SAT-based Approach for Learning Optimal Decision Trees with Non-Binary Features

POUYA SHATI, ELDAN COHEN, SHEILA MCILRAITH
UNIVERSITY OF TORONTO
CP 2021
SEPTEMBER 13, 2021
Overview

- **Decision trees** are popular classification models
  - provide **interpretability** and **accuracy**
  - constructed via **greedy heuristics** or **exact methods**
  - exact optimization methods largely focus on **binary features**

- **Our contribution**: an approach to handle **non-binary** features effectively
  - outperforms the state of the art on **non-binary** datasets with two popular objectives
Background
Classification

- A popular application of **machine learning**

- Labelling function learned from labelled data set

- The goal is to achieve high accuracy on unseen data points
Decision Trees

- Decision trees are interpretable:
  - human-readable
  - amenable to further (logical) reasoning

- Prime candidates for safety-critical applications
Decision Trees

- **Branching nodes** perform a *split* on a given feature and threshold
- **Leaf nodes** assign a label

Decision Trees:
- $f: \text{age}, t: 30$
- $f: \text{GPA}, t: 3.0$
- $f: \text{Exp}, t: 5$
- $l: \text{Reject}$
- $l: \text{Hire}$
- $l: \text{Reject}$
- $l: \text{Hire}$
Decision Trees

- A set of features $F$ and integer labels $C$
- A decision tree: $D = (T, \beta, \alpha, \theta)$:
  - $T$ tree structure ($T_{\beta}, T_L, \delta, p, l, r$)
  - $\beta$ feature selection function
  - $\alpha$ threshold selection function
  - $\theta$ leaf labelling function

- Recursive prediction for point $x_i$:
  $\Theta(t, x_i) = \begin{cases} 
  \theta(t) & \text{if } t \in T_L \\
  \Theta(l(t), x_i) & \text{else if } x_i[\beta(t)] \leq \alpha(t) \\
  \Theta(r(t), x_i) & \text{else}
  \end{cases}$
Decision Trees

Ways to construct decision trees:

1. **Local search** and **heuristics**

2. **Combinatorial optimization:**
   - optimality guarantees
   - additional constraints
Optimization Problem

**SAT:**
- A set of variables $\mathcal{X} = \{x_0, x_1, ..., x_n\}$ and a set of clauses $\mathcal{C} = \{C_1, C_2, ..., C_k\}$
- Find an assignment $\mathcal{M}: \mathcal{X} \rightarrow \{false, true\}$ that satisfies all clauses

**MaxSAT:**
- All hard clauses $\mathcal{C}_h$ should be satisfied
- The number of satisfied soft clauses $\mathcal{C}_s$ needs to be maximized
Encoding
Encoding Components

- It is straight-forward to encode:
  - feature selection
  - leaf labelling
  - presence at leaves

- The challenging component is the split

- How can we model a numerical threshold?
Split Encoding

- Existing approach:
  - only support binary features!
  - transform numerical and categorical features into a set of binary ones
  - can lead to a huge number of features

- Avellaneda’s [2020], Hu et al.’s [2020], and Verhaeghe et al.’s [2020] employ this approach

<table>
<thead>
<tr>
<th>Numerical</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>001</td>
</tr>
<tr>
<td>9</td>
<td>011</td>
</tr>
<tr>
<td>1</td>
<td>000</td>
</tr>
<tr>
<td>4</td>
<td>001</td>
</tr>
<tr>
<td>12</td>
<td>111</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Categorical</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>001</td>
</tr>
<tr>
<td>B</td>
<td>010</td>
</tr>
<tr>
<td>A</td>
<td>001</td>
</tr>
<tr>
<td>C</td>
<td>100</td>
</tr>
</tbody>
</table>
Split Encoding

- New idea:
  - encode the **direction** for each Point instead
  - validate the directions according to the **order** of values
  - the directions for **absent** points are encoded as well
Our Encoding

Variables:

- $a_{t,j}$: feature $j$ is chosen at node $t$
- $s_{i,t}$: point $i$ is directed left at node $t$
- $z_{i,t}$: point $i$ ends up at leaf $t$
- $g_{t,c}$: label $c$ is assigned to leaf $t$

Parameters:

- set of features $F$ and integer labels $C$
- set of training examples $\mathcal{X}$
- labelling $\gamma: X \rightarrow C$
- tree structure $T = \{\delta, T_B, T_L, p, l, r\}$
Our Encoding

Variables:

- $a_{t,j}$: feature $j$ is chosen at node $t$
- $s_{i,t}$: point $i$ is directed left at node $t$
- $z_{i,t}$: point $i$ ends up at leaf $t$
- $g_{t,c}$: label $c$ is assigned to leaf $t$

Clauses:

- Exactly one feature is chosen at each branching node

\[ \left( \neg a_{t,j}, \neg a_{t,j'} \right) \quad t \in T_B, j \neq j' \in F \]

\[ \left( \bigvee_{j \in F} a_{t,j} \right) \quad t \in T_B \]
Our Encoding

Variables:
- $a_{t,j}$: feature $j$ is chosen at node $t$
- $s_{i,t}$: point $i$ is directed left at node $t$
- $z_{i,t}$: point $i$ ends up at leaf $t$
- $g_{t,c}$: label $c$ is assigned to leaf $t$

Clauses:
- The directions for splits are in order

\[
(\neg a_{t,j}, s_{i,t}, \neg s_{i',t}) \quad t \in T_B, j \in F, (i, i') \in O_j(X)
\]

\[
(\neg a_{t,j}, \neg s_{i,t}, s_{i',t}) \quad t \in T_B, j \in F, (i, i') \in O_j(X), x_i[j] = x_{i'}[j]
\]
Our Encoding

Variables:

- $a_{t,j}$: feature $j$ is chosen at node $t$
- $s_{i,t}$: point $i$ is directed left at node $t$
- $z_{i,t}$: point $i$ ends up at leaf $t$
- $g_{t,c}$: label $c$ is assigned to leaf $t$

Clauses:

- The splits are non-trivial

\[ \neg a_{t,j}, s_{\#^1_j,t} \quad t \in \mathcal{T}_B, j \in F \]

\[ \neg a_{t,j}, s_{\#^{\mid x\mid}_j,t} \quad t \in \mathcal{T}_B, j \in F \]
Our Encoding

Variables:

- $a_{t,j}$: feature $j$ is chosen at node $t$
- $s_{i,t}$: point $i$ is directed left at node $t$
- $z_{i,t}$: point $i$ ends up at leaf $t$
- $g_{t,c}$: label $c$ is assigned to leaf $t$

Clauses:

- **Presence at leaves** matches the split directions

\[
\begin{align*}
& (\neg z_{i,t}, s_{i,t'}) & t \in T_L, x_i \in X, t' \in A_l(t) \\
& (\neg z_{i,t}, \neg s_{i,t'}) & t \in T_L, x_i \in X, t' \in A_r(t) \\
& (z_{i,t}, \bigvee_{t' \in A_l(t)} \neg s_{i,t'}, \bigvee_{t' \in A_r(t)} s_{i,t'}) & t \in T_L, x_i \in X
\end{align*}
\]
Our Encoding

Variables:

◦ $a_{t,j}$: feature $j$ is chosen at node $t$
◦ $s_{i,t}$: point $i$ is directed left at node $t$
◦ $z_{i,t}$: point $i$ ends up at leaf $t$
◦ $g_{t,c}$: label $c$ is assigned to leaf $t$

Clauses:

◦ At most one label is chosen at each leaf

$$(\neg g_{t,c}, \neg g_{t,c'}) \quad t \in T_L, c \neq c' \in C$$
Learning Decision Trees

Two main objectives:

- **Min-depth:**
  - correctly classify **all** of the training points
  - find the **lowest depth** possible
  - solved by iterative **SAT** instances

- **Max-accuracy:**
  - **maximize** the number of correct classifications
  - use a fixed **depth**
  - solved via **MaxSAT**
Extension to Categorical Features

- Use the same idea for **categorical splits**:
  - no need to validate **order** in directions, checking **equality** is enough
  - enables **power set** branching:
    - **min-depth**: potentially more **shallow** solution
    - **max-accuracy**: potentially more **accurate** solution
Experimental
The chosen baselines are the state-of-the-art algorithms for their respective objectives.
Goals

- The benefits and applications of the two objectives are well-studied

- Focus on optimization performance:
  - find the solutions faster
  - find near-optimal solutions in time-out scenarios
Datasets

- Three types of datasets:
  - mostly **numerical** features
  - mostly **categorical** features
  - mostly **binary** features

| Type | Name         | $|X|$ | $|F_N|$ | $|F_D|$ | $|F_C|$ | $\tilde{f}$ | $|C|$ |
|------|--------------|-----|------|------|------|--------|------|
| N    | Banknote     | 1372| 4    | 0    | 0    | 5016   | 2    |
|      | Breast Cancer| 116 | 9    | 0    | 0    | 891    | 2    |
|      | Cryotherapy  | 90  | 5    | 1    | 0    | 93     | 2    |
|      | Immunotherapy| 91  | 6    | 1    | 0    | 166    | 2    |
|      | Ionosphere   | 351 | 32   | 2    | 0    | 8114   | 2    |
|      | Iris         | 150 | 4    | 0    | 0    | 119    | 3    |
|      | User Knowledge| 258 | 5    | 0    | 0    | 431    | 4    |
|      | Vertebral Column| 310 | 6    | 0    | 0    | 1741   | 2    |
|      | Wine         | 178 | 13   | 0    | 0    | 1263   | 3    |
| B    | Car$^1$      | 1728| 6    | 0    | 0    | 15     | 2    |
|      | Monk2        | 169 | 4    | 2    | 0    | 11     | 2    |
Min-Depth Results

- On non-binary datasets, our approach is significantly faster than the baseline.

- As expected, the existing approach works better on binary datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Min Depth</th>
<th>Time (s)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Ours</td>
<td>SAT [3]</td>
<td></td>
</tr>
<tr>
<td>Banknote</td>
<td>4</td>
<td>5.82</td>
<td>T/O [4]</td>
<td></td>
</tr>
<tr>
<td>Cryotherapy</td>
<td>4</td>
<td>0.08</td>
<td>0.24</td>
<td></td>
</tr>
<tr>
<td>Immunotherapy</td>
<td>4</td>
<td>0.18</td>
<td>1.3</td>
<td></td>
</tr>
<tr>
<td>Iris</td>
<td>4</td>
<td>0.04</td>
<td>0.17</td>
<td></td>
</tr>
<tr>
<td>User Knowledge</td>
<td>5</td>
<td>1.31</td>
<td>59.44</td>
<td></td>
</tr>
<tr>
<td>Vertebral Column</td>
<td>5</td>
<td>87.35</td>
<td>T/O [5]</td>
<td></td>
</tr>
<tr>
<td>Wine</td>
<td>3</td>
<td>0.11</td>
<td>14.75</td>
<td></td>
</tr>
<tr>
<td>Car</td>
<td>8</td>
<td>T/O [8]</td>
<td></td>
<td>89.1</td>
</tr>
<tr>
<td>Monk2</td>
<td>6</td>
<td>2.73</td>
<td></td>
<td>0.28</td>
</tr>
</tbody>
</table>
Max-Accuracy Results

- On **non-binary** datasets, our approach is significantly faster than the baselines.
- As expected, existing approaches work better on **binary** datasets.
- Our approach still finds **optimal solutions** for **binary** datasets most of the time.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Depth</th>
<th>Solution Cost</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banknote</td>
<td>2</td>
<td>100</td>
<td>176</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>23</td>
<td>550</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0</td>
<td>88</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>2</td>
<td>19</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>9</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>Cryotherapy</td>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Immunotherapy</td>
<td>2</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>2</td>
<td>29</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>21</td>
<td>186</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>10</td>
<td>76</td>
</tr>
<tr>
<td>Iris</td>
<td>2</td>
<td>6</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>User Knowledge</td>
<td>2</td>
<td>35</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Vertebral Column</td>
<td>2</td>
<td>45</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>32</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>15</td>
<td>39</td>
</tr>
<tr>
<td>Wine</td>
<td>2</td>
<td>6</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Car</td>
<td>2</td>
<td>250</td>
<td>250</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>182</td>
<td>182</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>122</td>
<td>122</td>
</tr>
<tr>
<td>Monk2</td>
<td>2</td>
<td>57</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>32</td>
<td>31</td>
</tr>
</tbody>
</table>
Summary

- Novel MaxSAT-based encoding for constructing optimal decision trees for datasets with numerical and categorical Features

- Can be employed by both min-depth and max-accuracy objectives

- Supports power set splitting on categorical features to achieve compactness

- Significantly outperforms the state of the art for non-binary datasets
Thank You!

Questions & Answers

POUYA SHATI, ELDAN COHEN, SHEILA MCILRAITH
UNIVERSITY OF TORONTO
CP 2021
SEPTEMBER 13, 2021