Boosting with the L2 Loss: Regression and Classification

Bin Yu

`www.stat.berkeley.edu/ binyu/publications.html`

joint work with Peter Bühlmann (ETH, Zurich)

1. What is Boosting?

Boosting is general technique to potentially improve a given prediction or classification scheme, especially useful for large data set problems.

References: Schapire (1989), Freund (1990); Freund and Schapire (1995)...

An improvement in the range of 30 to 50% in classification problems.

Computer does the work!

Why are they suited for large data set problems?

For a large data set,

- it is easy to get a sensible procedure to start with;
- it is hard to get a sensible model under which to optimize.

Boosting (cf. Freund & Schapire, 1996):

- starting with a *weak* classifier (learner);
- resampling data with weights w_i for data point i;
- re-application of the initial procedure to the new sample to get a new procedure f_j;
- iteration;
- at the end of M iter, averaging and taking sign to get the classifier F_M.

Let err_j the classification error at iter. j, for $c_j = log((1 - err_j)/err_j)$:

$$w_i(j) \propto w_i(j-1)\exp(c_j I_{\{y_i \neq f_j(x_i)\}}), \quad \sum w_i = 1.$$

Heavily misclassified examples getting large weights!

A sequential algorithm...

After M iterations,

$$C_M(x) := I_{\{P_M^{Boost}(x)>1/2\}}, \quad P_M^{Boost}(x) = \frac{\exp\left(2F_M(x)\right)}{1+\exp(2F_M(x))}$$

where

$$F_M(x) = \sum_{j=1}^{M} c_j f_j(x),$$

$$f_j(x) = \text{estimate from ‘weak learner’ of } \frac{1}{2}\log\frac{P(x)}{1-P(x)}.$$

Bag-boosting (BY, 2000b): use the bagged estimator/learner as the weak learner in boosting.

State of affaris:

- big improvements when initial is tree procedures like CART;
- resistance to overfitting for most data sets tried;
- Freund and Schapire et al: VC bounds/distribution of margins on generalization errors.
- gradient-descent interpretation (Breiman, Mason et al, Friedman et al (FHT)) of boosting
  - $f_j(x)$ is a step of Newton method for minimizing a surrogate exp. loss function $J(F) = \mathbb{E}[\exp(-YF(X))]$;
  - in every Newton step the expectation in J is approximated using the current estimate of $P(x)$.

Boosting (or gradient descent) in terms of other loss functions (FHT, 2000; Friedman, 1999)

$$C(y,f) = \exp(-yf) \text{ with } y \in \{-1,1\}\text{: AdaBoost cost function,}$$
$$C(y,f) = \log(1+\exp(-2yf)) \text{ with } y \in \{-1,1\}\text{: LogitBoost cost function,}$$
$$C(y,f) = (y-f)^2/2 \text{ with } y \in \mathbb{R}\text{: } L_2\text{Boost cost function.} \tag{1}$$

Their population minimizers are

$$F(x) = \frac{1}{2}\log(\frac{\mathbb{P}[Y=1|X=x]}{\mathbb{P}[Y=-1|X=x]}) \text{ for AdaBoost and LogitBoost cost,}$$
$$F(x) = \mathbb{E}[Y|X=x] \text{ for } L_2\text{Boost cost.} \tag{2}$$

Generalization to L^2 regression (Friedman, 1999) – more tractable analytically

Under regression model:

$$Y_i = f(x_i) + \epsilon_i \tag{3}$$

where $f = (f(x_1), ..., f(x_n))^T$; $\epsilon = (\epsilon_1, ..., \epsilon_n)^T$ iid $N(0, \sigma^2)$.

BY (2001): Denote the weak learner by $\mathcal{S}Y$, then the boosting estimate in iteration m can be represented as:

$$\hat{F}_m = \mathcal{B}_m Y \;\; \text{where } \mathcal{B}_m = \sum_{j=0}^{m} \mathcal{S}(I - \mathcal{S})^j = (I - (I - \mathcal{S})^{m+1}).$$

No weighting!

$m = 1$ corresponds to Tukey's twicing.

Theorem (Linear case) Consider a linear, symmetric weak learner $\mathcal{S}$ with eigenvalues $\{\lambda_k;\ k = 1, \ldots, n\}$, satisfying $0 \le \lambda_k \le 1,\ k = 1, \ldots, n$ and eigenvectors building the columns of the orthonormal matrix U. Then, the bias, variance and averaged mean squared error for L_2Boost are

$$\begin{aligned}
bias &= \sum_{i=1}^{n} (\mathbb{E}[\hat{F}_m(x_i)] - f(x_i))^2 n^{-1} = f^T U \text{diag}((1-\lambda_k)^{2m+2}) U^T f n^- \\
variance &= \sum_{i=1}^{n} \text{Var}(\hat{F}_m(x_i)) n^{-1} = \sigma^2 \sum_{k=1}^{n} (1 - (1-\lambda_k)^{m+1})^2 n^{-1}, \\
MSE &= \text{bias} + \text{variance}.
\end{aligned}$$

Example 1: $\mathcal{S}$ is projection then L2Boosting has no effect.

Example 2: $\mathcal{S}$ = smoothing spline.

For any fixed n, $\lim_{m \to \infty} MSE = \sigma^2$: overfitting in the boosting limit.

Denote

$$\mathcal{G}^{(p)} = \{f : \int [f^{(p)}(x)]^2 dx < \infty\}.$$

and $\mathcal{S}Y = g_p(\lambda)$ is the smoothing spline solution to the penalized Least Squares problem

$$g_p(\lambda) = \text{argmin}_{g \in \mathcal{G}^{(p)}} \frac{1}{n} \sum_i [Y_i - g(X_i)]^2 + \lambda \int [f^{(p)}(x)]^2 dx$$

Theorem (optimality of L2Boost for smoothing splines) Suppose $\mathcal{S}$ is a smoothing spline linear learner $g_p(\lambda_0)$ of degree p corresponding to a fixed smoothing parameter λ_0. If $\nu \geq p$, then there is an $m = m(n) = O(n^{2p/(2\nu+1)}) \to \infty$ such that $\hat{F}_{m(n)}$ achieves the optimal minimax rate $n^{-2\nu/(2\nu+1)}$ of the smoother function class in terms of MSE.

Remarks:

Boosting smoothing splines is optimal for a given smoothness class and it adapts to any arbirtrary higher order smoothness.

Gu (1987) analyzes twicing (m=1) and shows that twicing can adapt to a higher order smoothness $\nu < 2p$.

For cubic smoothing spline with $\nu = p = 2$, the optimal rate $n^{-4/5}$ is achieved by $m = O(n^{4/5})$. If the underlying smoothness is $3 > 2$, then the boosted cubic smoothing spline can achieve the optimal rate $n^{-6/7}$for the smoother class.

A simulation example:

$$\begin{aligned} & f(x) = 0.8x + sin(6x), x \in \mathbb{R}^1, \\ & x_1, \ldots, x_n \text{ i.i.d. realizations from } \mathcal{N}(0,1),\ n = 100, \\ & \varepsilon_i \sim \mathcal{N}(0, \sigma^2),\ \sigma^2 = 2. \end{aligned} \tag{4}$$

$\mathcal{S}$ is a cubic smoothing spline weak learner.

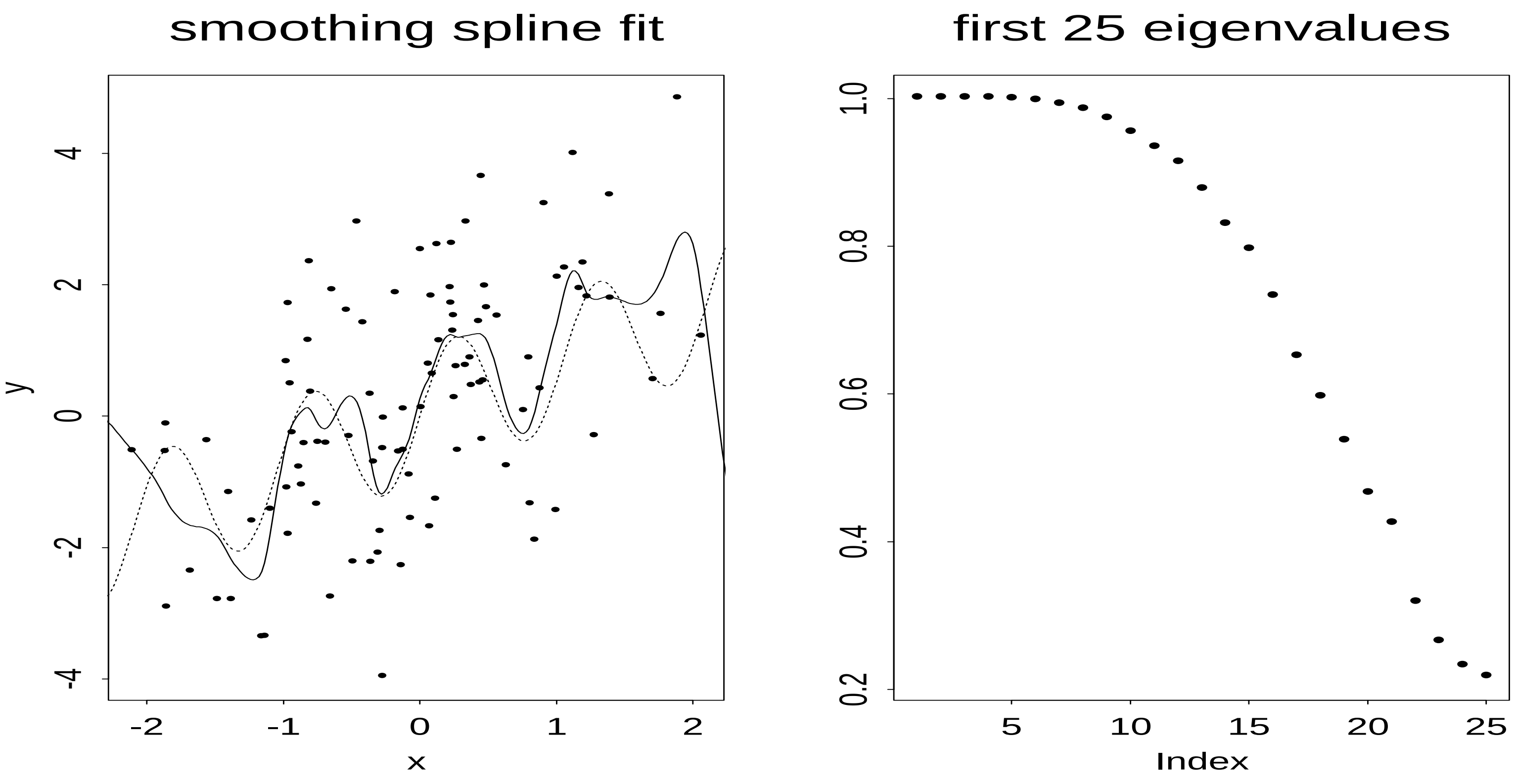
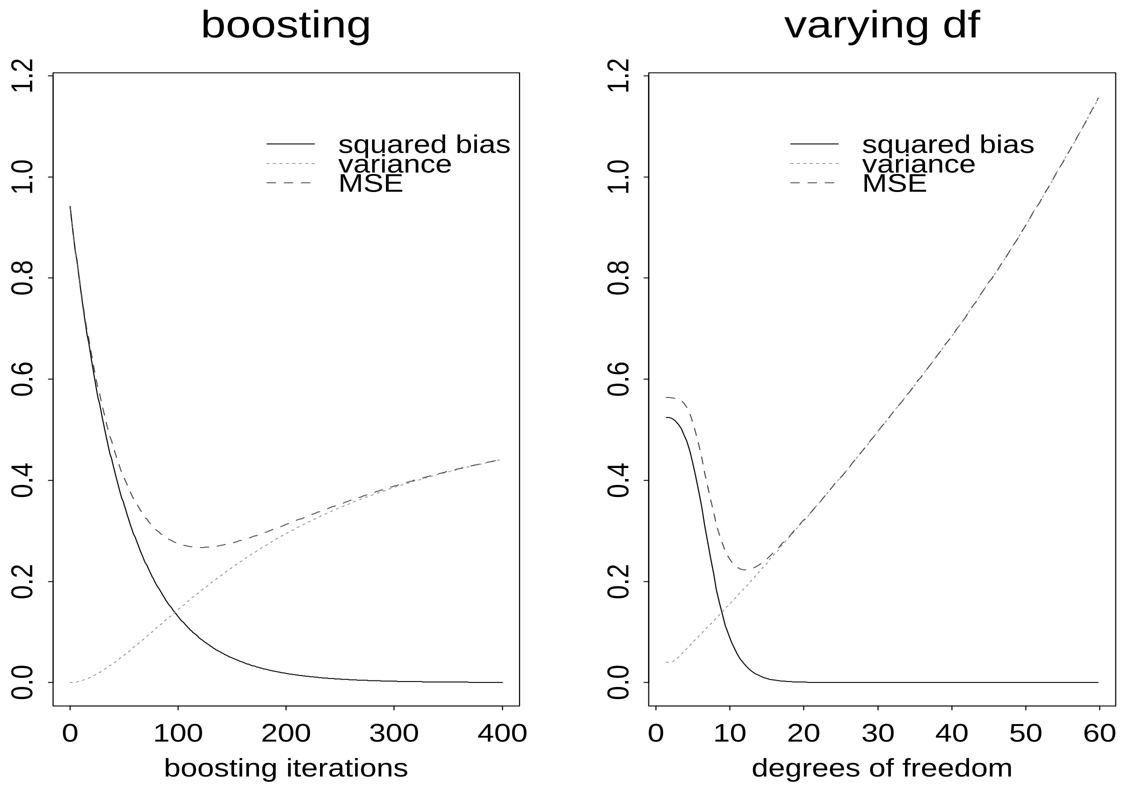


Figure 1: Left: realization of model (3) with (4) (dots), cubic smoothing spline fit (solid line) and true function $f(\cdot)$ (dotted line). Right: first ordered 25 eigenvalues of cubic smoothing spline operator $\mathcal{S}$ (with 20 degrees of freedom).

For boosting, iteration m is the "smoothing parameter".

Weak learner: a (shrinked) cubic smoothing spline with 20 degrees of freedom.



The boosting is a lot flatter or doesn't go up as much after the optimal point; the var/complexity term is bounded – hence good resistance to overfit!

A real data set: Ozone

L2Boost with comp-wise cubic smoothing splines	17.495 (5)
L2Boost with comp-wise stumps	20.957 (26)
MARS (in S-Plus Lib(mda))	18.092
Linear modeling (in S-Plus)	20.799

Table 1: Estimated test set MSE’s. Optimal number of boosts is given in parentheses.

2. Boosting in 2-Class Problems

0-1 loss is a different creature from $L2$ loss...

2.1 Expanding smoothed 0-1 loss

Given new $(Y, X) \in \{-1, 1\} \times \mathbb{R}^p$, indep. of training set, the misclassification rate or 0-1 loss for $\hat{F}_m$:

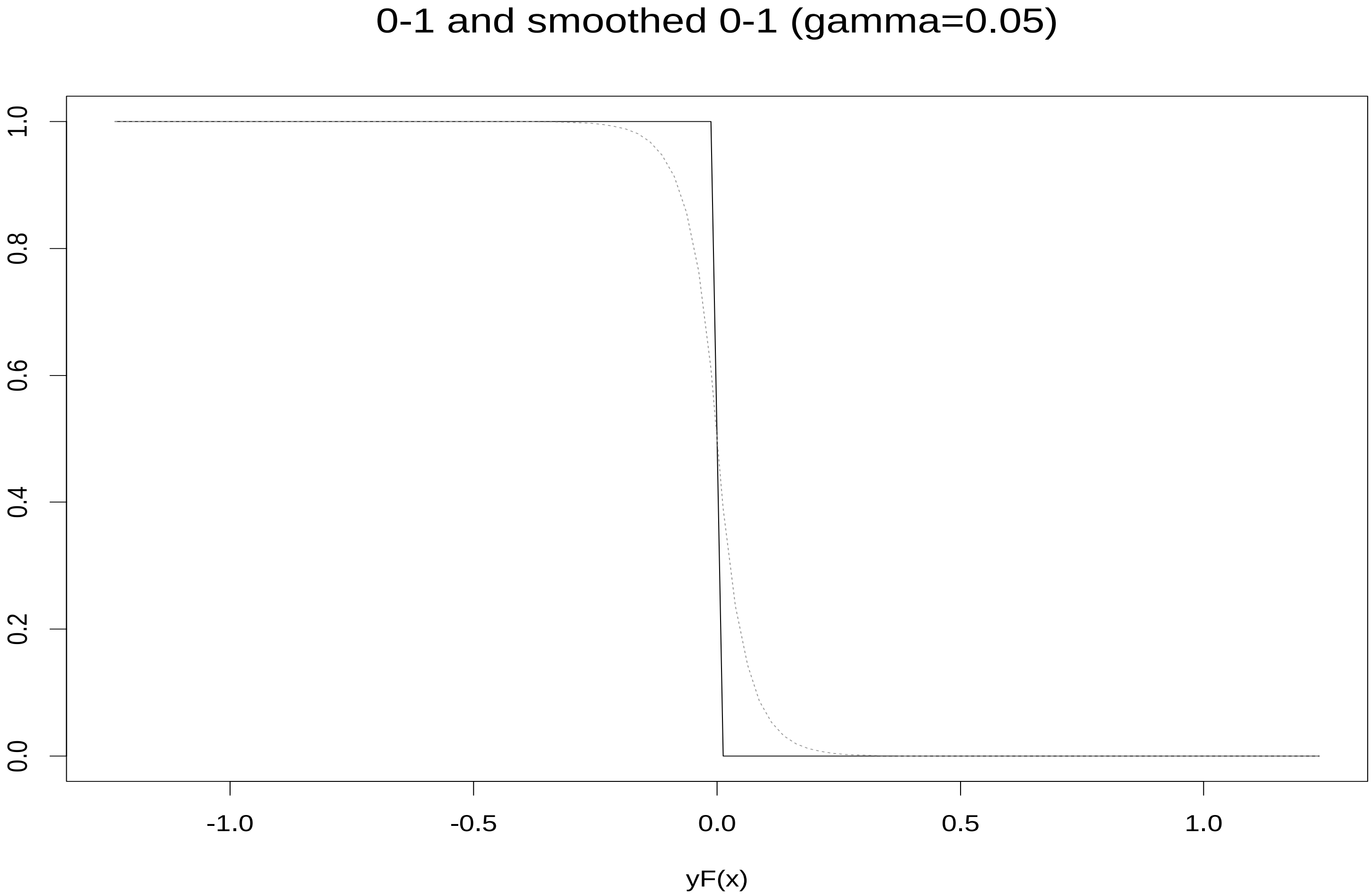
$$\mathbb{P}[Y\hat{F}_m(X) < 0] = \mathbb{E}[\mathbf{1}_{[Y\hat{F}_m(X)<0]}].$$

0-1 loss can be approximated by a smoothed version:

$$|\mathbb{P}[Y\hat{F}_m(X) < 0] - \mathbb{E}[C_\gamma(Y\hat{F}_m(X))]| = O(\gamma \log(\gamma^{-1})\ (\gamma \to 0),$$

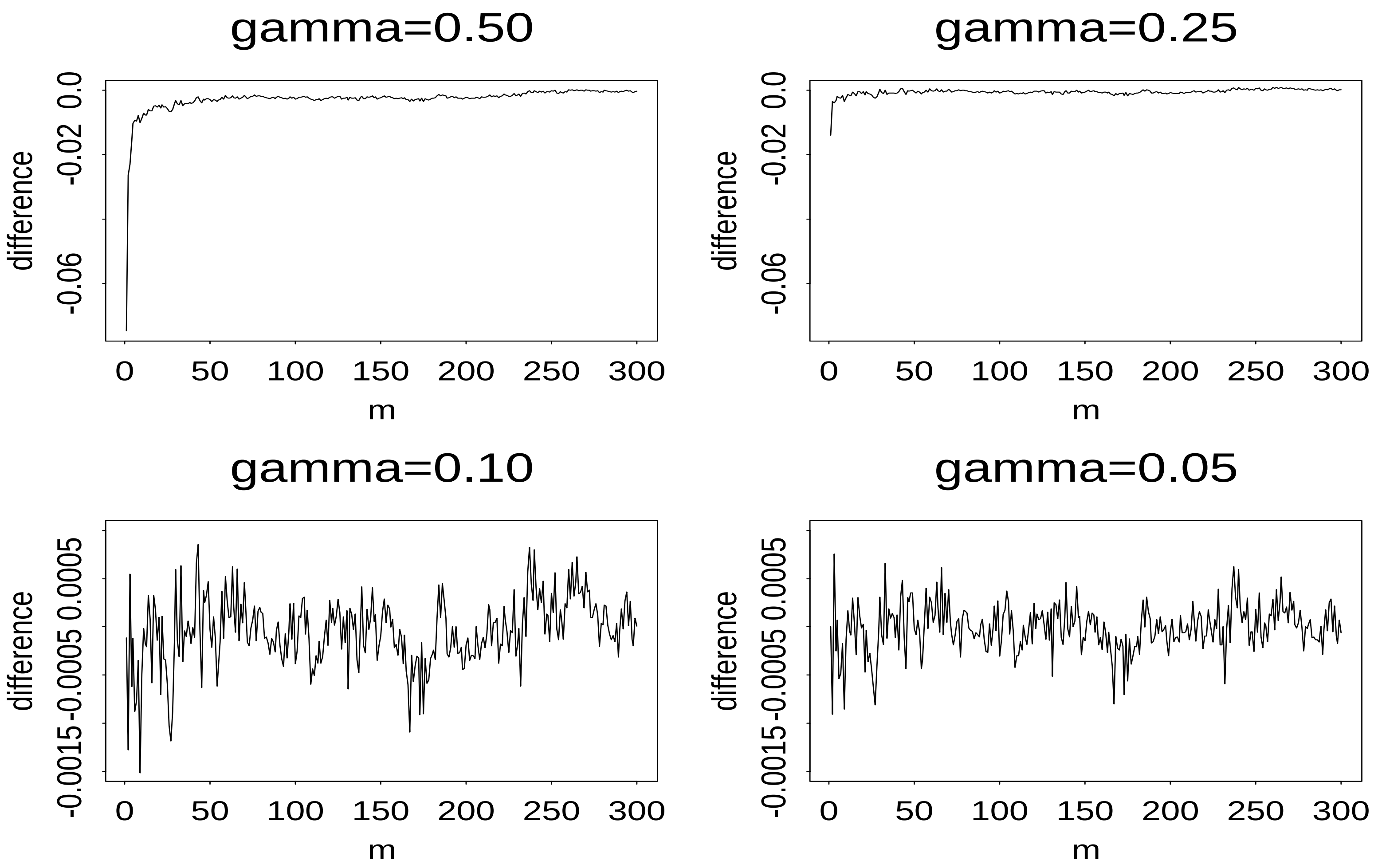
where

$$C_\gamma(z) = (1 - \exp(z/\gamma)/2)\mathbf{1}_{[z<0]} + \exp(-z/\gamma)/2\mathbf{1}_{[z\geq 0]},\ \gamma > 0.$$



0-1 and smoothed 0-1 (gamma=0.05)

Test set difference of $\mathbb{P}[Y\hat{F}_m(X) < 0] - (\mathbb{E}[C_\gamma(Y\hat{F}_m(X))]$ with LogitBoosting stumps and breast cancer data:



Expanding $C_\gamma(\cdot)$ around $Z^* = YF(X)$, i.e. the margin with the true $F(\cdot)$, and for

$$C_\gamma^{(k)}(z) = \frac{1}{\gamma^k}\exp(\frac{-|z|}{\gamma})(-\mathbf{1}_{[z<0]} + (-1)^k\mathbf{1}_{[z\geq 0]}) : \qquad (5)$$

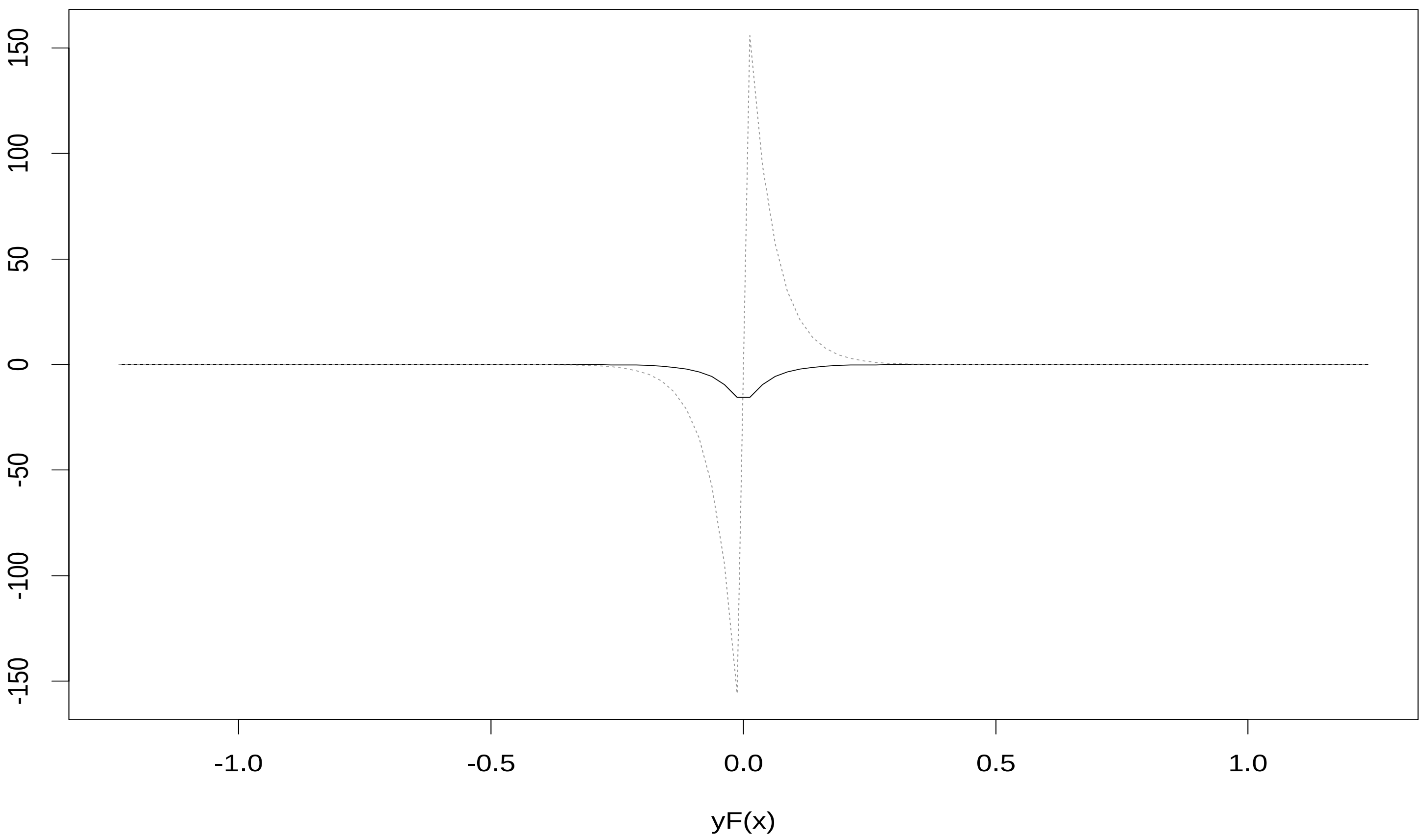
$$\begin{aligned}\mathbb{E}[C_\gamma(Z)] &= \mathbb{E}[C_\gamma(Z^*)] + \sum_{k=1}^{\infty}\frac{1}{k!}\mathbb{E}[C_\gamma^{(k)}(Z^*)(Z-Z^*)^k] \\ &= \text{(smoothed) Bayes risk} + \ tapered_1 \text{ bias } + \ tapered_2 \text{ MSE} \\ &+ \quad \text{tapered interactions between the bias and moments of random term,}\end{aligned}$$

where random term is defined as $\hat{F}_m - \mathbb{E}\hat{F}_m$, and each term has a different tapering function.

Moreover

$$tapered_2 MSE = \ tapered_2 \text{ bias}^2 + \ tapered_2 \text{ Variance} .$$

first two tapering functions



Remarks:

1. Tapering functions $C_\gamma^{(k)}/k!$ add much robustness to overfitting on top of the sub-linear complexity increase from L2 story – mostly only the classification outcome around the class boundary matters.
2. Bias matters more in classification. Boosting reduces the bias (BY, 2001, Breiman, 2000) hence behaves better than bagging in classification.
3. Complexity is not straightforward since the bias term interacts intimiately with the random term (whose variance has been the conventional complexity term) so no simple breakdowns into the bias and variance terms in an additive fashion, unless the first two terms give a good approximation. (In the latter case, the bias term has two parts involving two tapering functions.)

Acceleration of F and classification noise

If the true $F(\cdot)$ moves away quickly from the classification boundary $\{x; F(x) = 0\}$, the relevant tapering weights $C_{\gamma}^{(k)}(yF(x)$ decay very fast.

This can be measured with grad$(F(x))|_{x=0}$, the gradient of F at zero.

$F(\cdot)$: a large acceleration if its gradient is large.

Mammen and Tsybakov (1999): under local constraints on $F(\cdot)$ near class boundary, the minimax rate of convergence for the generalization error to approach the Bayes risk can be faster than the parametric rate $n^{-1/2}$!

Key: treating the problem as estimating the Bayes decision set, not function estimation. Hence their minimax optimal classifier is not a plug-in, but the minimizer of the empirical 0-1 risk over a set class of regulated size. Computationally difficult...

2.2 $L2Boost$ in Classification

Boosting algorithms use "nice" bounds on the empirical risk to minimize and stick with the "plug-in" philosophy.

L^2 is a very good bound if we estimate $E(Y = 1|x)$!

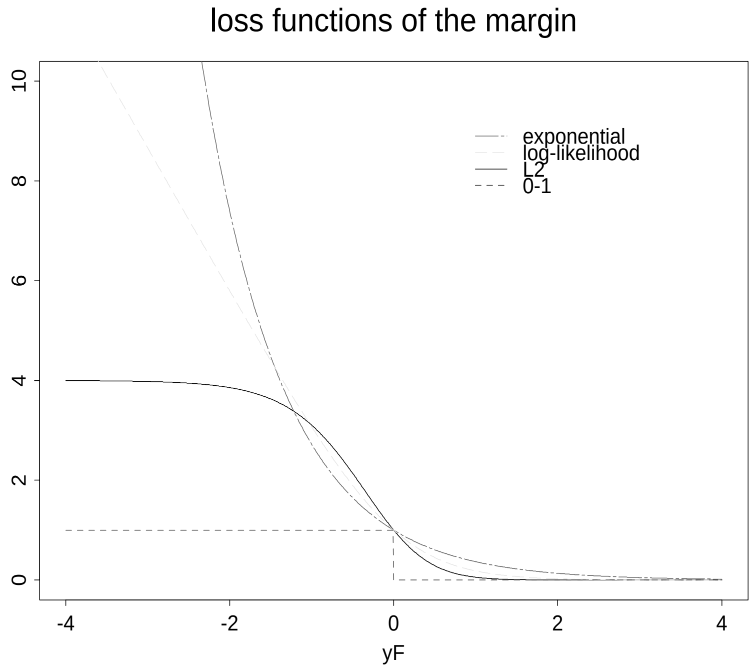


Figure 2: Various loss functions of the margin $YF(X)$.

Estimating $\mathbb{E}[Y|X = x] = 2p(x) - 1$ in the classification can be seen as estimating the regression function in a hetereoscadestic model:

$$Y_i = 2p(x_i) - 1 + \epsilon_i$$

where ϵ_i are independent, mean zero, but with variance $4p(x_i)(1 - p(x_i))$.

Similarly as in $L2$ regression...

Optimal rates for smoothing splines hold, which are known to be the optimal rates to approach Bayes risk if global smoothness classes are assumed (cf. Marron, 1982).

FHT: L2Boost has a slightly worse performance than LogitBoost...

Smoothing splines are appropriate as weak learners if the predictors are continuous.

Some Experimental Results

$L2WCBoost$: for each L2Boost iteration, impose the bound of $[-1, 1]$.

Comparing cubic spline with stumps as a weak learner:

dataset	n	p	learner	L_2WCBoost	LogitBoost
Breast cancer	699	9	stumps	0.040 (275)	0.039 (27)
Breast cancer	699	9	cubic smoothing spline	0.036 (126)	
Sonar	210	60	stumps	0.190 (335)	0.158 (228)
Sonar	210	60	cubic smoothing spline	0.168 (47)	0.158 (80)

Table 2: Estimated test set errors for L_2WCBoost and LogitBoost. Optimal number of boosts is given in parentheses.

Open Problems:

1. Analysis of L2Boost for trees (on-going research).

2. Can boosting achieve the rates in Mammen and Tsybakov (1999) for the locally constrained classes?

3. Breiman's conjecture (2000):

Adaboost is an equalizer of margins so weighting doesn't play much a role.

Our L^2 boosting has no weighting hence it supports this conjecture.