Budding Trees

Ozan İrsoy[1], Olcay Taner Yıldız[2], Ethem Alpaydın[3]

[1]Department of Computer Science, Cornell University, 14853, Ithaca, NY
[2]Department of Computer Engineering, Işık University, TR-34980, Istanbul, Turkey
[3]Department of Computer Engineering, Boğaziçi University, TR-34342, Istanbul, Turkey

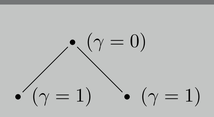
Budding Tree

- Softening the notion of *being a leaf*, a bud node redirects instances to all its children (as in an internal node), as well as makes a contribution itself (as in a leaf node).

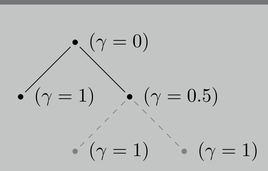
$$F_m(\mathbf{x}) = \gamma\rho_m + (1-\gamma)(F_m^L(\mathbf{x})g_m(\mathbf{x}) + F_m^R(\mathbf{x})(1-g_m(\mathbf{x}))$$
$$g_m(\mathbf{x}) = \frac{1}{1+\exp[-(\mathbf{w}_m^T\mathbf{x}+w_{m0})]}$$

- $\gamma \in [0,1]$ is the (soft) *leafness* parameter.
- $g_m(x) \in (0,1)$ is the gating function that defines a soft selection among two children.
- Regression: ρ is a scalar. The objective is MSE.
- Binary Classification: ρ is a scalar, and there is an additional sigmoid application to the root response. The objective is cross-entropy.
- k-class Classification: ρ is a k dimensional vector, and there is an additional softmax application to the root response. The objective is cross-entropy.

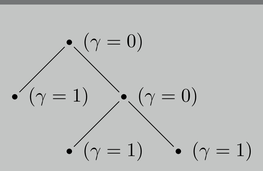
Bud Node



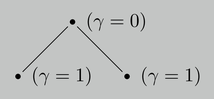
1. Budding tree with one internal and two leaf nodes. For an input, internal node makes a soft selection among its children.



2. Right child makes a smooth transition to an internal node. At this state, it redirects half of the input responsibility to its children, and the other half to itself, still making a contribution to the response as a partial leaf.



3. Right child continues its transition to a pure internal node. Currently, it does not have any leafness left, so it redirects all of the input responsibility to its children.



4. Node can go back to being a leaf during training. In general, any node can go back an forth, becoming a leaf or internal, depending on the signal.

Training a Budding Tree

Objective function is $J = E + \lambda\sum_m(1-\gamma_m)$ where E is the error (cross-entropy for classification and MSE for regression) and the other term is the regularizing term that favors leaf nodes (hence, small trees).
We use stochastic gradient descent with the following gradient equations:

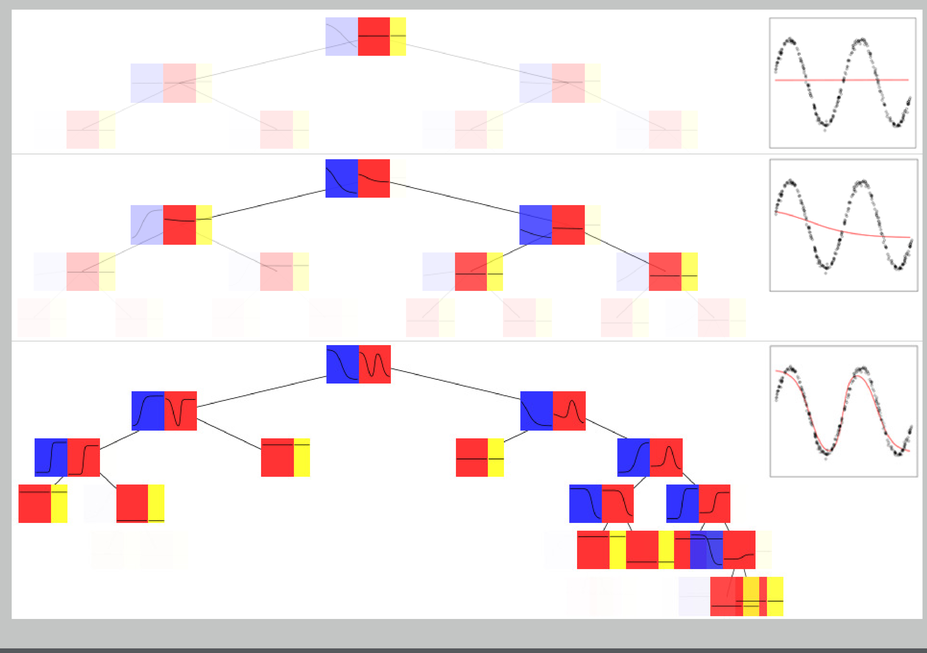
$$\frac{\partial J^t}{\partial w_{mi}} = \delta_m^t(1-\gamma_m)g_m(\mathbf{x}^t)(1-g_m(\mathbf{x}^t))(y_{ml}(\mathbf{x}^t)-y_{mr}(\mathbf{x}^t))x_i^t$$
$$\frac{\partial J^t}{\partial \rho_m} = \delta_m^t\gamma_m$$
$$\frac{\partial J^t}{\partial \gamma_m} = \delta_m^t[-g_m(\mathbf{x}^t)y_{ml}(\mathbf{x}^t) - (1-g_m(\mathbf{x}^t))y_{mr}(\mathbf{x}^t) + \rho_m] - \lambda$$

with

$$\delta_m^t = \begin{cases} y_r(\mathbf{x}) - r^t & \text{if } m \text{ is root } r \\ \delta_{pa(m)}^t(1-\gamma_m)g_m(\mathbf{x}^t) & \text{if } m \text{ is a left child} \\ \delta_{pa(m)}^t(1-\gamma_m)(1-g_m(\mathbf{x}^t)) & \text{if } m \text{ is a right child} \end{cases}$$

where $pa(m)$ is the parent node of node m.
Note that γ_m are constrained to be in $[0,1]$ during updates.

Visualization of Budding Tree Training on Toy Data



Summary and Conclusions

- Budding trees are trained jointly, the parameters of all nodes are learned together, whereas soft trees (as well as hard trees) are trained incrementally.
- The model is shown to have better or comparable performance to hard and soft trees, while having fewer nodes.
- Budding trees solve the optimization problem over all node parameters, unlike soft and hard trees which solve incremental subproblems.
- Continuous induction of budding trees allow them to be used in an online learning setting, with small updates.
- Floating search over the tree space allow internal nodes to become leaves and leaf nodes to become internal nodes, unlike separate training and pruning phases in traditional decision tree construction.
- Budding trees retain the advantages of soft trees over hard trees: Soft response function provides smoother fits and less bias near the decision boundaries. A linear gating function enables oblique splits in contrast to axis-orthogonal splits of univariate trees.

Results: Regression, Binary Classification & Multiclass Classification

	MSE			Node count		
	Hard	Soft	Budding	Ha.	So.	Bu.
ABA	0.541	0.421	0.416	44	**21**	35
ADD	0.244	0.070	**0.046**	327	49	**35**
BOS	0.342	0.273	0.218	19	32	19
CAL	0.311	0.251	**0.240**	300	146	**94**
COM	0.036	0.023	**0.019**	110	31	**19**
CON	0.268	0.208	**0.156**	101	39	38
8FH	0.416	0.381	**0.378**	47	**5**	13
8FM	0.068	0.051	**0.050**	164	**9**	17
8NH	0.394	0.358	**0.342**	77	**24**	27
8NM	0.066	0.049	**0.036**	272	**23**	37

	Accuracy				Node count			
	C4.5	LDT	Soft	Bud.	C.	L.	So.	Bu.
BRE	93.2	95.0	**96.5**	94.9	7	4	3	12
GER	70.0	74.1	**75.9**	68.7	**1**	3	8	56
MAG	82.5	83.0	85.3	**86.3**	53	38	56	122
MUS	94.5	93.5	95.6	**97.0**	62	11	33	15
PIM	72.1	**76.8**	75.0	67.1	8	5	7	68
POL	69.4	77.4	77.1	72.5	34	**3**	18	61
RIN	87.7	77.2	**90.1**	88.5	93	**3**	76	61
SAT	84.5	83.3	87.5	86.8	25	**9**	27	38
SPA	90.0	89.8	**92.4**	91.4	36	13	12	49
TWO	82.9	98.0	97.9	96.7	163	**3**	7	29

	Accuracy				Node count			
	C4.5	LDT	Soft	Budding	C4.5	LDT	So.	Bu.
BAL	61.91	88.46	89.85	**92.44**	5	**3**	10	29
CMC	50.00	46.64	52.03	53.23	24	**3**	21	28
DER	94.00	93.92	93.6	94.80	15	11	11	11
ECO	77.47	81.39	76.78	**83.56**	9	11	10	24
GLA	56.62	53.37	54.05	53.78	20	9	11	21
OPT	84.85	93.73	90.97	**94.57**	120	**31**	58	40
PAG	**96.71**	94.65	95.7	96.51	23	29	**16**	37
PEN	92.95	96.60	96.64	**98.13**	169	66	54	54
SEG	94.48	91.96	93.99	**95.63**	41	33	**22**	33
YEA	54.61	56.66	55.82	**59.31**	24	22	34	41

Code, data and more visualization examples available at http://www.cs.cornell.edu/∼oirsoy/budding.htm.

oirsoy@cs.cornell.edu, olcaytaner@isikun.edu.tr, alpaydin@boun.edu.tr