ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

ChatDT: Simplifying Constraint Integration in Decision Trees

Abiola Paterne Chokki and Benoît Frénay *

University of Namur - NaDI - Faculty of Computer Science - PReCISE
Rue Grandgagnage 21, B-5000 Namur - Belgium

Abstract. Decision trees help domain experts, such as doctors and bankers, rationalize system decisions. However, existing methods lack user-friendly ways to integrate multiple constraints and identify branches for pruning. This paper introduces ChatDT, a prototype developed with a new domain-specific language and an enhanced version of the CART algorithm to address these challenges. An evaluation involving 22 participants highlights ChatDT's effectiveness, confirming its role in facilitating decision tree creation tailored to domain-specific constraints and identifying branches for pruning.

1 Introduction

Decision trees serve as fundamental tools in machine learning, renowned for their simplicity and interpretability [1]. Despite their advantages, existing algorithms [2, 3, 4], approaches [5, 6, 7] and tools [11, 12, 13] for constructing and refining decision trees often fall short in user-friendliness, particularly when it comes to integrating multiple constraints or domain specific constraints (e.g., fairness, or privacy) and identifying branches that should be pruned to avoid overfitting.

This paper addresses these gaps by introducing ChatDT, a novel prototype built upon a domain-specific language (DSL) integrated into CART. Our contributions include the formulation of a new DSL and enhancements to the CART algorithm, aimed at easing the integration of diverse types of constraints into decision trees, the integration of a user interface to simplify constraint definition, and the process of branch identification and pruning. Through user experimentation, we confirm the ease of use and effectiveness of ChatDT in simplifying constraint integration and identifying branches for pruning in decision trees.

The paper's structure is as follows: Section 2 examines decision tree constraints and underscores the shortcomings associated with approaches integrating them. Section 3 presents our ChatDT prototype. Section 4 discusses experiments, and Section 5 concludes with future research insights.

2 Background

Inspired by [8], three distinct types of constraints can be identified in the context of decision tree learning. *Structure-level constraints* focus on the organization of the decision tree itself, governing properties such as size, depth, and the number

*This work was supported by the *ARIAC by Digital Wallonia4.AI* project number 2010235 of the SPW Recherche / Wallonia Research.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

of leaf nodes [8]. These constraints guide the learning algorithm in constructing a tree that adheres to predetermined structural properties. *Attribute-level constraints* are directly associated with dataset features, influencing the rules and parameters of the decision tree. These constraints, often derived from expert knowledge or algorithms, encompass properties such as monotonicity, attribute costs, and hierarchy constraints, including privacy and fairness considerations [8]. *Instance-level constraints* are imposed on specific instances within the dataset, dictating relationships between individual data points. Examples include "must-link" or "cannot-link" constraints, which determine whether certain examples must belong to the same class or not [8].

To the best of our knowledge, while various approaches [5, 6, 7, 9, 10] and tools [11, 12, 13] have been proposed for integrating the mentioned constraints (as detailed further in [8]), none of them have introduced a DSL or/and a user interface to simplify the process of defining constraints for users. Moreover, none have offered assistance in identifying nodes that could be removed from decision trees to prevent overfitting. Additionally, many of these approaches tend to prioritize certain types of constraints over others, sometimes neglecting some constraints in the process.

3 ChatDT prototype

To tackle the aforementioned challenges, this paper introduces a DSL designed to facilitate the definition of constraints by users. To our knowledge, this paper is the first to propose integrating a DSL for defining constraints within a decision tree. The choice of DSL was based on its appropriateness for addressing domain-specific requirements and its perceived user-friendliness, even for individuals with limited technical expertise [11]. The proposed DSL comprises three syntax categories: creation syntax for initializing decision trees and related to structure-level constraints, modification syntax for adjusting existing trees and associated with attribute-level constraints, and visual syntax for altering the tree's appearance. Table 1 provides a comprehensive overview of these syntax categories, with a usage example depicted in Figure 1.

After defining the DSL, we integrated it into the ChatDT, a prototype built using React for the frontend and FastAPI for the API (source code available at `https://github.com/chokkipaterne/ChatDT`). ChatDT's architecture for creating classification and regression trees involves users selecting or uploading a dataset in CSV format and defining domain constraints using DSL commands. ChatDT then generates decision trees with accuracy (for classification) or mean squared error (for regression) information using an enhanced CART algorithm, which validates user-defined constraints during node splitting. If constraints are not met, a leaf node is returned. Additionally, ChatDT offers gradient-colored decision trees displaying sample count and Gini impurity or variance reduction per node, aiding node identification for removal to mitigate overfitting. Users can also customize the appearance of the decision tree by specifying visual constraints like node color and size.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Update commands	
(u1) set node [node_number] to [column_name] with a threshold of [threshold] (u2) set node [node_number] to one of the following features: [column_names separated by comma] (u3) set node [node_number] to any feature except the following: [column_names separated by comma] (u4) remove tree from node [node_number]	
Creation commands	**Visual commands**
(c1) set features to [column_names separated by comma] (c2) set target to [column_name] (c3) set training data size to [train_size] (c4) set max depth to [max_depth] (c5) set min samples split to [min_samples_split]	(v1) set root node color to [color] (v2) set root node size to [size] (v3) set branch node color to [color] (v4) set branch node size to [size] (v5) set leaf node color to [color] (v6) set leaf node size to [size]

Table 1: DSL syntax for decision tree constraints definition.

Figure 1 depicts a scenario where a user used ChatDT to generate a decision tree from the iris dataset *(a)*, add a constraint to a specific node *(b)*, and detect *(b, red circles)* and remove *(c)* unnecessary nodes. The full interface of ChatDT showcasing its utilization with the iris dataset is depicted in Figure 2, and an associated demo video is available at https://tinyurl.com/chatdtvideo.

4 Evaluation and analysis

After implementing the prototype, we recruited 22 users from a data science course at the University of Namur. We conducted user testing, employing a think-aloud approach lasting up to 1 hour, along with an online survey (with details available at https://tinyurl.com/chatdtsurvey), to evaluate the prototype's ease of use, usefulness/effectiveness, and gather feedback. Some questions on ease of use assess users' agreement with statements such as: "Learning to use this prototype was easy for me", "I find this prototype clear and understandable", and "It would be easy for me to become proficient in using this prototype". Other questions evaluate the usefulness and effectiveness of the prototype, including whether users believe it is effective to: "Add constraints to a specific node", "See the impact of changes on the decision tree (in terms of accuracy or error)", and "Detect and remove unnecessary leaves in the tree".

Table 2 showcases the median, mean, and standard deviation (SD) of the 7-point Likert scale questions from the online survey. The findings from Table 2 yield the following insights. Firstly, the prototype is user-friendly and intuitive, indicated by both the median and mean scores of these questions exceeding 5 out of 7 (the highest value). Secondly, the prototype proves beneficial and effective in simplifying the creation and modification of decision trees, with the median

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

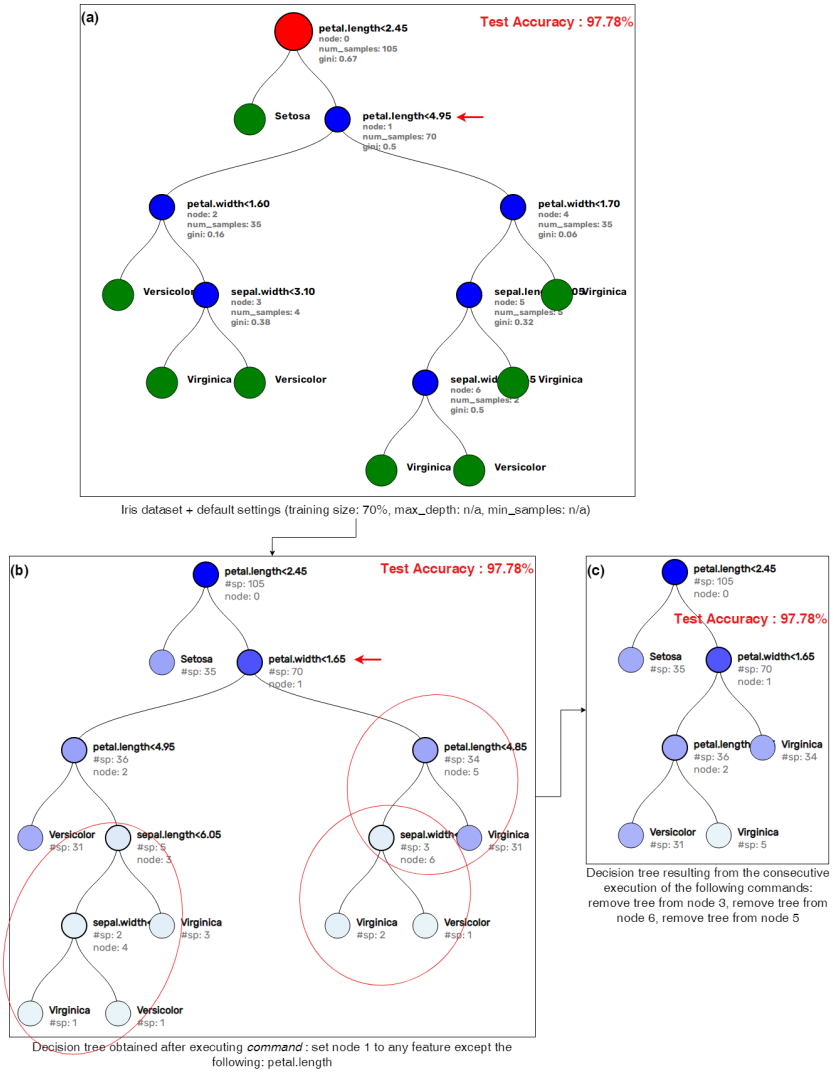

Fig. 1: Scenario demonstrating the use of the DSL integrated into ChatDT. In (a), the root node is represented in red, branch nodes in blue, and leaf nodes in green. In (b) and (c), a gradient color scale (from dark blue to light blue) indicates the number of instances in each node, aiding users in identifying potential nodes for removal. Initially, in (a), the user opts to replace node 1 with a different feature instead of petal.length, leading to the outcome seen in (b). Following this, the user chooses to eliminate nodes 3, 5, and 6 due to the low number of instances in these nodes, resulting in the outcome seen in (c).

and mean scores for these questions each reaching or exceeding 6 out of 7.

The results are supported by the observations and feedback obtained from

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

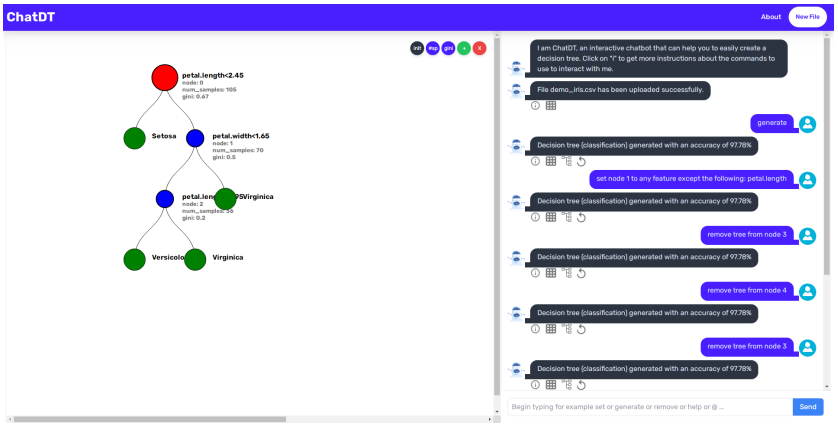

Fig. 2: Interface of ChatDT prototype.

	Ease of use	**Usefulness/Effectiveness**
Median	6	7
Mean (SD)	5.82 (1.12)	6.27 (1.07)

Table 2: Median, mean and standard deviation (SD) of survey scores.

users throughout and following the testing phase. During testing, all participants demonstrated proficiency in generating and manipulating their decision trees. They successfully applied various constraints to specific nodes or the entire decision tree, and the outputs provided by ChatDT consistently aligned with their intentions. This indicates the effectiveness of the prototype in meeting user requirements and expectations. Additionally, participants consistently praised the user-friendliness, intuitiveness, and ergonomic layout of the prototype. Despite facing initial challenges, participants recognized their potential to attain proficiency with prolonged usage. These positive sentiments contribute to the high scores attributed to ease of use, confirming that the design and functionality of the prototype were well-received and contributed to a smooth user experience.

Following the evaluation, participants were prompted with inquiries regarding which features they believed should be retained, enhanced, eliminated, or introduced. Universally, all participants advocated for the retention of the current features. However, there was a consensus among many users regarding the necessity to enhance the UX/UI design of the prototype for improved usability and intuitiveness. Suggestions included implementing a step-by-step guide to streamline the process of creating and updating decision trees. Additionally, users expressed interest in new functionalities, including exporting decision trees in PNG or PDF formats, manipulating datasets for better understanding, viewing the best decision tree, accessing the history of parameters utilized in various configurations, and enhancing the functionality of the prototype. Specifically, users proposed an improvement allowing them to input their requirements di-

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

rectly into the prototype using natural language rather than predefined instructions. The system would then match these inputs with existing instructions.

5 Conclusion and future work

This paper presents ChatDT, a prototype designed to streamline the integration of constraints into classification and regression decision trees. Additionally, ChatDT assists users in identifying nodes that can be removed to prevent overfitting. The contributions of this work include the formulation of a novel domain-specific language, enhancements to the CART algorithm to facilitate constraint integration, and the implementation of a user-friendly interface to easily add constraints and prune decision trees. Through user experimentation involving 22 participants, ChatDT's effectiveness in simplifying constraint integration is demonstrated, affirming its role as a facilitator for crafting decision trees tailored with domain-specific constraints. Future work may involve leveraging generative AI to enable users to manipulate decision trees using natural language, eliminating the need for predefined instructions. Additionally, we aim to understand user preferences between using natural language and graphical interfaces for interacting with decision trees, and to enhance the representativeness of participants in our evaluations.

References

[1] A. A. Freitas. Comprehensible classification models: A position paper. SIGKDD Explor. Newsl. 15, 1, 1–10, 2014.

[2] J. R. Quinlan. Induction of decision trees. Mach. Learn. 1, 1 (Mar. 1986), 81–106, 1986.

[3] J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.

[4] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. Classification and Regression Trees. Wadsworth and Brooks, Monterey, CA, 1984.

[5] L. Buitinck et al. API design for machine learning software: Experiences from the Scikit-learn project. In Proc. ECML PKDD Workshop, 108–122, 2013.

[6] G. Nanfack, V. Delchevalerie, and B. Frenay. Boundary-based fairness constraints in decision trees and random forests. In Proc. ESANN 21, 2021.

[7] N. Wang et al. Accurate decision trees with cost constraints. In Proc. 1st Int. Conf. Adv. Hybrid Inf. Process. (ADHIP 17), Springer, 154–165, 2018.

[8] G. Nanfack, P. Temple, and B. Frenay. Constraint Enforcement on Decision Trees: A Survey. ACM Comput. Surv. 54, 10s, Article 201, 36 pages, 2022.

[9] S. Calzavara et al. Treant: Training evasion-aware decision trees. Data Min. Knowl. Discov. 34, 5 (2020), 1390–1420, 2020.

[10] H. Chen et al. Robust decision trees against adversarial examples. In Proc. 36th Int. Conf. Mach. Learn. (ICML), PMLR, 1122–1131, 2019.

[11] A. Raja and D. Lakshmanan. Domain Specific Languages. International Journal of Computer Applications, 1, 105-111, 2010.

[12] S. van den Elzen and J. J. van Wijk. BaobabView: Interactive construction and analysis of decision trees. IEEE Conference on VAST. 2011.

[13] S. Pauwel, S. Moens and Bart Goethals. Interactive and manual construction of classification trees. 2014.

[14] R. Rakotomalala. Sipina: Decision Trees. http://sipina-arbres-de-decision.blogspot.com/. Accessed 21/06/2024.