Regularizing Soft Decision Trees

Olcay Taner Yıldız[1] Ethem Alpaydın[2]

[1]Dept of Computer Engineering, Işık University, TR-34980, Istanbul, Turkey

[2]Dept of Computer Engineering, Boğaziçi University, TR-34342, Istanbul, Turkey

ISCIS 2013

Outline

Hard Decision Tree

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

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

- 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*: $g_m(\boldsymbol{x}) = x_j + w_{m0} > 0$.
- *Multivariate linear tree*: $g_m(\boldsymbol{x}) = \boldsymbol{w}_m^T \boldsymbol{x} + w_{m0} > 0$.
- *Multivariate nonlinear tree*: $g_m(\boldsymbol{x}) = \sum_{j=1}^{k} w_j \phi_j(\boldsymbol{x}) > 0$.
- *Omnivariate tree*: $g_m(\boldsymbol{x})$ can be any of the above, chosen by a statistical model selection procedure.

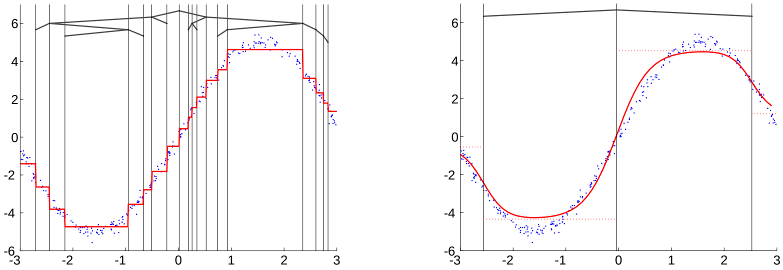
Soft Decision Tree

- Soft decision node redirects instances to all its children with probabilities calculated by a *gating function* $g_m(\boldsymbol{x})$.

$$
\begin{aligned}
F_m(\boldsymbol{x}) &= F_m^L(\boldsymbol{x})g_m(\boldsymbol{x}) + F_m^R(\boldsymbol{x})(1 - g_m(\boldsymbol{x})) \\
g_m(\boldsymbol{x}) &= \frac{1}{1 + \exp[-(\boldsymbol{w}_m^T\boldsymbol{x} + w_{m0})]}
\end{aligned}
$$

- 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

function $F_m(\boldsymbol{x})$
    if m is leaf node
        $y = z_m$ /* leaf value at m */
    else
        $g_m(\boldsymbol{x}) = sigmoid(\boldsymbol{w}_m^T \boldsymbol{x} + w_{m0})$
        $y = F_m^L(\boldsymbol{x}) g_m(\boldsymbol{x}) + F_m^R(\boldsymbol{x})(1 - g_m(\boldsymbol{x}))$
    return y

- Classification: $E = r \log y + (1 - r) \log(1 - y)$

Training the Soft Decision Tree

function LearnSoftTree(m, $\mathcal{X}$, $\mathcal{V}$)
    $E_{before} = \text{ErrorOfTree}(\mathcal{V})$
    initialize w_{mj}, z_m^L, and z_m^R
    repeat
        for all $(\boldsymbol{x}, r) \in \mathcal{X}$
            $\delta(\boldsymbol{x}) = (F_{root}(\boldsymbol{x}) - r)(g_p(\boldsymbol{x}))^{left}(1 - g_p(\boldsymbol{x}))^{right}$
            for $j = 0, \ldots, d$
                $w_{mj} = w_{mj} - \eta\delta(\boldsymbol{x})(F_m^L(\boldsymbol{x}) - F_m^R(\boldsymbol{x}))v_m(\boldsymbol{x})(1 - v_m(\boldsymbol{x}))x_j$
            $z_m^L = z_m^L - \eta\delta(\boldsymbol{x})v_m(\boldsymbol{x})$
            $z_m^R = z_m^R - \eta\delta(\boldsymbol{x})(1 - v_m(\boldsymbol{x}))$
    until convergence
    $E_{after} = \text{ErrorOfTree}(\mathcal{V})$
    if $E_{after} < E_{before}$
        LearnSoftTree(m.left, $\mathcal{X}$, $\mathcal{V}$)
        LearnSoftTree(m.right, $\mathcal{X}$, $\mathcal{V}$)

Regularization

- Local dimensionality reduction for better generalization
- L_1 regularization:

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

- L_2 regularization:

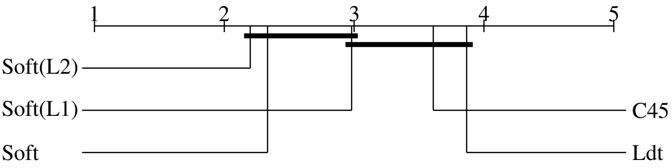
$$E_{L_2} = (1 - \lambda) CrossEntropy + \lambda \sum_{i=0}^{d} w_{mi}^2$$

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 5×2-fold cross validation, 1/3 test set.
- Parametric 5×2 paired F-test used for comparison on a single data set and nonparametric Nemenyi's test for overall comparison.

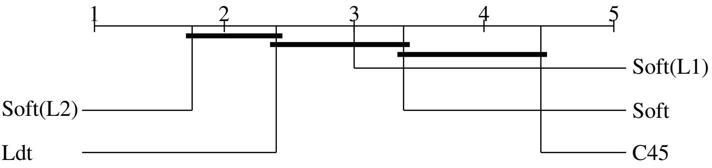
Results: Accuracy

	Hard	Ldt	Soft	Soft(L_1)	Soft(L_2)
Hard		6	0	5	4
Ldt	5		0	1	1
Soft	9	11		5	4
Soft(L_1)	7	4	1		0
Soft(L_2)	8	4	1	2	



Results: Node Counts

	Hard	Ldt	Soft	Soft(L_1)	Soft(L_2)
Hard		0	1	1	1
Ldt	11		4	2	0
Soft	10	2		1	0
Soft(L_1)	12	3	2		0
Soft(L_2)	13	4	4	1	



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 unnnecessary complexity.
- Both versions improve accuracy slightly and decrease complexity significantly.
- L_2 regularization seems to work slightly better than L_1.