Published in final edited form as:

Biometrics. 2014 September ; 70(3): 710–713. doi:10.1111/biom.12188.

Discussion on “Combining Biomarkers to Optimize Patient Treatment Recommendations”

Lu Tian

Department of Health Research and Policy, Stanford University, Palo Alto, California 94305, U.S.A

I would like to congratulate Kang, Janes, and Huang for their interesting article on a novel boosting algorithm for combining multiple biomarkers to optimize patient treatment recommendations. Their results show the potential of modern machine learning methods especially when one cannot correctly specify a regression model for the complex underlying relationship of interest. Since the publication of the AdaBoosting procedure by Freund and Schapire (1997), there has been great interest in seeking an explanation for its excellent performance and generalizing the method in various settings. For example, AdaBoosting can be viewed as a numerical algorithm performing forward stagewise regression with a set of weak classifiers to minimize a regularized exponential loss function. Along this line, several versions of boosting procedures have been proposed by changing the loss function to be minimized (Friedman, Hastie, and Tibshirani, 2000). Boosting is arguably one of the best off-the-shelf methods and it is not a surprise that when appropriately adapted, it can perform competitively well for differentiating patients with positive treatment benefit from those without. On the other hand, after carefully examining the proposed boosting algorithm in the article, I see some crucial departures from the classical version of boosting. Therefore, I will first discuss the differences between the proposed method and the conventional boosting algorithm and their potential implications. Secondly, I will propose simple alternatives to resolve some of the difficulties. To simplify the discussion, I will assume $P(T = 1) = P(T = 0) = 0.5$ throughout.

1. The Difference between the Proposal and Convention Boosting

Suppose for the time being that the directions of the treatment effect, $S_i = I\{Y_i > 0\}$, $i = 1, \dots, n$ are observed, and the real AdaBoosting (a generalized version of AdaBoosting returns a class probability) proceeds as follows:

1. Start with weights $w_i^{(0)} = 1/n$ for subjects $i = 1, \dots, n$, and fit a working model to calculate $\tilde{d}^{(0)}(y)$, the estimated probability $P(S = 1 | Y = y)$ based on the working model.
2. For $m = 1, \dots, M$,
 - update the weights according to $w_i^{(m)} = w_i^{*(m)} / \sum_{i=1}^n w_i^{*(m)}$, where

$$w_i^{*(m)} = w_i^{(m-1)} \exp \left(-\frac{1}{2} (2S_i - 1) \log \left[\frac{\tilde{\Delta}^{(m-1)}(y_i)}{1 - \tilde{\Delta}^{(m-1)}(y_i)} \right] \right)$$

for $i=1, \dots, n$.

- Refit the working model with updated weights $w_i^{(m)}$ to obtain $\tilde{\Delta}^{(m)}(y)$, the updated estimated probability $P(S=1|Y=y)$.

3. After the last iteration, the estimated treatment rule is

$$\hat{\phi}(Y_i) = I \left(\sum_{m=1}^M \log \left[\frac{\tilde{\Delta}^{(m)}(y_i)}{1 - \tilde{\Delta}^{(m)}(y_i)} \right] \geq 0 \right).$$

Although the assumption that S_i are all observed is artificial, it reveals an important message in comparing the proposed boosting algorithm with the ideal counterpart above. Firstly, while the updated weight in the ideal Real AdaBoosting depends on the fitted probability, the old weight and the true outcomes S_i , the weight in the proposed algorithm depends only on the fitted probability. Intuitively, the Real AdaBoosting always tries to upweight misclassified observations in the previous step to ensure that the new classification rule likely adds value to existing ones, especially on subgroup of observations for which the performance of the old classifiers is unsatisfactory. Since S_i is unobserved, the new algorithm upweights observations close to the estimated decision boundary, whose classification (optimal treatment recommendation) is relatively ambiguous. It is clear that there is a genuine difference in the reweighting scheme: following the same logic of the new algorithm, the Real AdaBoosting would upweights observations with the estimated class probability close to 0.5, which may or may not be the same observations misclassified in the previous step. One certainly can imagine that if the initial working model is a poor approximation to the truth, then some misclassified observations could mistakenly have an estimated class probability close to 0 or 1 and receive less weight. Furthermore, even when the working model yields reasonable class probability estimates, the misclassification that occurred near the decision boundary affects the value-function θ the least. Therefore, one may wonder if the new proposal sometimes fails to upweight the “right” observations. Secondly, while the Real AdaBoosting can be viewed as an algorithm to minimize a regularized loss function, the proposed algorithm can not be easily fit into the same framework. One consequence is that it is not clear if the algorithm provides a consistent estimate to the optimal treatment rule $I\{Y \geq 0\}$. To gain some insight the consistency issue, let's consider the convergence limit of $\tilde{P}^{(1)}(D=1|T=t, Y)$ in the proposed boosting procedure. The final estimator $\hat{\Delta}(Y)$ should be close to the difference between the two limits after sufficient number of iterations. When the simple logistic working model is used, the limit of $\tilde{P}^{(1)}(D=1|T=t, Y)$ is $\{1 + \exp(-\beta_t' Y)\}^{-1}$ and β_t solves the equation

$$E \left[Y \tilde{w} \left(\frac{\exp(\beta'_1 Y)}{1 + \exp(\beta'_1 Y)} - \frac{\exp(\beta'_0 Y)}{1 + \exp(\beta'_0 Y)} \right) \right] \\ \times \left[P(D = 1 | T = t, Y) - \frac{\exp(\beta'_t Y)}{1 + \exp(\beta'_t Y)} \right] = 0, \quad t = 0, 1,$$

as $n \rightarrow \infty$, which implies that

$$E[Y \tilde{w}(1 \tilde{A}(Y)) \{ \Delta(Y) - \tilde{A}(Y) \}] = 0,$$

where $\tilde{A}(Y) = \{1 + \exp(-\beta'_0 Y)\}^{-1} - \{1 + \exp(-\beta'_1 Y)\}^{-1}$ can be viewed as a reasonable approximation to $-\Delta(Y)$, especially in the region where $\beta'_0 Y \approx \beta'_1 Y$. However, it seems that in general $\tilde{A}(Y)$ is different from $-\Delta(Y)$ unless the simple logistic working model is correctly specified. The extra weight function does not solve this problem. This is in contrast to the appealing “consistency” property of the original AdaBoosting procedure, which does not require the “weak” learners to be “strong” or “correct.”

2. Alternative Boosting Procedure

From the discussion in the previous section, it seems that the new proposal may sacrifice some important components in the AdaBoosting procedure due to the simple fact that S_i , $i = 1, \dots, n$ are not observable in practice. This may cause concerns about its performance in settings beyond those investigated in the comprehensive simulation study. On the other hand, noticing the fact that

$$E\{(2D - 1)(2T - 1) | Y\} = -\Delta(Y),$$

one may treat the binary variable $\tilde{S}_i = (2D_i - 1)(2T_i - 1)$ as the surrogate to the unobserved $S_i = I\{-\Delta(Y_i) \geq 0\}$ (Signorovitch, 2007) and simply replace S_i by $\tilde{S}_i$ in the Real AdaBoosting procedure described above. To justify this approach, one notes that AdaBoosting can be viewed as a numerical algorithm to minimize the empirical version of the loss function

$$J(d) = E[\exp\{-\tilde{S}d(Y)\}]$$

in terms of $d(\cdot)$, a function of y . In this case, the minimizer of $J(d)$ is

$$d_1(y) = \frac{1}{2} \log \left\{ \frac{P(\tilde{S} = 1 | Y = y)}{P(\tilde{S} = -1 | Y = y)} \right\} = \frac{1}{2} \log \left\{ \frac{1 - \Delta(y)}{1 + \Delta(y)} \right\}.$$

It suggests that $I\{\tilde{S}_i(Y) \geq 0\}$, the treatment assignment rule based on $d_1(\cdot)$, is equivalent to our target $I\{-\Delta(Y) \geq 0\}$. Therefore, as a numerical algorithm to estimate $I\{\tilde{S}_i(Y) \geq 0\}$, the output of AdaBoosting with $\tilde{S}_i$, $i = 1, \dots, n$, as responses can be used to assign patients according to their individual treatment used to assign patients according to effects.

The second alternative is to estimate the optimal treatment rule by maximizing the mean outcome. As the authors advocated, the mean outcome is a clinically relevant and interpretable quantity to measure the value of a given treatment assignment rule. In the current setting, maximizing the mean outcome is equivalent to minimizing

$$\sum_{i=1}^n D_i I\{T_i \neq g(Y_i)\} = \sum_{i: D_i=1} I\{T_i \neq g(Y_i)\},$$

the misclassification error for predicting T_i among the subgroup of patients having outcome $D_i = 1$, with respect to the binary classification rule $g: Y \rightarrow \{0, 1\}$. This again provides a very natural platform for applying the AdaBoosting algorithm. Specifically, one may use AdaBoosting procedure to construct a classification rule to classify a patient to his/her actual randomized treatment assignment based on Y for the subgroup of patients with $D = 1$. To justify the validity of this boosting procedure, one only needs to note that the minimizer of the corresponding loss function $J(d) = E[\exp\{-(2T - 1)d(Y)\} \mid D = 1]$ is

$$\begin{aligned} d_2(y) &= \frac{1}{2} \log \left\{ \frac{P(T = 1 \mid Y = y, D = 1)}{P(T = 0 \mid Y = y, D = 1)} \right\} \\ &= \frac{1}{2} \log \left\{ \frac{P(D = 1 \mid T = 1, Y = y)}{P(D = 1 \mid T = 0, Y = y)} \right\}. \end{aligned}$$

Thus, the $d_2(\cdot)$ -based treatment assignment rule $I\{d_2(Y) \geq 0\} = I\{Y \geq 0\}$ and the corresponding boosting procedure provides a flexible algorithm to approximate the decision boundary of the optimal treatment rule.

I have selected the two most challenging scenarios (4 and 6) in the simulation study performed by Kang, Janes, and Huang to examine the relative performance of the above two proposals. For both cases, the “ada” function with default parameters in R is used to perform the boosting algorithm. The sample size of the training set is 500 and final results are based on 1000 Monte-Carlo replications (Table 1). It seems that the empirical performances of these two simple alternatives are comparable to those proposed by Kang, Janes, and Huang with classification tree as the base learner.

3. Remarks

We statisticians can learn a great deal from the rapid development of machine learning techniques, which oftentimes offer robust performance for a broad range of problems. The authors convincingly demonstrated the power of a version of the generalized boosting method for estimating the optimal strategy for assigning treatment. On the other hand, the proposed boosting method is different from its conventional counterpart in several key aspects and new explanations are needed to account for its good performance. Furthermore, there are alternative boosting procedures, which in my opinion can circumvent the difficulties caused by unobservable class labels $S_i = I\{Y_i \geq 0\}$, $i = 1, \dots, n$, in more natural ways. Further research in these directions is certainly warranted.

References

- Freund Y and Schapire RE (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55, 119–139.
- Friedman J, Hastie T, and Tibshirani R (2000). Additive logistic regression: A statistics view of boosting (with discussion and a rejoinder by the authors). *Annals of Statistics*, 28, 337–407.
- Signorovitch JE (2007). Identifying informative biological markers in high-dimensional genomic data and clinical trials. Ph.D. thesis, Harvard University.

The results of simulation study with a sample size of $n = 500$ for scenarios 4 and 6 from Table 2 of the article by Kang, Janes, and Huang. The results of the two new methods are summarized in the last two columns under the names “ $\tilde{S}$ ” and “value-function.”

Table 1

Scenario	True θ	Working model		Linear logistic			Classification tree		Separate AdaBoosting	$\tilde{S}$	New boosting	
		Algorithm	MLE	Boosting	IPWE	AIPWE	Single tree	Boosting				
4	0.0657	θ	Mean	0.0574	0.0607	0.0561	0.0567	0.0221	0.0378	0.0352	0.0396	0.0351
		MCR	Mean	0.1511	0.1206	0.1719	0.1653	0.5397	0.3251	0.5304	0.3270	0.3374
6	0.1393	θ	Mean	0.0236	0.0438	0.0498	0.0544	0.0978	0.1186	0.1010	0.1006	0.1011
		MCR	Mean	0.3865	0.3542	0.3452	0.3330	0.2697	0.1762	0.2433	0.2290	0.2227