Soft Decision Trees

Ozan İrsoy[1] Olcay Taner Yıldız[2] Ethem Alpaydın[1]

[1]Department of Computer Engineering, Boğaziçi University, TR-34342, Istanbul, Turkey

[2]Department of Computer Engineering, Işık University, TR-34980, Istanbul, Turkey

ICPR 2012

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*: $g_m(\boldsymbol{x}) = x_j + w_{m0} > 0$ (Quinlan, 1993).
- *Multivariate linear tree*: $g_m(\boldsymbol{x}) = \boldsymbol{w}_m^T \boldsymbol{x} + w_{m0} > 0$ (Murthy and Salzberg, 1994), (Yildiz and Alpaydin, 2005).
- *Multivariate nonlinear tree*: $g_m(\boldsymbol{x}) = \sum_{j=1}^{k} w_j \phi_j(\boldsymbol{x}) > 0$ (Guo and Gelfand, 1992).
- *Omnivariate tree*: $g_m(\boldsymbol{x})$ can be any of the above, chosen by a statistical model selection procedure (Yildiz and Alpaydin, 2001).

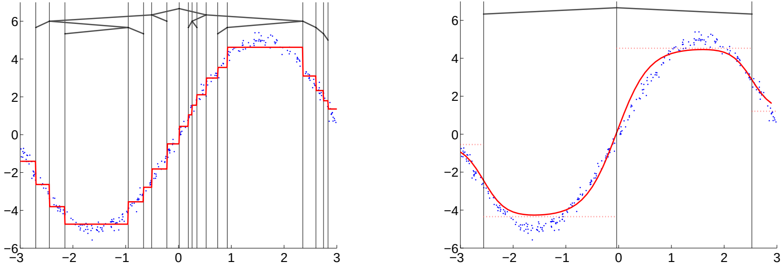
Soft Decision Tree

- 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
        $g_m(\boldsymbol{x}) = 1/(1 + \exp(-(\boldsymbol{w}_m^T \boldsymbol{x} + w_{m0})))$
        $y = F_m^L(\boldsymbol{x}) g_m(\boldsymbol{x}) + F_m^R(\boldsymbol{x})(1 - g_m(\boldsymbol{x}))$
    return y

- Classification: $E = r \log y + (1 - r) \log(1 - y)$
- Regression: $E = (r - y)^2$

Training 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, $\mathcal{X}$, $\mathcal{V}$)

Results: Regression

	Mean Square Error		Tree Size	
	Soft	Hard	Soft	Hard
ABA	**0.439**	0.557	**7**	32
ADD	**0.094**	0.267	**15**	202
BOS	0.271	0.344	**11**	18
CAL	**0.312**	0.326	**3**	201
COM	0.037	0.046	**5**	30
CON	0.264	0.286	**13**	69
8FH	**0.383**	0.418	**3**	40
8FM	**0.057**	0.074	**3**	92
8NH	0.388	0.416	**9**	52
8NM	**0.054**	0.084	**13**	144

Results: Classification

	Accuracy			Tree Size		
	Soft	Hard	LDT	Soft	Hard	LDT
BRE	95.34	93.80	95.09	<u>17</u>	47	**4**
GER	<u>75.74</u>	69.07	74.16	<u>16</u>	142	**7**
MAG	81.27	<u>84.09</u>	83.07	<u>17</u>	1072	40
MUS	92.25	94.62	93.59	<u>22</u>	202	15
PIM	70.85	69.41	**76.89**	<u>26</u>	111	**5**
POL	<u>77.41</u>	69.81	77.45	<u>21</u>	558	**5**
RIN	88.94	87.54	77.25	368	354	**4**
SAT	83.90	84.01	83.30	<u>11</u>	163	9
SPA	78.38	<u>90.14</u>	89.86	<u>22</u>	155	**13**
TWO	<u>97.92</u>	87.59	98.00	<u>41</u>	429	**3**

Summary

- Proposed decision tree model with soft decisions, which makes use of a soft gating function to merge the decisions of the subtrees.
- The model is shown to have better or comparable performance to hard trees, while having fewer nodes.
- One drawback of soft trees is gradient-descent which is prone to get stuck at local minima.

Conclusions: Soft Trees vs. Hard Trees

- 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.