Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Computer Science

Mixed-integer Programming in Machine Learning: Decision Trees and Neural Networks

Bc. Jiří Němeček

Supervisor: Mgr. Jakub Mareček, Ph.D.
Field of study: Open Informatics
Subfield: Artificial Intelligence
May 2023

Acknowledgements

Firstly, I would like to thank my supervisor Mgr. Jakub Mareček, Ph.D., for his guidance and pragmatic approach. I also thank Doc. Ing. Tomáš Pevný, Ph.D., for his insights regarding the implementation.

Declaration

I declare that the presented work was developed independently, and I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

In Prague, 20. May 2023

. . . .

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 20. 5. 2023

. . . .

Abstract

Keywords:

Supervisor: Mgr. Jakub Mareček, Ph.D.

Abstrakt

Klíčová slova:

Překlad názvu: Smíšené celočíselné programování ve strojovém učení

Contents

Figures

Tables

Chapter 1

Introduction

Artificial Intelligence (AI) is a topic that seems to be getting ever more important over time. In recent months, with the rise of Large Language Models, everyone is getting interested in the capabilities of AI. As the broad society learns to work with AI models, the models need to gain the trust of users. The requirements on the capabilities of models are expanding from pure quantitative performance to a more complex optimization goal. It no longer suffices to show that model is accurate. In many cases, society demands fair treatment, as well as transparency and accountability. These are the broad issues addressed by this thesis.

cite something that shows this interest

With great power comes great responsibility. So as the interest in AI is rising, so does the interest in Explainable Artificial Intelligence (XAI). XAI is the broad

cite some definition here

O cem je tadyta thesis teda

1.1 Why Mixed-integer programming (MIP)?

MIP is NP-complete framework for problem specification, meaning that any NP-hard problem can be stated using MIP.

cite!

This enables us to set up the model using constraints and then optimize a different function.

1.2 Goals of the thesis

1.2.1 Use of MIP methods in Machine learning (ML)

1.2.2 Counterfactual generation using MIP

1.2.3 Decision tree optimization using MIP

Chapter 2

State of the art

Chapter 3

Counterfactual generation

Chapter 4

Most interpretable classification trees

Classification trees are commonly considered to be well-interpretable. That is because of their inherent structure, which is very logical and simple. This is paid for by some non-trivial error on every classification. This means that every explanation is supported by accuracy, that is typically incomparable to currently used methods. While the tree is trained to minimize the total error it provides, it often means that some leaves are left with almost random accuracy. This is because more homogeneous parts of the data are cropped away while the most difficult areas of the data space remain. In those areas, accuracy is very low.

cite

We argue, that if the user is classified by a leaf, where the decision is supported by very low accuracy, the explanation is essentially meaningless. When a leaf has low accuracy on training data, its decision can be flipped by introducing just a few points of a different class. This is an obvious issue, and such leaf does not provide a true explanation of anything.

add image of a tree that has 50% acc somewhere

To address this issue, we aim to find a tree with the highest leaf accuracy. By leaf accuracy, we mean the worst accuracy in a single leaf.

4.1 MIP formulation

We start with the Optimal Classification Tree (OCT) model introduced by Bertsimas and Dunn (2017) and extend it into a model that focuses on best interpretability. This means that each interpretation is supported by the highest possible accuracy. We will call that model eXplained Classification Tree (XCT).

4.1.1 OCT formulation

Let us first explain the MIP formulation of OCT according to Bertsimas and Dunn (2017) in this section.

We consider a K classification problem, with n data points with p features. We will refer to i-th data point as $\boldsymbol{x}_i$. Each $\boldsymbol{x}_i$ has a corresponding class y_i which is one of the K classes.

Basic structural variables

A classification tree consists of branching nodes and of leaf nodes. We call these sets of nodes $\mathcal{T}_B$ and $\mathcal{T}_L$ respectively. In branching nodes, we will model decision making. Since we focus on interpretability, we consider so called axis-aligned trees. That means that in every node we require only the value of a single feature and a threshold. This is a simpler, more restrictive kind of decision-making, for example compared to decision making based on a linear combination of all features.

This will be achieved using a set of binary variables a_{tj} for all branch nodes t and all features j. We restrict them in a way that allows a single feature to be activated for each node:

$$\sum_{j=1}^{p} a_{tj} = 1, \quad \forall t \in \mathcal{T}_B$$

After we select a variable, we need to optimize the threshold. This will be represented by a set of variables b_t that will take continuous value in interval $[0, 1]$ because we consider our data to be normalized to this range.

We now have classification trees, where a branch node decision can now be conveniently written as $\boldsymbol{x}_i^\top \boldsymbol{a}_t \lesseqgtr b_t$. We further presume (in alignment with Bertsimas and Dunn (2017)) that in case of equality, point is assigned to the right branch. This means, that a point x_i arrives to the left child of t if $\boldsymbol{x}_i^\top \boldsymbol{a}_t < b_t$ and to the right child if $\boldsymbol{x}_i^\top \boldsymbol{a}_t \geq b_t$

Since strict inequalities cannot be used in MIP, the authors use the minimal difference between different consecutive values of each feature as a small epsilon that replaces the need for a strict inequality. It is precisely defined as follows:

$$\epsilon_j = \min \left\{ x_j^{(i+1)} - x_j^{(i)} \middle| x_j^{(i+1)} \neq x_j^{(i)}, \forall i \in \{1, \dots, n-1\} \right\}$$

where $x_j^{(i)}$ is the i-th largest value in the j-th feature.

We also need a variable signaling whether a point is assigned to a certain leaf node. For this we have a set of binary variables z_{it}. It will hold that $z_{it} = 1 \iff x_i$ is assigned to a leaf t.

Putting all of these things together, we can see the two most important constraints which will ensure the correct function of the branching:

$$\boldsymbol{a}_m^\top \boldsymbol{x}_i \geq b_m - (1 - z_{it}) \tag{4.1}$$

$$\boldsymbol{a}_m^\top (\boldsymbol{x}_i + \boldsymbol{\epsilon}) \leq b_m + (1 + \epsilon_{\max})(1 - z_{it}) \tag{4.2}$$

where $i \in \{1, \dots, n\}, t \in \mathcal{T}_L$ and $m \in \mathcal{T}_B$ is the node in which we decide to go right (4.1) or left (4.2). We will call these sets of ancestors of a node t with the node t to their left and right $A_L(t)$ and $A_R(t)$, respectively. These sets are disjoint, and their union is the set of all ancestors of a leaf t. $\epsilon_{\max}$ is the highest value of ϵ_j over all j and serves as the best big-M bound. $\boldsymbol{\epsilon}$ is a p-dimensional vector consisting of ϵ_j.

We must further ensure that points are assigned to exactly one leaf. This is done by the following constraint:

$$\sum_{t \in \mathcal{T}_L} z_{it} = 1$$

Extra structural variables

We want to allow the optimization to ignore certain nodes. Since we represent the complete binary tree, we will add binary variables to signal if a certain node is used or not.

For branching nodes, these will be named d_t and will represent whether the node is present in the tree. We thus limit $\boldsymbol{a}_t$ and b_t to be equal to zero if $d_t = 0$. This should lead to all points being sent to the right subtree according to Bertsimas and Dunn (2017), but more on that in Section 4.1.1.

$$\sum_{j=1}^{p} a_{jt} = d_t, \quad \forall t \in \mathcal{T}_B \tag{4.3}$$

$$b_t \leq d_t, \quad \forall t \in \mathcal{T}_B \tag{4.4}$$

We also ensure that child nodes of an unused node are also unused with the following constraint:

$$d_t \leq d_{p(t)}, \quad \forall t \in \mathcal{T}_B \backslash \{\text{root}\} \tag{4.5}$$

where $p(t)$ is the parent node of node t.

For leaf nodes, we introduce variables l_t,

Classification variables

Issues encountered

Until now, the full described formulation was the work of Bertsimas and Dunn (2017). However, after we have implemented this formulation, it turned out that the formulation is ill-defined, because it is trivially solved with 0% error by $d_m = 0, \forall m \in \mathcal{T}_B$.

Let us consider this explanation: A data point x_i is assigned to some leaf t. This means, that $z_{it} = 1$, and thus branch decision constraints simplify to $\boldsymbol{a}_m^\mathsf{T} \boldsymbol{x}_i \geq b_m$ and $\boldsymbol{a}_m^\mathsf{T}(\boldsymbol{x}_i + \boldsymbol{\epsilon}) \leq b_m$ for any ancestor m for which it holds that $d_m = 0$. There is nothing in the way of trivially satisfying either of these with $\boldsymbol{a}_m = \boldsymbol{0}$ and $b_m = 0$. Solver thus sets all d_m to 0 to enable all $\boldsymbol{a}_m = \boldsymbol{0}$ and proceeds by assigning all points of one class to a single leaf, achieving seemingly 100% accuracy.

A potential solution of this issue would be to change the decision constraint from (4.2) to

$$\begin{aligned} \boldsymbol{a}_m^\mathsf{T}(\boldsymbol{x}_i + \bar{\boldsymbol{\epsilon}}) + \epsilon_{\min} \leq b_m + (1 + \epsilon_{\max})(1 - z_{it}), \\ \forall i \in \{1, \dots, n\}, \forall t \in \mathcal{T}_L, \forall m \in A_L(t) \end{aligned} \tag{4.6}$$

where $\epsilon_{\min}$ is the lowest value of ϵ_j and $\bar{\epsilon}_j = \epsilon_j - \epsilon_{\min}$. $\bar{\boldsymbol{\epsilon}}$ is essentially the same vector as $\boldsymbol{\epsilon}$ but with all values decreased by $\epsilon_{\min}$.

This way, this is not satisfiable if $d_m = 0$ and any leaf t such that $m \in A_L(t)$ is assigned any data point. This holds because all $\epsilon_j > 0$ and thus also $\epsilon_{\min} > 0$. This results in the grouping of datapoints in the rightmost leaf of a subtree that has m as a root. This is what Bertsimas and Dunn (2017) claimed should happen.

This change in formulation will not influence the results otherwise, since if $d_m = 1$, there is $a_{mj} = 1$ for exactly one j, and the removed and added $\epsilon_{\min}$ will be canceled out.

Although this is a functioning correction of the issue, we were not interested in optimizing the number of decision tree nodes, so we went for a simpler solution to this issue and discarded the d_t variable altogether in further extending formulations. In all places where d_t was present, we swapped it for 1, essentially requiring all nodes to be branching, and rather pruned the leaves after optimization.

Now that we have gone through all parts of the OCT formulation let us see it in full:

RECHECK THE FORMULATION, ADD my corrections, and mention it

$$
\begin{aligned}
\min\ & \frac{1}{\hat{L}} \sum_{t \in \mathcal{T}_L} L_t + \alpha \sum_{t \in \mathcal{T}_B} d_t \\
\text{s. t.}\ & L_t \geq N_t - N_{kt} - n(1 - c_{kt}) && \forall k \in \{1, \dots, K\}, \quad \forall t \in \mathcal{T}_L \\
& L_t \leq N_t - N_{kt} - n c_{kt} && \forall k \in \{1, \dots, K\}, \quad \forall t \in \mathcal{T}_L \\
& L_t \geq 0 && \forall t \in \mathcal{T}_L \\
& N_{kt} = \sum_{i=1}^{n} Y_{ik} z_{it} && \forall k \in \{1, \dots, K\}, \quad \forall t \in \mathcal{T}_L \\
& N_t = \sum_{i=1}^{n} z_{it} && \forall t \in \mathcal{T}_L \\
& l_t = \sum_{k=1}^{K} c_{kt} && \forall t \in \mathcal{T}_L \\
& \boldsymbol{a}_m^\mathsf{T} \boldsymbol{x}_i \geq b_m - (1 - z_{it}) && \forall i \in \{1, \dots, n\}, \forall t \in \mathcal{T}_L, \forall m \in A_R(t) \\
& \boldsymbol{a}_m^\mathsf{T} (\boldsymbol{x}_i + \boldsymbol{\epsilon}) \leq b_m + (1 + \epsilon_{\max})(1 - z_{it}) && \forall i \in \{1, \dots, n\}, \forall t \in \mathcal{T}_L, \forall m \in A_L(t) \\
& \sum_{t \in \mathcal{T}_L} z_{it} = 1 && \forall i \in \{1, \dots, n\} \\
& z_{it} \leq l_t && \forall i \in \{1, \dots, n\}, \quad \forall t \in \mathcal{T}_L \\
& \sum_{i=1}^{n} z_{it} \geq N_{\min} l_t && \forall t \in \mathcal{T}_L \\
& \sum_{j=1}^{p} a_{jt} = d_t && \forall t \in \mathcal{T}_B \\
& 0 \leq b_t \leq d_t && \forall t \in \mathcal{T}_B \\
& d_t \leq d_{p(t)} && \forall t \in \mathcal{T}_B \backslash \{\text{root}\} \\
& z_{it}, l_t \in \{0, 1\} && \forall i \in \{1, \dots, n\}, \quad \forall t \in \mathcal{T}_L \\
& a_{jt}, d_t \in \{0, 1\} && \forall j \in \{1, \dots, p\}, \quad \forall t \in \mathcal{T}_B
\end{aligned}
\tag{4.7}
$$

add $c_k t$

4.1.2 Initial extended formulation

Initial approach to the problem consisted of creating a constraint that would be used as a lower limit on leaf accuracy.

4.1.3 Soft accuracy formulation

Next direction of extending the formulation focused on creating a model that would optimize leaf accuracy directly.

4.1.4 Optimizable formulation

A lot of effort was put into creating the next iteration of formulation of the problem. A good

4.1.5 Final XCT formulation

Finally, the last iteration of the development process was fueled by the realization, that one can use a reference variable to which all variables can relate their value, instead of all variables needing to have constraints between each other. This helped tremendously decrease the memory and time inefficiency from $O(n^2)$ to $O(n)$.

The extended formulation is thus as follows:

$$
\begin{array}{lll}
\max Q & & \\
\text{s. t.} & Q \leq \sum_{i=1}^{n} S_{it} + (1 - l_t) & \forall t \in \mathcal{T}_L \\
& s_{it} \leq z_{it} & \forall i \in \{1, \ldots, n\}, \quad \forall t \in \mathcal{T}_L \\
& r_t \leq s_{it} + (1 - z_{it}) & \forall i \in \{1, \ldots, n\}, \quad \forall t \in \mathcal{T}_L \\
& r_t \geq s_{it} + (z_{it} - 1) & \forall i \in \{1, \ldots, n\}, \quad \forall t \in \mathcal{T}_L \\
& S_{it} \leq s_{it} & \forall i \in \{1, \ldots, n\}, \quad \forall t \in \mathcal{T}_L \\
& S_{it} \leq \sum_{k=1}^{K} Y_{ik} c_{kt} & \forall i \in \{1, \ldots, n\}, \quad \forall t \in \mathcal{T}_L \\
& S_{it} \geq s_{it} + \sum_{k=1}^{K} Y_{ik} c_{kt} - 1 & \forall i \in \{1, \ldots, n\}, \quad \forall t \in \mathcal{T}_L \\
& l_t = \sum_{i=1}^{n} s_{it} & \forall t \in \mathcal{T}_L \\
& l_t = \sum_{k=1}^{K} c_{kt} & \forall t \in \mathcal{T}_L \\
& \boldsymbol{a}_m^\mathsf{T} \boldsymbol{x}_i \geq b_t - (1 - z_{it}) & \forall i \in \{1, \ldots, n\}, \forall t \in \mathcal{T}_L, \forall m \in A_R(t) \\
& \boldsymbol{a}_m^\mathsf{T} (\boldsymbol{x}_i + \boldsymbol{\epsilon}) \leq b_t + (1 + \epsilon_{\max})(1 - z_{it}) & \forall i \in \{1, \ldots, n\}, \forall t \in \mathcal{T}_L, \forall m \in A_L(t) \\
& \sum_{t \in \mathcal{T}_L} z_{it} = 1 & \forall i \in \{1, \ldots, n\} \\
& z_{it} \leq l_t & \forall i \in \{1, \ldots, n\}, \quad \forall t \in \mathcal{T}_L \\
& \sum_{i=1}^{n} z_{it} \geq N_{\min} l_t & \forall t \in \mathcal{T}_L \\
& \sum_{j=1}^{p} a_{jt} = d_t & \forall t \in \mathcal{T}_B \\
& 0 \leq b_t \leq d_t & \forall t \in \mathcal{T}_B \\
& d_t \leq d_{p(t)} & \forall t \in \mathcal{T}_B \backslash \{\text{root}\} \\
& z_{it}, l_t \in \{0, 1\} & \forall i \in \{1, \ldots, n\}, \quad \forall t \in \mathcal{T}_L \\
& a_{jt}, d_t \in \{0, 1\} & \forall j \in \{1, \ldots, p\}, \quad \forall t \in \mathcal{T}_B \\
& 0 \leq Q, r_t, S_{it}, s_{it} \leq 1 & \forall i \in \{1, \ldots, n\}, \quad \forall t \in \mathcal{T}_L
\end{array}
\tag{4.8}
$$

4.2 Further model improvements

4.2.1 Tree reduction

4.2.2 Leaves extension

4.3 Experiments

4.3.1 Datasets used

4.3.2 Results

Extended eXplainability Addressing Classification Tree EXACT

Chapter 5

Conclusion

Appendix A

Acronyms

AI	Artificial Intelligence. 1
MIP	Mixed-integer programming. v, 1, 2, 8, 9, 11, 13
ML	Machine learning. v, 2
OCT	Optimal Classification Tree. v, 8, 11
XAI	Explainable Artificial Intelligence. 1
XCT	eXplained Classification Tree. v, 8, 13

Appendix B

Bibliography

Dimitris Bertsimas and Jack Dunn. Optimal classification trees. *Machine Learning*, 106(7):1039–1082, July 2017. ISSN 1573-0565. doi: 10.1007/s10994-017-5633-9.

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Němeček Jiří**	Personal ID number:	**475701**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Computer Science**		
Study program:	**Open Informatics**		
Specialisation:	**Artificial Intelligence**		

II. Master’s thesis details

Master’s thesis title in English:

Mixed-integer Programming in Machine Learning: Decision Trees and Neural Networks

Master’s thesis title in Czech:

Smíšené celočíselné programování ve strojovém učení

Guidelines:

Mixed-integer Programming (MIP) is a framework for formulating a number of problems in machine learning (ML) and prototyping therein. With the increased interest in methods combining symbolic and statistical approaches, this seems particularly useful. Consider, for instance, training a decision tree with neural networks (NN) with ReLU activations in the leaf nodes (Zhou & Chen, 2002), whose training can clearly be cast as a MIP using the techniques of (Bertsimas & Dunn, 2017) and (Anderson et al., 2020).

Bibliography / sources:

Zhi-Hua Zhou & Zhao-Qian Chen: Hybrid decision tree. Knowledge-Based Systems, volume 15, Issue 8, pages 515-528 (2002).
Dimitris Bertsimas & Jack Dunn: Optimal classification trees. Machine Learning, volume 106, pages 1039–1082 (2017).
Ross Anderson, Joey Huchette, Will Ma, Christian Tjandraatmadja & Juan Pablo Vielma: Strong mixed-integer programming formulations for trained neural networks. Mathematical Programming, volume 183, pages 3–39 (2020).

Name and workplace of master’s thesis supervisor:

Mgr. Jakub Mareček, Ph.D. Artificial Intelligence Center FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **31.01.2023** Deadline for master's thesis submission: **26.05.2023**

Assignment valid until: **22.09.2024**

Mgr. Jakub Mareček, Ph.D.	Head of department’s signature	prof. Mgr. Petr Páta, Ph.D.
Supervisor’s signature		Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt Student’s signature