Online Learning of Decision Trees with Thompson Sampling

Ayman Chaouki [1 2] Jesse Read [1] Albert Bifet [2]

[1]LIX, Ecole Polytechnique, IP Paris

[2]AI Institute, The University of Waikato

AISTATS 2022

Table of Contents

Section 1

Introduction

Decision Trees

- Hierarchical rules partitioning the feature space.
- Each element of the partition is assigned a class label.
- Interpretable models, useful for sensitive domains.

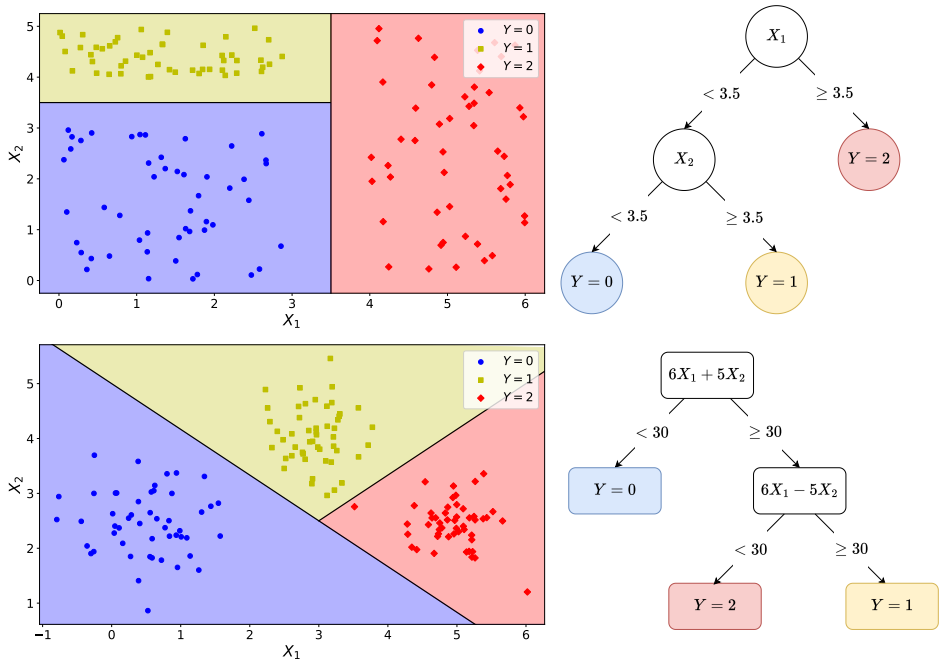


Figure: Example of Decision Tree classifiers on the feature space $\mathbb{R}^2$.

Optimising Decision Trees is challenging

Objective

Complex Decision Trees (i.e with many rules) lose interpretability. As such, it is desirable to find optimal Decision Trees both in the sense of maximising accuracy and also minimising complexity.

Optimising Decision Trees is challenging

Objective

Complex Decision Trees (i.e with many rules) lose interpretability. As such, it is desirable to find optimal Decision Trees both in the sense of maximising accuracy and also minimising complexity.

⚠ Unfortunately, this optimisation problem is NP-Hard.

Online Learning

- The continuous storage of data requires larger and larger memories to keep up.
- Many modern applications have an influx of data.
- Batch models are obsolete for handling data streams.

Objective:

- Find optimal Decision Trees in the online setting.
- We formulate the problem within a Reinforcement Learning framework.
- We devise TSDT to solve the Reinforcement Learning problem.

Section 2

Thompson Sampling Decision Trees (TSDT)

Online Classification problem

- $X \in \mathcal{X} = \prod_{i=1}^{q} \{1, \ldots, C_i\}$, $Y \in \{1, \ldots, K\}$.
- Data (X_i, Y_i) arrive incrementally, and are i.i.d. following a joint probability distribution.

Online Classification problem

- $X \in \mathcal{X} = \prod_{i=1}^{q} \{1, \ldots, C_i\}$, $Y \in \{1, \ldots, K\}$.
- Data (X_i, Y_i) arrive incrementally, and are i.i.d. following a joint probability distribution.
- For a Decision Tree T and $X \in \mathcal{X}$, $l(X)$ is the leaf containing X.
- Prediction: $T(X) = \text{Argmax}_{1 \leq k \leq K} \mathbb{P}[Y = k | X \in l(X)]$.

Online Classification problem

- $X \in \mathcal{X} = \prod_{i=1}^{q} \{1, \ldots, C_i\}$, $Y \in \{1, \ldots, K\}$.
- Data (X_i, Y_i) arrive incrementally, and are i.i.d. following a joint probability distribution.
- For a Decision Tree T and $X \in \mathcal{X}$, $l(X)$ is the leaf containing X.
- Prediction: $T(X) = \text{Argmax}_{1 \leq k \leq K} \mathbb{P}[Y = k | X \in l(X)]$.
- Accuracy: $\mathcal{H}(T) = \mathbb{P}[T(X) = Y]$.

Maximising Accuracy is not enough

The complete Decision Tree, that partitions $\mathcal{X}$ into its unit-subsets, maximises Accuracy. Nonetheless, it is an uninterpretable solution that just corresponds to point-wise classification.

Regularised objective function

We seek to maximise Accuracy while penalising solutions with large complexities.

$$\mathcal{H}_r(T) = \mathcal{H}(T) - \lambda \mathcal{S}(T)$$

- $\lambda \geq 0$ is a penalty parameter.
- $\mathcal{S}(T)$ is the number of splits of T.
- Find T^* maximising $\mathcal{H}_r(T)$.

Remark

This objective is used in CART during pruning. It is also employed by the OCT algorithm.

Problem Formulation

Markov Decision Process (MDP)

Problem Formulation

Markov Decision Process (MDP)

- **State:** Decision Tree. R denotes the root, the initial state.

Problem Formulation

Markov Decision Process (MDP)

- **State:** Decision Tree. R denotes the root, the initial state.
- **Action:** two types of actions: split actions split a leaf with respect to some attribute, and the terminal action ends the episode.

Problem Formulation

Markov Decision Process (MDP)

- **State:** Decision Tree. R denotes the root, the initial state.
- **Action:** two types of actions: split actions split a leaf with respect to some attribute, and the terminal action ends the episode.
- **Transition Dynamics:** When taking an action, we transition from state T to state T' deterministically, we denote the transition $T \rightarrow T'$. The set of next states from T is denoted $\mathrm{Ch}(T)$. We also denote $T \rightarrow \overline{T}$ when taking the terminal action.

Problem Formulation

Markov Decision Process (MDP)

- **State:** Decision Tree. R denotes the root, the initial state.
- **Action:** two types of actions: split actions split a leaf with respect to some attribute, and the terminal action ends the episode.
- **Transition Dynamics:** When taking an action, we transition from state T to state T' deterministically, we denote the transition $T \rightarrow T'$. The set of next states from T is denoted $\mathrm{Ch}(T)$. We also denote $T \rightarrow \overline{T}$ when taking the terminal action.
- **Reward:** Split actions yield $-\lambda$ reward, the terminal action yields reward $\mathcal{H}(T)$, which is unknown.

Problem Formulation

Search Tree representation

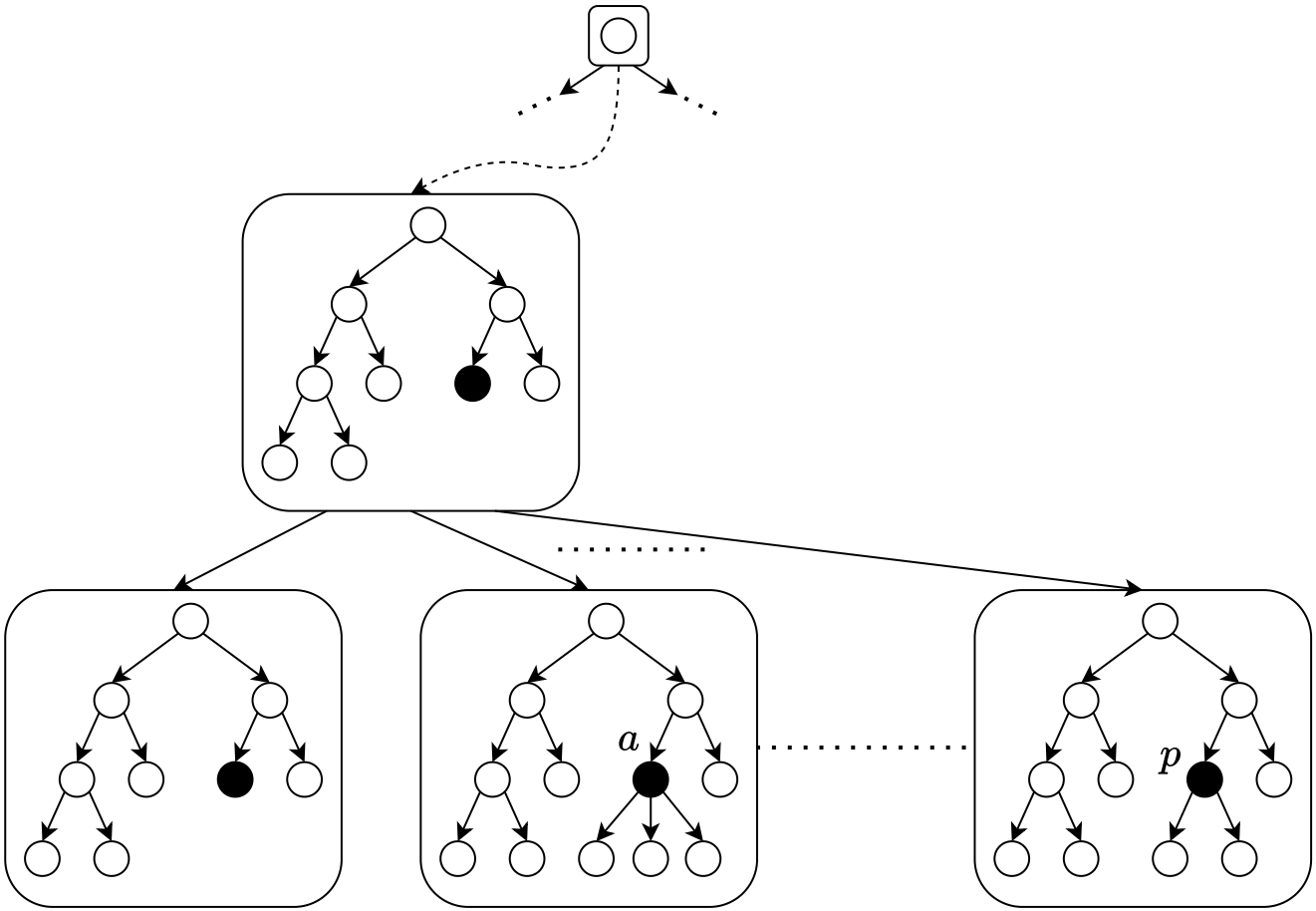


Figure: Graph representation of the state-action space. The (squared) nodes represent states and the edges represent actions.

Problem Formulation

Linking the MDP to online Decision Tree optimisation

- Let π be a policy, $\forall T : \pi(T) \in \mathrm{Ch}(T)$.
- $T = T^{(0)} \xrightarrow{\pi} T^{(1)} \xrightarrow{\pi} \ldots, \xrightarrow{\pi} T^{(N)} = \overline{T^{(N-1)}}$:

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

Proposition

The optimal policy $\pi^* = \mathrm{Argmax}_{\pi}\mathcal{V}^{\pi}(R)$ exists and it leads to T^*:

$$R \xrightarrow{\pi^*} \ldots, \xrightarrow{\pi^*} T^* \xrightarrow{\pi^*} \overline{T^*}$$

$$T^* = \mathrm{Argmax}_T \mathcal{H}_r(T)$$

Bellman Optimality Equation

In any non-terminal state T, the optimal policy satisfies:

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

- By estimating $\mathcal{V}^{\pi^*}(\overline{T})$, TSDT estimates $\mathcal{V}^{\pi^*}(T)$ for non-terminal states by Backward Induction.

Bellman Optimality Equation

In any non-terminal state T, the optimal policy satisfies:

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

- By estimating $\mathcal{V}^{\pi^*}(\overline{T})$, TSDT estimates $\mathcal{V}^{\pi^*}(T)$ for non-terminal states by Backward Induction.
- TSDT estimates $\mathcal{V}^{\pi^*}(\overline{T})$ within a Bayesian framework, and backpropagate the posterior distributions up the Search Tree.
- TSDT's exploration policy is Thompson Sampling.

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

Posterior distribution on $\mathcal{V}^{\pi^*}(\overline{T})$:

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

$$\mathcal{V}^{\pi^*}(\overline{T}) = \mathcal{H}(T) = \sum_{I \in \mathcal{L}(T)} \mathbb{P}[X \in I] \, \mathbb{E}[\mathbb{1}\{T(X) = Y\} | X \in I]$$

Posterior distribution on $\mathcal{V}^{\pi^*}(\overline{T})$:

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

$$\theta_{T,I} \sim \text{Beta}(\alpha_{T,I}, \beta_{T,I})$$

$$\alpha_{T,I} = 1 + \sum_{i=1}^{N} \mathbb{1}\{X_i \in I\} \mathbb{1}\{T(X_i) = Y_i\};$$

$$\beta_{T,I} = 1 + \sum_{i=1}^{N} \mathbb{1}\{X_i \in I\} - \sum_{i=1}^{N} \mathbb{1}\{X_i \in I\} \mathbb{1}\{T(X_i) = Y_i\}$$

$$\mathcal{V}^{\pi^*}(\overline{T}) = \mathcal{H}(T) = \sum_{I \in \mathcal{L}(T)} \mathbb{P}[X \in I]\, \mathbb{E}[\mathbb{1}\{T(X) = Y\} | X \in I]$$

Posterior distribution on $\mathcal{V}^{\pi^*}(\overline{T})$:

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

$$\theta_{T,I} \sim \text{Beta}(\alpha_{T,I}, \beta_{T,I})$$

$$\alpha_{T,I} = 1 + \sum_{i=1}^{N} \mathbb{1}\{X_i \in I\} \mathbb{1}\{\underbrace{T(X_i)}_{\text{unknown}} = Y_i\};$$

$$\beta_{T,I} = 1 + \sum_{i=1}^{N} \mathbb{1}\{X_i \in I\} - \sum_{i=1}^{N} \mathbb{1}\{X_i \in I\} \mathbb{1}\{\underbrace{T(X_i)}_{\text{unknown}} = Y_i\}$$

$$T(X) = \text{Argmax}_{1 \leq k \leq K} \mathbb{P}[Y = k | X \in l(X)]$$

$$\forall l \in \mathcal{L}(T) : \hat{p}_k^{(i)}(l) = \frac{\sum_{j=1}^{i} \mathbb{1}\{X_j \in l\} \mathbb{1}\{Y_j = k\}}{\sum_{j=1}^{i} \mathbb{1}\{X_j \in l\}}$$

$$\hat{T}_i(l) = \text{Argmax}_{1 \leq k \leq K} \{\hat{p}_k^{(i)}(l)\}$$

$$T(X) = \text{Argmax}_{1 \leq k \leq K} \mathbb{P}[Y = k | X \in l(X)]$$

$$\forall l \in \mathcal{L}(T) : \hat{p}_k^{(i)}(l) = \frac{\sum_{j=1}^{i} \mathbb{1}\{X_j \in l\} \mathbb{1}\{Y_j = k\}}{\sum_{j=1}^{i} \mathbb{1}\{X_j \in l\}}$$

$$\hat{T}_i(l) = \text{Argmax}_{1 \leq k \leq K} \{\hat{p}_k^{(i)}(l)\}$$

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

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

? What is the distribution of $\theta_{\overline{T}}$?

? What is the distribution of $\theta_{\overline{T}}$?

Posterior distribution on $\mathcal{V}^{\pi^*}(\overline{T})$:

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

$$\forall l \in \mathcal{L}(T) : \theta_{T,l} \sim \mathrm{Beta}(\alpha_{T,l}, \beta_{T,l})$$

- $\theta_{\overline{T}}$ is a linear combination of independent Beta variables.

? What is the distribution of $\theta_{\overline{T}}$?

Posterior distribution on $\mathcal{V}^{\pi^*}(\overline{T})$:

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

$$\forall l \in \mathcal{L}(T) : \theta_{T,l} \sim \mathrm{Beta}(\alpha_{T,l}, \beta_{T,l})$$

- $\theta_{\overline{T}}$ is a linear combination of independent Beta variables.
- Approximate each Beta distribution with a Normal distribution.

Match the first and second moments of the Beta distribution with its Normal approximation.

$$\theta_{T,I} \sim \mathcal{N}\left(\mu_{T,I}, (\sigma_{T,I})^2\right)$$

$$\mu_{T,I} = \frac{\alpha_{T,I}}{\alpha_{T,I} + \beta_{T,I}}$$

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

Normal Approximation of the Beta prior $\theta_{\overline{T}}$

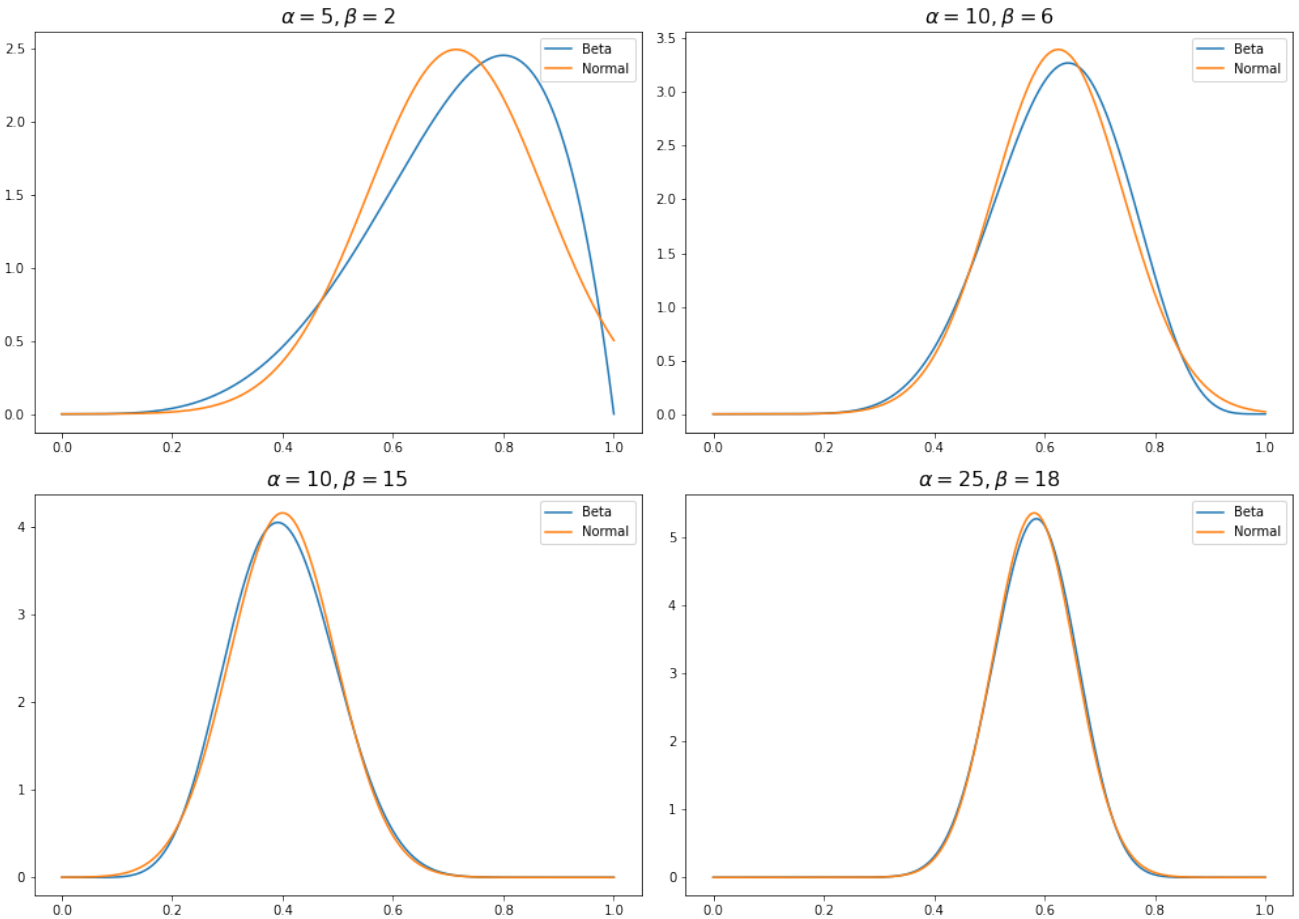


Figure: Probability density functions of a Beta distribution and its Normal approximation for different parameters α, and β.

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

$$\theta_{\overline{T}} \sim \mathcal{N}\left(\mu_{\overline{T}}, \left(\sigma_{\overline{T}}\right)^2\right)$$

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

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

Bellman Optimality Equation

In any non-terminal state T, the optimal policy satisfies:

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

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

$$\widetilde{T} = \mathrm{Argmax}_{T' \in \mathrm{Ch}(T)} \{ -\lambda \mathbb{1}\{ T' \neq \overline{T} \} + \mu_{T'} \}$$

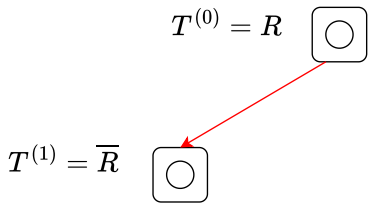
$$\theta_T \sim \mathcal{N}\left(-\lambda \mathbb{1}\{ \widetilde{T} \neq \overline{T} \} + \mu_{\widetilde{T}}, (\sigma_{\widetilde{T}})^2 \right)$$

$T^{(0)} = R$ 

Initialise $\pi(T) = \overline{T}$

Iteration $t = 1$

- *Selection*
- *Simulation*
- *Expansion*
- *Backpropagation*



Initialise $\pi(T) = \overline{T}$

Iteration $t = 1$

- *Selection*
- *Simulation*
- *Expansion*
- *Backpropagation*

Initialise $\pi(T) = \overline{T}$

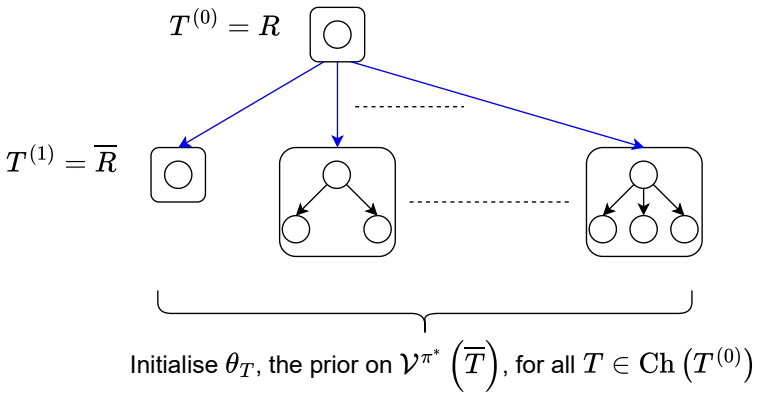
Iteration $t = 1$

- *Selection*
- *Simulation*
- *Expansion*
- *Backpropagation*

Initialise $\pi(T) = \overline{T}$

Iteration $t = 1$

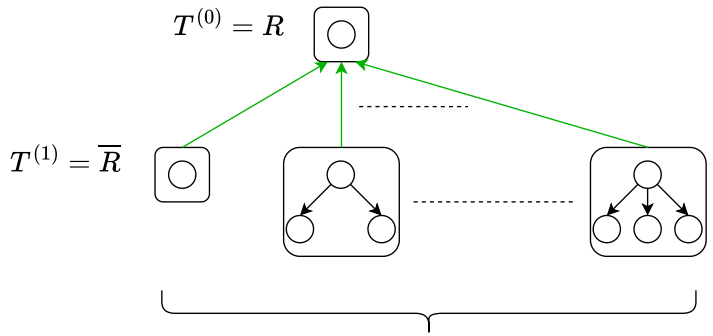
- *Selection*
- *Simulation*
- *Expansion*
- *Backpropagation*



Initialise $\pi(T) = \overline{T}$

Iteration $t = 1$

- *Selection*
- *Simulation*
- *Expansion*
- *Backpropagation*



Update $\theta_{T^{(0)}} \approx \max_{T \in \mathrm{Ch}(T^{(0)})} \{-\lambda \mathbb{1}\{T \neq \overline{T^{(0)}}\} + \theta_T\}$

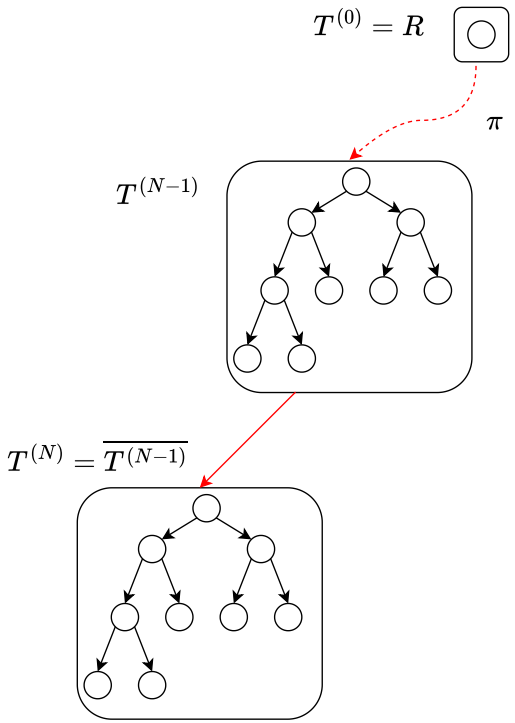
Update $\pi\left(T \middle| T^{(0)}\right) = \mathbb{P}\left[-\lambda \mathbb{1}\{T \neq \overline{T^{(0)}}\} + \theta_T = \max_{T' \in \mathrm{Ch}(T^{(0)})} \{-\lambda \mathbb{1}\{T' \neq \overline{T^{(0)}}\} + \theta_{T'}\}\right]$

Iteration t

- *Selection*
- *Simulation*
- *Expansion*
- *Backpropagation*

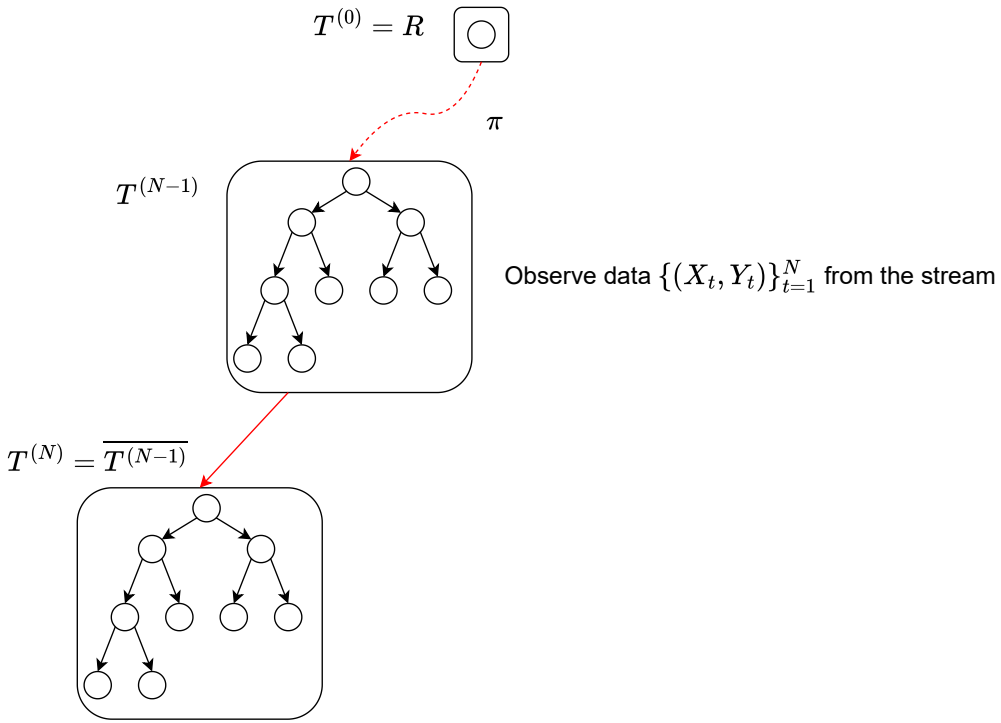
Iteration t

- *Selection*
- *Simulation*
- *Expansion*
- *Backpropagation*



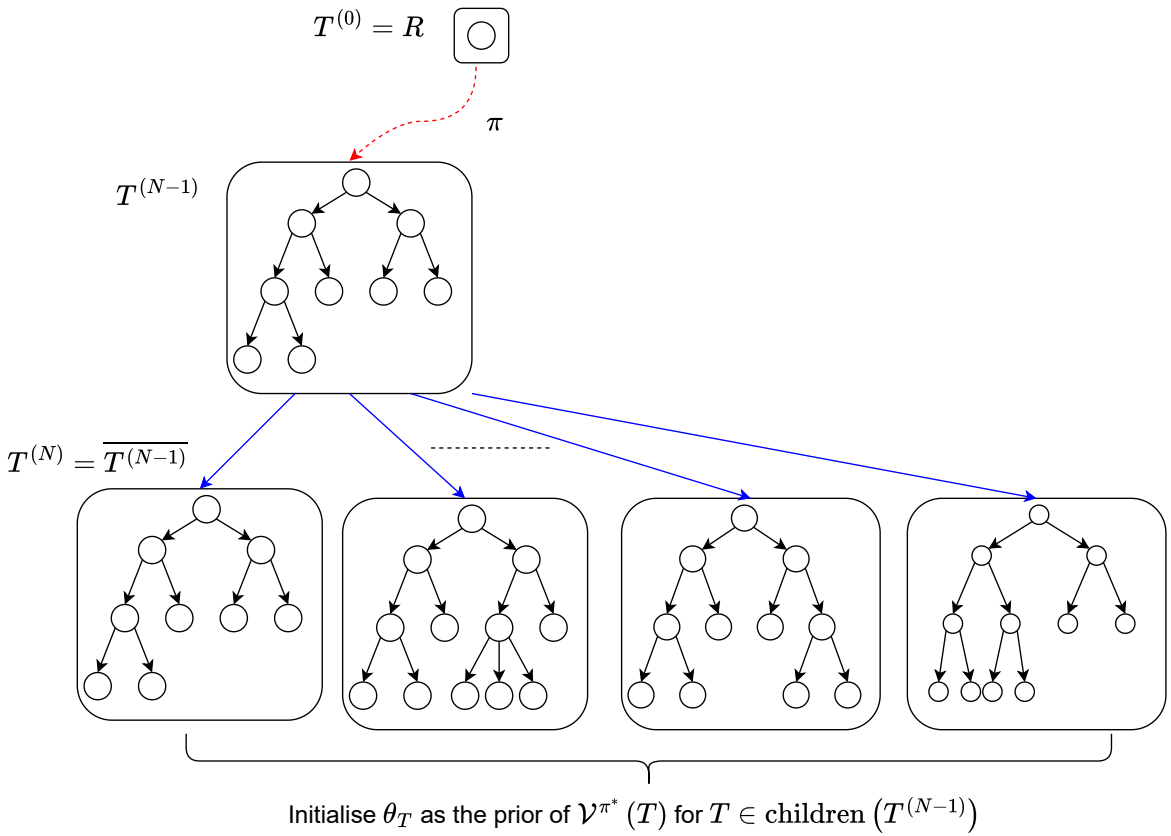
Iteration t

- *Selection*
- *Simulation*
- *Expansion*
- *Backpropagation*



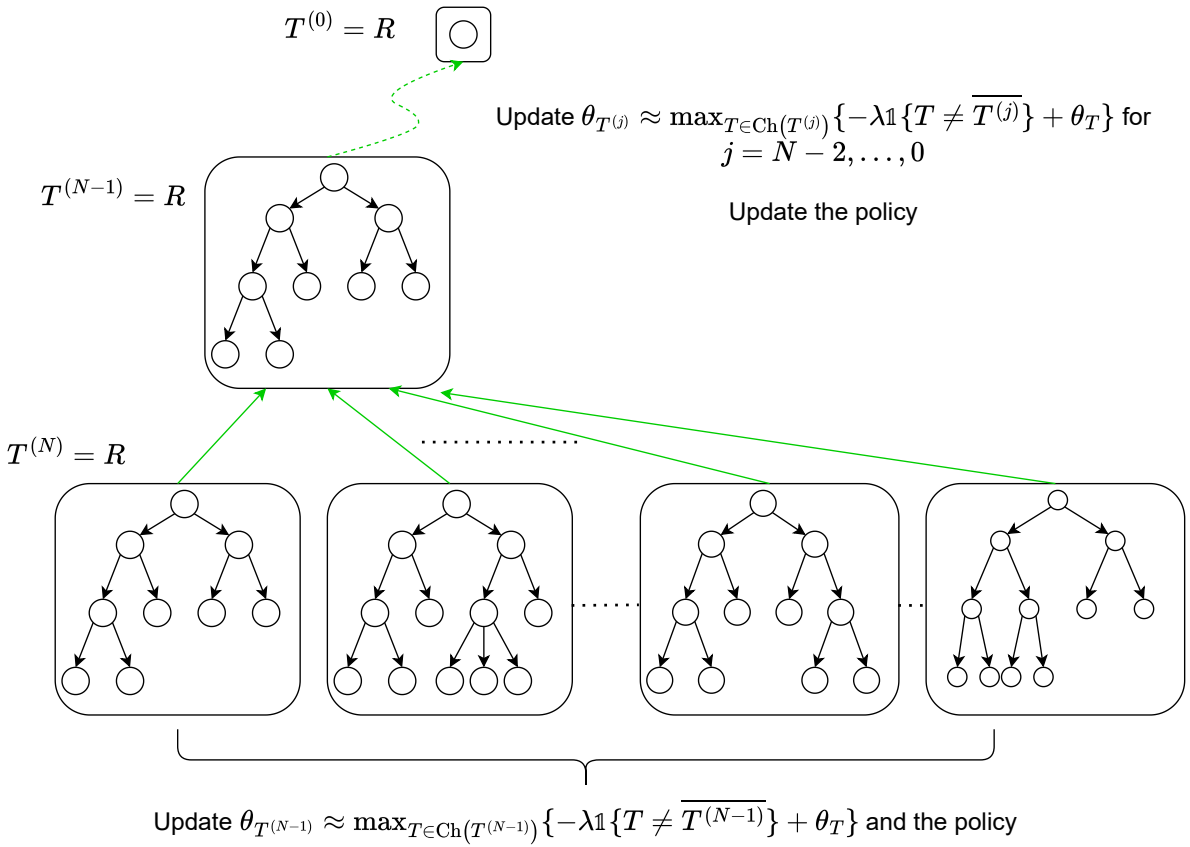
Iteration *t*

- *Selection*
- *Simulation*
- *Expansion*
- *Backpropagation*



Iteration t

- *Selection*
- *Simulation*
- *Expansion*
- *Backpropagation*



Theorem

For any state T, TSDT satisfies the following:

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

For any non-terminal state T:

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

Section 3

Experiments

Experiments

Test accuracies

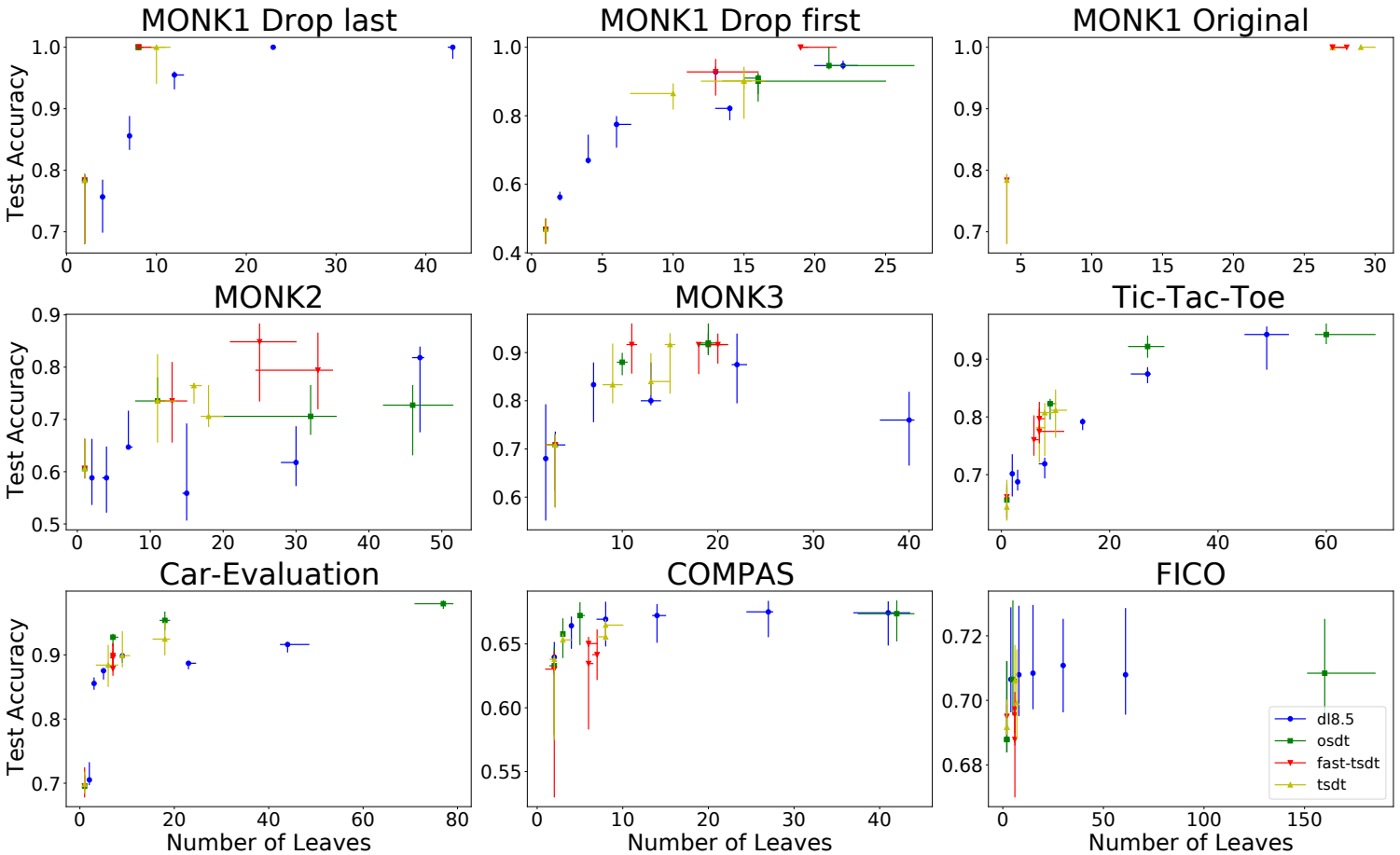


Figure: Cross-validation test accuracy comparison between TSDT, Fast-TSDT, OSDT and DL8.5 as a function of the number of leaves.

Experiments

Execution Times

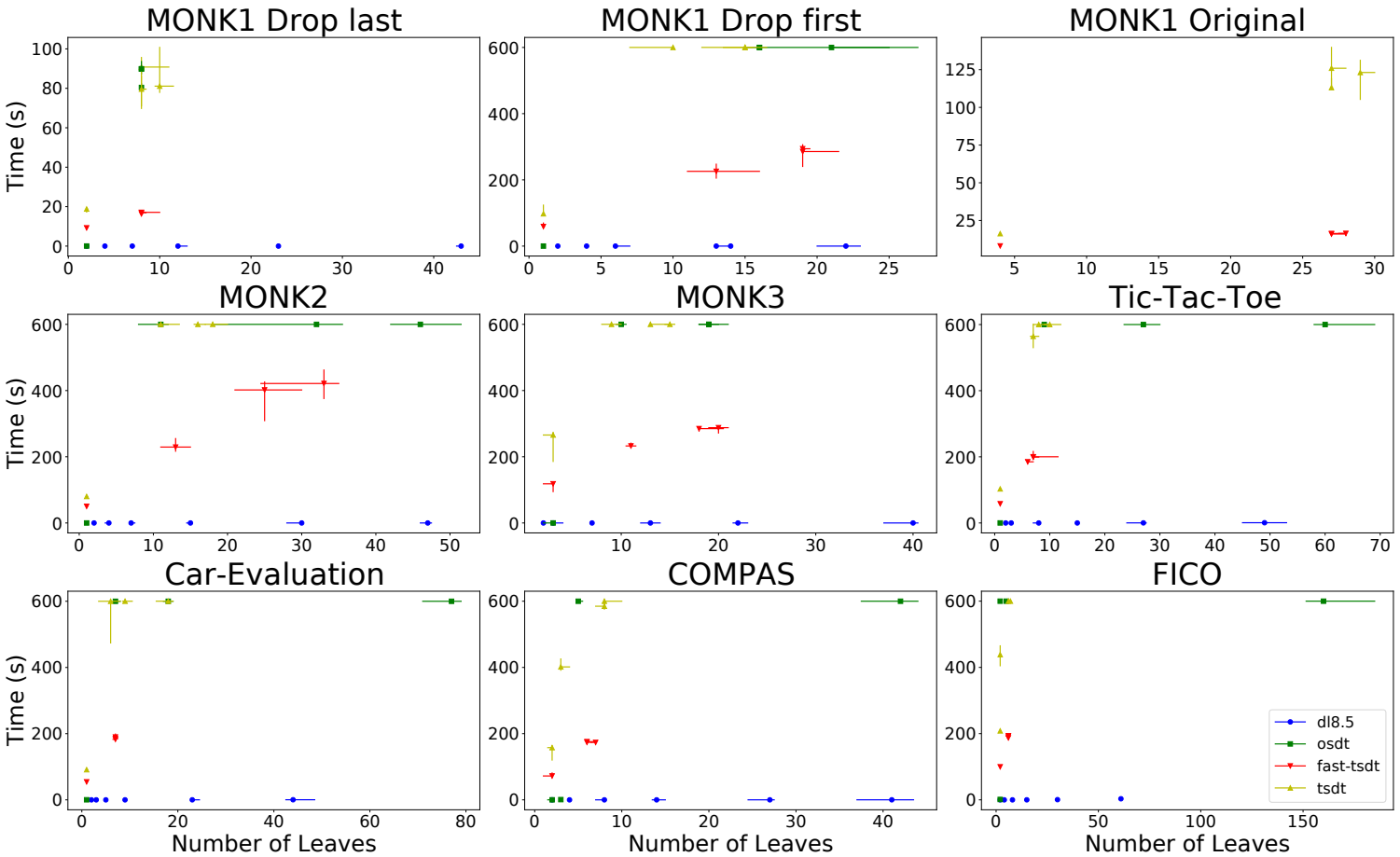


Figure: Cross-validation execution times of TSDT, OSDT and DL8.5 as a function of the number of leaves.

Section 4

Limitations and Future Work

Limitations

- TSDT is limited to categorical features for now.
- TSDT assumes i.i.d. data, it could suffer if the data distribution is non-stationary.
- No finite-time theoretical guarantees.
- The Normal approximations of the Beta prior could cause TSDT to be suboptimal in terms of the number of iterations it takes to find T^*.

Future Work

- Extend TSDT to handling numerical features.
- Extend this Value Iteration method to frequentist policies like UCB and Epsilon-greedy.
- Derive finite-time guarantees in the form of PAC-style bounds or rates of convergence.

Thank you.