$, and $(r, s-1)$. A tedious but simple calculation shows that

$$\left|\frac{rs}{r+s} - \frac{r's'}{r'+s'}\right| \le 1 \ .$$

Therefore, we may apply Lemma 2 with $c = \frac{4}{m}$ and obtain that

$$\left|\widehat{\Phi}_{i,F} - \mathbb{E}\,\widehat{\Phi}_{i,F}\right| \le \sqrt{\frac{8}{m}\ln\frac{2}{\delta}} \tag{11}$$

holds with probability at least $1-\delta$.

Next, we control the bias of $\text{HM}\big(\widehat{p}_k, \widehat{q}_k\big)$ as follows,

$$\begin{aligned}
0 &\le \text{HM}(p_k, q_k) - \mathbb{E}\Big[\text{HM}\big(\widehat{p}_k, \widehat{q}_k\big)\Big] \\
&= 2\frac{p_k q_K}{p_k + q_k} - 2\mathbb{E}\left[\frac{\widehat{p}_k\widehat{q}_k}{\widehat{p}_k + \widehat{q}_k}\right] \qquad (12)\\
&= 2\mathbb{E}\left[\frac{p_k\widehat{p}_k(q_k - \widehat{q}_k) + q_k\widehat{q}_k(p_k - \widehat{p}_k)}{(p_k + q_k)\big(\widehat{p}_k + \widehat{q}_k\big)}\right] \\
&\le 2\mathbb{E}\,|q_k - \widehat{q}_k| + 2\mathbb{E}|p_k - \widehat{p}_k| \\
&\le 2\sqrt{\mathbb{E}\left[\big(q_k - \widehat{q}_k\big)^2\right]} + 2\sqrt{\mathbb{E}\left[\big(p_k - \widehat{p}_k\big)^2\right]} \\
&\le \frac{2}{\sqrt{m}} \qquad (13)
\end{aligned}$$

where the first inequality is due to the concavity of HM. Combining (11) and (13) concludes the proof.

Appendix C
Proof Theorem 3

Lemma 3: Let B binomially distributed with parameters (m, p). Then

$$\left|\sqrt{\frac{B}{m}} - \sqrt{p}\right| \le \sqrt{\frac{1}{m}\ln\frac{2}{\delta}}$$

is true with probability at least $1-\delta$.

Proof: The result is an immediate consequence of the bounds in [22] —see also [3, Exercise 2.13]. In particular,

$$\mathbb{P}\left(\left|\frac{B}{m} - p\right| \ge \varepsilon\right) \le e^{-mD(p+\varepsilon\|p)} + e^{-mD(p-\varepsilon\|p)}$$

where $D(q\|p) = q\ln\frac{q}{p} + (1-q)\ln\frac{1-q}{1-p}$ is the KL divergence, and

$$\begin{aligned}
D(p+\varepsilon\|p) &\ge \big(\sqrt{p+\varepsilon} - \sqrt{p}\big)^2 \\
D(p-\varepsilon\|p) &\ge 2\big(\sqrt{p-\varepsilon} - \sqrt{p}\big)^2 \ .
\end{aligned}$$

Simple algrebraic manipulation concludes the proof. ■

Similarly to the proof of Theorem 2, note that

$$\begin{aligned}
&Q(Y_{|i} \mid F) \\
&= \mathbb{P}(F=1)\sqrt{\frac{\mathbb{P}(Y_{|i}=1, F=1)}{\mathbb{P}(F=1)}\,\frac{\mathbb{P}(Y_{|i}=0, F=1)}{\mathbb{P}(F=1)}} \\
&+ \mathbb{P}(F=0)\sqrt{\frac{\mathbb{P}(Y_{|i}=1, F=0)}{\mathbb{P}(F=0)}\,\frac{\mathbb{P}(Y_{|i}=0, F=0)}{\mathbb{P}(F=0)}} \\
&= \sqrt{\mathbb{P}(Y_{|i}=1, F=1)\,\mathbb{P}(Y_{|i}=0, F=1)} \\
&+ \sqrt{\mathbb{P}(Y_{|i}=1, F=0)\,\mathbb{P}(Y_{|i}=0, F=0)} \\
&= \sqrt{p_1 q_1} + \sqrt{p_0 q_0}
\end{aligned}$$

Then

$$\begin{aligned}
&\left|\sqrt{\widehat{p}_1\widehat{q}_1} + \sqrt{\widehat{p}_0\widehat{q}_0} - \sqrt{p_1 q_1} - \sqrt{p_0 q_0}\right| \\
&\le \left|\sqrt{\widehat{p}_1\widehat{q}_1} - \sqrt{p_1 q_1}\right| + \left|\sqrt{\widehat{p}_0\widehat{q}_0} - \sqrt{p_0 q_0}\right| \le 4\varepsilon
\end{aligned}$$

whenever $\left|\sqrt{\widehat{p}_k} - \sqrt{p_k}\right| \le \varepsilon$ and $\left|\sqrt{\widehat{q}_k} - \sqrt{q_k}\right| \le \varepsilon$ for $k \in \{0, 1\}$. Using the union bound and Lemma 3 we immediately get

$$\left|\sqrt{\widehat{p}_1\widehat{q}_1} + \sqrt{\widehat{p}_0\widehat{q}_0} - \sqrt{p_1 q_1} - \sqrt{p_0 q_0}\right| \le 4\sqrt{\frac{1}{m}\ln\frac{8}{\delta}}$$

thus concluding the proof.

Appendix D
Proof Theorem 4

Fix some arbitrary tree T of depth H and let $\mathcal{D}_h$ be the set of internal nodes at depth h. Clearly,

$$\sum_{i \in \mathcal{D}_h} \mathbb{P}(\boldsymbol{X} \to i) \le 1 \ .$$

Now, for any internal node i of T, let F_i be the split used at that node. We have

$$\begin{aligned}
&\mathbb{P}\big(\boldsymbol{X} \text{ routed via a } \tau\text{-suboptimal split}\big) \\
&= \mathbb{P}\big(\exists i \,:\, \boldsymbol{X} \to i,\ G_{i,F_i} + \tau < \max_{F \in \mathcal{F}} G_{i,F}\big) \\
&\le \sum_i \mathbb{P}\big(G_{i,F_i} + \tau < \max_{F \in \mathcal{F}} G_{i,F} \mid \boldsymbol{X} \to i\big)\mathbb{P}(\boldsymbol{X} \to i) \\
&= \sum_{h=0}^{H} \sum_{i \in \mathcal{D}_h} \mathbb{P}\big(G_{i,F_i} + \tau < \max_{F \in \mathcal{F}} G_{i,F} \mid \boldsymbol{X} \to i\big)\mathbb{P}(\boldsymbol{X} \to i) \\
&\le \sum_{h=0}^{H} \sum_{i \in \mathcal{D}_h} \mathbb{P}\Bigg(\widehat{\Phi}_{i|\widehat{F}} > \widehat{\Phi}_{i|F} \\
&\qquad - 2\varepsilon\left(m_{i,t}, \frac{\delta/\big[(h+1)(h+2)\big]}{tdm_{i,t}}\right) \Bigg|\, \boldsymbol{X} \to i\Bigg)\mathbb{P}(\boldsymbol{X} \to i) \\
&\le \sum_{h=0}^{H} \sum_{i \in \mathcal{D}_h} \sum_{s=0}^{t-1} \mathbb{P}\Bigg(\widehat{\Phi}_{i|\widehat{F}} > \widehat{\Phi}_{i|F} \\
&\qquad - 2\varepsilon\left(s, \frac{\delta/\big[(h+1)(h+2)\big]}{tds}\right) \Bigg|\, \boldsymbol{X} \to i\Bigg)\mathbb{P}(\boldsymbol{X} \to i) \\
&\le \sum_{h=0}^{H} \sum_{i \in \mathcal{D}_h} \sum_{s=0}^{t-1} \frac{\delta}{(h+1)(h+2)t}\mathbb{P}(\boldsymbol{X} \to i) \quad \text{(by Lemma 1)} \\
&\le \sum_{h=0}^{H} \sum_{s=0}^{t-1} \frac{\delta}{(h+1)(h+2)t} \le \delta\ .
\end{aligned}$$

REFERENCES

[1] András Antos and Ioannis Kontoyiannis. Convergence properties of functional estimates for discrete distributions. *Random Structures & Algorithms*, 19(3-4):163–193, 2001.

[2] Albert Bifet, Geoff Holmes, Bernhard Pfahringer, Richard Kirkby, and Ricard Gavaldà. New ensemble methods for evolving data streams. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 139–148, New York, NY, USA, 2009. ACM.

[3] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities*. Oxford University Press, 2013.

[4] Tom Dietterich, Michael Kearns, and Yishay Mansour. Applying the weak learning framework to understand and improve C4.5. In *ICML*, pages 96–104. Citeseer, 1996.

[5] Pedro Domingos and Geoff Hulten. Mining high-speed data streams. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 71–80. ACM, 2000.

[6] Piotr Duda, Maciej Jaworski, Lena Pietruczuk, and Leszek Rutkowski. A novel application of hoeffding's inequality to decision trees construction for data streams. In *Neural Networks (IJCNN), 2014 International Joint Conference on*. IEEE, 2014.

[7] João Gama, Pedro Medas, and Ricardo Rocha. Forest trees for on-line data. In *Proceedings of the 2004 ACM symposium on Applied computing*, SAC '04, pages 632–636, New York, NY, USA, 2004. ACM.

[8] João Gama, Ricardo Rocha, and Pedro Medas. Accurate decision trees for mining high-speed data streams. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 523–528. ACM, 2003.

[9] João Gama, Petr Kosina, et al. Learning decision rules from data streams. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1255, 2011.

[10] Jonathan Gratch. Sequential inductive learning. In *In Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 779–786. AAAI Press, 1995.

[11] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[12] Geoff Hulten, Laurie Spencer, and Pedro Domingos. Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 97–106. ACM, 2001.

[13] Elena Ikonomovska, João Gama, and Sašo Džeroski. Learning model trees from evolving data streams. *Data mining and knowledge discovery*, 23(1):128–168, 2011.

[14] Ruoming Jin and Gagan Agrawal. Efficient decision tree construction on streaming data. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 571–576. ACM, 2003.

[15] Michael Keams and Yishay Mansour. On the boosting ability of top-down decision tree learning algorithms. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 459–468. ACM, 1996.

[16] Richard Kirkby, Remco R Bouckaert, Milan Studen, Chi-San Althon Lin, Ashraf M Kibriya, Eibe Frank, Michael Mayo, Michael Mayo, S Mutter, B Pfahringer, et al. Improving hoeffding trees. *Int. J. Approx. Reasoning*, 45:39–48, 2007.

[17] Petr Kosina and João Gama. Very fast decision rules for multi-class problems. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 795–800. ACM, 2012.

[18] Jing Liu, Xue Li, and Weicai Zhong. Ambiguous decision trees for mining concept-drifting data streams. *Pattern Recognition Letters*, 30(15), 2009.

[19] Pawel Matuszyk, Georg Krempl, and Myra Spiliopoulou. Correcting the usage of the hoeffding inequality in stream mining. In *Advances in Intelligent Data Analysis XII*, pages 298–309. Springer, 2013.

[20] Colin McDiarmid. On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188, 1989.

[21] Ron Musick, Jason Catlett, and Stuart J. Russell. Decision theoretic subsampling for induction on large databases. In *ICML*, pages 212–219. Morgan Kaufmann, 1993.

[22] Masashi Okamoto. Some inequalities relating to the partial sum of binomial probabilities. *Annals of the institute of Statistical Mathematics*, 10(1):29–35, 1959.

[23] Liam Paninski. Estimation of entropy and mutual information. *Neural Computation*, 15(6):1191–1253, 2003.

[24] Bernhard Pfahringer, Geoffrey Holmes, and Richard Kirkby. New options for hoeffding trees. In *Proceedings of the 20th Australian joint conference on Advances in artificial intelligence*, AI'07, pages 90–99. Springer-Verlag, 2007.

[25] Leszek Rutkowski, Lena Pietruczuk, Piotr Duda, and Maciej Jaworski. Decision trees for mining data streams based on the McDiarmid's bound. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1272–1279, 2013.

[26] Christophe Salperwyck and Vincent Lemaire. Incremental decision tree based on order statistics. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–8. IEEE, 2013.

[27] Eiji Takimoto and Akira Maruoka. Top-down decision tree learning as information based boosting. *Theoretical Computer Science*, 292(2):447–464, 2003.

[28] Paul E Utgoff. Incremental induction of decision trees. *Machine learning*, 4(2):161–186, 1989.

[29] Wenhua Xu, Zheng Qin, Hao Hu, and Nan Zhao. Mining uncertain data streams using clustering feature decision trees. In *Advanced Data Mining and Applications*, pages 195–208. Springer, 2011.

[30] Hang Yang and Simon Fong. Incrementally optimized decision tree for noisy big data. In *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, BigMine '12, pages 36–44, New York, NY, USA, 2012. ACM.