Online Learning of Decision Trees with Thompson Sampling

Ayman Chaouki
LIX, Ecole Polytechnique, IP Paris
AI Institute, University of Waikato

Jesse Read
LIX, Ecole Polytechnique, IP Paris

Albert Bifet
AI Institute, University of Waikato
LTCI, Télécom Paris, IP Paris

Abstract

Decision Trees are prominent prediction models for interpretable Machine Learning. They have been thoroughly researched, mostly in the batch setting with a fixed labelled dataset, leading to popular algorithms such as C4.5, ID3 and CART. Unfortunately, these methods are of heuristic nature, they rely on greedy splits offering no guarantees of global optimality and often leading to unnecessarily complex and hard-to-interpret Decision Trees. Recent breakthroughs addressed this suboptimality issue in the batch setting, but no such work has considered the online setting with data arriving in a stream. To this end, we devise a new Monte Carlo Tree Search algorithm, Thompson Sampling Decision Trees (TSDT), able to produce optimal Decision Trees in an online setting. We analyse our algorithm and prove its almost sure convergence to the optimal tree. Furthermore, we conduct extensive experiments to validate our findings empirically. The proposed TSDT outperforms existing algorithms on several benchmarks, all while presenting the practical advantage of being tailored to the online setting.

1 INTRODUCTION

Interpretable Machine Learning is crucial in sensitive domains, like medicine, where high-stakes decisions have to be justified. Due to their extraction of simple decision rules, Decision Trees (DTs) are very popular in this context. Unfortunately, finding the optimal DT is NP-complete (Laurent and Rivest, 1976), and for this reason, popular batch algorithms greedily construct a DT by splitting its leaves according to some local gain metric, such approach is used in ID3 (Quinlan, 1986), C4.5 (Quinlan, 2014) and CART (Breiman et al., 1984) to name a few. Due to this heuristic nature, these approaches offer no optimality guarantees. In fact, they often lead to suboptimal DTs that are unnecessarily complex and hard to interpret, contradicting the main motivation behind DTs.

In many modern applications, data is supplied through a stream rather than a fixed data set, this renders most batch algorithms obsolete, which led to the emergence of the data stream (or online) learning paradigm (Bifet and Kirkby, 2009). The classic batch DT algorithms are ill-suited for online learning since they calculate a splitting gain metric on a whole data set. In response, Domingos and Hulten (2000) introduced the VFDT algorithm, which constructs DTs in an online fashion. VFDT estimates the gain of each split using a statistical test based on Hoeffding's inequality. This approach yielded a principled algorithm and laid the foundation for subsequent developments (Hulten et al., 2001; Bifet and Gavalda, 2009; Manapragada et al., 2018), with advances primarily focusing on improving the quality of the statistical tests (Jin and Agrawal, 2003; Rutkowski et al., 2012, 2013). Much like their batch counterparts, these online methods are heuristic in nature, and consequently, they are susceptible to the suboptimality issue.

In this work, we propose a method that circumvents these limitations, yielding an online algorithm proven to converge to the optimal DT. We consider online classification problems with categorical attributes, for which we seek the optimal DT balancing between the accuracy and the number of splits. To achieve this, we frame the problem as a Markov Decision Process (MDP) where the optimal policy leads to the optimal DT. We solve this MDP with a novel Monte Carlo Tree Search (MCTS) algorithm that we call Thompson Sampling Decision Trees (TSDT). TSDT employs a Thompson Sampling policy that converges almost surely to the optimal policy. In

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

our experiments, we highlight the limitations of traditional greedy online DT methods, such as VFDT and EFDT (Manapragada et al., 2018), and demonstrate how TSDT effectively circumvents these shortcomings. Due to the lack of literature on optimal online DTs, we compare TSDT with recent successful batch optimal DT algorithms by feeding benchmark datasets to TSDT as streams, TSDT clearly outperforms DL8.5 (Aglin et al., 2020) and matches or surpasses the performance of OSDT (Hu et al., 2019).

2 RELATED WORK

In the batch setting, the suboptimality issue of DTs has been the subject of multiple research papers focused mainly on mathematical programming, see (Bennett, 1994; Bennett and Blue, 1996; Norouzi et al., 2015; Bertsimas and Dunn, 2017; Verwer and Zhang, 2019). These methods optimise internal splits within a fixed DT structure, making the problem more manageable but potentially missing the optimal DT. Recently, branch and bound methods were proposed to mitigate this issue and yielded the DL8.5 algorithm (Aglin et al., 2020) and OSDT (Hu et al., 2019) among others. A subsequent algorithm, GOSDT (Lin et al., 2020), generalises OSDT to other objective functions including F-score, AUC and partial area under the ROC convex hull. However, these methods are limited to binary attributes, necessitating a preliminary binary encoding of the data. Moreover, the choice of this binary encoding may significantly influence the complexity of the solution, as demonstrated in our experiments. All the aforementioned methods operate solely in the batch learning paradigm, lacking a straightforward extension to the online setting.

The closest work to ours is perhaps (Nunes et al., 2018) since the authors use MCTS, see (Browne et al., 2012) for a survey about MCTS. Nunes et al. (2018) define a rollout policy that completes the selected DT with C4.5 on an induction set, then it estimates the value of the selected DT by evaluating its performance on a validation set. This approach does not differentiate between DTs of different complexities, in fact, the authors rely on a custom definition of terminal states in terms of predefined maximum depth and number of instances, alongside C4.5's pruning strategy. Additionally, by virtue of using C4.5, a pure batch algorithm, this algorithm is not applicable for data streams.

Our proposed method is a Value Iteration approach (Sutton and Barto, 2018) that uses Thompson Sampling policy within a MCTS framework. Unlike Temporal Difference methods, such as Q-Learning and SARSA, general convergence results for Monte Carlo methods remain an open theoretical question, noted in (Sutton and Barto, 2018, p. 99 and p. 103) as "one of the most fundamental open theoretical questions". Some convergence results were established under specific assumptions. Wang et al. (2020) prove almost sure optimal convergence of the policy for Monte Carlo with Exploring Starts, while Dong et al. (2022) show a similar result for Monte Carlo UCB. These results pertain to MDPs with finite random length episodes where the optimal policy does not revisit states. In our case, although our MDP features similar properties, the rewards are unknown and merely estimated. We investigate the convergence properties of MCTS with Thompson Sampling policy within our specific MDP. To the best of our knowledge, no prior work has carried such analysis under this assumption. The closest related work, found in (Bai et al., 2013), only considers discounted MDPs with finite fixed horizons, and does not provide a formal convergence proof, see (Bai et al., 2013, Section 3.5).

3 PROBLEM FORMULATION

Let $X = \left(X^{(1)}, \ldots, X^{(q)}\right)$ be the input with categorical attributes and $Y \in \{1, \ldots, K\}$ the class to predict. Data samples (X_i, Y_i) arrive incrementally through a stream, they are i.i.d. and follow a joint probability distribution $P_{X,Y}$. Let T be a DT and $\mathcal{L}(T)$ the set of leaves of T, for each leaf $l \in \mathcal{L}(T)$ and class $k \in \{1, \ldots, K\}$, let $p(l) = \mathbb{P}[X \in l]$ denote the probability of event "The subset of the input space, described by leaf l, contains X" and $p_k(l) = \mathbb{P}[Y = k | X \in l]$ the probability that $Y = k$ given $X \in l$. For any input X we also denote $l(X)$ the leaf l that contains X, i.e. the leaf l such that $X \in l$. Let $\mathcal{H}(T) = \mathbb{P}[T(X) = Y]$ be the accuracy of T where $T(X) = T(l(X)) = \text{Argmax}_k\{p_k(l(X))\}$ is the predicted class of X according to T. If we define our objective to maximise as $\mathcal{H}(T)$, then the full DT that exhaustively employs all the possible splits is a trivial solution. However, this solution is an uninterpretable DT of maximum depth that just classifies the inputs point-wise, as such, it is of no interest. We seek to balance between maximising the accuracy and minimising the complexity, the latter condition is for interpretation purposes. To this end, we introduce the regularised objective:

$$\mathcal{H}_r(T) = \mathbb{P}[T(X) = Y] - \lambda \mathcal{S}(T)$$

Where $\lambda \geq 0$ is a penalty parameter and $\mathcal{S}(T)$ is the number of splits in T. We note that Hu et al. (2019) introduce a similar objective, but the authors penalise the number of leaves $|\mathcal{L}(T)|$ rather than the number of splits $\mathcal{S}(T)$. Our choice is motivated by the MDP we define in the following section.

3.1 Markov Decision Process (MDP)

We introduce our undiscounted episodic MDP with finite random length episodes as follows:
State: Our state space is the space of DTs.
Action: There are two types of actions, split actions split a leaf with respect to an attribute and the terminal action ends the episode, in which case we transition from the current state T to the terminal state denoted $\overline{T}$ which represents the same DT as T.
Transition Dynamics: When taking an action, we transition from state T to state T' deterministically, in which case we denote the transition $T \to T'$. The set of next states from T is denoted $\mathrm{Ch}(T)$, which signifies the children of T.
Reward: In any non-terminal state T, split actions yield reward $-\lambda$ and the terminal action yields reward $\mathcal{H}(T) = \mathbb{P}[T(X) = Y]$, *which is unknown.* $r(T, T')$ is the reward of the transition $T \to T'$.

In the next paragraph, we link the search for the optimal policy to that of the optimal DT.
A stochastic policy π maps each non-terminal state T to a distribution over $\mathrm{Ch}(T)$, for any $T' \in \mathrm{Ch}(T)$ we denote $\pi(T'|T)$ the probability of the transition $T \to T'$ according to π. If π is deterministic, $\pi(T)$ is the next state from T according to π. Let $T = T^{(0)} \xrightarrow{\pi} T^{(1)} \xrightarrow{\pi} \ldots, \xrightarrow{\pi} T^{(N)}$ be the episode that stems from following policy π starting from state T; $T^{(N)} = \overline{T^{(N-1)}}$ is terminal. The value of π at T is defined as:

$$\mathcal{V}^{\pi}(T) = \mathbb{E}\left[\sum_{j=1}^{N} r\left(T^{(j-1)}, T^{(j)}\right)\right]$$

For convenience, we also define, for all terminal states $\overline{T}$, $\mathcal{V}^{\pi}(\overline{T}) = \mathcal{H}(\overline{T}) = \mathcal{H}(T)$. If π is deterministic, then we get:

$$\mathcal{V}^{\pi}(T) = \lambda \mathcal{S}(T) + \mathcal{H}_r\left(T^{(N-1)}\right)$$

Let R denote the root state (DT with only one leaf), we have $\mathcal{S}(R) = 0$ and therefore:

$$\mathcal{V}^{\pi}(R) = \lambda \mathcal{S}(R) + \mathcal{H}_r\left(T^{(N-1)}\right) = \mathcal{H}_r\left(T^{(N-1)}\right)$$

Let $\pi^* \in \mathrm{Argmax}_{\pi} \mathcal{V}^{\pi}(R)$, the optimal policy π^* exists and is deterministic because our MDP is finite. Let $R \xrightarrow{\pi^*} \ldots, \xrightarrow{\pi^*} T^* \xrightarrow{\pi^*} \overline{T^*}$, then $\mathcal{H}_r(T^*) = \mathcal{V}^{\pi^*}(R) \geq \mathcal{V}^{\pi}(R)$ for any policy π. On the other hand, any DT T is constructed from a series of splits of the root R, thus there always exists a policy π such that $R \xrightarrow{\pi} \ldots, \xrightarrow{\pi} T \xrightarrow{\pi} \overline{T}$, and consequently $\mathcal{V}^{\pi}(R) = \mathcal{H}_r(T)$. As a result, $\mathcal{H}_r(T^*) \geq \mathcal{H}_r(T)$ for any DT T, establishing the optimality of T^*. We can find T^* by deriving π^* first and then following it.

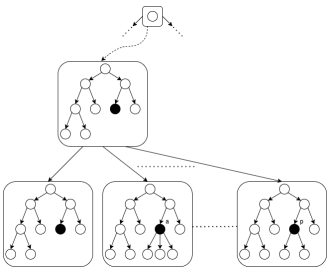

Figure 1: Each Search Node is a state and each edge an action. The left-most edge is the terminal action, hence why both the parent and child Search Nodes represent the same DT. The remaining edges are split actions with respect to the black leaf.

3.2 Tree representation of the State-Action Space

As is custom in MCTS, the State-Action space is represented as a Tree called the Search Tree. **We refer to the nodes of the Search Tree as Search Nodes to avoid confusion with the nodes of DTs**. These Search Nodes serve as representations of states, which are DTs in the context of our MDP. The edges within the Search Tree correspond to actions. Throughout this work, we refer to DTs, states and Search Nodes interchangeably. The root of the Search Tree is the initial state R (the root DT), and its leaves, called Search Leaves, are the terminal states. Figure 1 depicts a segment of the Search Tree.

4 TSDT

In this section, we introduce our method. If we know $\mathcal{V}^{\pi^*}(\overline{T}) = \mathcal{H}(T)$ for all Search Leaves $\overline{T}$, then we can Backpropagate these values up the Search Tree and recursively deduce $\mathcal{V}^{\pi^*}(T)$ for all internal Search Nodes with the Bellman Optimality Equation:

$$\mathcal{V}^{\pi^*}(T) = \max_{T' \in \mathrm{Ch}(T)} \left\{ -\lambda \mathbb{1}\left\{T' \neq \overline{T}\right\} + \mathcal{V}^{\pi^*}(T') \right\} \quad (1)$$

Unfortunetely, the values $\mathcal{V}^{\pi^*}(\overline{T})$ are unknown, this prompts us to estimate them, but which Search Leaves should we prioritise? We need a policy with an efficient Exploration-Exploitation trade-off. Several notable options can be considered, among which UCB (Auer et al., 2002), that was popularised, in the context of MCTS, by the UCT algorithm (Kocsis and Szepesvári, 2006). UCB is out of the scope of this paper, we analyse it in one of our ongoing works. We consider Thompson Sampling instead. To use this policy, we need to estimate the values within a Bayesian framework.

4.1 Estimating $\mathcal{V}^{\pi^*}(\overline{T})$ for a Search Leaf $\overline{T}$

For any Search Leaf $\overline{T}$, we have:

$$\begin{aligned}\mathcal{V}^{\pi^*}(\overline{T}) &= \mathcal{H}(T) = \mathbb{P}[T(X) = Y] \\ &= \sum_{l \in \mathcal{L}(T)} p(l)\, \mathbb{E}[\mathbb{1}\{T(X) = Y\} | X \in l] \quad (2)\end{aligned}$$

From Equation (2), given observed data $\{(X_i, Y_i)\}_{i=1}^N$ in T, we define the posterior on $\mathcal{V}^{\pi^*}(\overline{T})$ with:

$$\theta_{\overline{T}} = \sum_{l \in \mathcal{L}(T)} \hat{p}(l)\, \theta_{T,l} \quad (3)$$

Where $\hat{p}(l)$ is an estimator of $p(l)$ and $\theta_{T,l}$ follows a posterior distribution on $\mathbb{E}[\mathbb{1}\{T(X) = Y\} | X \in l]$ given $\{(X_i, Y_i)\}_{i=1}^N$. Since $\mathbb{E}[\mathbb{1}\{T(X) = Y\} | X \in l]$ is the mean of a Bernoulli variable, we are tempted to use its Beta conjugate prior with the following updates:

$$\begin{aligned}&\theta_{T,l} \sim \text{Beta}(\alpha_{T,l}, \beta_{T,l}) \\ &\alpha_{T,l} = 1 + \sum_{i=1}^N \mathbb{1}\{X_i \in l\}\mathbb{1}\{T(X_i) = Y_i\}; \\ &\beta_{T,l} = 1 + \sum_{i=1}^N \mathbb{1}\{X_i \in l\} - \sum_{i=1}^N \mathbb{1}\{X_i \in l\}\mathbb{1}\{T(X_i) = Y_i\}\end{aligned}$$

However, the challenge pertains to the unknown nature of $T(X_i) = \text{Argmax}_k\{p_k(l(X_i))\}$. We solve this issue with the empirical average estimates:

$$\begin{aligned}\forall l \in \mathcal{L}(T) : \hat{p}_k^{(i)}(l) &= \frac{\sum_{j=1}^i \mathbb{1}\{X_j \in l\}\mathbb{1}\{Y_j = k\}}{\sum_{j=1}^i \mathbb{1}\{X_j \in l\}} \\ \hat{T}_i(l) &= \text{Argmax}_k\{\hat{p}_k^{(i)}(l)\} \quad (4)\end{aligned}$$

Then we rather update $\alpha_{T,l}$ and $\beta_{T,l}$ with:

$$\alpha_{T,l} = 1 + \sum_{i=2}^N \mathbb{1}\{X_i \in l\}\mathbb{1}\{\hat{T}_{i-1}(X_i) = Y_i\}; \quad (5)$$

$$\begin{aligned}\beta_{T,l} &= 1 + \sum_{i=2}^N \mathbb{1}\{X_i \in l\} \\ &\quad - \sum_{i=2}^N \mathbb{1}\{X_i \in l\}\mathbb{1}\{\hat{T}_{i-1}(X_i) = Y_i\} \quad (6)\end{aligned}$$

Now $\theta_{\overline{T}}$ (Equation (3)) is a linear combination of the Beta variables $\theta_{T,l}$, and its distribution is not easy to infer. For this reason, we consider the common Normal approximation of the Beta distribution where we match the first two moments.

$$\theta_{T,l} \sim \mathcal{N}\left(\mu_{T,l}, (\sigma_{T,l})^2\right);\ \mu_{T,l} = \frac{\alpha_{T,l}}{\alpha_{T,l} + \beta_{T,l}} \quad (7)$$

$$(\sigma_{T,l})^2 = \frac{\alpha_{T,l}\beta_{T,l}}{(\alpha_{T,l} + \beta_{T,l})^2 (1 + \alpha_{T,l} + \beta_{T,l})} \quad (8)$$

This makes $\theta_{\overline{T}} = \sum_{l \in \mathcal{L}(T)} \hat{p}(l)\, \theta_{T,l}$ a linear combination of Normal random variables, and therefore:

$$\theta_{\overline{T}} \sim \mathcal{N}\left(\mu_{\overline{T}}, (\sigma_{\overline{T}})^2\right);\ \mu_{\overline{T}} = \sum_{l \in \mathcal{L}(T)} \hat{p}(l)\, \mu_{T,l} \quad (9)$$

$$(\sigma_{\overline{T}})^2 = \sum_{l \in \mathcal{L}(T)} \hat{p}(l)^2 (\sigma_{T,l})^2 \quad (10)$$

We recall that this posterior distribution is conditioned on the observed data $\{(X_i, Y_i)\}_{i=1}^N$ in T. To finalise the definition of $\theta_{\overline{T}}$, we still need to define $\hat{p}(l)$. We defer this task to Section 4.4 as we are currently lacking some key insights from the Algorithm. For the time being, we assume having such estimator that is completely defined by $\{X_i\}_{i=1}^N$ and consistent $\hat{p}(l) \xrightarrow[N \to \infty]{\text{a.s}} p(l)$.

4.2 Estimating $\mathcal{V}^{\pi^*}(T)$ for an internal Search Node T

Let T be an internal Search Node, which is a non-terminal state. Value Iteration updates the estimate of $\mathcal{V}^{\pi^*}(T)$ according to the Bellman Optimality Equation (1). Thus, we define the posterior on $\mathcal{V}^{\pi^*}(T)$ given all the observed data $\{(X_i, Y_i)\}_{i=1}^N$ in T as:

$$\theta_T = \max_{T' \in \text{Ch}(T)} \left\{ -\lambda \mathbb{1}\left\{T' \neq \overline{T}\right\} + \theta_{T'} \right\} \quad (11)$$

What is the posterior distribution of θ_T given $\{(X_i, Y_i)\}_{i=1}^N$?

We suppose $\forall T' \in \text{Ch}(T) : \theta_{T'} \sim \mathcal{N}\left(\mu_{T'}, (\sigma_{T'})^2\right)$. This is motivated by an inductive reasoning, indeed, if we show that θ_T is Normally distributed, then since the posteriors of all Search Leaves are Normal, as defined in Section 4.1, we would recursively infer that the posteriors of all internal Search Nodes are Normal. Unfortunately, the maximum of Normal variables, as defined in Equation (11), is not Normal, this observation is documented in (Clark, 1961; Sinha et al., 2007). To solve this issue, a first approach is to use a Normal approximation of the distribution of the maximum in Equation (11), this is achieved by recursively applying Clark's mean and variance formula for the maximum of two Normal variables as demonstrated in (Tesauro et al., 2010, Section 4.1). This leads to our first version of TSDT, we present the details of this Backpropagation scheme in Appendix C. Nevertheless, this approximation incurs a substantial computational cost as the number of children $|\text{Ch}(T)|$ increases. Moreover, as outlined by Sinha et al. (2007), the order in which the recursive approximations are executed may significantly affect the quality of the overall approximation. For these reasons, we introduce an alternative, more straightforward approach to Backpropagating the posterior distributions $\theta_{T'}$ to θ_T. Specifically, we assign

to θ_T the posterior distribution of the child with maximum posterior mean:

$$\widetilde{T} = \text{Argmax}_{T' \in \text{Ch}(T)} \{\mu_{T'}\};\ \theta_T \sim \mathcal{N}\left(\mu_{\widetilde{T}}, \left(\sigma_{\widetilde{T}}\right)^2\right) \tag{12}$$

We call this second version of the Algorithm Fast-TSDT, it is more practical than the first one and exhibits better computational efficiency. While in general, the distribution in formula (12) may not serve as an accurate approximation for the distribution of the maximum in Equation (11), it progressively improves as $\theta_{T'}$, for children T', concentrate around their means. This concentration occurs as more data is accumulated within T', as exemplified in Equations (10) and (8) for Search Leaves. Furthermore, it is worth noting that Fast-TSDT displays a substantial gain in computational efficiency and also (surprisingly!) in performance compared to TSDT.

4.3 The Algorithm

Algorithm 1 is an abstract description of TSDT and Fast-TSDT. In the following, we view it from the perspective of an incremental construction of the Search Tree representation. Initially, our Search Tree representation contains the root R only, then at each iteration t, we follow the steps illustrated in Figure 2.

- **Selection:** Line 4. Starting from R, descend the Search Tree by choosing a child according to the current policy π_t until reaching a Search Leaf.
- **Simulation:** Line 5. We observe new m incoming data from the stream in $T^{(N-1)}$. The objective is to use the accumulated observed data in $T^{(N-1)}$ to, either initialise the posteriors of children $T' \in \text{Ch}\left(T^{(N-1)}\right)$ with $\theta_{T'} = \theta_{\overline{T'}}$ (Line 11), or to just update the posterior of $\theta_{T^{(N)}}$ (Line 14). These posteriors will in turn update the posterior of θ_T, as per Section 4.2, during Backpropagation.
- **Expansion:** Line 7. If $T^{(N-1)}$ is visited for the first time, we add its children Search Nodes $T' \in \text{Ch}\left(T^{(N-1)}\right)$ to our Search Tree representation.
- **Backpropagation:** Loop from Line 16 to 21. Recursively update the posterior of the ancestors $\theta_{T^{(j)}}$ (Line 17), for $j = N-1$ to 0, with the internal Search Nodes posterior updates as per Section 4.2 (formula (12) for Fast-TSDT, and the recursive Normal approximation of the maximum for TSDT). At the same time, update the Thompson Sampling policy (Loop from Line 18 to 20).

In line 23, after M iterations of TSDT, we define the greedy policy π with respect to the posterior means. Then we unroll π: $R = T^{(0)} \xrightarrow{\pi} T^{(1)} \xrightarrow{\pi} \ldots, \xrightarrow{\pi} T^{(N)}$ and return the proposed solution $T^{(N)}$ (Lines 24, 25).

To complete the description of Algorithm 1, we still need to answer the following question: **How does the Simulation step allow us to initialise the posteriors for the children?** To answer this question, we introduce some new statistics.

Let T be a Search Node that was simulated, and suppose we observe data $\{(X_s, Y_s)\}_{s=1}^N$ in T. Let $l \in \mathcal{L}(T)$ be a leaf of T and $T^{(l,i)} \in \text{Ch}(T)$ the child that stems from splitting l with respect to attribute $X^{(i)}$. Our objective here is to use data $\{(X_s, Y_s)\}_{s=1}^N$ to initialise $\theta_{T^{(l,i)}} = \theta_{\overline{T^{(l,i)}}}$. According to Equations (7), (8), (9) and (10), this is achieved by calculating $\alpha_{T^{(l,i)},l'}, \beta_{T^{(l,i)},l'}$ for all leaves $l' \in \mathcal{L}\left(T^{(l,i)}\right)$ (and also $\hat{p}(l')$ which, we remind the reader, is deferred to Section 4.4). Let $l' \in \mathcal{L}\left(T^{(l,i)}\right)$, if l' is not a child of l, then l' is a common leaf between $T^{(l,i)}$ and T, thus $\alpha_{T^{(l,i)},l'} = \alpha_{T,l'}$ and $\beta_{T^{(l,i)},l'} = \beta_{T,l'}$, these are straightforwardly calculated with the observed data $\{(X_i, Y_i)\}_{i=1}^N$ in T using Equations (4), (5) and (6). If l' is a child of l, then there exists j such that $l' = l_{ij}$ is the child of l that corresponds to attribute $X^{(i)}$ being equal to j. For $N' \leq N$, we define:

$$n_{ijk}(N', l) = \sum_{s=1}^{N'} \mathbb{1}\{Y_s = k\}\mathbb{1}\{X_s^{(i)} = j\}\mathbb{1}\{X_s \in l\}$$

On the subset $\{(X_s, Y_s)\}_{s=1}^{N'}$, $n_{ijk}(N', l)$ is the number of inputs $X_s \in l$ of class k satisfying $X_s^{(i)} = j$. Then we can track the estimates:

$$\hat{T}_{N'}^{(l,i)}(l_{ij}) = \text{Argmax}_k \left\{ \frac{n_{ijk}(N', l)}{\sum_k n_{ijk}(N', l)} \right\}$$

$$m(l_{ij}) = \sum_{s=2}^{N} \mathbb{1}\{\hat{T}_{s-1}^{(l,i)}(X_s) = Y_s\}\mathbb{1}\{X_s^{(i)} = j\}\mathbb{1}\{X_s \in l\}$$

$$\alpha_{T^{(l,i)},l_{ij}} = 1 + m(l_{ij});$$

$$\beta_{T^{(l,i)},l_{ij}} = 1 + \sum_{ijk} n_{ijk}(N, l) - m(l_{ij})$$

With this, the initialisation $\theta_{T^{(l,i)}} = \theta_{\overline{T^{(l,i)}}}$ is now complete. In summary, the introduced $n_{ijk}(N', l)$ statistics at the leaves $l \in \mathcal{L}(T)$ allow us to use the accumulated observed data $\{(X_i, Y_i)\}_{i=1}^N$ in T to initialise the posterior $\theta_{T'} = \theta_{\overline{T'}}$ for all children $T' \in \text{Ch}(T)$.

In the following, we provide the optimal convergence result for both TSDT and Fast-TSDT.

Theorem 1. *Let time t denote the number of iterations of* TSDT *and Fast-*TSDT*, then any Search Node T satisfies the following:*

$$\mu_T \xrightarrow[t\to\infty]{a.s} \mathcal{V}^{\pi^*}(T), (\sigma_T)^2 \xrightarrow[t\to\infty]{a.s} 0$$

Algorithm 1 TSDT, Fast-TSDT

Input: M number of iterations, m number of observed samples per Simulation, $\lambda \geq 0$.
Initialise $\pi_0\left(\overline{T}\middle|T\right) = 1$ and *fully_expanded*(T) = *False* for all non-terminal states T
for $t = 1$ to M **do**
  Unroll π_t: $R = T^{(0)} \xrightarrow{\pi_t} T^{(1)} \xrightarrow{\pi_t} \ldots, \xrightarrow{\pi_t} T^{(N)}$
  Simulate$(T^{(N-1)})$
  if not *fully_expanded*$(T^{(N-1)})$ **then**
    Expand$(T^{(N-1)})$
    fully_expanded$(T^{(N-1)}) = True$
    for $T' \in \text{Ch}\left(T^{(N-1)}\right)$ **do**
      Update the posterior of $\theta_{\overline{T'}}$
      Initialise $\theta_{T'} = \theta_{\overline{T'}}$
    end for
  else
    Update the posterior of $\theta_{T^{(N)}}$
  end if
  for $j = N-1$ to 0 **do**
    Update the posterior of $\theta_{T^{(j)}}$
    for $T' \in \text{Ch}\left(T^{(j)}\right)$ **do**
      Update the policy at $T^{(j)} \to T'$

$$\pi_{t+1}\left(T'\middle|T^{(j)}\right) = \mathbb{P}\left[\theta_{T'} = \max_{T'' \in \text{Ch}\left(T^{(j)}\right)} \{\theta_{T''}\}\right]$$

    end for
  end for
end for
Define $\pi(T) = \text{Argmax}_{T' \in \text{Ch}(T)}\left\{-\lambda \mathbb{1}\{T' \neq \overline{T}\} + \mu_{T'}\right\}$ for all non-terminal states T
Unroll π: $R = T^{(0)} \xrightarrow{\pi} T^{(1)} \xrightarrow{\pi} \ldots, \xrightarrow{\pi} T^{(N)}$
return $T^{(N)}$

and any internal Search Node T satisfies:

$$\pi_t\left(\pi^*(T)\middle|T\right) \xrightarrow[t\to\infty]{a.s} 1$$

Theorem 1 states the concentration of θ_T around the optimal value $\mathcal{V}^{\pi^*}(T)$ for any Search Node T. Additionally, it asserts the concentration of the policy $\pi_t(.|T)$ around the optimal action $\pi^*(T)$ for any internal Search Node T. This Theorem provides an asymptotic guarantee of optimality, which is valuable. However, it would be ideal to have some finite time guarantees in the form of PAC-bounds or rates of convergence. Unfortunately, such guarantees are primarily derived for simpler settings like bandit problems. In the context of MDPs, most convergence guarantees are asymptotic, and the issue of finite-time guarantees remains open in many cases. We believe that the tail inequality we derive in Theorem 2 marks a first step towards achieving this goal in a future work. The idea

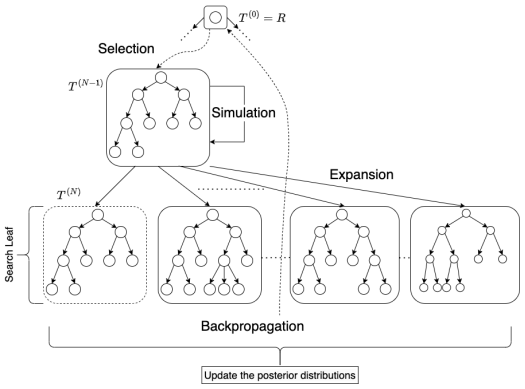

Figure 2: One iteration of TSDT. The Search Node in dashed lines is the Search Leaf $T^{(N)} = \overline{T^{(N-1)}}$.

is that by controlling the concentration of the posteriors at the level of Search Leaves, it may be possible to propagate this control up the Search Tree to the root state R. This would allow us to derive a time-dependent concentration probability of the posterior of θ_R around the true $\mathcal{V}^{\pi^*}(R)$. However, this Backpropagation reasoning is a non-trivial challenge that warrants further exploration in future work.

Theorem 2. *Let T be an internal Search Node with $Ch(T) = \{T_1, \ldots, T_w\}$ Search Leaves and $w \geq 2$. Let t denote the number of visits of T and $N_{T_j}(t)$ the number of visits of T_j up to t. Define $M_w = 1 + \sqrt{\frac{2}{\sqrt{3}}} erfc^{-1}\left(\frac{1}{w-1}\right)$, then $\forall T_j \in Ch(T)$:*

$$\mathbb{P}\left[N_{T_j}(t)\, m \leq \frac{\log t}{4|\mathcal{L}(T_j)|M_w}\right] \leq \exp\left[-\frac{2}{t}\left(\frac{t^{3/4}}{\sqrt{\pi}\phi(t)} - \frac{\log t}{4m|\mathcal{L}(T_j)|M_w^2}\right)^2\right]$$

Where $\phi(t) = \sqrt{\frac{\log t}{4}} + \sqrt{\frac{\log t}{4} + 2}$

4.4 Estimator $\hat{p}(l)$

Let T be a DT and $l \in \mathcal{L}(T)$ some leaf. We recall from Equations (2) and (3) that we need an estimator $\hat{p}(l)$ of $p(l) = \mathbb{P}[X \in l]$. Suppose we observe data $\{(X_i, Y_i)\}_{i=1}^N$ in T, the most straightforward estimator is the empirical average. However, Algorithm 1 incorporates mechanisms that make it sample efficient. It does not copy Decision Tree nodes, thus the $n_{ijk}(N, \eta)$, of any node η, are updated whenever data (X_s, Y_s) is observed (during Simulation) with $X_s \in \eta$, regardless of which Search Node was simulated. In addition, even though $n_{ijk}(N, l)$ can be used to calculate

the empirical averages:

$$\hat{p}(l) = \frac{\sum_{ijk} n_{ijk}(N,l)}{\sum_{l\in\mathcal{L}(T)}\sum_{ijk} n_{ijk}(N,l)}$$
$$\hat{p}(l_{ij}) = \frac{\sum_{k} n_{ijk}(N,l)}{\sum_{l\in\mathcal{L}(T)}\sum_{ijk} n_{ijk}(N,\eta)}$$

for any leaf $l \in \mathcal{L}(T)$ and any child node l_{ij} of l. $n_{ijk}(N,l)$ cannot calculate the empirical average for the children of l_{ij}. This limited scope, along with not copying Decision Tree nodes, cause the observed marginal distribution of input X to shift from the true marginal distribution. We explain this phenomenon in the next paragraph and illustrate it in Figure 3.

When expanding R, the observed data in r update the empirical averages for nodes a and b, then when A is selected, simulated and expanded, node a in C has its empirical average estimator updated using this newly observed data (during the Simulation of A) accumulated with the old data (from the Simulation of R). On the other hand however, for nodes c and d, the update only involves the newly observed data in b (during the Simulation of A). This leads to skewed estimators, $\hat{p}(a)$ overestimates $p(a)$ while $\hat{p}(c)$ and $\hat{p}(d)$ underestimate $p(c)$ and $p(d)$, and the greater the difference in depth between a and c,d, the greater the overestimation and underestimation effects are, we call this phenomenon **"Weights Degeneracy"**.
A second source of Weights Degeneracy is not copying the DT nodes, indeed $\hat{p}(a)$ is updated not only when A or C are simulated but also when D is simulated, in the latter case, the sufficient statistics of $\hat{p}(c)$ and $\hat{p}(d)$ are not updated at all, making the overestimation/underestimation wider!

We avoid both Weights Degeneracy causes by defining a new estimator based on the chain rule:

$$\begin{aligned}\mathbb{P}[X\in l] =& \mathbb{P}[X\in l, X\in \text{Parent}(l),\dots,X\in\text{root}]\\ =& \mathbb{P}[X\in l|X\in\text{Parent}(l)]\times\dots\times\mathbb{P}[X\in\text{root}]\end{aligned}$$

We estimate each term $\mathbb{P}[X\in\eta|X\in\text{Parent}(\eta)]$ with $\hat{p}(\eta|\text{Parent}(\eta)) = \frac{n(N,\eta)}{\sum_{\psi\in\text{Sib}(\eta)} n(N,\psi)}$, where $\text{Sib}(\eta)$ is the set of siblings of node η, including η itself, and $n(N,\eta)$ is the number of observed samples with inputs in η. This yields the product estimator:

$$\hat{p}(l) = \hat{p}(l|\text{Parent}(l))\times\dots\times 1$$

Since estimates $\hat{p}(\eta|\text{Parent}(\eta))$ only involve nodes at the same depth (η and its siblings), both sources of Weights Degeneracy are avoided, and by the Strong Law of Large numbers, we have the consistency:

$$\hat{p}(\eta|\text{Parent}(\eta)) \xrightarrow{\text{a.s}} \mathbb{P}[X\in\eta|X\in\text{Parent}(\eta)]$$

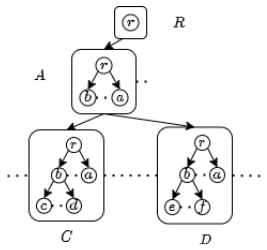

Figure 3: The Weights Degeneracy phenomenon.

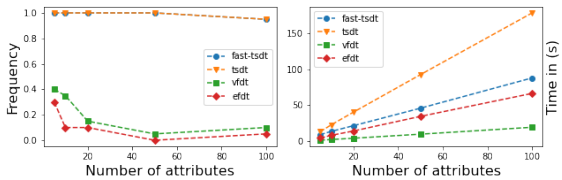

Figure 4: Comparison of VFDT, EFDT, TSDT and Fast-TSDT. Left: Frequency of perfect convergence; Right: Average running time in seconds.

As more and more samples are observed in Parent(η), and we deduce the consistency of our new estimator $\hat{p}(l) \xrightarrow[N\to\infty]{\text{a.s}} p(l)$.

5 EXPERIMENTS

Our first experiment highlights an important weakness of the classic greedy DT methods, and showcases how TSDT and Fast-TSDT circumvent this shortcoming. In the second experiment, we compare our methods with recent optimal batch DT algorithms DL8.5 and OSDT on standard real world benchmarks. All the experimental details and additional results are provided in Appendix B. Furthermore, our implemented code is available at https://github.com/Chaoukia/Thompson-Sampling-Decision-Trees along with the datasets we used.

We construct a challenging classification problem for greedy DT methods. In this setting, all the attributes are uninformative, in the sense that their true splitting gain metrics are all equal, making greedy methods VFDT and EFDT choose an attribute arbitrarily with tie-break. Concretely, We consider a binary classification problem with binary i.i.d. uniform attributes where $Y = 1$ if $X^{(1)} = 0, X^{(2)} = 0$ or $X^{(1)} = 1, X^{(2)} = 1$ and $Y = 0$ otherwise (Figure 6 provides a visualisation of the optimal DT). Attributes $X^{(3)},\dots,X^{(q)}$ are irrelevant, resulting in uniform class distributions in both leaves, regardless of the attribute chosen for the root split, even when selecting one of the relevant attributes $X^{(1)}$ or $X^{(2)}$. Consequently, VFDT and EFDT arbitrarily choose the attribute to

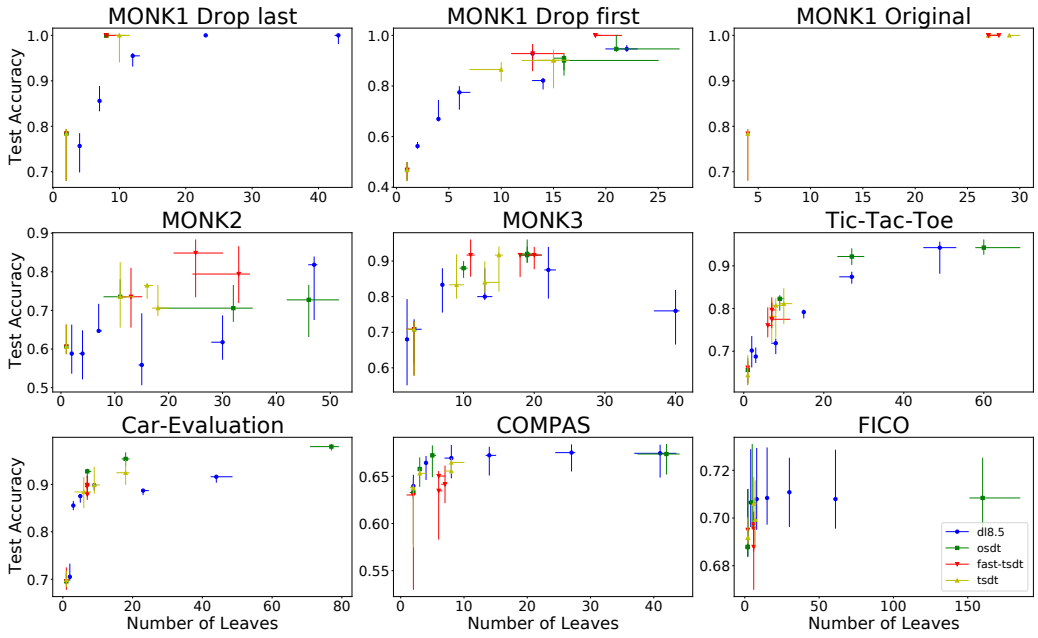

Figure 5: Cross-validation test accuracy comparison between TSDT, Fast-TSDT, OSDT and DL8.5 as a function of the number of leaves.

employ for the first split, and will continue with these arbitrary choices until attribute $X^{(1)}$ or $X^{(2)}$ is chosen, resulting in unnecessarily deep DTs. We compare VFDT, EFDT and TSDT on settings with different numbers of attributes $q = 5, 10, 20, 50, 100$. We perform 20 runs and report the frequency of perfect convergence, i.e, convergence to the optimal DT that only employs $X^{(1)}$ and $X^{(2)}$, and the average running time. Figure 4 presents a clear trend: VFDT and EFDT rarely attain perfect convergence, especially when q is large. In contrast, both TSDT and Fast-TSDT consistently achieve perfect convergence. Additionally, as anticipated, Fast-TSDT demonstrates superior computational efficiency compared to TSDT.

In our second experiment, we conduct a comparison between TSDT, Fast-TSDT, OSDT, and DL8.5, even though the latter two are batch algorithms. This choice is due to the absence of prior research on optimal online DTs, to the best of our knowledge. Furthermore, the source codes of both DL8.5 and OSDT are publicly available. Unfortunately, we could not include a comparison with the work by Nunes et al. (2018) as the authors have not made their code publicly accessible. In this regard, we also draw attention to (Nunes et al., 2018, Table 1), which demonstrates how that algorithm is prohibitively slow, taking up to 105 hours in an instance. It is worth noting that our experiments exclude GOSDT because our focus is on accuracy and complexity comparison, and GOSDT is primarily an extension of OSDT to other objective functions beyond accuracy. We follow the experimental protocol from (Hu et al., 2019) with its datasets. We perform a 5-fold crossvalidation with different values of the hyperparameters (maximum depth for DL8.5 and λ for OSDT and TSDT), and we report in Figure 5 the quartiles of the test accuracy and the number of leaves. For MONK1, Hu et al. (2019) utilised a One-Hot Encoding that excludes the last category of each attribute, which results in an optimal DT with 8 leaves. However, when the first category is dropped instead, the problem becomes significantly more challenging, with an optimal DT having over 18 leaves (further details are available in Appendix B). On this latter problem, Figure 5 clearly indicates that Fast-TSDT outperforms all other methods, followed by TSDT. In fact, Fast-TSDT is the only method that achieves 100% test accuracy with 19 leaves. Moreover, unlike OSDT and DL8.5, our methods do not necessitate binary attributes. Therefore, they can be directly applied to the original MONK1 data. In this scenario, the optimal DT representation of Y with the lowest complexity consists of 27 leaves, and it is successfully retrieved by Fast-TSDT. TSDT, on the other hand, identifies a slightly more complex DT with 28 leaves. For MONK2 and MONK3, Fast-TSDT demonstrates the best accuracy to number of leaves frontier. On the remaining datasets, our methods do not produce DTs with a large number of leaves, and OSDT performs slightly better. In Appendix B, we provide the execution times, DL8.5 is the fastest algorithm but also performs the least effectively, while OSDT and

TSDT frequently reach their time limit of 10 minutes. On the other hand, Fast-TSDT strikes a good balance between speed and performance.

6 CONCLUSIONS, LIMITATIONS AND FUTURE WORK

We devised TSDT, a new family of MCTS algorithms for constructing optimal online Decision Trees. We provided strong convergence results for our method and highlighted how it circumvents the sub-optimality issue of the standard methods. Furthermore, TSDT showcases similar or better accuracy-complexity trade-off compared to recent successful batch optimal DT algorithms, all while being tailored to handling data streams. For now, we are still limited to categorical attributes and our theoretical analysis only provides asymptotic convergence results. It would be desirable to derive some finite-time guarantees, in the form of PAC-bounds or rates of convergence, this is the aim of our future work. To conclude, this paper opens further possibilities for defining other MCTS algorithms with different policies, such as UCB and ϵ-greedy, in the context of optimal online DTs.

Acknowledgements

We thank Emilie Kaufmann for her insightful discussions, Otmane Sakhi for his helpful comments on the writing of the paper, and the anonymous reviewers for their valuable feedback.

References

Aglin, G., Nijssen, S., and Schaus, P. (2020). Learning optimal decision trees using caching branch-and-bound search. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3146–3153.

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256.

Bai, A., Wu, F., and Chen, X. (2013). Bayesian mixture modelling and inference based Thompson Sampling in Monte-Carlo tree search. *Advances in neural information processing systems*, 26.

Bennett, K. P. (1994). Global tree optimization: A non-greedy decision tree algorithm. *Computing Science and Statistics*, pages 156–156.

Bennett, K. P. and Blue, J. A. (1996). Optimal decision trees. *Rensselaer Polytechnic Institute Math Report*, 214:24.

Bertsimas, D. and Dunn, J. (2017). Optimal classification trees. *Machine Learning*, 106(7):1039–1082.

Bifet, A. and Gavalda, R. (2009). Adaptive learning from evolving data streams. In *International Symposium on Intelligent Data Analysis*, pages 249–260. Springer.

Bifet, A. and Kirkby, R. (2009). Data stream mining: a practical approach. *The university of Waikato*.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. (2012). A survey of Monte Carlo Tree Search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43.

Clark, C. E. (1961). The greatest of a finite set of random variables. *Operations Research*, 9(2):145–162.

Domingos, P. and Hulten, G. (2000). Mining high-speed data streams. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 71–80.

Dong, Z., Wang, C., and Ross, K. (2022). On the Convergence of Monte Carlo UCB for Random-Length Episodic MDPs. *arXiv preprint arXiv:2209.02864*.

Hu, X., Rudin, C., and Seltzer, M. (2019). Optimal sparse decision trees. *Advances in Neural Information Processing Systems*, 32.

Hulten, G., Spencer, L., and Domingos, P. (2001). Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 97–106.

Jin, R. and Agrawal, G. (2003). 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.

Kocsis, L. and Szepesvári, C. (2006). Bandit based Monte-Carlo planning. In *European conference on machine learning*, pages 282–293. Springer.

Kschischang, F. R. (2017). The complementary error function. *Online, April*.

Laurent, H. and Rivest, R. L. (1976). Constructing optimal binary decision trees is NP-complete. *Information processing letters*, 5(1):15–17.

Lin, J., Zhong, C., Hu, D., Rudin, C., and Seltzer, M. (2020). Generalized and scalable optimal sparse decision trees. In *International Conference on Machine Learning*, pages 6150–6160. PMLR.

Manapragada, C., Webb, G. I., and Salehi, M. (2018). Extremely fast decision tree. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1953–1962.

Norouzi, M., Collins, M., Johnson, M. A., Fleet, D. J., and Kohli, P. (2015). Efficient non-greedy optimization of decision trees. *Advances in neural information processing systems*, 28.

Nunes, C., De Craene, M., Langet, H., Camara, O., and Jonsson, A. (2018). A Monte Carlo Tree Search approach to learning decision trees. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 429–435. IEEE.

Quinlan, J. (2014). *C4. 5: programs for machine learning*. Elsevier.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

Rutkowski, L., Jaworski, M., Pietruczuk, L., and Duda, P. (2013). Decision trees for mining data streams based on the Gaussian approximation. *IEEE Transactions on Knowledge and Data Engineering*, 26(1):108–119.

Rutkowski, L., Pietruczuk, L., Duda, P., and Jaworski, M. (2012). Decision trees for mining data streams based on the mcdiarmid's bound. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1272–1279.

Sinha, D., Zhou, H., and Shenoy, N. V. (2007). Advances in computation of the maximum of a set of Gaussian random variables. *IEEE Transactions on Computer-Aided design of integrated circuits and systems*, 26(8):1522–1533.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Tesauro, G., Rajan, V. T., and Segal, R. (2010). Bayesian Inference in Monte-Carlo Tree Search. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, UAI'10, page 580–588, Arlington, Virginia, USA. AUAI Press.

Verwer, S. and Zhang, Y. (2019). Learning optimal classification trees using a binary linear program formulation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 1625–1632.

Wang, C., Yuan, S., Shao, K., and Ross, K. (2020). On the convergence of the Monte Carlo exploring starts algorithm for reinforcement learning. *arXiv preprint arXiv:2002.03585*.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes], Appendix for execution time.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes].

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes], Theorems 1 and 2.
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Not Applicable] all data are publicly available.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A Table of Notations

Table 1: Table of Notation

X	$=$	$\left(X^{(1)}, \ldots, X^{(q)}\right)$, the input.
Y	$\in$	$\{1, \ldots, K\}$, the class.
T	$\triangleq$	State, Decision Tree, Search Node.
$\overline{T}$	$\triangleq$	Terminal state that stems from taking the terminal action in T.
$T^{(l,i)}$	$\triangleq$	child Search Node of T that stems from splitting leaf $l \in \mathcal{L}(T)$ with respect to attribute $X^{(i)}$.
R	$\triangleq$	Root Decision Tree, initial state, root of the Search Tree.
$\mathcal{L}(T)$	$\triangleq$	Set of leaves of Decision Tree T.
$X \in l$	$\triangleq$	Event, the subset described by l contains X.
$p(l)$	$=$	$\mathbb{P}[X \in l]$, probability of $X \in l$.
$p_k(l)$	$=$	$\mathbb{P}[Y = k \vert X \in l]$, probability of $Y = k$ given $X \in l$.
$l(X)$	$\triangleq$	Leaf l such that $X \in l$.
$\mathcal{H}(T)$	$=$	$\mathbb{P}[T(X) = Y]$, accuracy of Decision Tree classifier T.
$T(X)$	$\triangleq$	$\text{argmax}_k\{p_k(l(X))\}$, predicted class of X according to DT T.
$\mathcal{H}_r(T)$	$\triangleq$	Regularized Objective function of Search Node T.
$\mathcal{H}_r(T)$	$=$	$\mathbb{P}[T(X) = Y] - \lambda \mathcal{S}(T)$.
$\mathcal{S}(T)$	$\triangleq$	Number of splits in DT T.
$T \to T'$	$\triangleq$	Transition from state T to state T'.
$\text{Ch}(T)$	$\triangleq$	Set of children of Search Node T, also set of next states from T.
$r(T, T')$	$\triangleq$	Reward of transition $T \to T'$.
π	$\triangleq$	Policy, maps each state T to a distribution over $\text{Ch}(T)$.
$\pi(T' \vert T)$	$\triangleq$	Probability of transition $T \to T'$ according to policy π.
$\pi(T)$	$\triangleq$	Next state from T according to policy π.
$T^{(0)} \xrightarrow{\pi} \ldots, \xrightarrow{\pi} T^{(N)}$	$\triangleq$	Episode that stems from following policy π starting from state T.
$\mathcal{V}^{\pi}(T)$	$\triangleq$	Value of following policy π starting from state T.
π^*	$\in$	$\text{Argmax}_{\pi} \mathcal{V}^{\pi}(R)$, optimal policy at the initial state R.
T^*	$\triangleq$	Optimal Decision Tree with respect to $\mathcal{H}_r$.
θ_T	$\triangleq$	Posterior distribution on $\mathcal{V}^{\pi^*}(T)$.
$\theta_{T,l}$	$\triangleq$	Posterior distribution on $\mathbb{E}[\mathbb{1}\{T(X) = Y\} \vert X \in l]$.
$\hat{p}(l)$	$\triangleq$	Estimator of $p(l)$.
$\hat{p}_k^{(i)}(l)$	$=$	$\frac{\sum_{j=1}^{i} \mathbb{1}\{X_j \in l\}\mathbb{1}\{Y_j = k\}}{\sum_{j=1}^{i} \mathbb{1}\{X_j \in l\}}$, estimator of $p_k(l)$.
$\hat{T}_i(l)$	$=$	$\text{Argmax}_k\{\hat{p}_k^{(i)}(l)\}$, estimator of $T(l)$.
$\alpha_{T,l}, \beta_{T,l}$	$=$	Parameters of random variable $\theta_{T,l}$.
$\mu_{T,l}, \sigma_{T,l}$	$=$	Mean and standard deviation of $\theta_{T,l}$ respectively.
μ_T, σ_T	$=$	Mean and standard deviation of θ_T respectively.
$n_{ijk}(N, \eta)$	$\triangleq$	Given observed samples $\{(X_s, Y_s)\}_{s=1}^{N}$, $n_{ijk}(N, \eta)$ is the number samples with $X_s \in \eta, Y_s = k$ satisfying $X_s^{(i)} = j$.

B Experiments

- All of the experiments were run on a personal Machine (2,6 GHz 6-Core Intel Core i7), they are easily reproducible.

- The codes for DL8.5 and OSDT are available at https://pypi.org/project/dl8.5/ and https://github.com/xiyanghu/OSDT.git respectively. We provide code for TSDT and Fast-TSDT at https://github.com/Chaoukia/Thompson-Sampling-Decision-Trees.

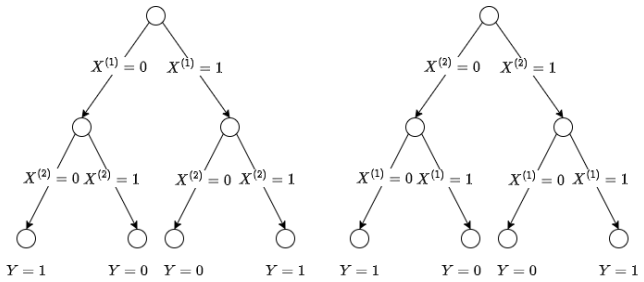

Figure 6: Equivalent representations of Y as a Decision Tree for the synthetic experiment.

- In practice, the variances in Equation (10) can collapse to 0 quickly, undermining exploration and slowing down the algorithm. We mitigate this issue by introducing an exponent $0 < \gamma < 1$ as follows:

$$(\sigma_{\overline{T}})^2 = \left(\sum_{l \in \mathcal{L}(T)} \hat{p}(l)^2 (\sigma_{T,l})^2 \right)^{\gamma}$$

  This is also discussed in (Kocsis and Szepesvári, 2006, Section 3.1). The authors of UCT introduce an exponent on the bias terms $c_{t,s}$ in practice.
  In all our experiments, we use $\gamma = 0.75$.

- During the Expansion step, creating all the children of the selected Search Node T can be computationally expensive. Therefore, whenever T is chosen, we expand it with respect to only one untreated leaf. Initially, all leaves of T are marked as untreated. Upon selecting T, we pick the untreated leaf $l \in \mathcal{L}(T)$ with the highest Gini impurity to prioritize exploring promising parts of the Search Tree. We then generate the subset of $\mathrm{Ch}(T)$ by considering all possible split actions exclusively with respect to leaf l, subsequently marking l as treated. We update π_t at T only when all leaves in $\mathcal{L}(T)$ have been treated. Only then, the children of T in $\mathrm{Ch}(T)$ become eligible to be considered for future Selection (and the subsequent) steps.

Synthetic Experiment: We use the default hyperparameters for VFDT and EFDT, for TSDT and Fast-TSDT, we set $M = 400, m = 100, \lambda = 0.05$.

Real World datasets: We ran both our methods with $M = 1000$ iterations for MONK1 Original and MONK1 Drop Last, and with $M = 10000$ iterations on the remaining datasets. All instances of TSDT and Fast-TSDT use $m = 100$ number of samples per Simulation step. To get our accuracy-complexity frontier figures, we run the experiments with multiple values of λ for TSDT, Fast-TSDT and OSDT, and different values of the depth limit for DL8.5.

- For DL8.5, the depth limits range from 1 to 6 exactly as reported by Lin et al. (2020).
- For TSDT, Fast-TSDT and OSDT, the set of λ values is $0.1, 0.01, 0.0025, 0.0001$, which is a subset of the values used by Lin et al. (2020). For all algorithms, we set a time limit of 10 minutes.

In Figure 8, OSDT reaches its time limit on all the datasets except MONK1, where the last category is dropped by the Binary Encoding. TSDT displays a similar behaviour but in less experiments than OSDT. On the other hand, while it is true that DL8.5 is clearly the fastest algorithm, it is also the algorithm that exhibits the least efficient accuracy-complexity frontier. In fact, when comparing the training accuracies in Figure 7 and the test accuracies in Figure 5, we notice a clear overfitting trend by DL8.5 unlike the remaining algorithms. To conclude, we argue that Fast-TSDT displays the best trade-off between optimality and execution time.

One of the main demonstrations of OSDTwas its Decision Tree solution to MONK1 Drop Last, and depicted in (Hu et al., 2019, Figure 6) in the comparison against BinOCT. Fast-TSDT consistently achieves the same solution with 8 leaves, represented in Figure 10, in much less time as documented in Figure 8. Moreover, as stated in Section 5, Hu et al. (2019) drop the last category of each attribute in their One-Hot Encoding of

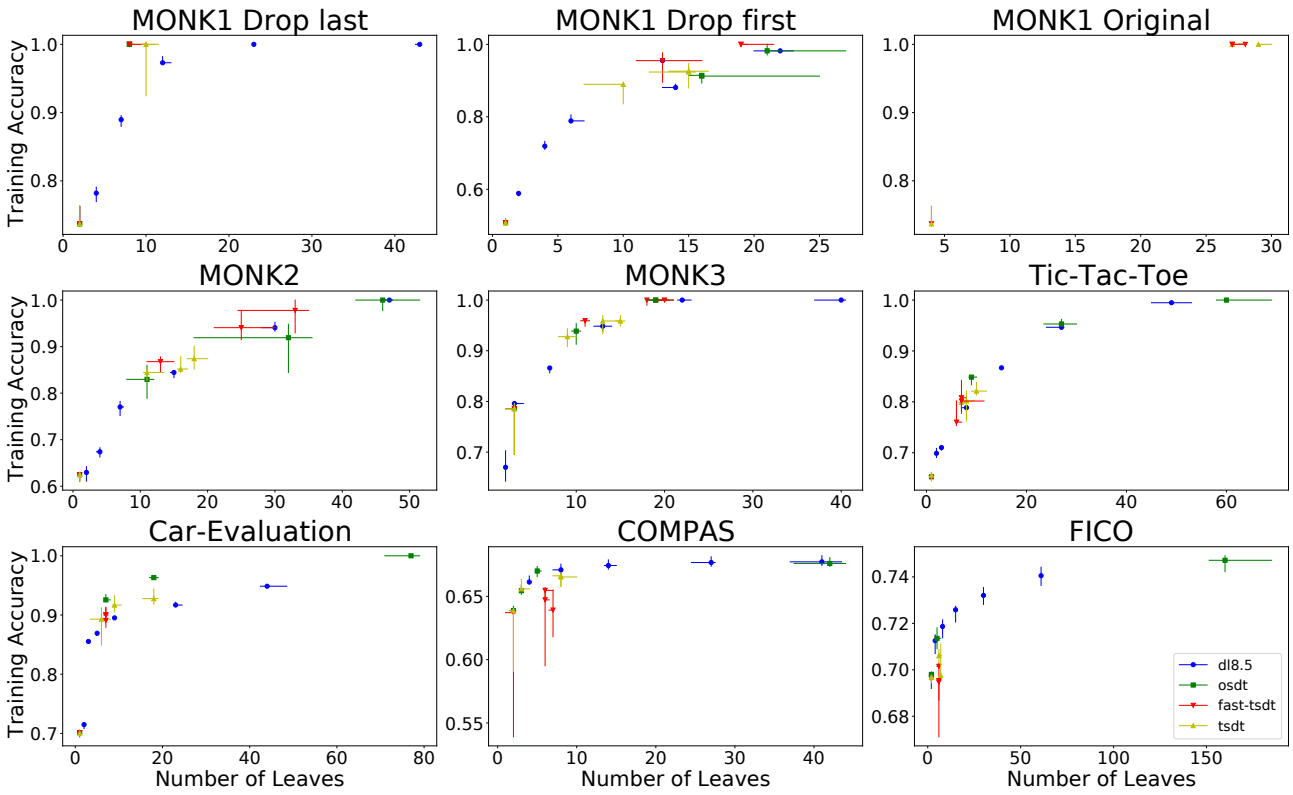

Figure 7: Cross-validation training accuracy of TSDT, OSDT and DL8.5 as a function of the number of leaves.

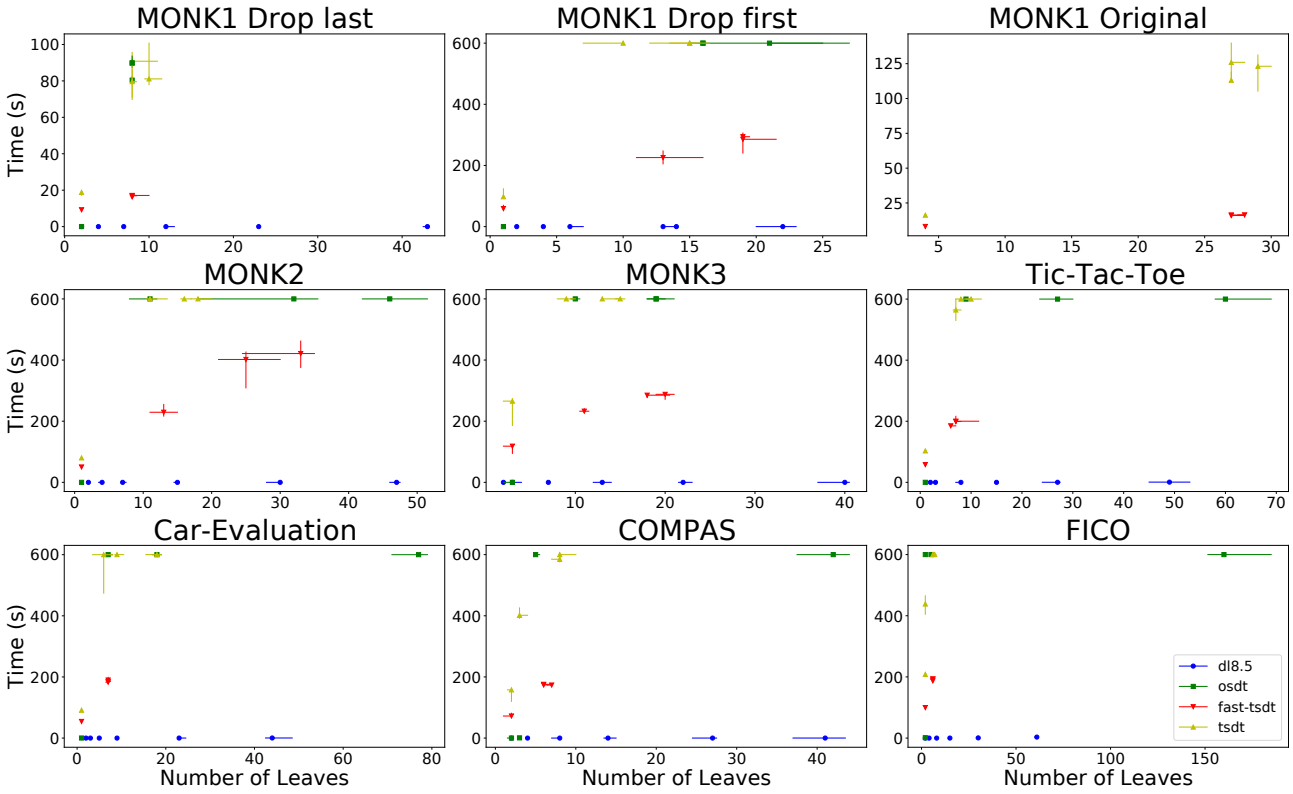

Figure 8: Cross-validation execution times of TSDT, OSDT and DL8.5 as a function of the number of leaves.

MONK1. However, without some form of prior knowledge, such choice is just arbitrary, and other options can be considered as well. The issue that arises from this is that these different options lead to solutions of different complexities as we explain in the following. In Figure 18, we aim at encoding some attribute variable X that takes three possible values (categories). One-Hot Encoding encodes X using two binary variables X_0 and X_1, if we drop the last category of X then we represent $X = 0$ with $X_0 = 1$, and $X = 1$ with $X_1 = 1$; but to encode $X = 2$ (the excluded category), we need $X_0 = 0, X_1 = 0$, which translates into a branch with two splits. Now, let us consider the following classification problem with $\mathbb{P}[Y = 1|X = 2] = 0, \mathbb{P}[Y = 1|X \neq 2] = 1$, then dropping the last category during One-Hot Encoding leads to an optimal Decision Tree with two splits, as represented in Figure 18, while dropping any other category instead leads to an optimal Decision Tree with only one split. In Monk 1, a similar phenomenon occurs where the choice of binary encoding has a significant impact on the complexity of the optimal DT. Indeed, dropping the last category leads to a simple solution with only 8 leaves, however, dropping the first category instead leads to a more complicated ad hard to find solution. In this case, Figures 7 and 5 clearly demonstrate that Fast-TSDT consistently achieves a better solution than DL8.5 and OSDT. Figure 11 illustrates a 19-leaf solution that Fast-TSDTretrieves, achieving 100% training and test accuracies, no similar solution has been found by DL8.5 and OSDT.

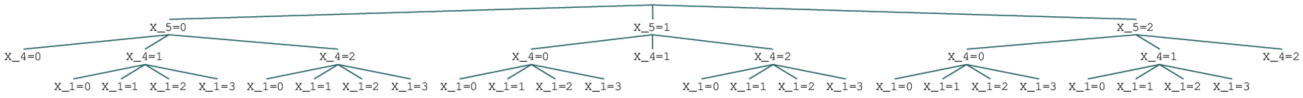

Figure 9: Fast-TSDT's solution to Monk 1 Original, the true optimal solution of least complexity.

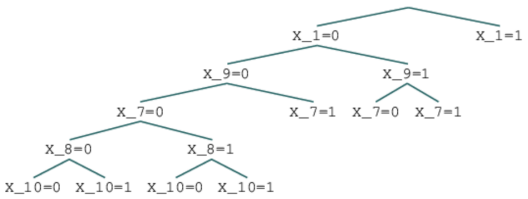

Figure 10: Fast-TSDT's solution to Monk 1 Drop Last, the true optimal solution of least complexity.

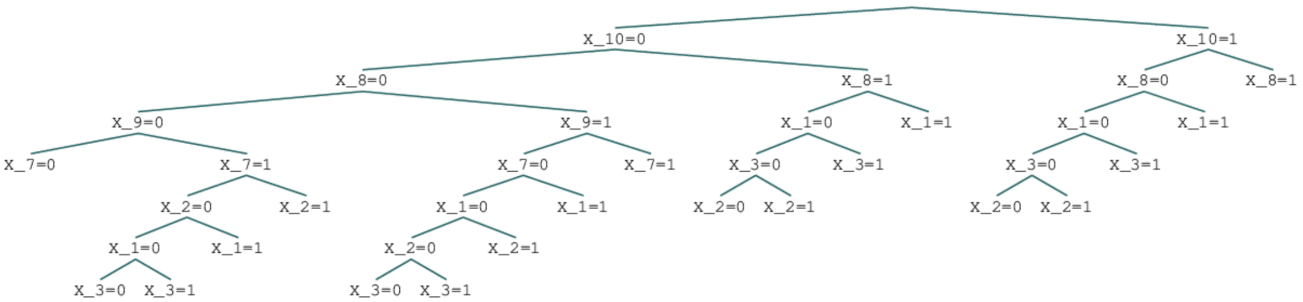

Figure 11: Fast-TSDT's solution to Monk 1 Drop First, this solution has 19 leaves and achieves 100% training and test accuracies; DL8.5 and OSDT were unsuccessful in retrieving such solution.

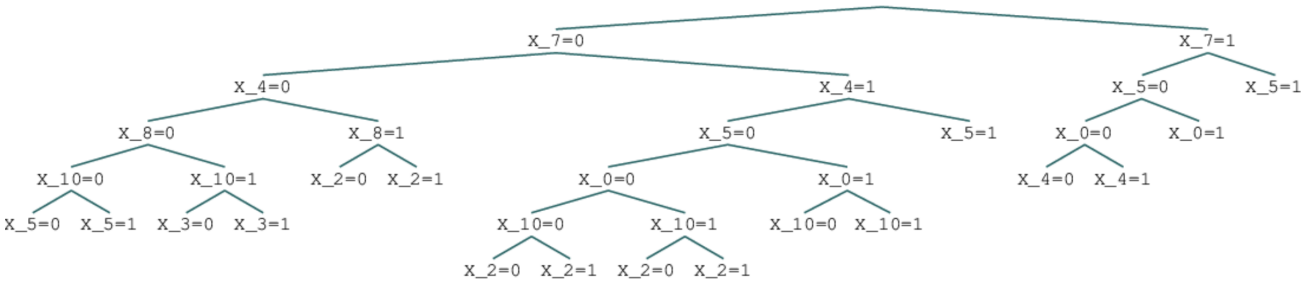

Figure 12: Fast-TSDT's solution to Monk 2, this solution has 17 leaves and achieves 89.6% training accuracy and 73.5% test accuracies.

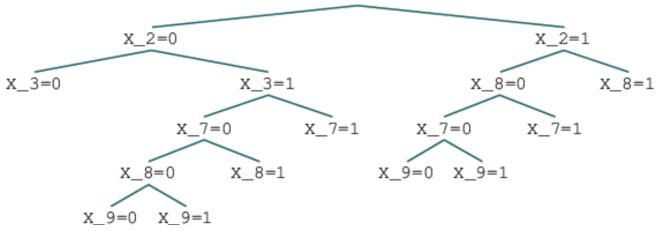

Figure 13: Fast-TSDT's solution to Monk 3, this solution achieves 93.8% training accuracy and 92% test accuracy.

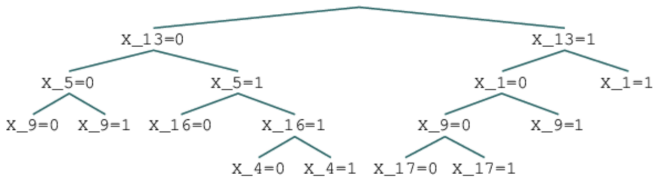

Figure 14: Fast-TSDT's solution to Tic-Tac-Toe, this solution achieves 79.8% training accuracy and 81.8% test accuracy.

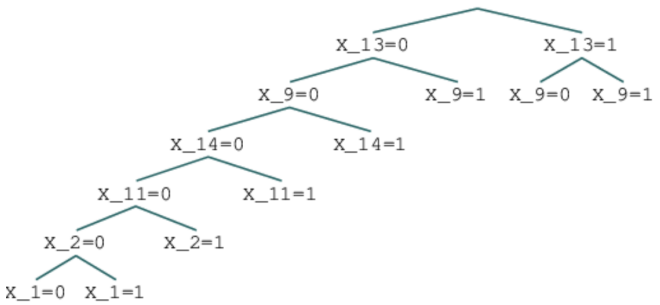

Figure 15: Fast-TSDT's solution to Car Evaluation, this solution achieves 89.9% training accuracy and 90.8% test accuracy.

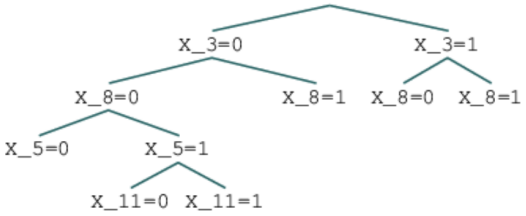

Figure 16: Fast-TSDT's solution to Compas, this solution achieves 66.4% training accuracy and 66% test accuracy.

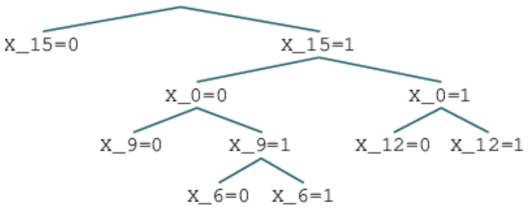

Figure 17: Fast-TSDT's solution to Fico, this solution achieves 69.4% training accuracy and 71.8% test accuracy.

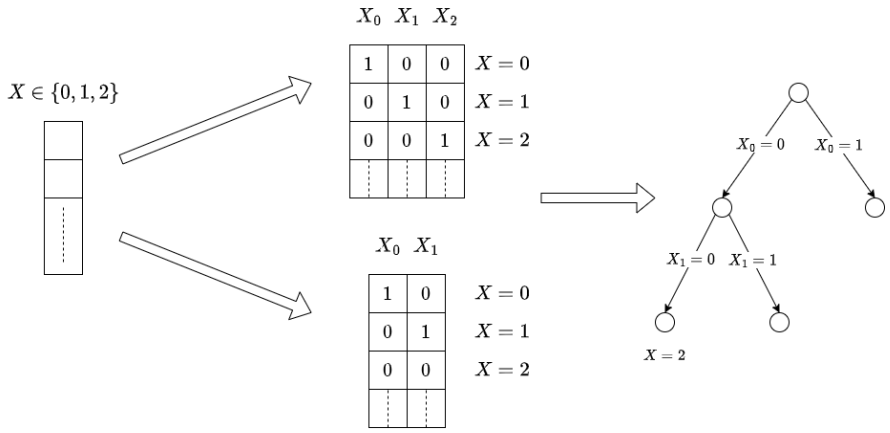

Figure 18: How the choice of which category to drop during One-Hot Encoding influences the resulting splits.

C TSDT's Backpropagation details

In this section, we present the details of the Backpropagation step performed by our first version of TSDT. Let T be an internal Search Node, which is an internal state, and let $\text{Ch}(T) = \{T_1, \dots, T_m\}$. We recall from Equation (11) that we have:

$$\theta_T = \max_{1 \le j \le m} \left\{ -\lambda \mathbb{1}\left\{T_j \neq \overline{T}\right\} + \theta_{T_j} \right\}$$

Following the Inductive reasoning in Section 4.2, we suppose that $\forall 1 \le j \le m : \theta_{T_j} \sim \mathcal{N}\left(\mu_{T_j}, \left(\sigma_{T_j}\right)^2\right)$. Now, θ_T is a maximum over Normal variables that are conditionally independent given the observed data $\{(X_i, Y_i)\}_{i=1}^N$ in T. We know that θ_T is not Normally distributed, but we can approximate its distribution with a Gaussian as is discussed in (Sinha et al., 2007). Let us first consider the case with two independent Normal variables $\theta_1 \sim \mathcal{N}\left(\mu_1, \sigma_1^2\right), \theta_2 \sim \mathcal{N}\left(\mu_2, \sigma_2^2\right)$. Let $\theta = \max\{\theta_1, \theta_2\}$, then the mean μ and variance σ^2 of θ satisfy:

$$\mu = \mu_1 \Phi(\alpha) + \mu_2 \Phi(-\alpha) + \phi(\alpha)\sigma_m \tag{13}$$

$$\sigma^2 = \left(\mu_1^2 + \sigma_1^2\right)\Phi(\alpha) + \left(\mu_2^2 + \sigma_2^2\right)\Phi(-\alpha) + (\mu_1 + \mu_2)\sigma_m \phi(\alpha) - \mu^2 \tag{14}$$

Where $\sigma_m = \sigma_1^2 + \sigma_2^2, \alpha = \frac{\mu_1 - \mu_2}{\sigma_m}$, and ϕ and Φ are respectively the probability density function and the cumulative distribution function of $\mathcal{N}(0,1)$ (see (Clark, 1961), (Tesauro et al., 2010)). The distribution of θ can be approximated with a Normal distribution by matching their first two moments, for short we call it Clark's approximation; Sinha et al. (2007) provide an error analysis of this approximation. This motivates rewriting θ_T as a nested pair-wise maximum:

$$\theta_T = \max\Big\{ -\lambda \mathbb{1}\{T_1 \neq \overline{T}\} + \theta_{T_1}, \max\{-\lambda \mathbb{1}\{T_2 \neq \overline{T}\} + \theta_{T_2}, \dots, \max\{-\lambda \mathbb{1}\{T_{m-1} \neq \overline{T}\} + \theta_{T_{m-1}}, \\ -\lambda \mathbb{1}\{T_m \neq \overline{T}\} + \theta_{T_m}\}\dots\Big\}$$

$\max\{-\lambda \mathbb{1}\Big\{T_{m-1} \neq \overline{T}\Big\} + \theta_{T_{m-1}}, -\lambda \mathbb{1}\Big\{T_m \neq \overline{T}\Big\} + \theta_{T_m}\}\}$ is a maximum of two (conditionally) independent Normal variables, thus we approximate its distribution as a Normal with Clark's approximation, and we recursively use Clark's approximation on the unfolding maximums of pairs to approximate the posterior distribution of θ_T as a Normal $\theta_T \sim \mathcal{N}\left(\mu_T, (\sigma_T)^2\right)$, μ_T and $(\sigma_T)^2$ are calculated with a recursive application of equations (13) and (14). A similar approximation was used by Tesauro et al. (2010) in the context of MCTS with a Bayesian approach, but the policy the authors considered is not Thompson Sampling but rather a form of UCB that uses the quantiles of the posterior distributions to derive the index of the UCB policy.

D Proofs

To prove Theorem 1, we use and inductive reasoning that starts from Search Leaves.

Lemma 3. *Let T be a Search Leaf and time t the number of times T is selected, then we have:*

$$\mu_T \xrightarrow[t\to\infty]{a.s} \mathcal{V}^{\pi^*}(T), (\sigma_T)^2 \xrightarrow[t\to\infty]{a.s} 0$$

Proof of Lemma 3. Let us start with the variance as it is a simpler result to prove. From the definition of $\alpha_T(t)$ and $\beta_T(t)$, we have $\alpha_T(t) + \beta_T(t) = tm + 1$, where we recall that m is the number of observed samples from the stream in a single Simulation step. Therefore we have:

$$\begin{aligned}(\sigma_T)^2 &= \frac{\alpha_T(t)\,\beta_T(t)}{(\alpha_T(t) + \beta_T(t))^2\,(\alpha_T(t) + \beta_T(t) + 1)} \\ &= \frac{\alpha_T(t)\,\beta_T(t)}{(tm+1)^2\,(tm+2)} \\ &\leq \frac{(tm+1)^2}{(tm+1)^2\,(tm+2)} \\ &\leq \frac{1}{tm+2}\end{aligned}$$

Therefore $(\sigma_T)^2$ converges, not only almost surely, but even surely to 0 as $t \to \infty$.

Now, let us show the result for the mean, we have:

$$\mu_T = \frac{\alpha_T(t)}{tm+1} = \frac{1}{1+tm} + \frac{1}{1+tm}\sum_{i=2}^{tm} \mathbb{1}\Big\{\hat{T}_{i-1}(X_i) = Y_i\Big\}$$

We know that, for any $l \in \mathcal{L}(T)$ such that $p(l) > 0$, which are the only leaves that can contain inputs X_i, the number of observed samples grows to infinity almost surely as $t \to \infty$, therefore, by the Strong Law of Large numbers we have:

$$\forall k \in \{1\dots,K\} : \hat{p}_k^{(i)}(l) \xrightarrow[i\to\infty]{\text{a.s}} p_k(l)$$

Which means that:

$$\mathbb{P}\left[\forall \epsilon > 0, \exists I > 0, \forall i \geq I, \forall k \in \{1,\dots,K\} : \left|\hat{p}_k^{(i)}(l) - p_k(l)\right| \leq \epsilon\right] = 1 \tag{15}$$

Take $\epsilon = \frac{1}{2}\min_{k\neq k'}\left\{\left|p_k(l) - p_{k'}(l)\right|\right\}$, in light of Equation (15) we have:

$$\begin{aligned}
&\mathbb{P}\left[\exists\tau>0, \forall i\geq\tau, \forall k\in\{1,\dots,K\} : \left|\hat{p}_k^{(i)}(l) - p_k(l)\right| \leq \epsilon\right] = 1\\
\implies &\mathbb{P}\left[\exists\tau>0, \forall i\geq\tau : \text{Argmax}_k\{p_k(l)\} = \text{Argmax}_k\{\hat{p}_k^{(i)}(l)\}\right] = 1\\
\implies &\mathbb{P}\left[\exists\tau>0, \forall i\geq\tau : \hat{T}_i(l) = T(l)\right] = 1 \qquad (16)
\end{aligned}$$

Let us define $\tau > 0$ the random time such that:

$$\forall i\geq\tau : \hat{T}_{i-1}(X_i) = T(X_i)$$

Since Equation (16) is satisfied for all leaves $l\in\mathcal{L}(T)$, we have $\mathbb{P}[\tau<\infty] = 1$.
Let $0<\epsilon'<\epsilon$, we want to show that:

$$\mathbb{P}\left[\exists\tau'>0, \forall t>\tau' : \left|\mu_T - \mathcal{V}^{\pi^*}(T)\right| \leq \epsilon'\right] = 1$$

By marginalising over $\tau>0$, we have:

$$\begin{aligned}
&\mathbb{P}\left[\exists\tau'>0, \forall t>\tau' : \left|\mu_T - \mathcal{V}^{\pi^*}(T)\right| \leq \epsilon'\right] \qquad (17)\\
&= \sum_{t'>0}\mathbb{P}\left[\exists\tau'>0, \forall t>\tau' : \left|\mu_T - \mathcal{V}^{\pi^*}(T)\right| \leq \epsilon' \middle| \tau = t'\right]\mathbb{P}[\tau = t'] \qquad (18)
\end{aligned}$$

Note that in this marginalisation, the term $\mathbb{P}[\tau=\infty]$ is absent because $\mathbb{P}[\tau=\infty] = 0$.
From the definition of τ, we can write μ_T as follows:

$$\forall t>\tau : \mu_T = \frac{1}{1+tm} + \frac{1}{1+tm}\sum_{i=2}^{\tau m}\mathbb{1}\left\{\hat{T}_{i-1}(X_i) = Y_i\right\} + \frac{1}{1+tm}\sum_{i=\tau m+1}^{tm}\mathbb{1}\left\{T(X_i) = Y_i\right\}$$

Now, given $\tau = t'$, then for all $t>t'$ we have the following:

$$\left|\mu_T - \mathcal{V}^{\pi^*}(T)\right| \leq \left|\frac{1}{1+tm} + \frac{1}{1+tm}\sum_{i=2}^{t'm}\mathbb{1}\left\{\hat{T}_{i-1}(X_i) = Y_i\right\}\right| + \left|\frac{1}{1+tm}\sum_{i=t'm+1}^{tm}\mathbb{1}\left\{T(X_i) = Y_i\right\} - \mathcal{V}^{\pi^*}(T)\right|$$

For the first term of the RHS, we have:

$$\left|\frac{1}{1+tm} + \frac{1}{1+tm}\sum_{i=2}^{t'm}\mathbb{1}\left\{\hat{T}_{i-1}(X_i) = Y_i\right\}\right| \leq \left|\frac{1}{1+tm} + \frac{t'm-1}{1+tm}\right| \underset{t\to\infty}{\longrightarrow} 0$$

Hence, there exists $\tau_1>0$ such that:

$$\forall t>\tau_1 : \left|\frac{1}{1+tm} + \frac{1}{1+tm}\sum_{i=2}^{t'm}\mathbb{1}\left\{\hat{T}_{i-1}(X_i) = Y_i\right\}\right| \leq \frac{\epsilon'}{3}$$

For the second term of the RHS, we bound it first as follows:

$$\begin{aligned}
\left|\frac{1}{1+tm}\sum_{i=t'm+1}^{tm}\mathbb{1}\left\{T(X_i) = Y_i\right\} - \mathcal{V}^{\pi^*}(T)\right| \leq &\left|\frac{1}{tm-t'm}\sum_{i=t'm+1}^{tm}\mathbb{1}\left\{T(X_i) = Y_i\right\} - \mathcal{V}^{\pi^*}(T)\right|\\
&+ \left|\left(\frac{1}{1+tm} - \frac{1}{tm-t'm}\right)\sum_{i=t'm+1}^{tm}\mathbb{1}\left\{T(X_i) = Y_i\right\}\right|
\end{aligned}$$

By the Strong Law of Large numbers, with probability 1, there exists $\tau_2>0$ such that:

$$\forall t>\tau_2 : \left|\frac{1}{tm-t'm}\sum_{i=t'm+1}^{tm}\mathbb{1}\left\{T(X_i) = Y_i\right\} - \mathcal{V}^{\pi^*}(T)\right| \leq \frac{\epsilon'}{3}$$

On the other hand:

$$\left|\left(\frac{1}{1+tm}-\frac{1}{tm-t'm}\right)\sum_{i=t'm+1}^{tm}\mathbb{1}\Big\{T\left(X_i\right)=Y_i\Big\}\right|\leq\frac{t'm+1}{1+tm}\underset{t\to\infty}{\longrightarrow}0$$

Therefore, there exists $\tau_3>0$ such that:

$$\forall t>\tau_3:\left|\left(\frac{1}{1+tm}-\frac{1}{tm-t'm}\right)\sum_{i=t'm+1}^{tm}\mathbb{1}\Big\{T\left(X_i\right)=Y_i\Big\}\right|\leq\frac{\epsilon'}{3}$$

Now take $\tau'=\max\{\tau_1,\tau_2,\tau_3\}$, then we have:

$$\left|\mu_T-\mathcal{V}^{\pi^*}\left(T\right)\right|\leq\frac{\epsilon'}{3}+\frac{\epsilon'}{3}+\frac{\epsilon'}{3}=\epsilon'$$

Therefore, conditionally on $\tau=t'$, with probability 1, there exists $\tau'>0$ such that $\forall t>\tau':\left|\mu_T-\mathcal{V}^{\pi^*}\left(T\right)\right|\leq\epsilon'$, i.e:

$$\mathbb{P}\left[\exists\tau'>0,\forall t>\tau':\left|\mu_T-\mathcal{V}^{\pi^*}\left(T\right)\right|\leq\epsilon'\middle|\tau=t'\right]=1$$

From Equation (18), we deduce that:

$$\begin{aligned}&\forall 0<\epsilon'<\epsilon:\mathbb{P}\left[\exists\tau'>0,\forall t>\tau':\left|\mu_T-\mathcal{V}^{\pi^*}\left(T\right)\right|\leq\epsilon'\right]=\sum_{t'>0}\mathbb{P}\left[\tau=t'\right]=1\\&\Longrightarrow\mathbb{P}\left[\forall 0<\epsilon'<\epsilon,\exists\tau'>0,\forall t>\tau':\left|\mu_T-\mathcal{V}^{\pi^*}\left(T\right)\right|\leq\epsilon'\right]=1\\&\Longrightarrow\mu_T\xrightarrow[t\to\infty]{\text{a.s}}\mathcal{V}^{\pi^*}\left(T\right)\end{aligned}$$

□

Lemma 4. *For any Search Node T with t the number of visits of its parent and $N_T(t)$ the number of times T has been visited up to time t, we have:*

$$N_T\left(t\right)\xrightarrow[t\to\infty]{a.s}\infty$$

Proof of Lemma 4. Let $P\left(T\right)$ denote the parent of T, for convenience, we denote $\text{Ch}\left(P\left(T\right)\right)=\{T_1,\ldots,T_n\}$ and $T=T_j$, now we want to show that $N_{T_j}\left(t\right)\xrightarrow[t\to\infty]{\text{a.s}}\infty$. To do so, we want to prove the following result: $\mathbb{P}\left[\theta_{T_j}\leq\max_{i\neq j}\{\theta_{T_i}\}\right]<1$ at all times t, i.e, the probability of choosing T_j is always non-zero. We consider the auxiliary problem with $\text{Ch}'\left(P\left(T\right)\right)=\{T_j,T'\}$ where

$$\theta_{T'}\sim\mathcal{N}\left(\mu_{T'},\left(\sigma_{T'}\right)^2\right),\mu_{T'}=\max_{i\neq j}\{\mu_{T_i}\}+f_n\left(t\right),\sigma_{T'}=\max_{i\neq j}\{\sigma_{T_i}\}$$

Where we will define $f_n\left(t\right)$ such that $\mathbb{P}\left[\theta_{T'}\geq\max_{i\neq j}\{\theta_{T_i}\}\right]\geq\frac{1}{2}$ at all times t. In this case, we have the following bound: $\mathbb{P}\left[\theta_{T_j}\leq\max_{i\neq j}\{\theta_{T_i}\}\right]\leq\mathbb{P}\left[\theta_{T_j}\leq\theta_{T'}\right]$. We use the union bound:

$$\mathbb{P}\left[\theta_{T'}\geq\max_{i\neq j}\{\theta_{T_i}\}\right]\geq1-\sum_{i\neq j}\mathbb{P}\left[\theta_{T'}<\theta_{T_i}\right]$$

Since $\forall i\neq j:\theta_{T'}-\theta_{T_i}\sim\mathcal{N}\left(\mu_{T'}-\mu_{T_i},\left(\sigma_{T'}\right)^2+\left(\sigma_{T_i}\right)^2\right)$, we have:

$$\begin{aligned}\mathbb{P}\left[\theta_{T'}<\theta_{T_i}\right]&=\frac{1}{2}\text{erfc}\left(\frac{\max_{k\neq j}\{\mu_{T_k}\}-\mu_{T_i}+f_n\left(t\right)}{\sqrt{2\left[\left(\sigma_{T'}\right)^2+\left(\sigma_{T_i}\right)^2\right]}}\right)\\&\leq\frac{1}{2}\text{erfc}\left(\frac{f_n\left(t\right)}{2\sigma_{T'}}\right)\end{aligned}$$

Hence:

$$\mathbb{P}\left[\theta_{T'} \geq \max_{i\neq j}\{\theta_{T_i}\}\right] \geq 1 - \frac{n-1}{2}\text{erfc}\left(\frac{f_n(t)}{2\sigma_{T'}}\right)$$

Thus, we want $f_n(t)$ satisfying $\text{erfc}\left(\frac{f_n(t)}{2\sigma_{T'}}\right) \leq \frac{1}{n-1}$.

Take $f_n(t) = g_n(t)\,\sigma_{T'}$, hence it suffices to take $g_n(t) = 2\text{erfc}^{-1}\left(\frac{1}{n-1}\right)$ and thus $f_n(t) = 2\sigma_{T'}\text{erfc}^{-1}\left(\frac{1}{n-1}\right)$.

On the other hand, we have the following:

$$\mathbb{P}\left[\theta_{T_j} > \theta_{T'}\right] = \frac{1}{\sqrt{\pi}}\int_{\frac{\mu_{T'}-\mu_{T_j}}{\sqrt{2\left[(\sigma_{T'})^2+\left(\sigma_{T_j}\right)^2\right]}}}^{\infty} e^{-u^2}du > 0$$

Therefore we deduce that:

$$\forall t > 0 : \mathbb{P}\left[\theta_{T_j} \leq \max_{i\neq j}\{\theta_{T_i}\}\right] \leq \mathbb{P}\left[\theta_{T_j} \leq \theta_{T'}\right] < 1$$

To show that $N_{T_j}(t) \xrightarrow[t\to\infty]{\text{a.s}} \infty$, we will equivalently prove:

$$\mathbb{P}\left[\exists M > 0, \forall t > 0 : N_{T_j}(t) < M\right] = 0$$

The event $\{\exists M > 0, \forall t > 0 : N_{T_j}(t) < M\}$ can be rewritten as the event $\{\exists \tau > 0, \forall t \geq \tau : \theta_{T_j} \leq \max_{i\neq j}\{\theta_i\}\}$, which means that there exists a time $\tau > 0$ such that, from then on, T_j will never be chosen again. Therefore:

$$\begin{aligned}
\mathbb{P}\left[\exists M > 0, \forall t > 0 : N_{T_j}(t) < M\right] &= \mathbb{P}\left[\exists \tau > 0, \forall t \geq \tau : \theta_{T_j} \leq \max_{i\neq j}\{\theta_i\}\right] \\
&\leq \sum_{\tau>0}\mathbb{P}\left[\forall t \geq \tau : \theta_{T_j} \leq \max_{i\neq j}\{\theta_i\}\right] \\
&\leq \sum_{\tau>0}\prod_{t\geq\tau}\mathbb{P}\left[\theta_{T_j} \leq \max_{i\neq j}\{\theta_i\}\right] \\
&\leq \sum_{\tau>0}\prod_{t\geq\tau}\mathbb{P}\left[\theta_{T_j} \leq \theta_{T'}\right]
\end{aligned}$$

The third line comes from the fact that $\{\theta_{T_i}\}$ are independent. We recall that for all $t \geq \tau$:

$$\mathbb{P}\left[\theta_{T_j} \leq \theta_{T'}\right] = \frac{1}{\sqrt{\pi}}\int_{-\infty}^{\frac{\mu_{T'}-\mu_{T_j}}{\sqrt{2\left[(\sigma_{T'})^2+\left(\sigma_{T_j}\right)^2\right]}}} e^{-u^2}du \leq \frac{1}{\sqrt{\pi}}\int_{-\infty}^{\frac{1+f_n(\tau)-\mu_{T_j}}{\sqrt{2}\sigma_{T_j}}} e^{-u^2}du < 1$$

This comes from the fact that $\forall 1 \leq i \leq n : 0 \leq \mu_i \leq 1$ and $f_n(t)$ is a decreasing function of t because $\sigma_{T'}$ decreases with t. Since $\forall t \geq \tau : T_j$ is not chosen, then μ_{T_j} and σ_{T_j} remain constant, and therefore:

$$\prod_{t\geq\tau}\frac{1}{\sqrt{\pi}}\int_{-\infty}^{\frac{1+f_n(\tau)-\mu_{T_j}}{\sqrt{2}\sigma_{T_j}}} e^{-u^2}du = 0$$

Thus we deduce that:

$$\mathbb{P}\left[\exists M > 0, \forall t > 0 : N_{T_j}(t) < M\right] = 0$$

Which concludes our proof. □

Corollary 5. *Let t be the number of iterations of* TSDT *, then the number of visits of any Search Node diverges almost surely to ∞ as $t \to \infty$.*

Proof of Corollary 5. Corollary 5 is straightforward to prove by Induction using Lemma 4 and the fact that t is the number of visits of the Root Search Node.

□

In what follows, let T denote an internal Search Node with $\text{Ch}(T) = \{T_1, \dots, T_n\}$ and T_1 the optimal child, i.e $\mathcal{V}^{\pi^*}(T_1) = \max_{1 \leq j \leq n} \{\mathcal{V}^{\pi^*}(T_j)\}$.

Lemma 6. *Let time t denote the number of times that T has been visited. Suppose that $\forall j \in \{1, \dots, n\}$: $\mathcal{V}^{\pi^*}(T_j) \xrightarrow[N_{T_j}(t) \to \infty]{a.s} \mu_{T_j}, \sigma_{T_j} \xrightarrow[N_{T_j}(t) \to \infty]{a.s} 0$, then we have:*

$$\lim_{t \to \infty} \pi_t(T_1|T) = 1$$

Note that we abuse the notation a little bit here. Indeed, in the main paper π_t is the policy after t iterations of TSDT, but here π_t denotes the policy after choosing T for t times.

Proof of Lemma 6. We define the following events at time t:

$$\mathcal{M}(t, \epsilon) = \bigcap_{j=1}^{n} \left\{ \left| \mu_{T_j} - \mathcal{V}^{\pi^*}(T_j) \right| \leq \epsilon \right\}$$
$$\mathcal{V}(t, \epsilon) = \bigcap_{j=1}^{n} \left\{ \left(\sigma_{T_j}\right)^2 < \frac{\epsilon}{2} \right\}$$

and $t(\epsilon) > 0$ the random time such that $\forall t > t(\epsilon) : \mathcal{M}(t, \epsilon)$ and $\mathcal{V}(t, \epsilon)$ happen. By Lemma 4, we have $\forall j \in \{1, \dots, n\} : N_{T_j}(t) \xrightarrow[t \to \infty]{\text{a.s}} \infty$, therefore $\mathbb{P}[t(\epsilon) < \infty] = 1$. Let $i(t)$ denote the chosen child at time t, we have $\mathbb{P}[i(t) = T_1] = \pi_t(T_1|T)$. The introduction of $i(t)$ is purely for convenience purposes. We write:

$$\begin{aligned}
\mathbb{P}[i(t) = T_1] &= \sum_{\tau > 0} \mathbb{P}\left[i(t) = T_1 \middle| t(\epsilon) = \tau \right] \mathbb{P}[t(\epsilon) = \tau] \\
&\geq \sum_{1 \leq \tau \leq t} \mathbb{P}\left[i(t) = T_1 \middle| t(\epsilon) = \tau \right] \mathbb{P}[t(\epsilon) = \tau]
\end{aligned}$$

Conditionally on $t(\epsilon) = \tau$, for $t \geq \tau$ we have the following:

$$\begin{aligned}
\mathbb{P}\left[i(t) = T_1 \middle| t(\epsilon) \right] &= \mathbb{P}\left[\forall j \neq 1 : \theta_{T_1} > \theta_{T_j} \middle| \mathcal{M}(t, \epsilon), \mathcal{V}(t, \epsilon) \right] \\
&\geq 1 - \sum_{j \neq 1} \mathbb{P}\left[\theta_{T_1} \leq \theta_{T_j} \middle| \mathcal{M}(t, \epsilon), \mathcal{V}(t, \epsilon) \right]
\end{aligned}$$

Before we continue, we introduce the following notation $\forall j \neq 1 : \Delta_j = \mathcal{V}^{\pi^*}(T_1) - \mathcal{V}^{\pi^*}(T_j) > 0, \Delta = \min_{j \neq 1} \Delta_j$. Let $C > 0$ and define $\epsilon = \frac{\Delta}{4C}$, then for all $j \neq 1$:

$$\begin{aligned}
&\mathbb{P}\left[\theta_{T_1} \leq \theta_{T_j} \middle| \mathcal{M}(t, \epsilon), \mathcal{V}(t, \epsilon) \right] \\
&\leq \mathbb{P}\left[\theta_{T_1} \leq \theta_{T_j} \middle| \left| \mu_{T_1} - \mathcal{V}^{\pi^*}(T_1) \right| < \frac{\Delta_j}{4C}, \left| \mu_{T_j} - \mathcal{V}^{\pi^*}(T_j) \right| < \frac{\Delta_j}{4C}, (\sigma_{T_1})^2 < \frac{\Delta_j}{8C}, \left(\sigma_{T_j}\right)^2 < \frac{\Delta_j}{8C} \right] \\
&\leq \frac{1}{\sqrt{\pi}} \int_{4C}^{\infty} e^{-u^2} du = \frac{1}{2} \text{erfc}(4C)
\end{aligned}$$

Hence, we deduce that:

$$\mathbb{P}\left[i(t) = T_1 \middle| t(\epsilon) \right] \geq 1 - \frac{|\text{Ch}(T)| - 1}{2} \text{erfc}(4C)$$

and thus:

$$\mathbb{P}[i(t) = T_1] \geq \left[1 - \frac{|\text{Ch}(T)| - 1}{2} \text{erfc}(4C) \right] \sum_{1 \leq \tau \leq t} \mathbb{P}[t(\epsilon) = \tau]$$

Since $\sum_{\tau \geq 1} \mathbb{P}[t(\epsilon) = \tau] = 1$ (because $\mathbb{P}[t(\epsilon) = \infty] = 0$), by taking the limit, we get:

$$\lim_{t \to \infty} \mathbb{P}[i(t) = T_1] \geq 1 - \frac{|\text{Ch}(T)| - 1}{2} \text{erfc}(4C)$$

Since this is satisfied for all $C > 0$ and $\lim_{C\to\infty} \operatorname{erfc}(4C) = 0$, we deduce that:

$$\lim_{t\to\infty} \mathbb{P}\left[i\left(t\right) = T_1\right] = \lim_{t\to\infty} \pi_t\left(T_1|T\right) = 1$$

□

Lemma 7. *Let time t denote the number of times that the parent of T has been visited. Under the same assumptions as Lemma 6, T satisfies:*

$$\mu_T \xrightarrow[t\to\infty]{a.s} \mathcal{V}^{\pi^*}\left(T\right), \left(\sigma_T\right)^2 \xrightarrow[t\to\infty]{a.s} 0$$

Proof of Lemma 7. By Induction on $n = |\mathrm{Ch}\left(T\right)|$, if $n = 2$ we have:

$$\begin{aligned}
\mu_T &= \mu_{T_1}\Phi\left(\alpha\right) + \mu_{T_2}\Phi\left(-\alpha\right) + \phi\left(\alpha\right)\sigma_m \\
\left(\sigma_T\right)^2 &= \left[\left(\mu_{T_1}\right)^2 + \left(\sigma_{T_1}\right)^2\right]\Phi\left(\alpha\right) + \left[\left(\mu_{T_2}\right)^2 + \left(\sigma_{T_2}\right)^2\right]\Phi\left(-\alpha\right) + \left(\mu_{T_1} + \mu_{T_2}\right)\sigma_m\phi\left(\alpha\right) - \left(\mu_T\right)^2
\end{aligned}$$

Where $\sigma_m = \left(\sigma_{T_1}\right)^2 + \left(\sigma_{T_2}\right)^2, \alpha = \frac{\mu_1 - \mu_2}{\sigma_m}$, and ϕ and Φ are respectively the probability density function and the cumulative distribution function of $\mathcal{N}\left(0,1\right)$. Using Lemma 4, we have $N_{T_1}\left(t\right) \xrightarrow[t\to\infty]{\text{a.s}} \infty$ and $N_{T_2}\left(t\right) \xrightarrow[t\to\infty]{\text{a.s}} \infty$, therefore $\mu_{T_1} \xrightarrow[t\to\infty]{\text{a.s}} \mathcal{V}^{\pi^*}\left(T_1\right), \sigma_{T_1} \xrightarrow[t\to\infty]{\text{a.s}} 0$ and $\mu_{T_2} \xrightarrow[t\to\infty]{\text{a.s}} \mathcal{V}^{\pi^*}\left(T_2\right), \sigma_{T_2} \xrightarrow[t\to\infty]{\text{a.s}} 0$, which yields $\sigma_m \xrightarrow[t\to\infty]{\text{a.s}} 0$ and $\alpha \xrightarrow[t\to\infty]{\text{a.s}} \infty$, using these results with the formulas above, we deduce that:

$$\mu_T \xrightarrow[t\to\infty]{\text{a.s}} \mathcal{V}^{\pi^*}\left(T_1\right) = \max\{\mathcal{V}^{\pi^*}\left(T_1\right), \mathcal{V}^{\pi^*}\left(T_2\right)\},\ \sigma_T \xrightarrow[t\to\infty]{\text{a.s}} 0$$

Now suppose the result holds true for some $n \geq 2$ and now consider $\mathrm{Ch}\left(T\right) = \{T_1, \ldots, T_{n+1}\}$. We define $\theta_{T'}\left(t\right) \sim \mathcal{N}\left(\mu_{T'}, \left(\sigma_{T'}\right)^2\right)$ which approximates recursively the maximum for $\{T_2, \ldots, T_{n+1}\}$. By the Induction hypothesis, we have:

$$\mu_{T'} \xrightarrow[t\to\infty]{\text{a.s}} \max\{\mathcal{V}^{\pi^*}\left(T_2\right), \ldots, \mathcal{V}^{\pi^*}\left(T_{n+1}\right)\},\ \sigma_{T'} \xrightarrow[t\to\infty]{\text{a.s}} 0$$

Now we have $\theta_T\left(t\right) \sim \mathcal{N}\left(\mu_T, \left(\sigma_T\right)^2\right)$ approximating the maximum for $\{T_1, T'\}$. By the Induction hypothesis again, we have:

$$\mu_T \xrightarrow[t\to\infty]{\text{a.s}} \mathcal{V}^{\pi^*}\left(T_1\right) = \max\{\mathcal{V}^{\pi^*}\left(T_1\right), \mathcal{V}^{\pi^*}\left(T'\right)\},\ \sigma_T \xrightarrow[t\to\infty]{\text{a.s}} 0$$

This concludes the Induction proof. □

Proof of Theorem 1. By backward induction starting from the Search Leaves and going up to the Root Search Node, using Lemmas 3, 4, 7 and Corollary 5, it is straightforward to deduce the main result.

□

Lemma 8. *Let T be a Search Leaf, with t the number of visits of its parent and $N_T\left(t\right)$ the number of visits of T up to time t. Then we have:*

$$\forall C > |\mathcal{L}\left(T\right)| : N_T\left(t\right) m \leq \frac{1}{2}\left(\frac{C}{|\mathcal{L}\left(T\right)|}\right)^{\frac{1}{2}} \implies \left(\sigma_T\right)^2 \geq \frac{1}{C}$$

Proof of Lemma 8. We have:

$$\left(\sigma_T\right)^2 = \sum_l \left(\hat{p}\left(l\right)\right)^2 \left(\sigma_{T,l}\right)^2; \left(\sigma_{T,l}\right)^2 = \frac{\alpha_{T,l}\beta_{T,l}}{\left(\alpha_{T,l} + \beta_{T,l}\right)^2\left(1 + \alpha_{T,l} + \beta_{T,l}\right)}$$

Let $n_{T,l} = \sum_{i=2}^{t} \mathbb{1}\{X_i \in l\}$ the number of samples effectively observed in leaf l of T up to the t^{th} visit of the parent of T. Then from Equation (5) and (6), we have the following $\alpha_{T,l} + \beta_{T,l} = 2 + n_{T,l}$ and $\alpha_{T,l}\beta_{T,l} = \alpha_{T,l}\left(2 + n_{T,l} - \alpha_{T,l}\right)$ knowing that $1 \leq \alpha_{T,l} \leq 1 + n_{T,l}$.

Consider the function $f : x \mapsto x\left(2 + n_{T,l} - x\right)$ defined on $[1, 1 + n_{T,l}]$, f has a minimum $f(1) = f\left(1 + n_{T,l}\right) = 1 + n_{T,l}$, and therefore:

$$\left(\sigma_{T,l}\right)^2 \geq \frac{1 + n_{T,l}}{\left(2 + n_{T,l}\right)^2 \left(3 + n_{T,l}\right)}$$

Now consider the function $g : x \mapsto \frac{1+x}{(2+x)^2(3+x)}$ defined for $x \geq 0$, g is differentiable on $\mathbb{R}_+^*$ and $g'(x) = \frac{-2-6x-2x^2}{(2+x)^3(3+x)^2} < 0$, hence g is decreasing. Therefore, for $\mathcal{C} > n_{T,l}$ we get:

$$\left(\sigma_{T,l}\right)^2 \geq g\left(n_{T,l}\right) \geq g\left(\mathcal{C}\right)$$

Furthermore, the total number of observed samples in T is obviously larger than the number of observed samples in leaf $l \in \mathcal{L}(l)$, thus $n_{T,l} \leq N_T(t)\, m \leq \frac{1}{2}\sqrt{\frac{C}{|\mathcal{L}(T)|}}$. Let us choose $\mathcal{C} > 5$, this leads to:

$$\begin{aligned}
\left(\sigma_T\right)^2 &\geq g\left(\mathcal{C}\right) \sum_{l \in \mathcal{L}(T)} \hat{p}\left(l\right) \\
\left(\sigma_T\right)^2 &\geq \frac{1}{|\mathcal{L}(T)|} \frac{1 + \mathcal{C}}{\left(2 + \mathcal{C}\right)^2 \left(3 + \mathcal{C}\right)} \\
&\geq \frac{1}{|\mathcal{L}(T)|} \frac{\mathcal{C}}{\left(\sqrt{2}\mathcal{C}\right)^2 \left(2\mathcal{C}\right)} \\
&\geq \frac{1}{4|\mathcal{L}(T)|\mathcal{C}^2}
\end{aligned}$$

The second inequality comes from the fact that the uniform distribution minimises the collision probability.

By taking $\mathcal{C} = \frac{1}{2}\left(\frac{C}{|\mathcal{L}(T)|}\right)^{\frac{1}{2}}$, we deduce the result of the Lemma.

□

Proof of Theorem 2. Let us first consider the case with $n = 2$, i.e $\mathrm{Ch}(T) = \{T_1, T_2\}$ and then we will generalise the result for an arbitrary $n \geq 2$. We have $M_2 = 1$, thus we want to show that:

$$\mathbb{P}\left[N_{T_2}(t)\, m \leq \frac{\log t}{4|\mathcal{L}(T_2)|}\right] \leq \exp\left[-\frac{2}{t}\left(\frac{t^{3/4}}{\sqrt{\pi}\left(\sqrt{\frac{\log t}{4}} + \sqrt{\frac{\log t}{4} + 2}\right)} - \frac{\log t}{4m|\mathcal{L}(T_2)|}\right)^2\right]$$

The result will be valid for T_1 as well without loss of generality.

At time t, child T_2 is chosen if $\theta_{T_2} \geq \theta_{T_1}$, which motivates us to study $\mathbb{P}\left[\theta_{T_2} \geq \theta_{T_1}\right]$:
We know that $\theta_{T_2} - \theta_{T_1} \sim \mathcal{N}\left(\mu_{T_2} - \mu_{T_1}, \left(\sigma_{T_2}\right)^2 + \left(\sigma_{T_1}\right)^2\right)$, hence:

$$\mathbb{P}\left[\theta_{T_2} \geq \theta_{T_1}\right] = \mathbb{P}\left[\theta_{T_2} \geq \theta_{T_1} \middle| \mu_{T_2} < \mu_{T_1}\right] \mathbb{P}\left[\mu_{T_2} < \mu_{T_1}\right] + \mathbb{P}\left[\theta_{T_2} \geq \theta_{T_1} \middle| \mu_{T_2} \geq \mu_{T_1}\right] \mathbb{P}\left[\mu_{T_2} \geq \mu_{T_1}\right]$$

Since $\mathbb{P}\left[\theta_{T_2} \geq \theta_{T_1} \middle| \mu_{T_2} < \mu_{T_1}\right] \leq \mathbb{P}\left[\theta_{T_2} \geq \theta_{T_1} \middle| \mu_{T_2} \geq \mu_{T_1}\right]$, we have:

$$\begin{aligned}
\mathbb{P}\left[\theta_{T_2} \geq \theta_{T_1}\right] &\geq \mathbb{P}\left[\theta_{T_2} \geq \theta_{T_1} \middle| \mu_{T_2} < \mu_{T_1}\right] \\
&\geq \frac{1}{\sqrt{\pi}} \int_{\frac{\mu_{T_1} - \mu_{T_2}}{\sqrt{2\left[\left(\sigma_{T_1}\right)^2 + \left(\sigma_{T_2}\right)^2\right]}}}^{\infty} e^{-u^2} du \\
&\geq \frac{1}{\sqrt{\pi}} \int_{\frac{\mu_{T_1} - \mu_{T_2}}{\sqrt{2\left(\sigma_{T_2}\right)^2}}}^{\infty} e^{-u^2} du
\end{aligned}$$

In what follows, we consider $N_{T_2}(t)\, m \leq \frac{C}{2|\mathcal{L}(T)|}$, and we will define C as a function of t later. According to Lemma 8 we have $(\sigma_{T_2})^2 \geq \frac{|\mu_{T_1}-\mu_{T_2}|}{C}$, thus $\frac{\mu_{T_1}-\mu_{T_2}}{\sqrt{2(\sigma_{T_2})^2}} \leq \sqrt{\frac{C(\mu_{T_1}-\mu_{T_2})}{2}} \leq \sqrt{\frac{C}{2}}$ since by definition, all the means $\mu_{T_i} \in [0,1]$. This leads to:

$$\begin{aligned}\mathbb{P}\left[\theta_{T_2} \geq \theta_{T_1} \middle| (\sigma_{T_2})^2 \geq \frac{|\mu_{T_1}-\mu_{T_2}|}{C}\right] &\geq \frac{1}{\sqrt{\pi}} \int_{\sqrt{\frac{C}{2}}}^{\infty} e^{-u^2} du \\ &\geq \frac{1}{2}\text{erfc}\left(\sqrt{\frac{C}{2}}\right)\end{aligned}$$

erfc (.) denotes the complementary error function. Using the lower bound in (Kschischang, 2017), we deduce that:

$$\mathbb{P}\left[\theta_{T_2} \geq \theta_{T_1} \middle| (\sigma_{T_2})^2 \geq \frac{|\mu_{T_1}-\mu_{T_2}|}{C}\right] \geq \frac{1}{2}\text{erfc}\left(\sqrt{\frac{C}{2}}\right) \geq \frac{\exp\left(\frac{-C}{2}\right)}{\sqrt{\pi}\left(\sqrt{\frac{C}{2}}+\sqrt{\frac{C}{2}+2}\right)} \tag{19}$$

Since $\forall t' \leq t : N_{T_2}(t') \leq N_{T_2}(t)$, we have $N_{T_2}(t)\, m \leq \frac{C}{2|\mathcal{L}(T)|} \implies N_{T_2}(t')\, m \leq \frac{C}{2|\mathcal{L}(T)|} \implies (\sigma_{T_2})^2 \geq \frac{\left|\mu_{T_1}-\mu_{T_2}\right|}{C}$, and Inequality (19) holds for all $1 \leq t' \leq t$. Hence we write:

$$\begin{aligned}\mathbb{P}\left[N_{T_2}(t)\, m \leq \frac{C}{2|\mathcal{L}(T_2)|}\right] &= \mathbb{P}\left[N_{T_2}(t)\, m \leq \frac{C}{2|\mathcal{L}(T_2)|}, (\sigma_{T_2})^2 \geq \frac{|\mu_{T_2}-\mu_{T_1}|}{C}\right] \\ &= \mathbb{P}\left[N_{T_2}(t)\, m \leq \frac{C}{2|\mathcal{L}(T_2)|}, \forall 1 \leq t' \leq t : (\sigma_{T_2})^2 \geq \frac{\left|\mu_{T_2}-\mu_{T_1}\right|}{C}\right] \\ &\leq \mathbb{P}\left[N_{T_2}(t)\, m \leq \frac{C}{2|\mathcal{L}(T_2)|} \middle| \forall 1 \leq t' \leq t : (\sigma_{T_2})^2 \geq \frac{\left|\mu_{T_2}-\mu_{T_1}\right|}{C}\right]\end{aligned}$$

Given the event $\left\{\forall 1 \leq t' \leq t : (\sigma_{T_2})^2 \geq \frac{\left|\mu_{T_2}-\mu_{T_1}\right|}{C}\right\}$, at each time $1 \leq t' \leq t : T_2$ is chosen with probability $\mathbb{P}\left[\theta_{T_2} \geq \theta_{T_1} \middle| (\sigma_{T_2})^2 \geq \frac{\left|\mu_{T_2}-\mu_{T_1}\right|}{C}\right]$.

Let $\mathcal{C} = \frac{\exp\left(\frac{-C}{2}\right)}{\sqrt{\pi}\left(\sqrt{\frac{C}{2}}+\sqrt{\frac{C}{2}+2}\right)}$ and define the i.i.d. random variables $Z_1, \ldots, Z_t$ such that $\forall i : Z_i \sim \text{Bernoulli}(\mathcal{C})$. Inequality (19) and Lemma 8 lead to:

$$\begin{aligned}\mathbb{P}\left[N_{T_2}(t)\, m \leq \frac{C}{2|\mathcal{L}(T_2)|}\right] &= \mathbb{P}\left[N_{T_2}(t)\, m \leq \frac{C}{2|\mathcal{L}(T_2)|}, \forall 1 \leq t' \leq t : (\sigma_{T_2})^2 \geq \frac{\left|\mu_{T_2}-\mu_{T_1}\right|}{C}\right] \\ &\leq \mathbb{P}\left[N_{T_2}(t)\, m \leq \frac{C}{2|\mathcal{L}(T_2)|} \middle| \forall 1 \leq t' \leq t : (\sigma_{T_2})^2 \geq \frac{\left|\mu_{T_2}-\mu_{T_1}\right|}{C}\right] \\ &\leq \mathbb{P}\left[\sum_{i=1}^{t} Z_i \leq \frac{C}{2m|\mathcal{L}(T_2)|}\right]\end{aligned}$$

Using Hoeffding's inequality, for $\epsilon > 0$ we have $\mathbb{P}\left[\sum_{i=1}^{t} Z_i - t\mathcal{C} \leq -\epsilon\right] \leq \exp\left(-\frac{2\epsilon^2}{t}\right)$.

Thus, by setting $\epsilon = t\mathcal{C} - \frac{C}{2m|\mathcal{L}(T_2)|} > 0$, we deduce:

$$\begin{aligned}\mathbb{P}\left[N_{T_2}(t)\, m \leq \frac{C}{2|\mathcal{L}(T_2)|}\right] &\leq \mathbb{P}\left[\sum_{i=1}^{t} Z_i \leq \frac{C}{2m|\mathcal{L}(T_2)|}\right] \\ &\leq \exp\left(-\frac{2\left(t\mathcal{C} - \frac{C}{2m|\mathcal{L}(T_2)|}\right)^2}{t}\right)\end{aligned}$$

Now let us find an adequate expression of C as a function of t, hence we will write $\mathcal{C}(t), C(t)$.
We recall that $\epsilon = t\mathcal{C} - \frac{C(t)}{2m|\mathcal{L}(T_2)|} > 0$, thus a first condition is to have $C(t) < 2m|\mathcal{L}(T_2)|t\mathcal{C}$, which means that $C(t)$ has to be sublinear.
Recall that $\mathcal{C}(t) = \frac{\exp\left(\frac{-C(t)}{2}\right)}{\sqrt{\pi}\left(\sqrt{\frac{C(t)}{2}}+\sqrt{\frac{C(t)}{2}+2}\right)}$, thus:

$$C(t) < 2m|\mathcal{L}(T_2)|t\frac{\exp\left(\frac{-C(t)}{2}\right)}{\sqrt{\pi}\left(\sqrt{\frac{C(t)}{2}}+\sqrt{\frac{C(t)}{2}+2}\right)}$$

$$\frac{\sqrt{\pi}}{2m|\mathcal{L}(T_2)|}C(t)\left(\sqrt{\frac{C(t)}{2}}+\sqrt{\frac{C(t)}{2}+2}\right) < t\exp\left(-\frac{C(t)}{2}\right)$$

For any $\alpha > 0$, we cannot have $C(t) = t^\alpha$ because the RHS would converge to 0 as $t \to \infty$ while the LHS would diverge to ∞. Hence, we consider $C(t) = a\log t$ for some $a > 0$.

$$\begin{aligned}\frac{\sqrt{\pi}}{2m|\mathcal{L}(T_2)|}a\log(t)\left(\sqrt{\frac{a\log t}{2}}+\sqrt{\frac{a\log t}{2}+2}\right) &< t\exp\left(-\frac{a\log t}{2}\right) \\ &< t^{1-\frac{a}{2}}\end{aligned}$$

Thus we must have $0 < a < 2$.

$$\mathbb{P}\left[N_{T_2}(t)\, m \leq \frac{a\log t}{2|\mathcal{L}(T_2)|}\right] \leq \exp\left(-\frac{2\left(\frac{t^{1-\frac{a}{2}}}{\sqrt{\pi}\left(\sqrt{\frac{a\log t}{2}}+\sqrt{\frac{a\log t}{2}+2}\right)} - \frac{a\log t}{2m|\mathcal{L}(T_2)|}\right)^2}{t}\right)$$

Since $\frac{2}{t}\left(\frac{t^{1-\frac{a}{2}}}{\sqrt{\pi}\left(\sqrt{\frac{a\log t}{2}}+\sqrt{\frac{a\log t}{2}+2}\right)} - \frac{a\log t}{2m|\mathcal{L}(T_2)|}\right)^2 = \mathcal{O}\left(t^{1-a}\right)$, we must have $0 < a < 1$; by taking $a = \frac{1}{2}$, we get:

$$\mathbb{P}\left[N_{T_2}(t)\, m \leq \frac{\log t}{4|\mathcal{L}(T_2)|}\right] \leq \exp\left[-\frac{2}{t}\left(\frac{t^{3/4}}{\sqrt{\pi}\left(\sqrt{\frac{\log t}{4}}+\sqrt{\frac{\log t}{4}+2}\right)} - \frac{\log t}{4m|\mathcal{L}(T_2)|}\right)^2\right]$$

Following the exact same steps, we can show that if $|\mu_{T_2} - \mu_{T_1}| \leq M$ where $M > 0$ some constant, we would have:

$$\mathbb{P}\left[N_{T_2}(t)\, m \leq \frac{\log t}{4|\mathcal{L}(T_2)|M}\right] \leq \exp\left[-\frac{2}{t}\left(\frac{t^{3/4}}{\sqrt{\pi}\left(\sqrt{\frac{\log t}{4}}+\sqrt{\frac{\log t}{4}+2}\right)} - \frac{\log t}{4m|\mathcal{L}(T_2)|M^2}\right)^2\right]$$

This result constitutes our induction hypothesis for the following generalisation.

Let us now address the setting with $\mathrm{Ch}(T) = \{T_1, \dots, T_n\}$ where $n \geq 2$, and let $i \in \{1, \dots, n\}$. Our idea is to transform this problem into a problem with two children and use the induction hypothesis.
We consider a new child T' with parameters $\theta_{T'} \sim \mathcal{N}\left(\mu_{T'}, (\sigma_{T'})^2\right)$ such that:

$$\begin{aligned}\mu_{T'} &= \max_{j \neq i}\{\mu_{T_j}\} + f_n(t)\\ \sigma_{T'} &= \max_{j \neq i}\{\sigma_{T_j}\}\end{aligned}$$

$f_n(t)$ is a function that we will derive later on.
Consider the setting with the new set of children $\mathrm{Ch}'(T) = \{T_i', T'\}$ where $\theta_{T_i} = \theta_{T_i'}$ and $|\mathcal{L}(T_i)| = |\mathcal{L}(T_i')|$.
For any $C > 0$, we want $\mathbb{P}[N_{T_i}(t)\, m \leq C] \leq \mathbb{P}\left[N_{T_i'}(t)\, m \leq C\right]$, to achieve this, it suffices to have $\mathbb{P}\left[\theta_{T'} \geq \max_{j \neq i}\{\theta_{T_j}\}\right] \geq \frac{1}{2}$ because it means that the probability of choosing T_i' in the problem with $\mathrm{Ch}'(T)$ is lower than the probability of choosing T_i in the problem with $\mathrm{Ch}(T)$, which leads to $\mathbb{P}[N_{T_i}(t)\, m \leq C] \leq \mathbb{P}\left[N_{T_i'}(t)\, m \leq C\right]$. Using the union bound, we have:

$$\mathbb{P}\left[\theta_{T'} \geq \max_{j \neq i}\{\theta_{T_j}\}\right] \geq 1 - \sum_{j \neq i} \mathbb{P}\left[\theta_{T'} < \theta_{T_j}\right]$$

Since $\forall j \neq i : \theta_{T'} - \theta_{T_j} \sim \mathcal{N}\left(\mu_{T'} - \mu_{T_j}, (\sigma_{T'})^2 + \left(\sigma_{T_j}\right)^2\right)$, we have:

$$\begin{aligned}\mathbb{P}\left[\theta_{T'} < \theta_{T_j}\right] &= \frac{1}{2}\mathrm{erfc}\left(\frac{\max_{k \neq i}\{\mu_{T_k}\} - \mu_{T_j} + f_n(t)}{\sqrt{2\left[(\sigma_{T'})^2 + \left(\sigma_{T_j}\right)^2\right]}}\right)\\ &\leq \frac{1}{2}\mathrm{erfc}\left(\frac{f_n(t)}{2\sigma_{T'}}\right)\end{aligned}$$

Hence:

$$\mathbb{P}\left[\theta_{T'} \geq \max_{j \neq i}\{\theta_{T_j}\}\right] \geq 1 - \frac{n-1}{2}\mathrm{erfc}\left(\frac{f_n(t)}{2\sigma_{T'}}\right)$$

Thus, we want $f_n(t)$ satisfying $\mathrm{erfc}\left(\frac{f_n(t)}{2\sigma_{T'}}\right) \leq \frac{1}{n-1}$.
Take $f_n(t) = g_n(t)\,\sigma_{T'}$, hence it suffices to take $g_n(t) = 2\mathrm{erfc}^{-1}\left(\frac{1}{n-1}\right)$ and thus

$$f_n(t) = 2\sigma_{T'}\mathrm{erfc}^{-1}\left(\frac{1}{n-1}\right)$$

In order to use the induction hypothesis, let us bound $\left|\mu_{T'} - \mu_{T_j}\right|$:

$$\begin{aligned}\left|\mu_{T'} - \mu_{T_j}\right| &= \left|\max_{j \neq i}\{\mu_{T_j}\} + f_n(t) - \mu_{T_i}(t)\right|\\ &\leq 1 + 2\sigma_{T'}\mathrm{erfc}^{-1}\left(\frac{1}{n-1}\right)\end{aligned}$$

For any $j \neq i$, we have $\left(\sigma_{T_j}\right)^2 \leq \sqrt{\frac{1}{12}}$, thus $\left|\mu_{T'} - \mu_{T_j}\right| \leq 1 + \sqrt{\frac{2}{\sqrt{3}}}\mathrm{erfc}^{-1}\left(\frac{1}{n-1}\right)$.
By defining $M_n = 1 + \sqrt{\frac{2}{\sqrt{3}}}\mathrm{erfc}^{-1}\left(\frac{1}{n-1}\right)$, we use the induction hypothesis to deduce that:

$$\mathbb{P}\left[N_{T_i'}(t)\, m \leq \frac{\log t}{4|\mathcal{L}(T_i')|M_n}\right] \leq \exp\left[-\frac{2}{t}\left(\frac{t^{3/4}}{\sqrt{\pi}\left(\sqrt{\frac{\log t}{4}} + \sqrt{\frac{\log t}{4} + 2}\right)} - \frac{\log t}{4m|\mathcal{L}(T_1')|M_n^2}\right)^2\right]$$

Since $\mathbb{P}\left[N_{T_i}(t)\, m \leq \frac{\log t}{4|\mathcal{L}(T_i)|M_n}\right] \leq \mathbb{P}\left[N_{T_i'}(t)\, m \leq \frac{\log t}{4|\mathcal{L}(T_i')|M_n}\right]$, we deduce that:

$$\mathbb{P}\left[N_{T_i}(t)\, m \leq \frac{\log t}{4|\mathcal{L}(T_i)|M_n}\right] \leq \exp\left[-\frac{2}{t}\left(\frac{t^{3/4}}{\sqrt{\pi}\left(\sqrt{\frac{\log t}{4}}+\sqrt{\frac{\log t}{4}+2}\right)}-\frac{\log t}{4m|\mathcal{L}(T_i)|M_n^2}\right)^2\right]$$

□