Boosting Flexible Learning Ensembles with Dynamic Feature Selection

Alexander Borisov, Victor Eruhimov, Eugene Tuv

Intel Corp.

Challenging models / data we face

- both regression and classification models are of interest
- mixed type variables, categorical predictors with very large number of levels (hundreds or thousands)
- blocks of non-randomly missing data
- often datasets are extremely saturated - small number of observations and huge number of variables (tens of thousands) , with only small number relevant for a specific problem
- data is not clean, noise and outliers both in Xs and Ys
- ability to understand nature of learned relationships is crucial

An universal learner is needed ...

- Recent advances in tree based methods such as MART (Freidman's Gradient Tree Boosting) and RF (Breiman's Random Forests) are proven to be effective in addressing most of the issues listed above
- Both ensembles are resistant to outliers in X-space, both have efficient mechanism to handle missing data, both are competitive in accuracy with the best known learning algorithms in regression and classification settings, mixed type data is handled naturally , both allow (to different degree) to look inside of black box

An universal learner ...

- MART (simplified view)

A) Regression :

1) Set $F_0 = average(y_i)$ For m = 1…M

2) Compute residuals : $r_{im} = y_i - F_{m-1}(x_i)$

3) Fit tree to residuals: $T_m(X)$

4) Update model as: $F_m(X) = F_{m-1}(X) + \eta T_m(X)$

B) Classification : build K=number of response classes regression tree sequences. *k*-th sequence fits log-odds $f_k(X), p_k(X) = e^{f_K(X)} \big/ e^{f_1(X)} + \ldots + e^{f_K(X)}$ using the above scheme with pseudo-residuals

$$r_{ikm} = y_{ik} - p_k(x_i), i = 1 \ldots N, k = 1 \ldots K$$

An universal learner ...

- RF :
  - builds parallel ensemble of trees
  - each tree is grown on a bootstrap sample of the training set
  - at each node, a fixed small number (comparing to total number) of variables is selected, then the best split on these variables is selected..
  - resulting prediction is obtained by averaging in regression or voting in classification.

But when dealing with very large numbers of predictors…

- MART uses exhaustive search on all input variables for every split and every tree in ensemble, and it becomes computationally extremely expensive to handle very large number of predictors.
- RF shows noticeable degradation in accuracy in the presence of many noise variables

A simple trick to improve both:

- only a small subset of features is considered at every construction step of an individual learner in ensemble (like in RF)
- sampling distribution of features is dynamically modified to reflect currently learned feature importance
- this distribution is initialized as uniform, and progresses with adjustable rate to prevent initial overweighting of a few variables.
- feature importance is dynamically recalculated over the current ensemble (we used reduction in impurity due to splits on the feature as measure of it's importance).

Dynamic variable reweighting :

- *MART regression* : the weight of *n*-th variable in *i*-th step

(**) $$(1-m/M)^{ai} I_0 + \sum_{j=1}^{i} V_n^{(j)}$$

where

m - # selected variables, *M* – total # variables

$V_n^{(j)}$ - importance of *n*-th feature in *j*-th tree in an ensemble (total reduction in impurity due to splits on the feature in *i*-th tree)

I_0 - root node impurity of the first tree

- first term dominates initial weights, second represents current variable importances. ***a*** - adjustable parameter controlling how fast initial weights decrease (empirically chosen in range 0.5-2.)

Dynamic variable reweighting :

- *MART K-class classification* : the weight of *n*-th variable in *i*-th step is given by (**)

where

$V_n^{(j)}$ - sum of importances of n-th feature in K trees corresponding to *j*-th iteration

I_0 - the sum of root node impurities for K trees corresponding to 1-st iteration

- *Random Forest* : weight of *n*-th variable in *i*-th step is calculated as $aI_0 + \sum_{j=1}^{i} V_n^{(j)}$

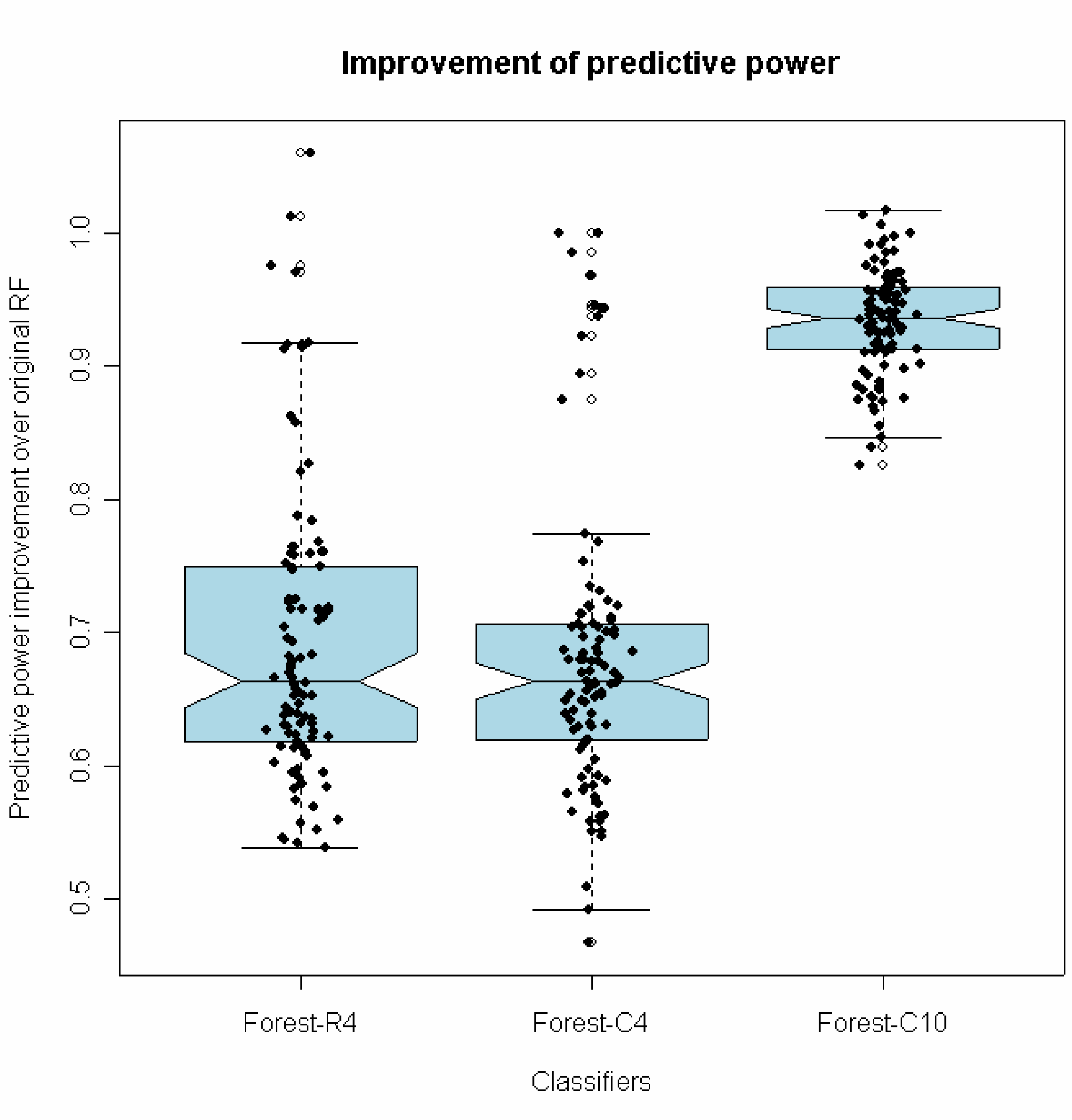
where I_0 is root node error for first tree,

a - adjustable parameter, taken usually as 5-10

Experiments

- Freidman’s (1999) random function generator was used
- 100 datasets with 50 vars generated: K significant inputs, 50-K noise inputs
- K=4,10
- data partitions train/test -3/2

Experiments (RF)



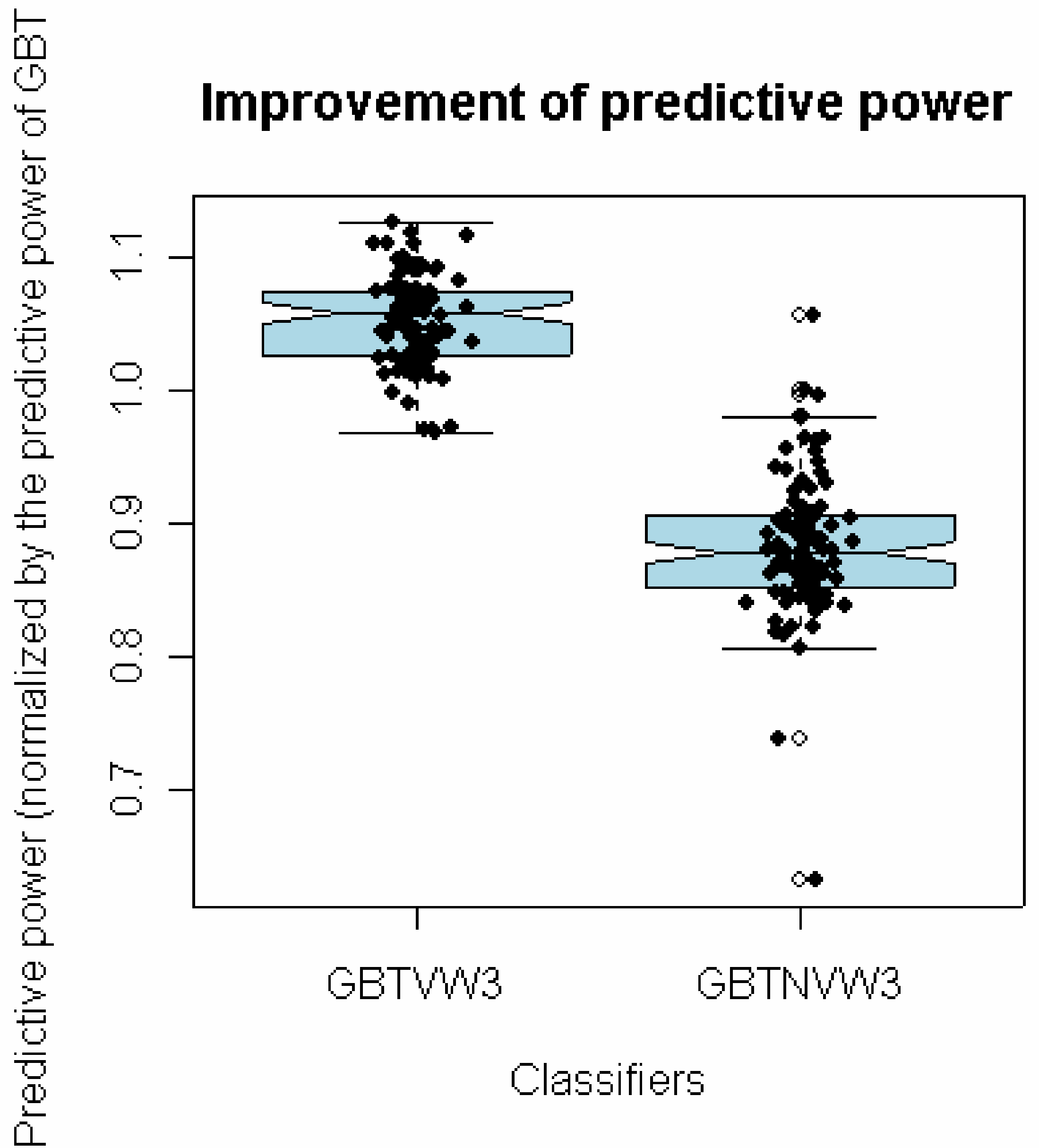
R4 – regression, K=4

C4/10 – classification, K=4,10

Error is relative to standard RF error

For 10/40 ratio of relevant/noise vars RF improvement is slight, where for 4/40 – very significant!

Experiments (MART)



Binary classification, K=10

GBTVW3 (variable weighting scheme applied, m=3, M=50)

GBTVW3 (m=3 selected uniformly, M=50)

Accuracy (1-err) is relative to standard GBT accuracy

GBTVW3 is slightly better than standard and 50/3 ~ 17 times faster!

Experiments

- UCI datasets : connect4, dna, letter-recognition, musk, segment
- RF, MART with/without dynamic variable weighting give similar accuracy (boosted Mart much faster)

Summary

- This method makes tree gradient boosting feasible (actually very fast) for the data with large number of predictors without loss of accuracy. It also adds bias correction element to RF in the presence of many noise variables.
- Our experiments showed slight improvement of predictive accuracy for MART on average and very significant for RF in the presence of noise.
- Note that RF with this method becomes a sequential ensemble and looses attractive computational parallelism.
- Feature selection challenge results were obtained using stochastic gradient boosting with dynamic feature selection implemented in IDEAL (internal tool) practically out of box with a few runs.
- IDEAL (interactive data exploration and learning) is optimized for IA, and will be available for non commercial use / educational purposes soon gratis.