Journal of Machine Learning Research 5 (2004) 941–973 Submitted 5/03; Revised 10/03; Published 8/04

Boosting as a Regularized Path to a Maximum Margin Classifier

Saharon Rosset SROSSET@US.IBM.COM
Data Analytics Research Group
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598, USA

Ji Zhu JIZHU@UMICH.EDU
Department of Statistics
University of Michigan
Ann Arbor, MI 48109, USA

Trevor Hastie HASTIE@STAT.STANFORD.EDU
Department of Statistics
Stanford University
Stanford, CA 94305,USA

Editor: Robert Schapire

Abstract

In this paper we study boosting methods from a new perspective. We build on recent work by Efron et al. to show that boosting approximately (and in some cases exactly) minimizes its loss criterion with an l_1 constraint on the coefficient vector. This helps understand the success of boosting with early stopping as regularized fitting of the loss criterion. For the two most commonly used criteria (exponential and binomial log-likelihood), we further show that as the constraint is relaxed—or equivalently as the boosting iterations proceed—the solution converges (in the separable case) to an "l_1-optimal" separating hyper-plane. We prove that this l_1-optimal separating hyper-plane has the property of maximizing the minimal l_1-margin of the training data, as defined in the boosting literature. An interesting fundamental similarity between boosting and kernel support vector machines emerges, as both can be described as methods for regularized optimization in high-dimensional predictor space, using a computational trick to make the calculation practical, and converging to margin-maximizing solutions. While this statement describes SVMs exactly, it applies to boosting only approximately.

Keywords: boosting, regularized optimization, support vector machines, margin maximization

1. Introduction and Outline

Boosting is a method for iteratively building an additive model

$$F_T(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_{j_t}(\mathbf{x}), \tag{1}$$

where $h_{j_t} \in \mathcal{H}$—a large (but we will assume finite) dictionary of candidate predictors or "weak learners"; and h_{j_t} is the basis function selected as the "best candidate" to modify the function at stage t. The model F_T can equivalently be represented by assigning a coefficient to each dictionary function $h \in \mathcal{H}$ rather than to the selected h_{j_t}'s only:

$$F_T(\mathbf{x}) = \sum_{j=1}^{J} h_j(\mathbf{x}) \cdot \beta_j^{(T)}, \tag{2}$$

where $J = |\mathcal{H}|$ and $\beta_j^{(T)} = \sum_{j_t=j} \alpha_t$. The "$\beta$" representation allows us to interpret the coefficient vector $\beta^{(T)}$ as a vector in $\mathcal{R}^J$ or, equivalently, as the hyper-plane which has $\beta^{(T)}$ as its normal. This interpretation will play a key role in our exposition.

Some examples of common dictionaries are:

- The training variables themselves, in which case $h_j(\mathbf{x}) = x_j$. This leads to our "additive" model F_T being just a linear model in the original data. The number of dictionary functions will be $J = d$, the dimension of $\mathbf{x}$.

- Polynomial dictionary of degree p, in which case the number of dictionary functions will be $J = \begin{pmatrix} p+d \\ d \end{pmatrix}$.

- Decision trees with up to k terminal nodes, if we limit the split points to data points (or mid-way between data points as CART does). The number of possible trees is bounded from above (trivially) by $J \leq (np)^k \cdot 2^{k^2}$. Note that regression trees do not fit into our framework, since they will give $J = \infty$.

The boosting idea was first introduced by Freund and Schapire (1995), with their AdaBoost algorithm. AdaBoost and other boosting algorithms have attracted a lot of attention due to their great success in data modeling tasks, and the "mechanism" which makes them work has been presented and analyzed from several perspectives. Friedman et al. (2000) develop a statistical perspective, which ultimately leads to viewing AdaBoost as a gradient-based incremental search for a good additive model (more specifically, it is a "coordinate descent" algorithm), using the exponential loss function $C(y,F) = \exp(-yF)$, where $y \in \{-1,1\}$. The gradient boosting (Friedman, 2001) and anyboost (Mason et al., 1999) generic algorithms have used this approach to generalize the boosting idea to wider families of problems and loss functions. In particular, Friedman et al. (2000) have pointed out that the binomial log-likelihood loss $C(y,F) = \log(1+\exp(-yF))$ is a more natural loss for classification, and is more robust to outliers and misspecified data.

A different analysis of boosting, originating in the machine learning community, concentrates on the effect of boosting on the margins $y_i F(\mathbf{x}_i)$. For example, Schapire et al. (1998) use margin-based arguments to prove convergence of boosting to perfect classification performance on the training data under general conditions, and to derive bounds on the generalization error (on future, unseen data).

In this paper we combine the two approaches, to conclude that gradient-based boosting can be described, in the separable case, as an approximate margin maximizing process. The view we develop of boosting as an approximate path of optimal solutions to regularized problems also justifies early stopping in boosting as specifying a value for "regularization parameter".

We consider the problem of minimizing non-negative convex loss functions (in particular the exponential and binomial log-likelihood loss functions) over the training data, with an l_1 bound on the model coefficients:

$$\hat{\beta}(c) = \arg \min_{\|\beta\|_1 \leq c} \sum_i C(y_i, h(\mathbf{x}_i)'\beta). \tag{3}$$

Where $h(\mathbf{x}_i) = [h_1(\mathbf{x}_i), h_2(\mathbf{x}_i), \ldots, h_J(\mathbf{x}_i)]'$ and $J = |\mathcal{H}|$.[1]

Hastie et al. (2001, Chapter 10) have observed that "slow" gradient-based boosting (i.e., we set $\alpha_t = \varepsilon$, $\forall t$ in (1), with ε small) tends to follow the penalized path $\hat{\beta}(c)$ as a function of c, under some mild conditions on this path. In other words, using the notation of (2), (3), this implies that $\|\beta^{(c/\varepsilon)} - \hat{\beta}(c)\|$ vanishes with ε, for all (or a wide range of) values of c. Figure 1 illustrates this equivalence between ε-boosting and the optimal solution of (3) on a real-life data set, using squared error loss as the loss function. In this paper we demonstrate this equivalence further and formally

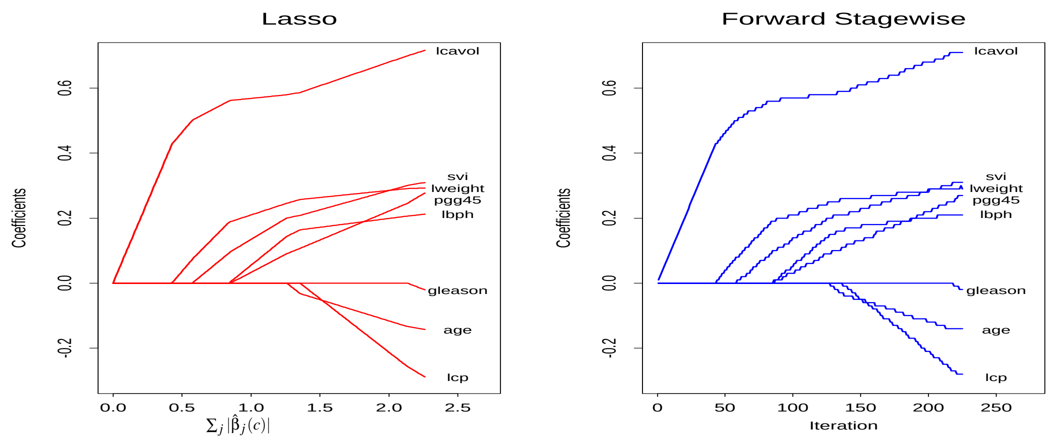

Figure 1: Exact coefficient paths(left) for l_1-constrained squared error regression and "boosting" coefficient paths (right) on the data from a prostate cancer study

state it as a conjecture. Some progress towards proving this conjecture has been made by Efron et al. (2004), who prove a weaker "local" result for the case where C is squared error loss, under some mild conditions on the optimal path. We generalize their result to general convex loss functions.

Combining the empirical and theoretical evidence, we conclude that boosting can be viewed as an approximate incremental method for following the l_1-regularized path.

We then prove that in the separable case, for both the exponential and logistic log-likelihood loss functions, $\hat{\beta}(c)/c$ converges as $c \to \infty$ to an "optimal" separating hyper-plane $\hat{\beta}$ described by

$$\hat{\beta} = \arg \max_{\|\beta\|_1 = 1} \min_i y_i \beta' h(\mathbf{x}_i). \tag{4}$$

In other words, $\hat{\beta}$ *maximizes the minimal margin* among all vectors with l_1-norm equal to 1.[2] This result generalizes easily to other l_p-norm constraints. For example, if $p = 2$, then $\hat{\beta}$ describes the optimal separating hyper-plane in the Euclidean sense, i.e., the same one that a non-regularized support vector machine would find.

Combining our two main results, we get the following characterization of boosting:

1. Our notation assumes that the minimum in (3) is unique, which requires some mild assumptions. To avoid notational complications we use this slightly abusive notation throughout this paper. In Appendix B we give explicit conditions for uniqueness of this minimum.
2. The margin maximizing hyper-plane in (4) may not be unique, and we show that in that case the limit $\hat{\beta}$ is still defined and it also maximizes the second minimal margin. See Appendix B.2 for details.

> ε-Boosting can be described as a gradient-descent search, approximately following the path of l_1-constrained optimal solutions to its loss criterion, and converging, in the separable case, to a "margin maximizer" in the l_1 sense.

Note that boosting with a large dictionary $\mathcal{H}$ (in particular if $n < J = |\mathcal{H}|$) guarantees that the data will be separable (except for pathologies), hence separability is a very mild assumption here.

As in the case of support vector machines in high dimensional feature spaces, the non-regularized "optimal" separating hyper-plane is usually of theoretical interest only, since it typically represents an over-fitted model. Thus, we would want to choose a good regularized model. Our results indicate that Boosting gives a natural method for doing that, by "stopping early" in the boosting process. Furthermore, they point out the fundamental similarity between Boosting and SVMs: both approaches allow us to fit regularized models in high-dimensional predictor space, using a computational trick. They differ in the regularization approach they take—exact l_2 regularization for SVMs, approximate l_1 regularization for Boosting—-and in the computational trick that facilitates fitting—the "kernel" trick for SVMs, coordinate descent for Boosting.

1.1 Related Work

Schapire et al. (1998) have identified the normalized margins as distance from an l_1-normed separating hyper-plane. Their results relate the boosting iterations' success to the minimal margin of the combined model. Rätsch et al. (2001b) take this further using an asymptotic analysis of AdaBoost. They prove that the "normalized" minimal margin, $\min_i y_i \sum_t \alpha_t h_t(\mathbf{x}_i) / \sum_t |\alpha_t|$, is asymptotically equal for both classes. In other words, they prove that the asymptotic separating hyper-plane is equally far away from the closest points on either side. This is a property of the margin maximizing separating hyper-plane as we define it. Both papers also illustrate the margin maximizing effects of AdaBoost through experimentation. However, they both stop short of proving the convergence to optimal (margin maximizing) solutions.

Motivated by our result, Rätsch and Warmuth (2002) have recently asserted the margin-maximizing properties of ε-AdaBoost, using a different approach than the one used in this paper. Their results relate only to the asymptotic convergence of infinitesimal AdaBoost, compared to our analysis of the "regularized path" traced along the way and of a variety of boosting loss functions, which also leads to a convergence result on binomial log-likelihood loss.

The convergence of boosting to an "optimal" solution from a loss function perspective has been analyzed in several papers. Rätsch et al. (2001a) and Collins et al. (2000) give results and bounds on the convergence of training-set loss, $\sum_i C(y_i, \sum_t \alpha_t h_t(\mathbf{x}_i))$, to its minimum. However, in the separable case convergence of the loss to 0 is inherently different from convergence of the linear separator to the optimal separator. Any solution which separates the two classes perfectly can drive the exponential (or log-likelihood) loss to 0, simply by scaling coefficients up linearly.

Two recent papers have made the connection between boosting and l_1 regularization in a slightly different context than this paper. Zhang (2003) suggests a "shrinkage" version of boosting which converges to l_1 regularized solutions, while Zhang and Yu (2003) illustrate the quantitative relationship between early stopping in boosting and l_1 constraints.

2. Boosting as Gradient Descent

Generic gradient-based boosting algorithms (Friedman, 2001; Mason et al., 1999) attempt to find a good linear combination of the members of some dictionary of basis functions to optimize a given loss function over a sample. This is done by searching, at each iteration, for the basis function which gives the "steepest descent" in the loss, and changing its coefficient accordingly. In other words, this is a "coordinate descent" algorithm in $\mathbb{R}^J$, where we assign one dimension (or coordinate) for the coefficient of each dictionary function.

Assume we have data $\{\mathbf{x}_i, y_i\}_{i=1}^n$ with $\mathbf{x}_i \in \mathbb{R}^d$, a loss (or cost) function $C(y, F)$, and a set of dictionary functions $\{h_j(\mathbf{x})\} : \mathbb{R}^d \to \mathbb{R}$. Then all of these algorithms follow the same essential steps:

Algorithm 1 *Generic gradient-based boosting algorithm*

1. *Set* $\beta^{(0)} = 0$.
2. *For* $t = 1 : T$,
   (a) *Let* $F_i = \beta^{(t-1)'} h(\mathbf{x}_i)$, $i = 1, \ldots, n$ *(the current fit).*
   (b) *Set* $w_i = \frac{\partial C(y_i, F_i)}{\partial F_i}$, $i = 1, \ldots, n$.
   (c) *Identify* $j_t = \arg\max_j |\sum_i w_i h_j(\mathbf{x}_i)|$.
   (d) *Set* $\beta_{j_t}^{(t)} = \beta_{j_t}^{(t-1)} - \alpha_t \, sign(\sum_i w_i h_{j_t}(\mathbf{x}_i))$ *and* $\beta_k^{(t)} = \beta_k^{(t-1)}, k \neq j_t$.

Here $\beta^{(t)}$ is the "current" coefficient vector and $\alpha_t > 0$ is the current step size. Notice that $\sum_i w_i h_{j_t}(\mathbf{x}_i) = \frac{\partial \sum_i C(y_i, F_i)}{\partial \beta_{j_t}}$.

As we mentioned, Algorithm 1 can be interpreted simply as a coordinate descent algorithm in "weak learner" space. Implementation details include the dictionary $\mathcal{H}$ of "weak learners", the loss function $C(y, F)$, the method of searching for the optimal j_t and the way in which α_t is determined.[3] For example, the original AdaBoost algorithm uses this scheme with the exponential loss $C(y, F) = \exp(-yF)$, and an implicit line search to find the best α_t once a "direction" j_t has been chosen (see Hastie et al., 2001; Mason et al., 1999). The dictionary used by AdaBoost in this formulation would be a set of candidate classifiers, i.e., $h_j(\mathbf{x}_i) \in \{-1, +1\}$—usually decision trees are used in practice.

2.1 Practical Implementation of Boosting

The dictionaries used for boosting are typically very large—practically infinite—and therefore the generic boosting algorithm we have presented cannot be implemented verbatim. In particular, it is not practical to exhaustively search for the maximizer in step 2(c). Instead, an approximate, usually greedy search is conducted to find a "good" candidate weak learner h_{j_t} which makes the first order decline in the loss large (even if not maximal among all possible models).

In the common case that the dictionary of weak learners is comprised of decision trees with up to k nodes, the way AdaBoost and other boosting algorithms solve stage 2(c) is by building a

3. The sign of α_t will always be $-sign(\sum_i w_i h_{j_t}(\mathbf{x}_i))$, since we want the loss to be reduced. In most cases, the dictionary $\mathcal{H}$ is negation closed, and so it can be assumed WLOG that the coefficients are always positive and increasing

decision tree to a re-weighted version of the data, with the weights $|w_i|$. Thus they first replace step 2(c) with minimization of

$$\sum_i |w_i| 1\{y_i \neq h_{j_t}(\mathbf{x}_i)\},$$

which is easily shown to be equivalent to the original step 2(c). They then use a greedy decision-tree building algorithm such as CART or C5 to build a k-node decision tree which minimizes this quantity, i.e., achieves low "weighted misclassification error" on the weighted data. Since the tree is built greedily—one split at a time—it will not be the global minimizer of weighted misclassification error among all k-node decision trees. However, it will be a good fit for the re-weighted data, and can be considered an approximation to the optimal tree.

This use of approximate optimization techniques is critical, since much of the strength of the boosting approach comes from its ability to build additive models in *very* high-dimensional predictor spaces. In such spaces, standard exact optimization techniques are impractical: any approach which requires calculation and inversion of Hessian matrices is completely out of the question, and even approaches which require only first derivatives, such as coordinate descent, can only be implemented approximately.

2.2 Gradient-Based Boosting as a Generic Modeling Tool

As Friedman (2001); Mason et al. (1999) mention, this view of boosting as gradient descent allows us to devise boosting algorithms for any function estimation problem—all we need is an appropriate loss and an appropriate dictionary of "weak learners". For example, Friedman et al. (2000) suggested using the binomial log-likelihood loss instead of the exponential loss of AdaBoost for binary classification, resulting in the LogitBoost algorithm. However, there is no need to limit boosting algorithms to classification—Friedman (2001) applied this methodology to regression estimation, using squared error loss and regression trees, and Rosset and Segal (2003) applied it to density estimation, using the log-likelihood criterion and Bayesian networks as weak learners. Their experiments and those of others illustrate that the practical usefulness of this approach—coordinate descent in high dimensional predictor space—carries beyond classification, and even beyond supervised learning.

The view we present in this paper, of coordinate-descent boosting as approximate l_1-regularized fitting, offers some insight into why this approach would be good in general: it allows us to fit regularized models directly in high dimensional predictor space. In this it bears a conceptual similarity to support vector machines, which exactly fit an l_2 regularized model in high dimensional (RKH) predictor space.

2.3 Loss Functions

The two most commonly used loss functions for boosting classification models are the exponential and the (minus) binomial log-likelihood:

$$\begin{aligned} Exponential: &\quad C_e(y,F) = \exp(-yF); \\ Loglikelihood: &\quad C_l(y,F) = \log(1+\exp(-yF)). \end{aligned}$$

These two loss functions bear some important similarities to each other. As Friedman et al. (2000) show, the population minimizer of expected loss at point $\mathbf{x}$ is similar for both loss functions and is

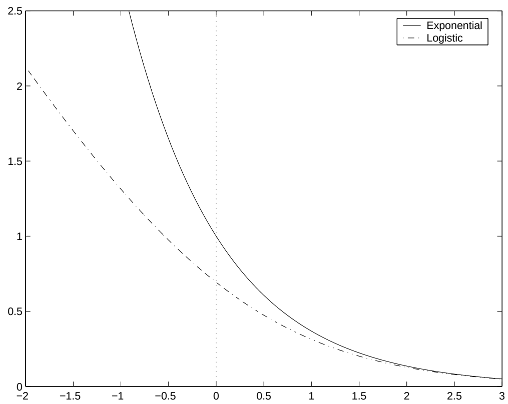

Figure 2: The two classification loss functions

given by

$$\hat{F}(\mathbf{x}) = c \cdot log\left[\frac{P(y=1|\mathbf{x})}{P(y=-1|\mathbf{x})}\right],$$

where $c_e = 1/2$ for exponential loss and $c_l = 1$ for binomial loss.

More importantly for our purpose, we have the following simple proposition, which illustrates the strong similarity between the two loss functions for positive margins (i.e., correct classifications):

Proposition 1

$$yF \geq 0 \Rightarrow 0.5C_e(y,F) \leq C_l(y,F) \leq C_e(y,F). \tag{5}$$

In other words, the two losses become similar if the margins are positive, and both behave like exponentials.

Proof Consider the functions $f_1(z) = z$ and $f_2(z) = log(1+z)$ for $z \in [0,1]$. Then $f_1(0) = f_2(0) = 0$, and

$$\frac{\partial f_1(z)}{\partial z} \equiv 1$$

$$\frac{1}{2} \leq \frac{\partial f_2(z)}{\partial z} = \frac{1}{1+z} \leq 1.$$

Thus we can conclude $0.5f_1(z) \leq f_2(z) \leq f_1(z)$. Now set $z = exp(-yf)$ and we get the desired result. ■

For negative margins the behaviors of C_e and C_l are very different, as Friedman et al. (2000) have noted. In particular, C_l is more robust against outliers and misspecified data.

2.4 Line-Search Boosting vs. ε-Boosting

As mentioned above, AdaBoost determines α_t using a line search. In our notation for Algorithm 1 this would be

$$\alpha_t = \arg\min_{\alpha} \sum_i C(y_i, F_i + \alpha h_{j_t}(\mathbf{x_i})).$$

The alternative approach, suggested by Friedman (2001); Hastie et al. (2001), is to "shrink" all α_t to a single small value ε. This may slow down learning considerably (depending on how small ε is), but is attractive theoretically: the first-order theory underlying gradient boosting implies that the weak learner chosen is the best increment only "locally". It can also be argued that this approach is "stronger" than line search, as we can keep selecting the same h_{j_t} repeatedly if it remains optimal and so ε-boosting dominates line-search boosting in terms of training error. In practice, this approach of "slowing the learning rate" usually performs better than line-search in terms of prediction error as well (see Friedman, 2001). For our purposes, we will mostly assume ε is infinitesimally small, so the theoretical boosting algorithm which results is the "limit" of a series of boosting algorithms with shrinking ε.

In regression terminology, the line-search version is equivalent to forward stage-wise modeling, infamous in the statistics literature for being too greedy and highly unstable (see Friedman, 2001). This is intuitively obvious, since by increasing the coefficient until it saturates we are destroying "signal" which may help us select other good predictors.

3. l_p Margins, Support Vector Machines and Boosting

We now introduce the concept of margins as a geometric interpretation of a binary classification model. In the context of boosting, this view offers a different understanding of AdaBoost from the gradient descent view presented above. In the following sections we connect the two views.

3.1 The Euclidean Margin and the Support Vector Machine

Consider a classification model in high dimensional predictor space: $F(\mathbf{x}) = \sum_j h_j(\mathbf{x})\beta_j$. We say that the model *separates* the training data $\{\mathbf{x}_i, y_i\}_{i=1}^n$ if $sign(F(\mathbf{x}_i)) = y_i, \ \forall i$. From a geometrical perspective this means that the hyper-plane defined by $F(\mathbf{x}) = 0$ is a separating hyper-plane for this data, and we define its (Euclidean) margin as

$$m_2(\beta) = \min_i \frac{y_i F(\mathbf{x}_i)}{\|\beta\|_2}. \tag{6}$$

The margin-maximizing separating hyper-plane for this data would be defined by β which maximizes $m_2(\beta)$. Figure 3 shows a simple example of separable data in two dimensions, with its margin-maximizing separating hyper-plane. The Euclidean margin-maximizing separating hyper-plane is the (non regularized) support vector machine solution. Its margin maximizing properties play a central role in deriving generalization error bounds for these models, and form the basis for a rich literature.

3.2 The l_1 Margin and Its Relation to Boosting

Instead of considering the Euclidean margin as in (6) we can define an "l_p margin" concept as

$$m_p(\beta) = \min_i \frac{y_i F(\mathbf{x}_i)}{\|\beta\|_p}. \tag{7}$$

Of particular interest to us is the case $p = 1$. Figure 4 shows the l_1 margin maximizing separating hyper-plane for the same simple example as Figure 3. Note the fundamental difference between

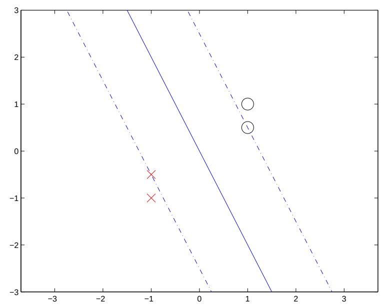

Figure 3: A simple data example, with two observations from class "O" and two observations from class "X". The full line is the Euclidean margin-maximizing separating hyper-plane.

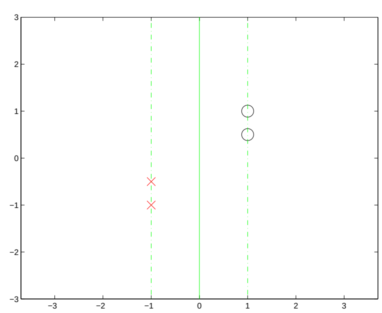

Figure 4: l_1 margin maximizing separating hyper-plane for the same data set as Figure 3. The difference between the diagonal Euclidean optimal separator and the vertical l_1 optimal separator illustrates the "sparsity" effect of optimal l_1 separation

the two solutions: the l_2-optimal separator is diagonal, while the l_1-optimal one is vertical. To understand why this is so we can relate the two margin definitions to each other as

$$\frac{yF(\mathbf{x})}{\|\beta\|_1} = \frac{yF(\mathbf{x})}{\|\beta\|_2} \cdot \frac{\|\beta\|_2}{\|\beta\|_1}. \tag{8}$$

From this representation we can observe that the l_1 margin will tend to be big if the ratio $\frac{\|\beta\|_2}{\|\beta\|_1}$ is big. This ratio will generally be big if β is sparse. To see this, consider fixing the l_1 norm of the vector and then comparing the l_2 norm of two candidates: one with many small components and the other—a sparse one—with a few large components and many zero components. It is easy to see that the second vector will have bigger l_2 norm, and hence (if the l_2 margin for both vectors is equal) a bigger l_1 margin.

A different perspective on the difference between the optimal solutions is given by a theorem due to Mangasarian (1999), which states that the l_p margin maximizing separating hyper plane maximizes the l_q distance from the closest points to the separating hyper-plane, with $\frac{1}{p} + \frac{1}{q} = 1$. Thus the Euclidean optimal separator ($p = 2$) also maximizes Euclidean distance between the points and the hyper-plane, while the l_1 optimal separator maximizes l_∞ distance. This interesting result gives another intuition why l_1 optimal separating hyper-planes tend to be coordinate-oriented (i.e., have sparse representations): since l_∞ projection considers only the largest coordinate distance, some coordinate distances may be 0 at no cost of decreased l_∞ distance.

Schapire et al. (1998) have pointed out the relation between AdaBoost and the l_1 margin. They prove that, in the case of separable data, the boosting iterations increase the "boosting" margin of the model, defined as

$$\min_i \frac{y_i F(\mathbf{x_i})}{\|\alpha\|_1}. \tag{9}$$

In other words, this is the l_1 margin of the model, except that it uses the α incremental representation rather than the β "geometric" representation for the model. The two representations give the same l_1 norm if there is sign consistency, or "monotonicity" in the coefficient paths traced by the model, i.e., if at every iteration t of the boosting algorithm

$$\beta_{j_t} \neq 0 \Rightarrow sign(\alpha_t) = sign(\beta_{j_t}). \tag{10}$$

As we will see later, this monotonicity condition will play an important role in the equivalence between boosting and l_1 regularization.

The l_1-margin maximization view of AdaBoost presented by Schapire et al. (1998)—and a whole plethora of papers that followed—is important for the analysis of boosting algorithms for two distinct reasons:

- It gives an intuitive, geometric interpretation of the model that AdaBoost is looking for—a model which separates the data well in this l_1-margin sense. Note that the view of boosting as gradient descent in a loss criterion doesn't really give the same kind of intuition: if the data is separable, then any model which separates the training data will drive the exponential or binomial loss to 0 when scaled up:

$$m_1(\beta) > 0 \implies \sum_i C(y_i, d\beta'\mathbf{x}_i) \to 0 \;\; as \;\; d \to \infty.$$

- The l_1-margin behavior of a classification model on its training data facilitates generation of generalization (or prediction) error bounds, similar to those that exist for support vector machines (Schapire et al., 1998). The important quantity in this context is not the margin but the "normalized" margin, which considers the "conjugate norm" of the predictor vectors:

$$\frac{y_i\beta' h(x_i)}{\|\beta\|_1 \|h(x_i)\|_\infty}.$$

When the dictionary we are using is comprised of classifiers then $\|h(x_i)\|_\infty \equiv 1$ always and thus the l_1 margin is *exactly* the relevant quantity. The error bounds described by Schapire et al. (1998) allow using the whole l_1 margin distribution, not just the minimal margin. However, boosting's tendency to separate well in the l_1 sense is a central motivation behind their results.

From a statistical perspective, however, we should be suspicious of margin-maximization as a method for building good prediction models in high dimensional predictor space. Margin maximization in high dimensional space is likely to lead to over-fitting and bad prediction performance. This has been observed in practice by many authors, in particular Breiman (1999). Our results in the next two sections suggest an explanation based on model complexity: margin maximization is the limit of parametric regularized optimization models, as the regularization vanishes, and the regularized models along the path may well be superior to the margin maximizing "limiting" model, in terms of prediction performance. In Section 7 we return to discuss these issues in more detail.

4. Boosting as Approximate Incremental l_1 Constrained Fitting

In this section we introduce an interpretation of the generic coordinate-descent boosting algorithm as tracking a path of approximate solutions to l_1-constrained (or equivalently, regularized) versions of its loss criterion. This view serves our understanding of what boosting does, in particular the connection between early stopping in boosting and regularization. We will also use this view to get a result about the asymptotic margin-maximization of regularized classification models, and by analogy of classification boosting. We build on ideas first presented by Hastie et al. (2001, Chapter 10) and Efron et al. (2004).

Given a convex non-negative loss criterion $C(\cdot,\cdot)$, consider the 1-dimensional path of optimal solutions to l_1 constrained optimization problems over the training data:

$$\hat{\beta}(c) = \arg\min_{\|\beta\|_1 \le c} \sum_i C(y_i, h(\mathbf{x}_i)'\beta). \tag{11}$$

As c varies, we get that $\hat{\beta}(c)$ traces a 1-dimensional "optimal curve" through $\mathbb{R}^J$. If an optimal solution for the non-constrained problem exists and has finite l_1 norm c_0, then obviously $\hat{\beta}(c) = \hat{\beta}(c_0) = \hat{\beta}$, $\forall c > c_0$. in the case of separable 2-class data, using either C_e or C_l, there is no finite-norm optimal solution. Rather, the constrained solution will always have $\|\hat{\beta}(c)\|_1 = c$.

A different way of building a solution which has l_1 norm c, is to run our ε-boosting algorithm for c/ε iterations. This will give an $\alpha^{(c/\varepsilon)}$ vector which has l_1 norm exactly c. For the norm of the geometric representation $\beta^{(c/\varepsilon)}$ to also be equal to c, we need the monotonicity condition (10) to hold as well. This condition will play a key role in our exposition.

We are going to argue that the two solution paths $\hat{\beta}(c)$ and $\beta^{(c/\varepsilon)}$ are very similar for ε "small". Let us start by observing this similarity in practice. Figure 1 in the introduction shows an example of

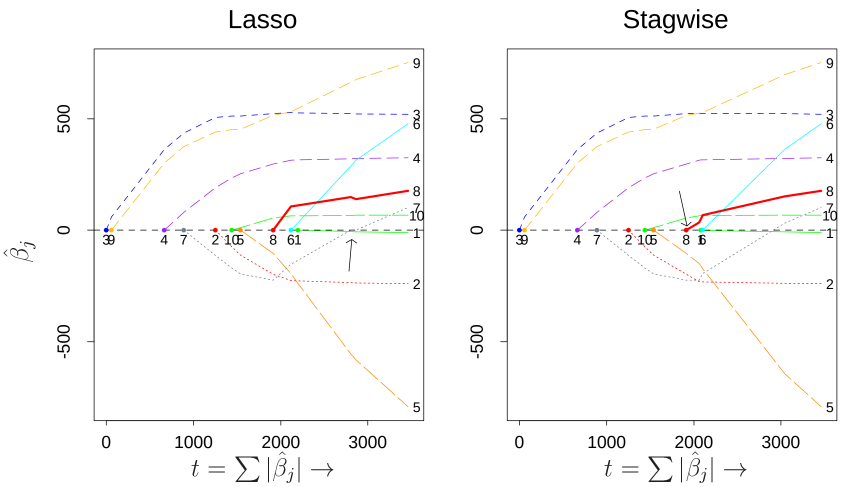

Figure 5: Another example of the equivalence between the Lasso optimal solution path (left) and ε-boosting with squared error loss. Note that the equivalence breaks down when the path of variable 7 becomes non-monotone

this similarity for squared error loss fitting with l_1 (lasso) penalty. Figure 5 shows another example in the same mold, taken from Efron et al. (2004). The data is a diabetes study and the "dictionary" used is just the original 10 variables. The panel on the left shows the path of optimal l_1-constrained solutions $\hat{\beta}(c)$ and the panel on the right shows the ε-boosting path with the 10-dimensional dictionary (the total number of boosting iterations is about 6000). The 1-dimensional path through $\mathbb{R}^{10}$ is described by 10 coordinate curves, corresponding to each one of the variables. The interesting phenomenon we observe is that the two coefficient traces are not completely identical. Rather, they agree up to the point where variable 7 coefficient path becomes *non monotone*, i.e., it violates (10) (this point is where variable 8 comes into the model, see the arrow on the right panel). This example illustrates that the monotonicity condition—and its implication that $\|\alpha\|_1 = \|\beta\|_1$—is critical for the equivalence between ε-boosting and l_1-constrained optimization.

The two examples we have seen so far have used squared error loss, and we should ask ourselves whether this equivalence stretches beyond this loss. Figure 6 shows a similar result, but this time for the binomial log-likelihood loss, C_l. We used the "spam" data set, taken from the UCI repository (Blake and Merz, 1998). We chose only 5 predictors of the 57 to make the plots more interpretable and the computations more accommodating. We see that there is a perfect equivalence between the exact constrained solution (i.e., regularized logistic regression) and ε-boosting in this case, since the paths are fully monotone.

To justify why this observed equivalence is not surprising, let us consider the following "l_1-locally optimal monotone direction" problem of finding the best monotone ε increment to a given model β_0:

$$\begin{aligned} \min \quad & C(\beta) \\ s.t. \quad & \|\beta\|_1 - \|\beta_0\|_1 \leq \varepsilon, \\ & |\beta| \succeq |\beta_0| \text{ (component-wise).} \end{aligned} \tag{12}$$

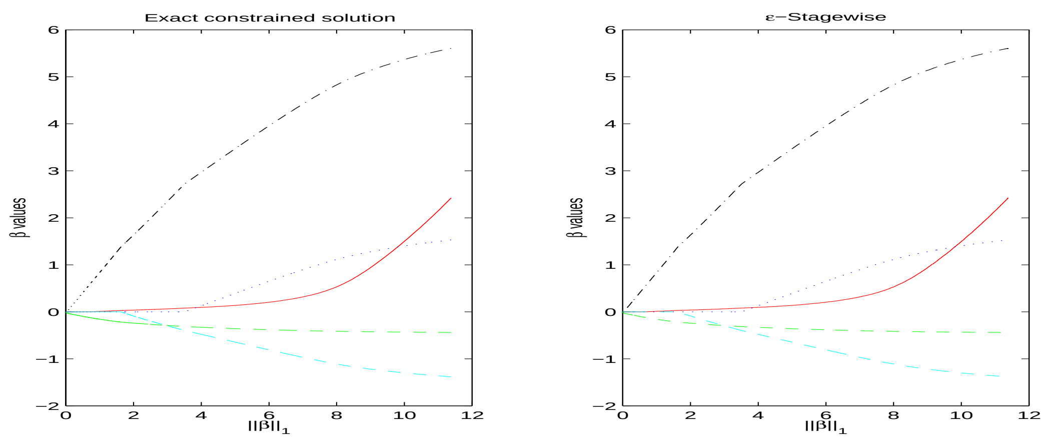

Figure 6: Exact coefficient paths (left) for l_1-constrained logistic regression and boosting coefficient paths (right) with binomial log-likelihood loss on five variables from the "spam" data set. The boosting path was generated using $\varepsilon = 0.003$ and 7000 iterations.

Here we use $C(\beta)$ as shorthand for $\sum_i C(y_i, h(\mathbf{x}_i)'\beta)$. A first order Taylor expansion gives us

$$C(\beta) = C(\beta_0) + \nabla C(\beta_0)'(\beta - \beta_0) + O(\varepsilon^2).$$

And given the l_1 constraint on the increase in $\|\beta\|_1$, it is easy to see that a first-order optimal solution (and therefore an optimal solution as $\varepsilon \to 0$) will make a "coordinate descent" step, i.e.

$$\beta_j \neq \beta_{0,j} \quad \Rightarrow \quad |\nabla C(\beta_0)_j| = \max_k |\nabla C(\beta_0)_k|,$$

assuming the signs match, i.e., $sign(\beta_{0j}) = -sign(\nabla C(\beta_0)_j)$.

So we get that if the optimal solution to (12) *without* the monotonicity constraint happens to be monotone, then it is equivalent to a coordinate descent step. And so it is reasonable to expect that if the optimal l_1 regularized path is monotone (as it indeed is in Figures 1,6), then an "infinitesimal" ε-boosting algorithm would follow the same path of solutions. Furthermore, even if the optimal path is not monotone, we can still use the formulation (12) to argue that ε-boosting would tend to follow an approximate l_1-regularized path. The main difference between the ε-boosting path and the true optimal path is that it will tend to "delay" becoming non-monotone, as we observe for variable 7 in Figure 5. To understand this specific phenomenon would require analysis of the true optimal path, which falls outside the scope of our discussion—Efron et al. (2004) cover the subject for squared error loss, and their discussion applies to any continuously differentiable convex loss, using second-order approximations.

We can employ this understanding of the relationship between boosting and l_1 regularization to construct l_p boosting algorithms by changing the coordinate-selection criterion in the coordinate descent algorithm. We will get back to this point in Section 7, where we design an "l_2 boosting" algorithm.

The experimental evidence and heuristic discussion we have presented lead us to the following conjecture which connects slow boosting and l_1-regularized optimization:

Conjecture 2 *Consider applying the* ε*-boosting algorithm to any convex loss function, generating a path of solutions* $\beta^{(\varepsilon)}(t)$*. Then if the optimal coefficient paths are monotone* $\forall c < c_0$*, i.e., if* $\forall j,\ |\hat{\beta}(c)_j|$ *is non-decreasing in the range* $c < c_0$*, then*

$$\lim_{\varepsilon \to 0} \beta^{(\varepsilon)}(c_0/\varepsilon) = \hat{\beta}(c_0).$$

Efron et al. (2004, Theorem 2) prove a weaker "local" result for the case of squared error loss only. We generalize their result to any convex loss. However this result still does not prove the "global" convergence which the conjecture claims, and the empirical evidence implies. For the sake of brevity and readability, we defer this proof, together with concise mathematical definition of the different types of convergence, to appendix A.

In the context of "real-life" boosting, where the number of basis functions is usually very large, and making ε small enough for the theory to apply would require running the algorithm forever, these results should not be considered directly applicable. Instead, they should be taken as an intuitive indication that boosting—especially the ε version—is, indeed, approximating optimal solutions to the constrained problems it encounters along the way.

5. l_p-Constrained Classification Loss Functions

Having established the relation between boosting and l_1 regularization, we are going to turn our attention to the regularized optimization problem. By analogy, our results will apply to boosting as well. We concentrate on C_e and C_l, the two classification losses defined above, and the solution paths of their l_p constrained versions:

$$\hat{\beta}^{(p)}(c) = \arg\min_{\|\beta\|_p \le c} \sum_i C(y_i, \beta' h(\mathbf{x_i})). \tag{13}$$

where C is either C_e or C_l. As we discussed below Equation (11), if the training data is separable in $span(\mathcal{H})$, then we have $\|\hat{\beta}^{(p)}(c)\|_p = c$ for *all* values of c. Consequently

$$\|\frac{\hat{\beta}^{(p)}(c)}{c}\|_p = 1.$$

We may ask what are the convergence points of this sequence as $c \to \infty$. The following theorem shows that these convergence points describe "l_p-margin maximizing" separating hyper-planes.

Theorem 3 *Assume the data is separable, i.e.,* $\exists \beta\ s.t. \forall i,\ y_i \beta' h(\mathbf{x}_i) > 0$*.*
Then for both C_e *and* C_l*,* every *convergence point of* $\frac{\hat{\beta}(c)}{c}$ *corresponds to an* l_p*-margin-maximizing separating hyper-plane.*
If the l_p*-margin-maximizing separating hyper-plane is unique, then it is the unique convergence points, i.e.*

$$\hat{\beta}^{(p)} = \lim_{c \to \infty} \frac{\hat{\beta}^{(p)}(c)}{c} = \arg\max_{\|\beta\|_p = 1} \min_i y_i \beta' h(\mathbf{x}_i). \tag{14}$$

Proof This proof applies to both C_e and C_l, given the property in (5). Consider two separating candidates β_1 and β_2 such that $\|\beta_1\|_p = \|\beta_2\|_p = 1$. Assume that β_1 separates better, i.e.

$$m_1 := \min_i y_i \beta_1' h(\mathbf{x}_i) > m_2 := \min_i y_i \beta_2' h(\mathbf{x}_i) > 0.$$

Then we have the following simple lemma:

Lemma 4 *There exists some $D = D(m_1, m_2)$ such that $\forall d > D$, $d\beta_1$ incurs smaller loss than $d\beta_2$, in other words:*

$$\sum_i C(y_i, d\beta_1' h(\mathbf{x}_i)) < \sum_i C(y_i, d\beta_2' h(\mathbf{x}_i)).$$

Given this lemma, we can now prove that any convergence point of $\frac{\hat{\beta}^{(p)}(c)}{c}$ must be an l_p-margin maximizing separator. Assume β^* is a convergence point of $\frac{\hat{\beta}^{(p)}(c)}{c}$. Denote its minimal margin on the data by m^*. If the data is separable, clearly $m^* > 0$ (since otherwise the loss of $d\beta^*$ does not even converge to 0 as $d \to \infty$).

Now, assume some $\tilde{\beta}$ with $\|\tilde{\beta}\|_p = 1$ has bigger minimal margin $\tilde{m} > m^*$. By continuity of the minimal margin in β, there exists some open neighborhood of β^*

$$N_{\beta^*} = \{\beta : \|\beta - \beta^*\|_2 < \delta\}$$

and an $\varepsilon > 0$, such that

$$\min_i y_i \beta' h(\mathbf{x_i}) < \tilde{m} - \varepsilon, \;\; \forall \beta \in N_{\beta^*}.$$

Now by the lemma we get that there exists some $D = D(\tilde{m}, \tilde{m} - \varepsilon)$ such that $d\tilde{\beta}$ incurs smaller loss than $d\beta$ for any $d > D$, $\beta \in N_{\beta^*}$. Therefore β^* *cannot be a convergence point of* $\frac{\hat{\beta}^{(p)}(c)}{c}$.

We conclude that any convergence point of the sequence $\frac{\hat{\beta}^{(p)}(c)}{c}$ must be an l_p-margin maximizing separator. If the margin maximizing separator is unique then it is the only possible convergence point, and therefore

$$\hat{\beta}^{(p)} = \lim_{c \to \infty} \frac{\hat{\beta}^{(p)}(c)}{c} = \arg \max_{\|\beta\|_p = 1} \min_i y_i \beta' h(\mathbf{x}_i).$$

■

Proof of Lemma Using (5) and the definition of C_e, we get for both loss functions:

$$\sum_i C(y_i, d\beta_1' h(\mathbf{x_i})) \le n \exp(-d \cdot m_1).$$

Now, since β_1 separates better, we can find our desired

$$D = D(m_1, m_2) = \frac{logn + log2}{m_1 - m_2}$$

such that

$$\forall d > D, \;\; n \exp(-d \cdot m_1) < 0.5 \exp(-d \cdot m_2).$$

And using (5) and the definition of C_e again we can write

$$0.5 \exp(-d \cdot m_2) \le \sum_i C(y_i, d\beta_2' h(\mathbf{x_i})).$$

Combining these three inequalities we get our desired result:

$$\forall d > D, \;\; \sum_i C(y_i, d\beta_1' h(\mathbf{x_i})) \le \sum_i C(y_i, d\beta_2' h(\mathbf{x_i})).$$

■

We thus conclude that if the l_p-margin maximizing separating hyper-plane is unique, the normalized constrained solution converges to it. In the case that the margin maximizing separating hyper-plane is not unique, we can in fact prove a stronger result, which indicates that the limit of the regularized solutions would then be determined by the second smallest margin, then by the third and so on. This result is mainly of technical interest and we prove it in Appendix B, Section 2.

5.1 Implications of Theorem 3

We now briefly discuss the implications of this theorem for boosting and logistic regression.

5.1.1 Boosting Implications

Combined with our results from Section 4, Theorem 3 indicates that the normalized boosting path $\frac{\beta^{(t)}}{\sum_{u\leq t}\alpha_u}$—with either C_e or C_l used as loss—"approximately" converges to a separating hyper-plane $\hat{\beta}$, which attains

$$\max_{\|\beta\|_1=1} \min_i y_i \beta' h(\mathbf{x_i}) = \max_{\|\beta\|_1=1} \|\beta\|_2 \min_i y_i d_i, \tag{15}$$

where d_i is the (signed) Euclidean distance from the training point i to the separating hyper-plane. In other words, it maximizes Euclidean distance scaled by an l_2 norm. As we have mentioned already, this implies that the *asymptotic* boosting solution will tend to be sparse in representation, due to the fact that for fixed l_1 norm, the l_2 norm of vectors that have many 0 entries will generally be larger. In fact, under rather mild conditions, the asymptotic solution $\hat{\beta} = \lim_{c\to\infty} \hat{\beta}^{(1)}(c)/c$, will have *at most n* (the number of observations) non-zero coefficients, if we use either C_l or C_e as the loss. See Appendix B, Section 1 for proof.

5.1.2 Logistic Regression Implications

Recall, that the logistic regression (maximum likelihood) solution is undefined if the data is separable in the Euclidean space spanned by the predictors. Theorem 3 allows us to define a logistic regression solution for separable data, as follows:

1. Set a high constraint value c_{max}

2. Find $\hat{\beta}^{(p)}(c_{max})$, the solution to the logistic regression problem subject to the constraint $\|\beta\|_p \leq c_{max}$. The problem is convex for any $p \geq 1$ and differentiable for any $p > 1$, so interior point methods can be used to solve this problem.

3. Now you have (approximately) the l_p-margin maximizing solution for this data, described by

$$\frac{\hat{\beta}^{(p)}(c_{max})}{c_{max}}.$$

   This is a solution to the original problem in the sense that it is, approximately, the convergence point of the normalized l_p-constrained solutions, as the constraint is relaxed.

Of course, with our result from Theorem 3 it would probably make more sense to simply find the optimal separating hyper-plane directly—this is a linear programming problem for l_1 separation and a quadratic programming problem for l_2 separation. We can then consider this optimal separator as a logistic regression solution for the separable data.

6. Examples

We now apply boosting to several data sets and interpret the results in light of our regularization and margin-maximization view.

6.1 Spam Data Set

We now know if the data are separable and we let boosting run forever, we will approach the same "optimal" separator for both C_e and C_l. However if we stop early—or if the data is not separable—the behavior of the two loss functions may differ significantly, since C_e weighs negative margins exponentially, while C_l is approximately linear in the margin for large negative margins (see Friedman et al., 2000). Consequently, we can expect C_e to concentrate more on the "hard" training data, in particular in the non-separable case. Figure 7 illustrates the behavior of ε-boosting with both

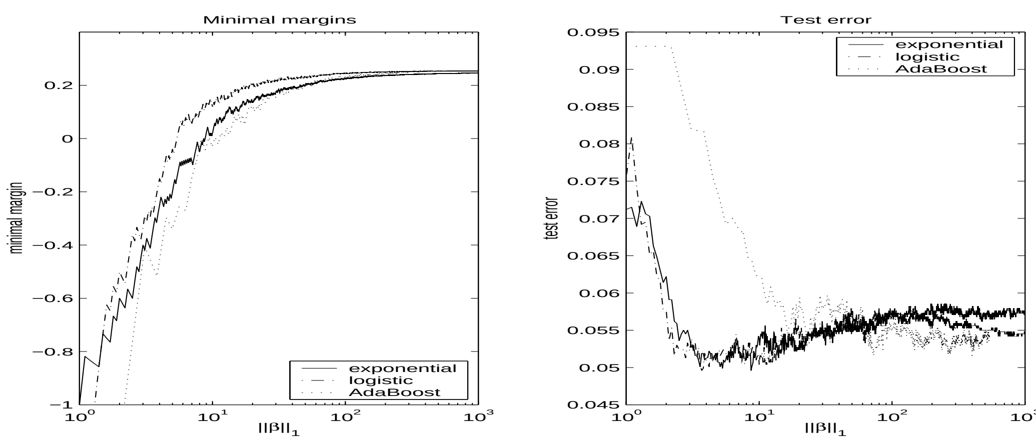

Figure 7: Behavior of boosting with the two loss functions on spam data set

loss functions, as well as that of AdaBoost, on the spam data set (57 predictors, binary response). We used 10 node trees and $\varepsilon = 0.1$. The left plot shows the minimal margin as a function of the l_1 norm of the coefficient vector $\|\beta\|_1$. Binomial loss creates a bigger minimal margin initially, but the minimal margins for both loss functions are converging asymptotically. AdaBoost initially lags behind but catches up nicely and reaches the same minimal margin asymptotically. The right plot shows the test error as the iterations proceed, illustrating that both ε-methods indeed seem to over-fit eventually, even as their "separation" (minimal margin) is still improving. AdaBoost did not significantly over-fit in the 1000 iterations it was allowed to run, but it obviously would have if it were allowed to run on.

We should emphasize that the comparison between AdaBoost and ε-boosting presented considers as a basis for comparison the l_1 norm, not the number of iterations. In terms of computational complexity, as represented by the number of iterations, AdaBoost reaches both a large minimal margin and good prediction performance much more quickly than the "slow boosting" approaches, as AdaBoost tends to take larger steps.

6.2 Simulated Data

To make a more educated comparison and more compelling visualization, we have constructed an example of separation of 2-dimensional data using a 8-th degree polynomial dictionary (45 functions). The data consists of 50 observations of each class, drawn from a mixture of Gaussians, and presented in Figure 8. Also presented, in the solid line, is the optimal l_1 separator for this data in this dictionary (easily calculated as a linear programming problem - note the difference from the l_2 optimal decision boundary, presented in Section 7.1, Figure 11). The optimal l_1 separator has only 12 non-zero coefficients out of 45.

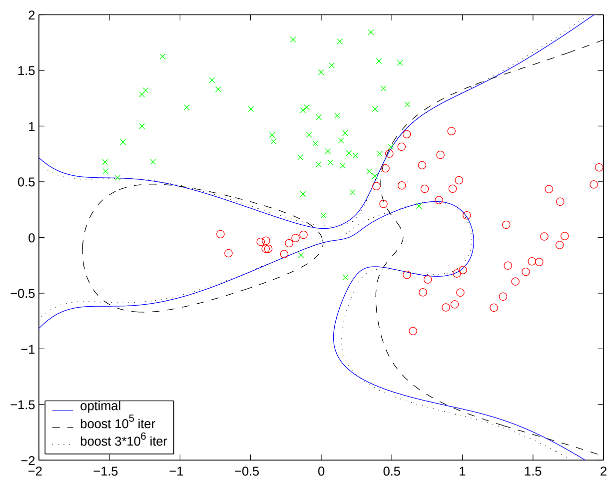

Figure 8: Artificial data set with l_1-margin maximizing separator (solid), and boosting models after 10^5 iterations (dashed) and 10^6 iterations (dotted) using $\varepsilon = 0.001$. We observe the convergence of the boosting separator to the optimal separator

We ran an ε-boosting algorithm on this data set, using the logistic log-likelihood loss C_l, with $\varepsilon = 0.001$, and Figure 8 shows two of the models generated after 10^5 and $3 \cdot 10^6$ iterations. We see that the models seem to converge to the optimal separator. A different view of this convergence is given in Figure 9, where we see two measures of convergence: the minimal margin (left, maximum value obtainable is the horizontal line) and the l_1-norm distance between the normalized models (right), given by

$$\sum_j \left| \hat{\beta}_j - \frac{\beta_j^{(t)}}{\|\beta^{(t)}\|_1} \right|,$$

where $\hat{\beta}$ is the optimal separator with l_1 norm 1 and $\beta^{(t)}$ is the boosting model after t iterations.

We can conclude that on this simple artificial example we get nice convergence of the logistic-boosting model path to the l_1-margin maximizing separating hyper-plane.

We can also use this example to illustrate the similarity between the boosted path and the path of l_1 optimal solutions, as we have discussed in Section 4.

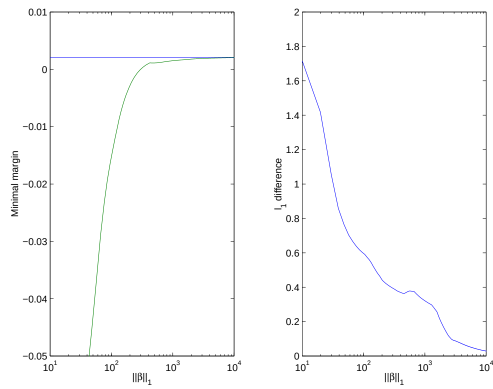

Figure 9: Two measures of convergence of boosting model path to optimal l_1 separator: minimal margin (left) and l_1 distance between the normalized boosting coefficient vector and the optimal model (right)

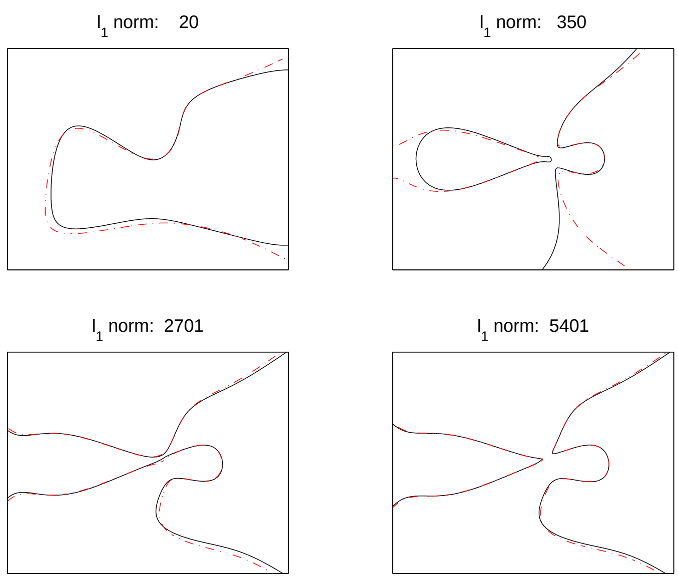

Figure 10: Comparison of decision boundary of boosting models (broken) and of optimal constrained solutions with same norm (full)

Figure 10 shows the class decision boundaries for 4 models generated along the boosting path, compared to the optimal solutions to the constrained "logistic regression" problem with the same bound on the l_1 norm of the coefficient vector. We observe the clear similarities in the way the solutions evolve and converge to the optimal l_1 separator. The fact that they differ (in some cases significantly) is not surprising if we recall the monotonicity condition presented in Section 4 for exact correspondence between the two model paths. In this case if we look at the coefficient paths (not shown), we observe that the monotonicity condition is consistently violated in the low norm ranges, and hence we can expect the paths to be similar in spirit but not identical.

7. Discussion

We can now summarize what we have learned about boosting from the previous sections:

- Boosting approximately follows the path of l_1-regularized models for its loss criterion
- If the loss criterion is the exponential loss of AdaBoost or the binomial log-likelihood loss of logistic regression, then the l_1 regularized model converges to an l_1-margin maximizing separating hyper-plane, if the data are separable in the span of the weak learners

We may ask, which of these two points is the key to the success of boosting approaches. One empirical clue to answering this question, can be found in Breiman (1999), who programmed an algorithm to *directly* maximize the margins. His results were that his algorithm consistently got significantly higher minimal margins than AdaBoost on many data sets (and, in fact, a "higher" margin distribution beyond the minimal margin), but had slightly worse prediction performance. His conclusion was that margin maximization is *not* the key to AdaBoost's success. From a statistical perspective we can embrace this conclusion, as reflecting the importance of regularization in high-dimensional predictor space. By our results from the previous sections, "margin maximization" can be viewed as the limit of parametric regularized models, as the regularization vanishes.[4] Thus we would generally expect the margin maximizing solutions to perform *worse* than regularized models. In the case of boosting, regularization would correspond to "early stopping" of the boosting algorithm.

7.1 Boosting and SVMs as Regularized Optimization in High-dimensional Predictor Spaces

Our exposition has led us to view boosting as an approximate way to solve the regularized optimization problem

$$\min_{\beta} \sum_i C(y_i, \beta' h(\mathbf{x}_i)) + \lambda \|\beta\|_1 \tag{16}$$

which converges as $\lambda \to 0$ to $\hat{\beta}^{(1)}$, if our loss is C_e or C_l. In general, the loss C can be any convex differentiable loss and should be defined to match the problem domain.

Support vector machines can be described as solving the regularized optimization problem (see Friedman et al., 2000, Chapter 12)

$$\min_{\beta} \sum_i (1 - y_i \beta' h(\mathbf{x}_i))_+ + \lambda \|\beta\|_2^2 \tag{17}$$

which "converges" as $\lambda \to 0$ to the non-regularized support vector machine solution, i.e., the optimal Euclidean separator, which we denoted by $\hat{\beta}^{(2)}$.

An interesting connection exists between these two approaches, in that they allow us to solve the regularized optimization problem in high dimensional predictor space:

4. It can be argued that margin-maximizing models are still "regularized" in some sense, as they minimize a norm criterion among all separating models. This is arguably the property which still allows them to generalize reasonably well in many cases.

- We are able to solve the l_1- regularized problem approximately in very high dimension via boosting by applying the "approximate coordinate descent" trick of building a decision tree (or otherwise greedily selecting a weak learner) based on re-weighted versions of the data.

- Support vector machines facilitate a different trick for solving the regularized optimization problem in high dimensional predictor space: the "kernel trick". If our dictionary $\mathcal{H}$ spans a Reproducing Kernel Hilbert Space, then RKHS theory tells us we can find the regularized solutions by solving an n-dimensional problem, in the space spanned by the kernel representers $\{K(\mathbf{x}_i, \mathbf{x})\}$. This fact is by no means limited to the hinge loss of (17), and applies to any convex loss. We concentrate our discussion on SVM (and hence hinge loss) only since it is by far the most common and well-known application of this result.

So we can view both boosting and SVM as methods that allow us to fit regularized models in high dimensional predictor space using a computational "shortcut". The complexity of the model built is controlled by regularization. These methods are distinctly different than traditional statistical approaches for building models in high dimension, which start by reducing the dimensionality of the problem so that standard tools (e.g., Newton's method) can be applied to it, and also to make over-fitting less of a concern. While the merits of regularization without dimensionality reduction—like Ridge regression or the Lasso—are well documented in statistics, computational issues make it impractical for the size of problems typically solved via boosting or SVM, without computational tricks.

We believe that this difference may be a significant reason for the enduring success of boosting and SVM in data modeling, i.e.:

> Working in high dimension and regularizing is statistically preferable to a two-step procedure of first reducing the dimension, then fitting a model in the reduced space.

It is also interesting to consider the differences between the two approaches, in the loss (flexible vs. hinge loss), the penalty (l_1 vs. l_2), and the type of dictionary used (usually trees vs. RKHS). These differences indicate that the two approaches will be useful for different situations. For example, if the true model has a sparse representation in the chosen dictionary, then l_1 regularization may be warranted; if the form of the true model facilitates description of the class probabilities via a logistic-linear model, then the logistic loss C_l is the best loss to use, and so on.

The computational tricks for both SVM and boosting limit the kind of regularization that can be used for fitting in high dimensional space. However, the problems can still be formulated and solved for different regularization approaches, as long as the dimensionality is low enough:

- Support vector machines can be fitted with an l_1 penalty, by solving the *1-norm* version of the SVM problem, equivalent to replacing the l_2 penalty in (17) with an l_1 penalty. In fact, the 1-norm SVM is used quite widely, because it is more easily solved in the "linear", non-RKHS, situation (as a linear program, compared to the standard SVM which is a quadratic program) and tends to give sparser solutions in the primal domain.

- Similarly, we describe below an approach for developing a "boosting" algorithm for fitting approximate l_2 regularized models.

Both of these methods are interesting and potentially useful. However they lack what is arguably the most attractive property of the "standard" boosting and SVM algorithms: a computational trick to allow fitting in high dimensions.

7.1.1 AN l_2 BOOSTING ALGORITHM

We can use our understanding of the relation of boosting to regularization and Theorem 3 to formulate l_p-boosting algorithms, which will approximately follow the path of l_p-regularized solutions and converge to the corresponding l_p-margin maximizing separating hyper-planes. Of particular interest is the l_2 case, since Theorem 3 implies that l_2-constrained fitting using C_l or C_e will build a regularized path to the optimal separating hyper-plane in the Euclidean (or SVM) sense.

To construct an l_2 boosting algorithm, consider the "equivalent" optimization problem (12), and change the step-size constraint to an l_2 constraint:

$$\|\beta\|_2 - \|\beta_0\|_2 \leq \varepsilon.$$

It is easy to see that the first order solution to this problem entails selecting for modification the coordinate which maximizes

$$\frac{\nabla C(\beta_0)_k}{\beta_{0,k}}$$

and that subject to monotonicity, this will lead to a correspondence to the locally l_2-optimal direction.

Following this intuition, we can construct an l_2 boosting algorithm by changing only step $2(c)$ of our generic boosting algorithm of Section 2 to

2(c)* Identify j_t which maximizes $\frac{|\sum_i w_i h_{j_t}(\mathbf{x}_i)|}{|\beta_{j_t}|}$.

Note that the need to consider the current coefficient (in the denominator) makes the l_2 algorithm appropriate for toy examples only. In situations where the dictionary of weak learner is prohibitively large, we will need to figure out a trick like the one we presented in Section 2.1, to allow us to make an approximate search for the optimizer of step 2(c)*.

Another problem in applying this algorithm to large problems is that we never choose the same dictionary function twice, until all have non-0 coefficients. This is due to the use of the l_2 penalty, where the current coefficient value affects the rate at which the penalty term is increasing. In particular, if $\beta_j = 0$ then increasing it causes the penalty term $\|\beta\|_2$ to increase at rate 0, to first order (which is all the algorithm is considering).

The convergence of our l_2 boosting algorithm on the artificial data set of Section 6.2 is illustrated in Figure 11. We observe that the l_2 boosting models do indeed approach the optimal l_2 separator. It is interesting to note the significant difference between the optimal l_2 separator as presented in Figure 11 and the optimal l_1 separator presented in Section 6.2 (Figure 8).

8. Summary and Future Work

In this paper we have introduced a new view of boosting in general, and two-class boosting in particular, comprised of two main points:

- We have generalized results from Efron et al. (2004) and Hastie et al. (2001), to describe boosting as approximate l_1-regularized optimization.

- We have shown that the exact l_1-regularized solutions converge to an l_1-margin maximizing separating hyper-plane.

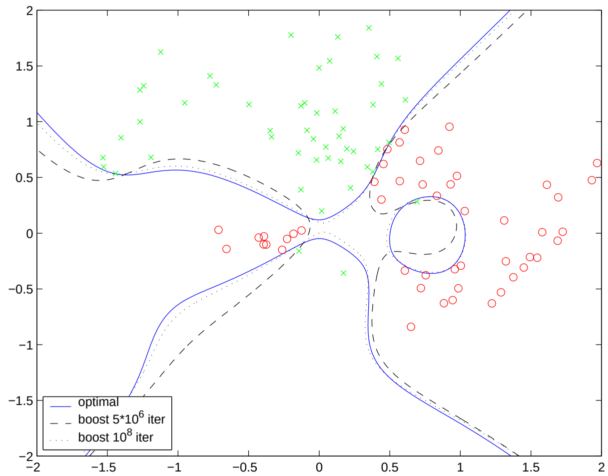

Figure 11: Artificial data set with l_2-margin maximizing separator (solid), and l_2-boosting models after $5 * 10^6$ iterations (dashed) and 10^8 iterations (dotted) using $\varepsilon = 0.0001$. We observe the convergence of the boosting separator to the optimal separator

We hope our results will help in better understanding how and why boosting works. It is an interesting and challenging task to separate the effects of the different components of a boosting algorithm:

- Loss criterion
- Dictionary and greedy learning method
- Line search / slow learning

and relate them to its success in different scenarios. The implicit l_1 regularization in boosting may also contribute to its success, as it has been shown that in some situations l_1 regularization is inherently superior to others (see Donoho et al., 1995).

An important issue when analyzing boosting is over-fitting in the noisy data case. To deal with over-fitting, Rätsch et al. (2001b) propose several regularization methods and generalizations of the original AdaBoost algorithm to achieve a *soft margin* by introducing slack variables. Our results indicate that the models along the boosting path can be regarded as l_1 regularized versions of the optimal separator, hence regularization can be done more directly and naturally by stopping the boosting iterations early. It is essentially a choice of the l_1 constraint parameter c.

Many other questions arise from our view of boosting. Among the issues to be considered:

- Is there a similar "separator" view of multi-class boosting? We have some tentative results to indicate that this might be the case if the boosting problem is formulated properly.
- Can the constrained optimization view of boosting help in producing generalization error bounds for boosting that would be more tight than the current existing ones?

Acknowledgments

We thank Stephen Boyd, Brad Efron, Jerry Friedman, Robert Schapire and Rob Tibshirani for helpful discussions. We thank the referees for their thoughtful and useful comments. This work was partially supported by Stanford graduate fellowship, grant DMS-0204612 from the National Science Foundation, and grant ROI-CA-72028-01 from the National Institutes of Health.

Appendix A. Local Equivalence of Infinitesimal ε-Boosting and l_1-Constrained Optimization

As before, we assume we have a set of training data $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots (\mathbf{x}_n, y_n)$, a smooth cost function $C(y, F)$, and a set of basis functions $(h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots h_J(\mathbf{x}))$.

We denote by $\hat{\beta}(s)$ be the optimal solution of the l_1-constrained optimization problem:

$$\min_{\beta} \quad \sum_{i=1}^{n} C(y_i, h(\mathbf{x}_i)'\beta) \tag{18}$$

$$\text{subject to} \quad \|\beta\|_1 \leq s. \tag{19}$$

Suppose we initialize the ε-boosting version of Algorithm 1, as described in Section 2, at $\hat{\beta}(s)$ and run the algorithm for T steps. Let $\beta(T)$ denote the coefficients after T steps.

The "global convergence" Conjecture 2 in Section 4 implies that $\forall \Delta s > 0$:

$$\beta(\Delta s/\varepsilon) \to \hat{\beta}(s + \Delta s) \text{ as } \varepsilon \to 0$$

under some mild assumptions. Instead of proving this "global" result, we show here a "local" result by looking at the derivative of $\hat{\beta}(s)$. Our proof builds on the proof by Efron et al. (2004, Theorem 2) of a similar result for the case that the cost is squared error loss $C(y, F) = (y - F)^2$. Theorem 1 below shows that if we start the ε-boosting algorithm at a solution $\hat{\beta}(s)$ of the l_1-constrained optimization problem (18)–(19), the "direction of change" of the ε-boosting solution will agree with that of the l_1-constrained optimization problem.

Theorem 1 *Assume the optimal coefficient paths $\hat{\beta}_j(s)$ $\forall j$ are monotone in s and the coefficient paths $\beta_j(T)$ $\forall j$ are also monotone as ε-boosting proceeds, then*

$$\frac{\beta(T) - \hat{\beta}(s)}{T \cdot \varepsilon} \to \nabla\hat{\beta}(s) \text{ as } \varepsilon \to 0, T \to \infty, T \cdot \varepsilon \to 0.$$

Proof First we introduce some notations. Let

$$\mathbf{h}_j = (h_j(\mathbf{x}_1), \ldots h_j(\mathbf{x}_n))'$$

be the jth basis function evaluated at the n training data.

Let

$$\mathbf{F} = (F(\mathbf{x}_1), \ldots F(\mathbf{x}_n))'$$

be the vector of current fit.

Let

$$\mathbf{r} = \left(-\frac{\partial C(y_1, F_1)}{\partial F_1}, \ldots - \frac{\partial C(y_n, F_n)}{\partial F_n}\right)'$$

be the current "generalized residual" vector as defined in Friedman (2001).

Let

$$c_j = \mathbf{h}_j'\mathbf{r}, \quad j = 1, \dots J$$

be the current "correlation" between $\mathbf{h}_j$ and $\mathbf{r}$.

Let

$$\mathcal{A} = \{j : |c_j| = \max_j |c_j|\}$$

be the set of indices for the maximum absolute correlation.

For clarity, we re-write this ε-boosting algorithm, starting from $\hat{\beta}(s)$, as a special case of Algorithm 1, as follows:

(1) Initialize $\beta(0) = \hat{\beta}(s), \mathbf{F}_0 = \mathbf{F}, \mathbf{r}_0 = \mathbf{r}$.

(2) For $t = 1 : T$

- (a) Find $j_t = \arg\max_j |\mathbf{h}_j'\mathbf{r}_{t-1}|$.
- (b) Update
$$\beta_{t,j_t} \leftarrow \beta_{t-1,j_t} + \varepsilon \cdot sign(c_{j_t})$$
- (c) Update $\mathbf{F}_t$ and $\mathbf{r}_t$.

Notice in the above algorithm, we start from $\hat{\beta}(s)$, rather than 0. As proposed in Efron et al. (2004), we consider an idealized ε-boosting case: $\varepsilon \to 0$. As $\varepsilon \to 0$, $T \to \infty$ and $T \cdot \varepsilon \to 0$, under the monotone paths condition, Section 3.2 and Section 6 of Efron et al. (2004) showed

$$\frac{\mathbf{F}_T - \mathbf{F}_0}{T \cdot \varepsilon} \quad \to \quad \mathbf{u}, \tag{20}$$

$$\frac{\mathbf{r}_T - \mathbf{r}_0}{T \cdot \varepsilon} \quad \to \quad \mathbf{v}, \tag{21}$$

where $\mathbf{u}$ and $\mathbf{v}$ satisfy two constraints:

(**Constraint 1**) $\mathbf{u}$ is in the convex cone generated by $\{sign(c_j)\mathbf{h}_j : j \in \mathcal{A}\}$, i.e.:

$$\mathbf{u} = \sum_{j \in \mathcal{A}} P_j sign(c_j)\mathbf{h}_j, P_j \geq 0.$$

(**Constraint 2**) $\mathbf{v}$ has equal "correlation" with $sign(c_j)\mathbf{h}_j, j \in \mathcal{A}$:

$$sign(c_j)\mathbf{h}_j'\mathbf{v} = \lambda_{\mathcal{A}} \text{ for } j \in \mathcal{A}.$$

The first constraint is true because the basis functions in $\mathcal{A}^C$ will not be able to catch up in terms of $|c_j|$ for sufficiently small $T \cdot \varepsilon$; the P_j's are non-negative because the coefficient paths $\beta_j(T)$ are monotone. The second constraint can be seen by taking a Taylor expansion of $C(y,F)$ around F_0 to the quadratic term, letting $T \cdot \varepsilon$ go to zero and applying the result for the squared error loss from Efron et al. (2004). Once the two constraints are established, we notice that

$$v_i = -\left.\frac{\partial^2 C(y_i, F)}{\partial F^2}\right|_{F_0(\mathbf{x}_i)} u_i.$$

Hence we can plug the constraint 1 into the constraint 2 and get the following set of equations:

$$\tilde{H}_{\mathcal{A}}^T W \tilde{H}_{\mathcal{A}} \mathbf{P} = \lambda_{\mathcal{A}} \mathbf{1},$$

where

$$\begin{aligned}
\tilde{H}_{\mathcal{A}} &= (\cdots sign(c_j)\mathbf{h}_j \cdots), \ j \in \mathcal{A}, \\
W &= diag\left(-\frac{\partial^2 C(y_i, F)}{\partial F^2}\bigg|_{F_0(\mathbf{x}_i)} \right), \\
\mathbf{P} &= (\cdots P_j \cdots)', \ j \in \mathcal{A}.
\end{aligned}$$

If $\tilde{H}$ is of rank $|\mathcal{A}|$ (we will get back to this issue in details in Appendix B), then $\mathbf{P}$, or equivalently $\mathbf{u}$ and $\mathbf{v}$, are uniquely determined up to a scale number.

Now we consider the l_1-constrained optimization problem (18)–(19). Let $\hat{\mathbf{F}}(s)$ be the fitted vector and $\hat{\mathbf{r}}(s)$ be the corresponding residual vector. Since $\hat{\mathbf{F}}(s)$ and $\hat{\mathbf{r}}(s)$ are smooth, define

$$\mathbf{u}^* \equiv \lim_{\Delta s \to 0} \frac{\hat{\mathbf{F}}(s+\Delta s) - \hat{\mathbf{F}}(s)}{\Delta s}, \tag{22}$$

$$\mathbf{v}^* \equiv \lim_{\Delta s \to 0} \frac{\hat{\mathbf{r}}(s+\Delta s) - \hat{\mathbf{r}}(s)}{\Delta s}. \tag{23}$$

Lemma 2 *Under the monotone coefficient paths assumption,* $\mathbf{u}^*$ *and* $\mathbf{v}^*$ *also satisfy constraints 1–2.*

Proof Write the coefficient β_j as $\beta_j^+ - \beta_j^-$, where

$$\begin{cases} \beta_j^+ = \beta_j, \beta_j^- = 0 & \text{if} \quad \beta_j > 0, \\ \beta_j^+ = 0, \beta_j^- = -\beta_j & \text{if} \quad \beta_j < 0. \end{cases}$$

The l_1-constrained optimization problem (18)–(19) is then equivalent to

$$\min_{\beta^+,\beta^-} \quad \sum_{i=1}^n C\left(y_i, h(\mathbf{x}_i)'(\beta^+ - \beta^-)\right), \tag{24}$$

$$\text{subject to} \quad \|\beta^+\|_1 + \|\beta^-\|_1 \le s, \beta^+ \ge 0, \beta^- \ge 0. \tag{25}$$

The corresponding Lagrangian dual is

$$L = \sum_{i=1}^n C\left(y_i, h(\mathbf{x}_i)'(\beta^+ - \beta^-)\right) + \lambda \sum_{j=1}^J (\beta_j^+ + \beta_j^-) \tag{26}$$

$$-\lambda \cdot s - \sum_{j=1}^J \lambda_j^+ \beta_j^+ - \sum_{j=1}^J \lambda_j^- \beta_j^-, \tag{27}$$

where $\lambda \ge 0, \lambda_j^+ \ge 0, \lambda_j^- \ge 0$ are Lagrange multipliers.

By differentiating the Lagrangian dual, we get the solution of (24)–(25) needed to satisfy the following Karush-Kuhn-Tucker conditions:

$$\frac{\partial L}{\partial \beta_j^+} = -\mathbf{h}_j'\hat{\mathbf{r}} + \lambda - \lambda_j^+ = 0, \tag{28}$$

$$\begin{aligned}
\frac{\partial L}{\partial \beta_j^-} &= \mathbf{h}_j'\hat{\mathbf{r}} + \lambda - \lambda_j^- = 0, &(29)\\
\lambda_j^+ \hat{\beta}_j^+ &= 0, &(30)\\
\lambda_j^- \hat{\beta}_j^- &= 0. &(31)
\end{aligned}$$

Let $c_j = \mathbf{h}_j'\hat{\mathbf{r}}$ and $\mathcal{A} = \{j : |c_j| = \max_j |c_j|\}$. We can see the following facts from the Karush-Kuhn-Tucker conditions:

(**Fact 1**) Use (28), (29) and $\lambda \geq 0, \lambda_j^+, \lambda_j^- \geq 0$, we have $|c_j| \leq \lambda$.

(**Fact 2**) If $\hat{\beta}_j \neq 0$, then $|c_j| = \lambda$ and $j \in \mathcal{A}$. For example, suppose $\hat{\beta}_j^+ \neq 0$, then $\lambda_j^+ = 0$ and (28) implies $c_j = \lambda$.

(**Fact 3**) If $\hat{\beta}_j \neq 0$, $sign(\hat{\beta}_j) = sign(c_j)$.

We also note that:

- $\hat{\beta}_j^+$ and $\hat{\beta}_j^-$ can not both be non-zero, otherwise $\lambda_j^+ = \lambda_j^- = 0$, (28) and (29) can not hold at the same time.
- It is possible that $\hat{\beta}_j = 0$ and $j \in \mathcal{A}$. This only happens for a finite number of s values, where basis $\mathbf{h}_j$ is about to enter the model.

For sufficiently small Δs, since the second derivative of the cost function $C(y,F)$ is finite, $\mathcal{A}$ will stay the same. Since $j \in \mathcal{A}$ if $\hat{\beta}_j \neq 0$, the change in the fitted vector is

$$\hat{\mathbf{F}}(s+\Delta s) - \hat{\mathbf{F}}(s) = \sum_{j\in\mathcal{A}} Q_j \mathbf{h}_j.$$

Since $sign(\hat{\beta}_j) = sign(c_j)$ and the coefficients $\hat{\beta}_j$ change monotonically, $sign(Q_j)$ will agree with $sign(c_j)$. Hence we have

$$\frac{\hat{\mathbf{F}}(s+\Delta s) - \hat{\mathbf{F}}(s)}{\Delta s} = \sum_{j\in\mathcal{A}} P_j sign(c_j)\mathbf{h}_j. \tag{32}$$

This implies $\mathbf{u}^*$ satisfies constraint 1. The claim $\mathbf{v}^*$ satisfies constraint 2 follows directly from fact 2, since both $\hat{\mathbf{r}}(s+\Delta s)$ and $\hat{\mathbf{r}}(s)$ satisfy constraint 2. ■

Completion of proof of Theorem (1): We further notice that in both the ε-boosting case and the constrained optimization case, we have $\sum_{j\in\mathcal{A}} P_j = 1$ by definition and the monotone coefficient paths condition, hence $\mathbf{u}$ and $\mathbf{v}$ are uniquely determined, i.e.:

$$\mathbf{u} = \mathbf{u}^* \text{ and } \mathbf{v} = \mathbf{v}^*.$$

To translate the result into $\hat{\beta}(s)$ and $\beta(T)$, we notice $F(\mathbf{x}) = h(\mathbf{x})'\beta$. Efron et al. (2004) showed that for $\nabla\hat{\beta}(s)$ to be well defined, $\mathcal{A}$ can have at most n elements, i.e., $|\mathcal{A}| \leq n$. We give sufficient conditions for when this is true in Appendix B.

Now Let

$$H_{\mathcal{A}} = (\cdots h_j(\mathbf{x}_i) \cdots), i = 1, \ldots n; j \in \mathcal{A}$$

be a $n \times |\mathcal{A}|$ matrix, which we assume is of rank $|\mathcal{A}|$. Then $\nabla\hat{\beta}(s)$ is given by

$$\nabla\hat{\beta}(s) = \left(H'_{\mathcal{A}} W H_{\mathcal{A}}\right)^{-1} H'_{\mathcal{A}} W \mathbf{u}^*,$$

and

$$\frac{\beta(T) - \hat{\beta}(s)}{T \cdot \varepsilon} \to \left(H'_{\mathcal{A}} W H_{\mathcal{A}}\right)^{-1} H'_{\mathcal{A}} W \mathbf{u}.$$

Hence the theorem is proved. ■

Appendix B. Uniqueness and Existence Results

In this appendix, we give some details on the properties of regularized solution paths. In section B.1 we formulate and prove sparseness and uniqueness results on l_1-regularized solutions for any convex loss. In section B.2 we extend Theorem 3 of Section 5—which proved the margin maximizing property of the limit of l_p-regularized solutions, as regularization varies—to the case that the margin maximizing solution is not unique.

B.1 Sparseness and Uniqueness of l_1-Regularized Solutions and Their Limits

Consider the l_1-constrained optimization problem:

$$\min_{\|\beta\|_1 \le c} \sum_{i=1}^{n} C(y_i, \beta' h(\mathbf{x}_i)). \tag{33}$$

In this section we give sufficient conditions for the following properties of the solutions of (33):

1. Existence of a sparse solution (with at most n non-zero coefficients),
2. Non-existence of non-sparse solutions with more than n non-zero coefficients,
3. Uniqueness of the solution,
4. Convergence of the solutions to sparse solution, as c increases.

Theorem 3 *Assume that the unconstrained solution for problem (33) has l_1 norm bigger than c. Then there exists a solution of (33) which has at most n non-zero coefficients.*

Proof As Lemma 2 in the Appendix A, we will prove the theorem using the Karush-Kuhn-Tucker (KKT) formulation of the optimization problem.

The chain rule for differentiation gives us that

$$\frac{\partial \sum_i C(y_i, \beta' h(\mathbf{x}_i))}{\partial \beta_j} = -\mathbf{h}'_j \mathbf{r}(\beta), \tag{34}$$

where $\mathbf{h}_j$ and $\mathbf{r}(\beta)$ are defined in the Appendix A; $\mathbf{r}(\beta)$ is the "generalized residual" vector. Using this simple relationship and fact 2 of Lemma 2 we can write a system of equations for all non-zero coefficients at the optimal constrained solution as follows (denote by $\mathcal{A}$ the set of indices for non-zero coefficients):

$$H'_{\mathcal{A}} \mathbf{r}(\beta) = \lambda \cdot \text{sign}\beta_{\mathcal{A}}. \tag{35}$$

In other words, we get $|\mathcal{A}|$ equations in $|\mathcal{A}|$ variables, corresponding to the non-zero β_j's.

However, each column of the matrix $H_{\mathcal{A}}$ is of length n, and so $H_{\mathcal{A}}$ can have at most n linearly independent columns, $\text{rank}(H_{\mathcal{A}}) \leq n$. Assume now that we have an optimal solution for (33) with $|\mathcal{A}| > n$. Then there exists $l \in \mathcal{A}$ such that

$$\mathbf{h}_l = \sum_{j \in \mathcal{A}, j \neq l} \alpha_j \mathbf{h}_j. \tag{36}$$

Substituting (36) into the l'th row in (35) we get

$$(\sum_{j \in \mathcal{A}, j \neq l} \alpha_j \mathbf{h}_j)' \mathbf{r}(\beta) = \lambda \cdot \text{sign}\beta_l. \tag{37}$$

But from (35) we know that $\mathbf{h}_j' \mathbf{r}(\beta) = \lambda \cdot \text{sign}\beta_j$, $\forall j \in \mathcal{A}$, meaning we can re-phrase (37) as

$$\sum_{j \in \mathcal{A}, j \neq l} \alpha_j \cdot \text{sign}\beta_j \cdot \text{sign}\beta_l = 1. \tag{38}$$

In other words, we get that $\mathbf{h}_l$ is a linear combination of the columns of $H_{\mathcal{A}-\{l\}}$ which must obey the specific numeric relation in (38).

Now we can construct an alternative optimal solution for (33) with one less non-zero coefficient, as follows:

1. Start from β
2. Define the direction γ in coefficient space implied by (36), that is:
   $\gamma_l = -\text{sign}\beta_l$, $\gamma_j = \alpha_j \cdot \text{sign}\beta_l$, $\forall j \in \mathcal{A} - \{l\}$
3. Move in direction γ until some coefficient in $\mathcal{A}$ hits zero, i.e., define:
   $$\delta^* = \min\left\{\delta > 0 \,:\, \exists j \in \mathcal{A} \text{ s.t. } \beta_j + \gamma_j \delta = 0\right\}$$
   (we know that $\delta^* \leq |\beta_l|$)
4. Set $\tilde{\beta} = \beta + \delta^* \gamma$

Then from (36) we get that $\tilde{\beta}' h(\mathbf{x}_i) = \beta' h(\mathbf{x}_i)$, $\forall i$ and from (38) we get that

$$\begin{aligned} \|\tilde{\beta}\|_1 &= \|\beta\|_1 - \sum_{j \in \mathcal{A}} [|\beta_j + \gamma_j \delta^*| - |\beta_j|] = \\ &= \|\beta\|_1 - \delta^* \cdot \left(1 - \sum_{j \in \mathcal{A} - l} \alpha_j \cdot \text{sign}\beta_j \text{sign}\beta_l\right) = \|\beta\|_1. \end{aligned} \tag{39}$$

So $\tilde{\beta}$ generates the same fit as β and has the same l_1 norm, therefore it is also an optimal solution, with at least one less non-zero coefficient (from the definition of δ^*).

We can obviously apply this process repeatedly until we get a solution with at most n non-zero coefficients. ■

This theorem has the following immediate implication:

Corollary 4 *If there is no set of more than n dictionary functions which obeys the equalities (36,38) on the training data, then* any *solution of (33) has at most n non-zero coefficients.*

This corollary implies, for example, that if the basis functions come from a "continuous non-redundant" distribution (which means that any equality would hold with probability 0) then with probability 1 any solution of (33) has at most n non-zero coefficients.

Theorem 5 *Assume that there is no set of more than n dictionary functions which obeys the equalities (36,38) on the training data. In addition assume:*

1. *The loss function C is strictly convex (squared error loss, C_l and C_e obviously qualify),*

2. *No set of dictionary functions of size $\leq n$ is linearly dependent on the training data.*

Then the problem (33) has a unique solution.

Proof The previous corollary tells us that any solution has at most n non-zero coefficients. Now assume β_1, β_2 are both solutions of (33). From strict convexity of the loss we get that

$$h(X)'\beta_1 = h(X)'\beta_2 = h(X)'(\alpha\beta_1 + (1-\alpha)\beta_2)\ ,\ \ \forall 0 \leq \alpha \leq 1; \tag{40}$$

and from convexity of the l_1 norm we get

$$\|\alpha\beta_1 + (1-\alpha)\beta_2\|_1 \leq \|\beta_1\|_1 = \|\beta_2\|_1 = c. \tag{41}$$

So $(\alpha\beta_1 + (1-\alpha)\beta_2)$ must also be a solution. Thus, the total number of variables with non-zero coefficients in either β_1 or β_2 cannot be bigger than n, since then $(\alpha\beta_1 + (1-\alpha)\beta_2)$, would have $> n$ non-zero coefficients for almost all values of α, contradicting Corollary 4. Thus, by ignoring all coefficients which are 0 in both β_1 and β_2 we get that both β_1 and β_2 can be represented in the same $n-dimensional$ (maximum) sub-space of $\mathbb{R}^J$. Which leads to a contradiction between (40) and assumption 2. ■

Corollary 6 *Consider a sequence $\{\frac{\hat{\beta}(c)}{c} : 0 \leq c \leq \infty\}$ of normalized solutions to the problem (33). Assume that all these solutions have at most n non-zero coefficients. Then any limit point of the sequence has at most n non-zero coefficients.*

Proof This is a trivial consequence of convergence. Assume by contradiction β^* is a convergence point with more than n non-zero coefficients. Let $k = \arg\min_j\{|\beta^*_j| : \beta^*_j \neq 0\}$. Then for *any* vector $\tilde{\beta}$ with at most n non-zero coefficients we know that $\|\tilde{\beta} - \beta^*\| \geq |\beta^*_j| > 0$ so we get a contradiction to convergence. ■

B.2 Uniqueness of Limiting Solution in Theorem 3 when Margin Maximizing Separator is not Unique

Recall, that we are interested in convergence points of the normalized regularized solutions $\frac{\hat{\beta}^{(p)}(c)}{c}$. Theorem 3 proves that any such convergence point corresponds to an l_p-margin maximizing separating hyper-plane. We now extend it to the case that this first-order separator is not unique, by extending the result to consider the second smallest margin as a "tie breaker". We show that any convergence point maximizes the second smallest margin among all models with maximal minimal margin. If there are also ties in the second smallest margin, then any limit point maximizes the third smallest margin among all models which still remain, and so on. It should be noted that the minimal margin is typically not attained by one observation only in margin maximizing models. In case of ties in the smallest margins our reference to "smallest", "second smallest" etc. implies arbitrary tie-breaking (i.e., our decision on which one of the tied margins is considered smallest, and which one second smallest is of no consequence).

Theorem 7 *Assume that the data is separable and that the margin-maximizing separating hyper-plane, as defined in (4) is not unique. Then any convergence point of* $\frac{\hat{\beta}^{(p)}(c)}{c}$ *will correspond to a margin-maximizing separating hyper-plane which also maximizes the second smallest margin.*

Proof The proof is essentially the same as that of Theorem 3. We outline it below.

From Theorem 3 we know that we only need to consider margin-maximizing models as limit points. Thus let β_1, β_2 be two margin maximizing models with l_p norm 1, but let β_1 have a bigger *second smallest* margin. Assume that β_1 attains its smallest margin on observation i_1 and β_2 attains the same smallest margin on observation i_2. Now define

$$m_1 = \min_{i \neq i_1} y_i h(\mathbf{x}_i)'\beta_1 > \min_{i \neq i_2} y_i h(\mathbf{x}_i)'\beta_2 = m_2.$$

Then we have that Lemma 4 of Theorem 3 holds for β_1 and β_2 (the proof is exactly the same, except that we ignore the smallest margin observation for each model, since these always contribute the same amount to the combined loss).

Let β^* be a convergence point. We know β^* maximizes the margin from Theorem 3. Now assume $\tilde{\beta}$ also maximizes the margin but has bigger second-smallest margin than β^*. Then we can proceed exactly as the proof of Theorem 3, considering only $n-1$ observations for each model and using our modified Lemma 4, to conclude that β^* cannot be a convergence point (again note that the smallest margin observation always contributes the same to the loss of both models). ■

In the case that the two smallest margins still do not define a unique solution, we can continue up the list of margins, applying this result recursively. The conclusion is that the limit of the normalized, l_p -regularized models "maximizes the margins", and not just the minimal margin. The only case when this convergence point is not unique is, therefore, the case that the whole order statistic of the optimal separator is not unique. It is an interesting research question to investigate under which conditions this scenario is possible.

References

C. L. Blake and C. J. Merz. Repository of machine learning databases. [http://www.ics.uci.edu/ mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science., 1998.

L. Breiman. Prediction games and arcing algorithms. *Neural Computation*, 11(7):1493–1517, 1999.

M. Collins, R. E. Schapire, and Y. Singer. Logistic regression, adaboost and bregman distances. In *Computational Learing Theory*, pages 158–169, 2000.

D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard. Wavelet shrinkage: Asymptopia? *J. R. Statist. Soc. B.*, 57(2):301–337, 1995.

B. Efron, T. Hastie, I. M. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2), 2004.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995.

J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 2001.

J. H. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28:337–407, 2000.

T. Hastie, T. Tibshirani, and J. H. Friedman. *Elements of Statistical Learning*. Springer-Verlag, New York, 2001.

O. Mangasarian. Arbitrary-norm separating plane. *Operations Research Letters*, 24(1–2):15–23, 1999.

L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent. In *Neural Information Processing Systems*, volume 12, 1999.

G. Rätsch, S. Mika, and M. K. Warmuth. On the convergence of leveraging. NeuroCOLT2 Technical Report 98, Royal Holloway College, London, 2001a.

G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. *Machine Learning*, 42(3): 287–320, March 2001b.

G. Rätsch and M. W. Warmuth. Efficient margin maximization with boosting. submitted to JMLR, December 2002.

S. Rosset and E. Segal. Boosting density estimation. NIPS-02, 2003.

R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. *Annals of Statistics*, 26:1651–1686, 1998.

T. Zhang. Sequential greedy approximation for certain convex optimization problems. *IEEE Transactions on Information Theory*, 49, 2003.

T. Zhang and B. Yu. Boosting with early stopping: Convergence and results. Techincal report, Dept. of Statistics, Univ. of California, Berkeley, 2003.