Regularizing Soft Decision Trees

Olcay Taner Yıldız¹  Ethem Alpaydın²

¹Dept of Computer Engineering, İşık University, TR-34980, Istanbul, Turkey
²Dept of Computer Engineering, Boğaziçi University, TR-34342, Istanbul, Turkey

ISCIS 2013
Hard Decision Tree

- Each decision node $m$ applies a test $g_m(x)$ and chooses one of the children accordingly.

  $$F_m(x) = \begin{cases} 
  F^L_m(x) & \text{if } g_m(x) > 0 /* true */ \\
  F^R_m(x) & \text{otherwise } /* false */
  \end{cases}$$

- Classification: Leaves carry the label of one of $K$ classes
- Regression: Leaves carry a constant which is the numeric regression value.
Hard Decision Tree Types

- **Univariate tree**: \( g_m(x) = x_j + w_{m0} > 0. \)
- **Multivariate linear tree**: \( g_m(x) = w_m^T x + w_{m0} > 0. \)
- **Multivariate nonlinear tree**: \( g_m(x) = \sum_{j=1}^{k} w_j \phi_j(x) > 0. \)
- **Omnivariate tree**: \( g_m(x) \) can be any of the above, chosen by a statistical model selection procedure.
Soft Decision Node redirects instances to all its children with probabilities calculated by a \textit{gating function} \( g_m(x) \).

\[
F_m(x) = F^L_m(x)g_m(x) + F^R_m(x)(1 - g_m(x))
\]

\[
g_m(x) = \frac{1}{1 + \exp[-(w^T_m x + w_{m0})]}
\]

Gating model implements a discriminative (logistic linear) model estimating the posterior probability of the left child.
Hard vs. Soft Tree (Toy Dataset)
Response and Error

1 function $F_m(x)$
2 if $m$ is leaf node
3 \[ y = z_m \text{ /* leaf value at } m */ \]
4 else
5 \[ g_m(x) = \text{sigmoid}(\mathbf{w}_m^T \mathbf{x} + w_{m0}) \]
6 \[ y = F_m^L(x) g_m(x) + F_m^R(x)(1 - g_m(x)) \]
7 return $y$

Classification: $E = r \log y + (1 - r) \log(1 - y)$
## Training the Soft Decision Tree

```latex
1 \textbf{function} \text{LearnSoftTree}(m, \mathcal{X}, \mathcal{V})
2 \quad E_{\text{before}} = \text{ErrorOfTree}(\mathcal{V})
3 \quad \text{initialize} \ w_{mj}, z^L_m, \text{and} \ z^R_m
4 \quad \text{repeat}
5 \quad \quad \text{for all} \ (x, r) \in \mathcal{X}
6 \quad \quad \quad \delta(x) = (F_{\text{root}}(x) - r)(g_p(x))^{\text{left}}(1 - g_p(x))^{\text{right}}
7 \quad \quad \quad \text{for} \ j = 0, \ldots, d
8 \quad \quad \quad \quad w_{mj} = w_{mj} - \eta \delta(x)(F^L_m(x) - F^R_m(x))v_m(x)(1 - v_m(x))x_j
9 \quad \quad \quad \quad z^L_m = z^L_m - \eta \delta(x)v_m(x)
10 \quad \quad \quad \quad z^R_m = z^R_m - \eta \delta(x)(1 - v_m(x))
11 \quad \text{until} \ \text{convergence}
12 \quad E_{\text{after}} = \text{ErrorOfTree}(\mathcal{V})
13 \quad \text{if} \ E_{\text{after}} < E_{\text{before}}
14 \quad \quad \text{LearnSoftTree}(m.\text{left}, \mathcal{X}, \mathcal{V})
15 \quad \text{LearnSoftTree}(m.\text{right}, \mathcal{X}, \mathcal{V})
```
Regularization

- Local dimensionality reduction for better generalization
- $L_1$ regularization:

\[ E_{L_1} = (1 - \lambda) \text{CrossEntropy} + \lambda \sum_{i=0}^{d} |w_{mi}| \]

- $L_2$ regularization:

\[ E_{L_2} = (1 - \lambda) \text{CrossEntropy} + \lambda \sum_{i=0}^{d} w_{mi}^2 \]
Experiments

- Hard decision tree (HDT), Linear discriminant tree (LDT), Soft decision tree (SDT) without regularization, SDT with $L_1$ regularization, SDT with $L_2$ regularization.

- Comparison on 27 data sets

- 2/3 training data with $5 \times 2$-fold cross validation, 1/3 test set.

- Parametric $5 \times 2$ paired $F$-test used for comparison on a single data set and nonparametric Nemenyi’s test for overall comparison.
Results: Accuracy

<table>
<thead>
<tr>
<th></th>
<th>Hard</th>
<th>Ldt</th>
<th>Soft</th>
<th>Soft($L_1$)</th>
<th>Soft($L_2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard</td>
<td>6</td>
<td>0</td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Ldt</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Soft</td>
<td>9</td>
<td>11</td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Soft($L_1$)</td>
<td>7</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Soft($L_2$)</td>
<td>8</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Olcay Taner Yıldız¹, Ethem Alpaydın²

Regularizing Soft Decision Trees
## Results: Node Counts

<table>
<thead>
<tr>
<th></th>
<th>Hard</th>
<th>Ldt</th>
<th>Soft</th>
<th>Soft($L_1$)</th>
<th>Soft($L_2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hard</strong></td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>Ldt</strong></td>
<td>11</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td><strong>Soft</strong></td>
<td>10</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td><strong>Soft($L_1$)</strong></td>
<td>12</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td><strong>Soft($L_2$)</strong></td>
<td>13</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

![Tree Diagram](image)

**Olcay Taner Yıldız¹, Ethem Alpaydın²**

Regularizing Soft Decision Trees
Conclusions: Soft Trees

- Proposed decision tree model with soft decisions, which makes use of a soft gating function to merge the decisions of the subtrees.
- Soft trees have smoother fits and hence lower bias around the split boundaries.
- Linear gating function enables soft trees to make oblique splits in contrast to the axis-orthogonal splits made by hard trees.
Conclusions: Regularization

- We extend the soft decision tree model by adding $L_1$ and $L_2$ regularization to penalize unnecessary complexity.
- Both versions improve accuracy slightly and decrease complexity significantly.
- $L_2$ regularization seems to work slightly better than $L_1$. 