Discussion.

Peter Bühlmann
ETH Zürich

Bin Yu
University of California, Berkeley

March 13, 2003

Jiang, Lugosi and Vayatis, and Zhang ought to be congratulated for their different works on the original AdaBoost algorithm with early stopping (Jiang), an ℓ_1-penalized version of boosting (Lugosi and Vayatis), and a convex minimization method which can be viewed as an ℓ_2-penalized version of boosting (Zhang).

1. A motivation for combining trees with boosting. The interesting and common part of all three papers is that Bayes risk consistency can be achieved by a linear or convex combination of simple classifiers. The most prominent examples, exhibiting good performance in a variety of datasets, are combinations of small or moderate-sized classification trees. Each of the trees is low-dimensional, but by linear or convex addition of trees we obtain a combined classifier whose complexity is (much) larger.

A problem with single classification trees is that they are often inflexible or cannot be constructed large enough for optimal classification performance. We show in Figure 1 the test set misclassification loss (0–1 loss) for $n = 100$ i.i.d. realizations of (X, Y) in the model

$$X = (X_1, \ldots, X_{10}) \sim \text{ Uniform}([-0.5, 0.5]^{10}),\ Y \in \{-1, 1\} \text{ with } \mathbb{P}[Y = 1] = p(X),$$

$$\text{logit}(p(x)) = \log(p(x)/(1 - p(x))) = 50 \sum_{j=1}^{10} x_j. \tag{1}$$

The trees are constructed in a greedy way (as usual) optimizing the Gini index fitting criterion. We tuned the size of a classification tree by the minimal number of observations that fall into the terminal nodes, and the largest trees are constructed under the constraint that there are at least two observations per terminal node. We see in Figure 1 that on average, the largest classification trees have about 10 or 11 terminal nodes. We also see that the test set error is smallest at our largest tree, but we cannot make the trees larger (more complex) to potentially decrease the test set error (we could enlarge them a bit by requiring at least one observation per terminal node, but this turns out to be rather unstable with low predictive power). This has to do with two things: first, it is the constrained nature of trees with splits parallel to coordinate axes; secondly, a greedily constructed classification tree is restrictive and hence involving much fewer degrees of freedom (less complexity) than when constructed in a non-greedy way. Regarding the first issue, other proposals with splits that are not parallel to axes have been proposed, cf. Kim and Loh (2001); the second issue is more difficult, but recently some progress has been made in constructing trees in a more exhaustive, less greedy way (Geman and Jedynak, 2001). The second remedy is non-trivial and with much higher computational costs.

Perhaps a conceptually simpler way, if we are concerned only with good classification performance, is given by boosting (AdaBoost which may be read as "ad a boost") which "boosts" a single classification tree to make it more flexible and more complex. Figure 1 also shows how much the test set error could be improved by using LogitBoost (with the log-likelihood loss function) with stumps, namely by about 30%. Thus, from a practical point of view, linear or convex combinations of trees overcome the limitation of "bounded" complexity of single trees. Moreover, as we understand from rigorous results in the L_2-boosting case with the squared error loss (Bühlmann and Yu, 2003), the increase of complexity is in a very gradual fashion (much slower than counting the number of the terms) which allows adaptation to problems of different complexity. Last but not least, boosting has also been found to have excellent performance in a wide range of real datasets. The papers under discussion are justifying such combination procedures which seem to act intelligently with the curse of dimensionality.

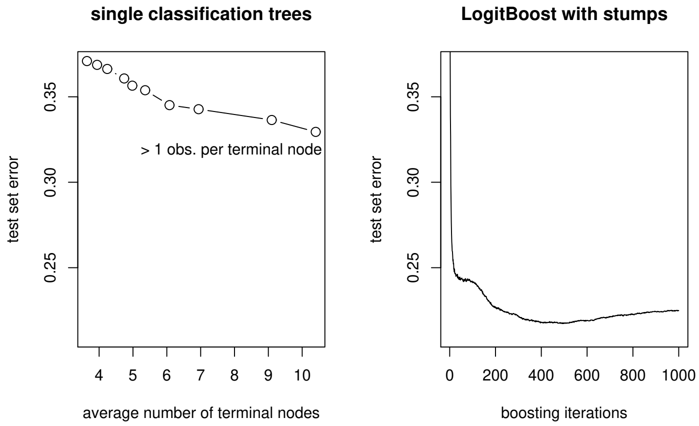

Figure 1: Test set misclassification errors in model (1). Left panel: classification trees with varying minimal numbers of observations per terminal nodes, displayed as a function of average number of terminal nodes; the lower right circle corresponds to classification trees with at least 2 observations per terminal node. Right panel: LogitBoost with stumps as a function of boosting iterations.

2. Boosting (with early stopping) versus regularized boosting. Jiang analyzes the original AdaBoost algorithm with early stopping, whereas versions of regularized boosting are considered by Lugosi and Vayatis (ℓ_1 constrained boosting) and also Zhang (ℓ_2 penalized boosting).

Computational advantage of boosting (with early stopping). The original boosting scheme specifies explicitly the numerical algorithm for optimization to be greedy, in contrast to many other classical statistical estimation schemes which are defined through an ideal optimization of an objective function. And we believe this original version of boosting (with early stopping) has an important computational advantage for coping with high-dimensional complex datasets with dimension of the predictor in the thousands. We think that it is exactly for such problems where boosting (using a learner which does variable selection) plays a significant role, since more traditional methods become very difficult to use and tune; for the latter, forward variable selection is still feasible, but assigning various smoothing parameters for selected predictors or terms is very difficult (see also the end of this section).

ℓ_1-constrained boosting is Lasso. The ℓ_1-constrained boosting algorithm proposed by Lugosi and Vayatis can be understood as seeking a combination of base learners with an ℓ_1-constraint on the combination weights, that is, one minimizes the empirical risk $A_n(f)$ under the constraint $\sum_{j=1}^{N} w_j \leq \lambda$ (notation as in Lugosi and Vayatis). This is best known in the statistics community as the Lasso method (Tibshirani, 1996), or also as basis pursuit (Chen et al., 1999) in signal processing.

Efficient computation of Lasso or basis pursuit is in general a non-trivial issue (Chen et al., 1999; Osborne et al., 2000). A notable point is that Lasso solutions are usually *not* computed using greedy algorithms which are in danger to be overly greedy and can get stuck in suboptimal solutions. Lugosi and Vayatis use in their examples the MarginBoost.L_1 algorithm from Mason et al. (2000) which is a gradient descent, greedy forward method, very similar to boosting, and it normalizes the ℓ_1-norm of the weights along the way. Interestingly, this MarginBoost.L_1 algorithm can be used for many base learners and is not restricted to specialized problems like linear regression or expansions from an over-complete wavelet basis. Lugosi and Vayatis do not discuss to what extent the MarginBoost.L_1 algorithm yields approximate solutions to Lasso-type problems or whether the MarginBoost.L_1 algorithm corresponds exactly to the ℓ_1-constrained boosting for which theoretical results are proven by Lugosi and Vayatis. In particular, at first sight, it seems that the greedy nature of the MarginBoost.L_1 algorithm used by Lugosi and Vayatis for their regularized boosting is in conflict with the non-greedy Lasso algorithms in Chen et al. (1999) and Osborne et al. (2000).

Using the LARS (least angle regression) algorithm for finite linear regression models, Efron et al. (2002) recently makes a connection between Lasso and boosting with infinitesimal shrinkage factor (or ϵ-boosting), or equivalently, linking non-greedy linear programming algorithms for Lasso with greedy, gradient descent methods for boosting with infinitesimal steps: ϵ-boosting (or "stagewise" as called by Efron et al. (2002)) adds normalized base learners to the current fit by an infinitesimal amount ϵ (but fixed among the boosting iterations). Under some positive cone conditions, Efron et al. (2002) show that Lasso and ϵ-boosting are equivalent. In practice, the ϵ or infinitesimal amount of shrinkage has to be chosen as a small constant as has been advocated by Friedman (2001). However, this is worth noting that ϵ-boosting is not the same as MarginBoost.L_1, but we believe they are closely related.

Although this connection in the finite predictor (or finite base learner) case is intriguing, it is unclear how to generalize the LARS algorithm to the infinite base learner case. (One such example is trees, although for a given n, there are only finitely many possible trees with any fixed number of terminal nodes and split points in the middle between observations. However, this finite number is already equal to dim.(predictor) $\cdot$ $(n-1)$ for stumps and even much bigger for larger trees; asymptotically, it is infinite). This infinite base learner scenario is the most relevant to the success of boosting with empirical datasets because the base learner fitted at each step is taken from a pool of infinitely many base learners. For this case, we believe boosting (with small steps) provides the most flexible solution and, in some sense, generalizes LARS from the finite learner case. It is interesting to note that the convergence analysis of Zhang and Yu (2003) suggests that small step sizes are necessary for the convergence of the boosting algorithm as the iteration goes to infinity. For good statistical performance however, we almost always stop before convergence and we believe that boosting is in general different from ℓ_1-constrained boosting or MarginBoost.L_1. This difference can also been seen in the experiments provided by Lugosi and Vayatis on AdaBoost and MarginBoost.L_1.

ℓ_2-regularization is Ridge. Zhang proposes convex combination of base learners: his way of estimation and regularization is via the more established ℓ_2-penalty. Because of the ℓ_2-penalty, his algorithm can be viewed as a Ridge method. In general, the solution is not expected to be sparse, an interesting exception is the case with the SVM loss function. On the other hand, boosting with a base learner that does variable selection can be shown to have the interesting feature to do variable selection *and* assigning varying amount of degrees of freedom to different selected variables (e.g. in a linear model) or terms in an expansion (e.g. in fitting an additive model). The same holds for Lasso in linear models due to the "equivalence" of ϵ-boosting and Lasso (Efron et al., 2002); we believe that it is also true for more general models.

Adaptivity of boosting (with early stopping). We have shown in Bühlmann and Yu (2003) that boosting with the squared error loss function, which we called L_2Boost, adapts to higher order smoothness for curve estimation in nonparametric regression. For example, when using cubic smoothing splines as base learners with a fixed conventional smoothing parameter λ_0, L_2Boost with a suitable number of boosting iterations achieves the minimax optimal MSE rates over Sobolev classes. Even though we are using a cubic smoothing spline as a base learner, L_2Boost achieves a faster MSE rate than $O(n^{-4/5})$ (the optimal rate for the Sobolev class of degree 2) if the underlying true function is in the Sobolev space of degree larger than 2 (essentially more than twice differentiable). With non-boosted smoothing splines, we would only get the minimax optimal MSE rates when knowing the smoothness of the underlying function. Thus, L_2Boost has the interesting theoretical property to adapt automatically to higher order smoothness, and interestingly, this is achieved by a greedy forward algorithm!

Because of the connection between the ℓ_2-penalized convex combination algorithm of Zhang, when used with the squared error loss, and the classical smoothing splines, we doubt that this adaptivity holds for Zhang's ℓ_2-regularized boosting algorithm. It remains to be seen whether the ℓ_1-constrained boosting has this adaptivity, but we conjecture that it does due to its connection to ϵ-boosting.

3. Final remarks. Jiang solved the problem of consistency for original boosting with early stopping which we think is a very effective statistical methodology and at the same time computationally feasible for high-dimensional data-sets. Breiman (2000) pointed already at the issue of consistency for AdaBoost but Jiang was the first to prove consistency of AdaBoost. Since we believe that boosting (with early stopping) is very useful in general, we have followed up on Jiang's work. In Bühlmann (2002), consistency of L_2Boost (with early stopping) is proved for regression or probability estimation in classification (which is more general than Bayes risk consistency). More recently, Zhang and Yu (2003) showed consistency of boosting with early stopping under general loss functions.

Lugosi and Vayatis present elegant consistency theorems which work under "minimal" assumptions. Since they analyze ℓ_1-constrained boosting, we may think that their result also hints at consistency for Lasso-type methods in classification.

Zhang's work has an interesting part on implementing loss functions for classification, providing consistency for Ridge-type methods in classification.

In summary, the three papers under discussion present some important recent understandings of boosting, as a result of the joint efforts of the statistics and the machine learning communities. We believe that this interaction of statistics and machine learning is bearing or will bear fruits on understanding many other procedures such as support vector machines and independent component analysis.

References

[1] Breiman, L. (2000). Some infinity theory for predictor ensembles. Tech. Report 579, Dept. of Statist., Univ. of Calif., Berkeley.

[2] Bühlmann, P. (2002). Consistency for L_2Boosting and matching pursuit with trees and tree-type basis functions. Preprint, available from http://stat.ethz.ch/~buhlmann/bibliog.html

[3] Bühlmann, P. and Yu, B. (2003). Boosting with the L_2 loss: regression and classification. To appear in J. Amer. Statist. Assoc.

[4] Chen, S.S., Donoho, D.L. and Saunders, M.A. (1999). Atomic decomposition by basis pursuit. SIAM J. Sci. Comp. **20**(1), 33–61.

[5] Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2002). Least angle regression. Preprint.

[6] Friedman, J.H. (2001). Greedy function approximation: a gradient boosting machine. Ann. Statist. **29**, 1189–1232.

[7] Geman, D. and Jedynak, B. (2001). Model-based classification trees. IEEE Transactions on Information Theory **47**, 1075–1082.

[8] Kim, H. and Loh, W.-Y. (2001). Classification trees with unbiased multiway splits. J. Amer. Statist. Assoc. **96**, 589–604.

[9] Mason, L., Baxter, J. Bartlett, P. and Frean, M. (2000). Functional gradient techniques for combining hypotheses. In *Advances in Large Margin Classifiers* (eds. A.J. Smola, P.J. Bartlett, B. Schölkopf and D. Schuurmans). MIT Press, Cambridge, MA.

[10] Osborne, M.R., Presnell, B. and Turlach, B.A. (2000). On the lasso and its dual. J. Comp. Graph. Statist. **9**, 319–337.

[11] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. J. Roy. Statist. Soc., Series B, **58**, 267–288.

[12] Zhang, T. and Yu, B. (2003). Boosting with early stopping: convergence and consistency. Tech. Report 635, Department of Statistics, UC Berkeley. available from http://www.stat.berkeley.edu/~binyu/publications.html