Globally Induced Forest: A Prepruning Compression Scheme

Jean-Michel Begon, Arnaud Joly, Pierre Geurts

jm.begon@ulg.ac.be @JmBegon

Systems and Modeling, Deptartment of EE and CS, University of Liege, Belgium

Motivations

What? Is it possible to build accurate yet lightweight decision forests without building the whole model first?

Why? Decision forests are heavy models memory-wise:

- $\propto$ Number of nodes in a tree is (at worst) linear with the size of the data;
- $\propto$ number of required trees grows with the problem complexity.

What for?

- Big data;
- small memory devices;
- better interpretability, less overfitting, faster prediction, ...

How? Build an additive model corresponding to a forest by introducing optimal decision nodes sequentially until a node budget constraint is met.

GIF: decision forest and additive model

The forest prediction $\hat{y}^{(t)}(x)$ at step t for instance x is given by:

$$\hat{y}^{(t)}(x) = \hat{y}^{(t-1)}(x) + \lambda w_{j_t} z_{j_t}(x) = w_0 + \lambda \sum_{\tau=1}^{t} w_{j_\tau} z_{j_\tau}(x)$$

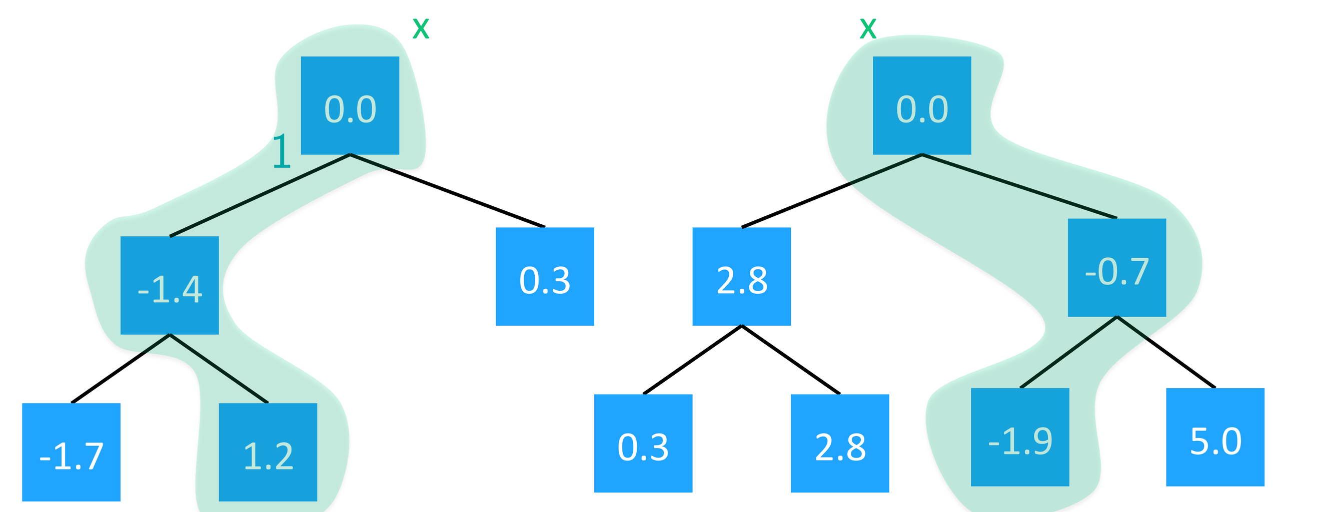
where

- j_τ is the node selected at step τ
- w_0 is some initial bias
- w_j is the weight of node j $(1 \leq j \leq t)$
- λ is the learning rate

$$z_j(x) = \begin{cases} 1, & \text{if } x \text{ reaches node } j \\ 0, & \text{otherwise} \end{cases}$$

$(1 \leq j \leq t)$

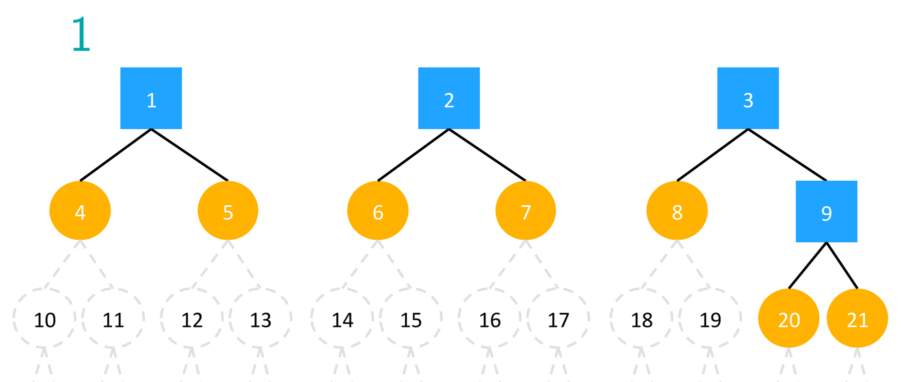
i.e. node j indicator function



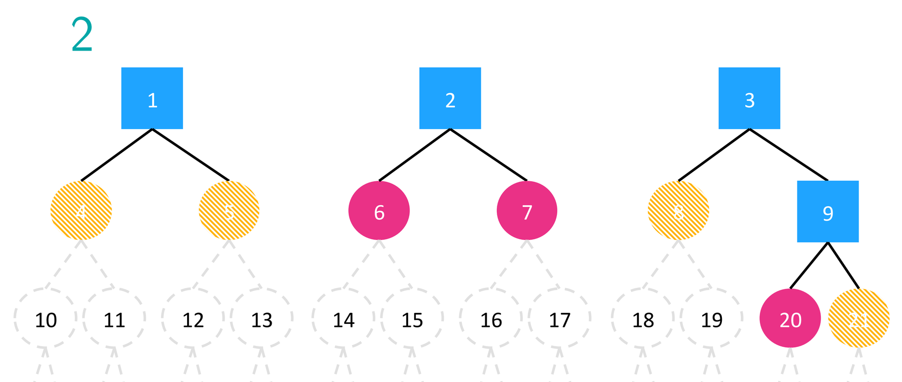
$$\hat{y}(x) = w_0 + \lambda(1.4 + 1.2) + \lambda(-0.7 + -1.9)$$

For classification, the sum of weights represents the class probability vector (i.e. the weights are multidimensional).

GIF algorithm

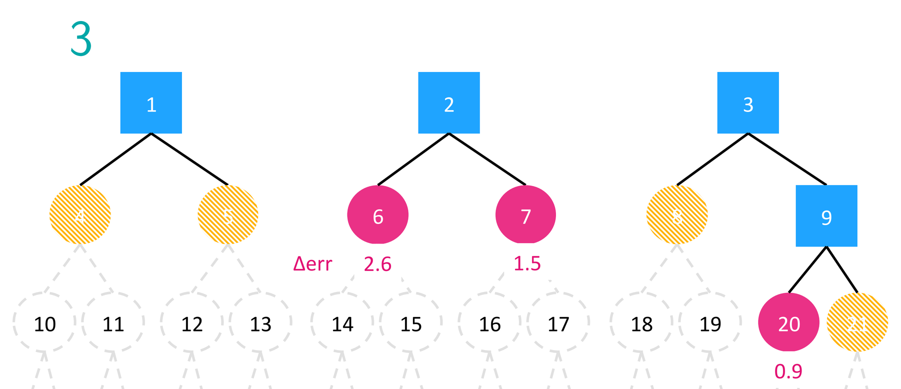


$$\hat{y}(.) = \bar{y} + \lambda w_9 z_9(.)$$



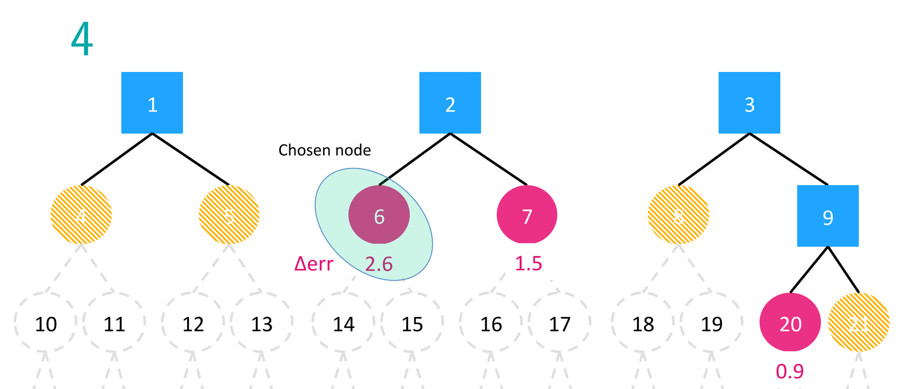
$$\hat{y}(.) = \bar{y} + \lambda w_9 z_9(.)$$

The actual candidates are sampled from the candidate list.



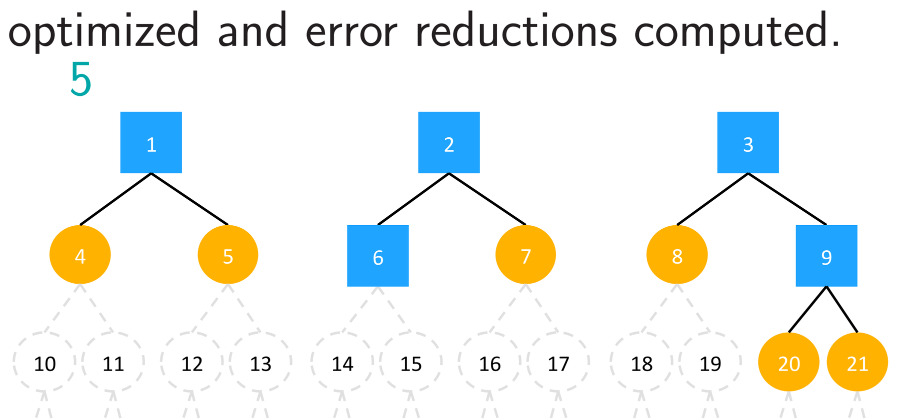
$$\hat{y}(.) = \bar{y} + \lambda w_9 z_9(.)$$

For all the actual candidates, weights are optimized and error reductions computed.



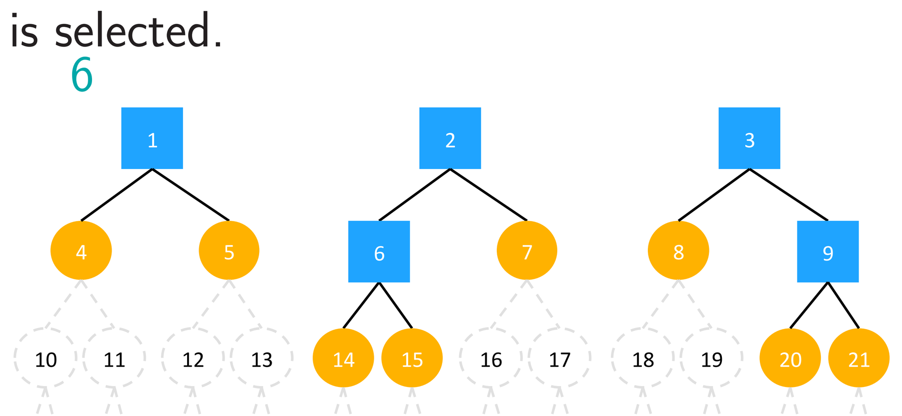
$$\hat{y}(.) = \bar{y} + \lambda w_9 z_9(.)$$

The node which reduces the error the most is selected.



$$\hat{y}(.) = \bar{y} + \lambda w_9 z_9(.) + \lambda w_6 z_6(.)$$

The best node is added to the model, together with its optimal weight.



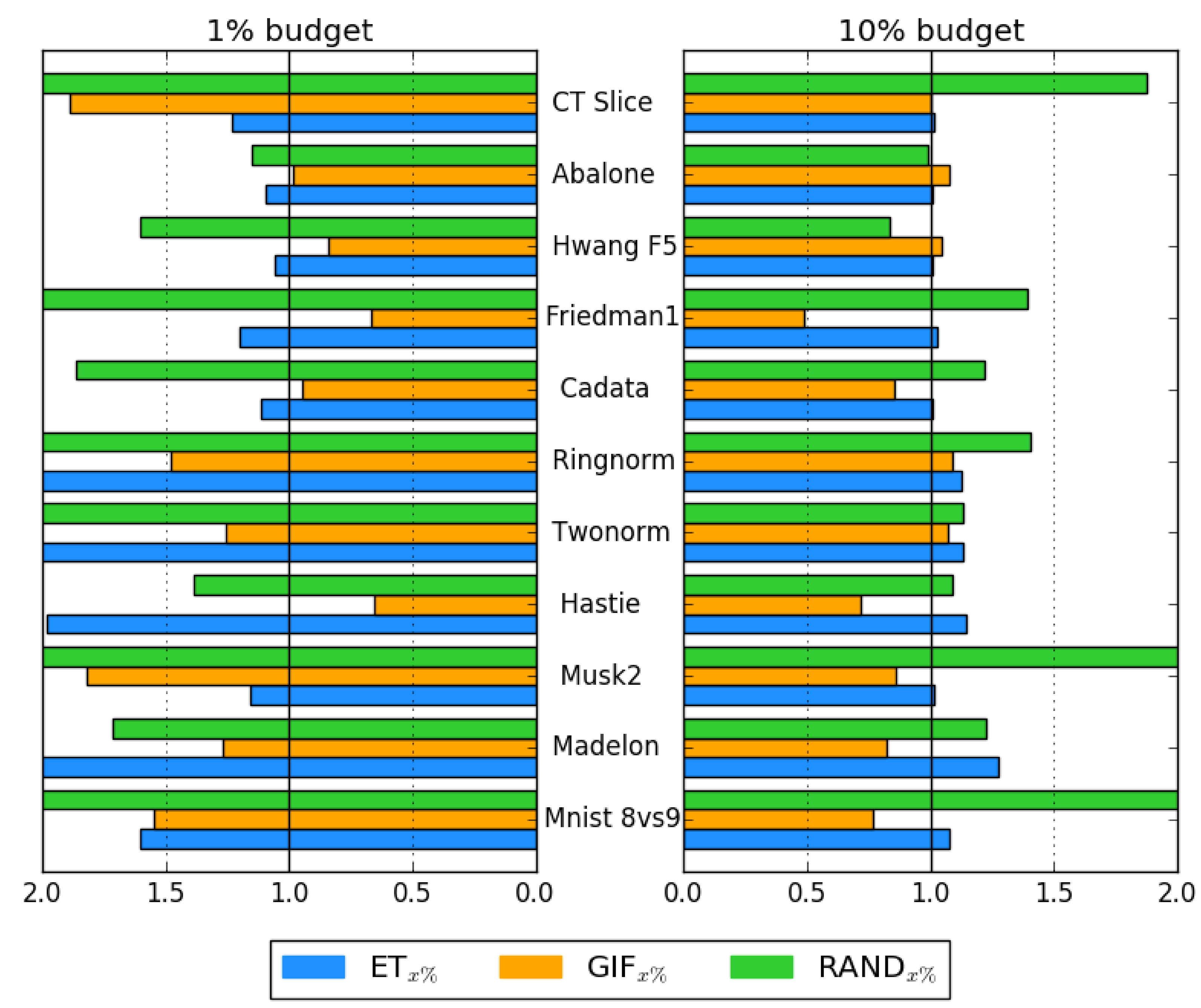
The chosen node is split according to a local criterion and its children are added to the candidate list.

GIF versus other prepruning methods

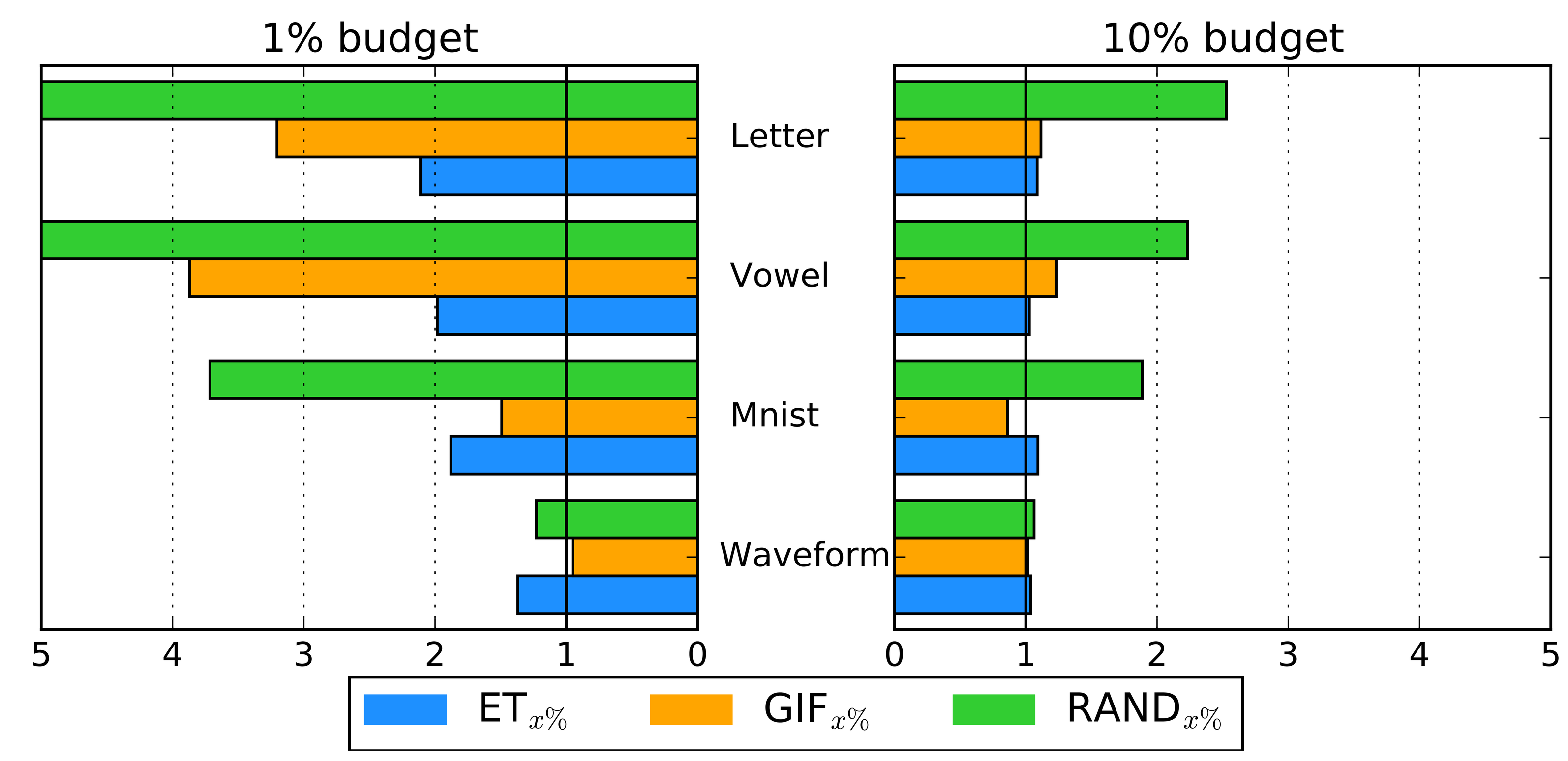
A reference forest of 1000 fully-developed extremely randomized trees ($\text{ET}_{100\%}$) was computed as reference and the total number of nodes B was extracted. Several baselines were tested under severe (1% of B) and mild (10% of B) constraints:

- **$\text{GIF}_{x\%}$** a forest of 1000 stumps developed according to the GIF algorithm ($\lambda = 10^{-1.5}$, $CW = 1$).
- **$\text{RAND}_{x\%}$** a forest of a 1000 trees where nodes are added randomly (the tree is selected uniformly, then is the node).
- **$\text{ET}_{x\%}$** a forest of 10x fully-developed ET.

Average results (over ten runs) are expressed relative to the original forest ($\text{ET}_{100\%}$).



Relative average error with respect to the original forest. Mean square error in regression, misclassification rate in classification



Relative average misclassification rate with respect to the original forest.

Take home message

- It is possible to build lightweight yet accurate decision forests directly.
- Global optimization helps.
- Optimizing the forest shape is hurtful: