Boosting Methodology for Regression Problems

Greg Ridgeway,
David Madigan,
Thomas Richardson

Department of Statistics
University of Washington

Outline

- Boosting algorithms
- Recent developments
- Regression as a classification problem
- The boosted naïve Bayes regression model
- Performance results

Boosting algorithms for classification

1. Learn a classifier from the data
2. Upweight observations poorly predicted, downweight observations well predicted
3. Refit the model using the new weighting
4. After *T* iterations, have each model vote on the final prediction.

Recent developments

- Friedman, Hastie, Tibshirani demonstrated that AdaBoost is a greedy, stepwise procedure to fit an additive logistic regression model.

$$J(F) = E\left(e^{yF(x)}\right)$$

$$J(F + cf) = E\left(e^{y(F(x)+cf(x))}\right)$$

A regression analogy

- With a current regressor, F(x), modify it in order to minimize

$$J(F+f)=E\left(y-\left(F(x)+f(x)\right)\right)^2$$

$$\Rightarrow \hat{f}(x)=E\left(y-F(x)\,|\,x\right)$$

Casting regression as classification

X_1	X_2	Y
0.6	0.4	0.3
0.8	0.5	0.9

Regression

$h(\underline{X}) \to Y$

Classification

$h(\underline{X}, S) \to Y^*$

	X_1	X_2	Y	S	$Y^* = I(S \geq Y)$
Obs. 1	0.6	0.4	0.3	0.00	0
	0.6	0.4	0.3	0.01	0
	0.6	0.4	0.3	¶	0
	0.6	0.4	0.3	0.29	0
	0.6	0.4	0.3	0.30	1
	0.6	0.4	0.3	0.31	1
	0.6	0.4	0.3	¶	1
	0.6	0.4	0.3	0.99	1
	0.6	0.4	0.3	1.00	1
Obs. 2	0.8	0.5	0.9	0.00	0
	0.8	0.5	0.9	0.01	0
	0.8	0.5	0.9	0.02	0
	0.8	0.5	0.9	¶	0
	0.8	0.5	0.9	0.89	0
	0.8	0.5	0.9	0.90	1
	0.8	0.5	0.9	0.91	1
	0.8	0.5	0.9	¶	1
	0.8	0.5	0.9	0.99	1
	0.8	0.5	0.9	1.00	1

Prediction

If $h(\underline{X},S)$ has the form $P(Y^*=1|\underline{X},S)$, we will predict Y as

$$\hat{Y} = \inf_s \left\{ s : \hat{P}(Y^* = 1 \mid \underline{X}, S = s) \geq \tfrac{1}{2} \right\}$$

The *inf* always exists by the construction of S and if the probability function is continuous in s, then

$$\tfrac{1}{2} = \hat{P}(Y^* = 1 \mid \underline{X}, S = \hat{Y}) = \hat{P}(Y \geq \hat{Y} \mid \underline{X}, S = \hat{Y})$$

So the predicted Y is the median of the predictive density of Y.

Why cast as classification?

- A classifier merely needs to “pitch” itself on the correct side of ½ to be accurate
- Exposes regression problems to models proposed for classification
- We can directly apply boosting (AdaBoost)

Naïve Bayes model

The naïve Bayes assumption

$$P(Y^* = 1 \mid \underline{X}, S = \hat{Y}) \propto P(Y^* = 1) P_{S|Y^*=1}(\hat{Y} \mid Y^* = 1) \prod_{j=1}^{d} P(X_j \mid Y^* = 1)$$

The prediction rule is an additive model for a transformation of Y

$$P(Y^* = 1 \mid \underline{X}, S = \hat{Y}) = \tfrac{1}{2}$$

$$\log \frac{P_{S|Y^*=0}(\hat{Y} \mid Y^* = 0)}{P_{S|Y^*=1}(\hat{Y} \mid Y^* = 1)} = \log \frac{P(Y^* = 1)}{P(Y^* = 0)} + \sum_{j=1}^{d} \log \frac{P(X_j \mid Y^* = 1)}{P(X_j \mid Y^* = 0)}$$

$$\Rightarrow l(\hat{Y}) = f_0 + \sum_{j=1}^{d} f_j(X_j)$$

Estimation with infinite datasets

- For finite datasets, naïve Bayes estimation is simple
- For example, if *Y* ϵ [0,1] estimation turns into simple limits

$$\hat{P}(Y^* = 1) = \lim_{m \to \infty} \frac{1}{N \times m} \sum_{i=1}^{N} \sum_{j=1}^{m} I(Y_i^*(S_j) = 1)$$

$$= 1 - \bar{y}$$

- Not so simple when *Y* ϵ ⍰

Weight functions

- When *Y* ϵ ⍰ we assign weight functions such that

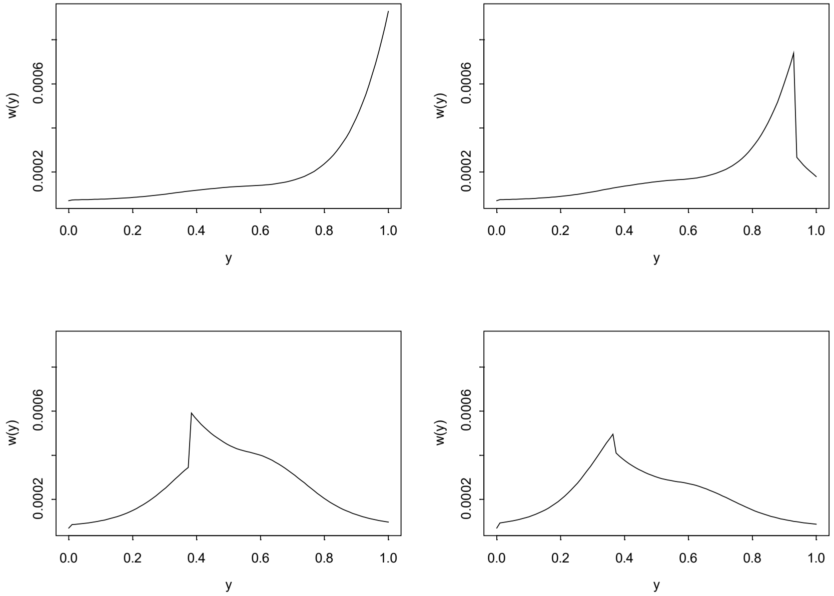
$$w_i(s) \geq 0 \text{ and } \sum_{i=1}^{N} \int_{-\infty}^{\infty} w_i(s)ds = 1$$

- Initially we set

$$\int_{-\infty}^{\infty} w_i(s)ds = \tfrac{1}{N}$$

Empirical weight functions

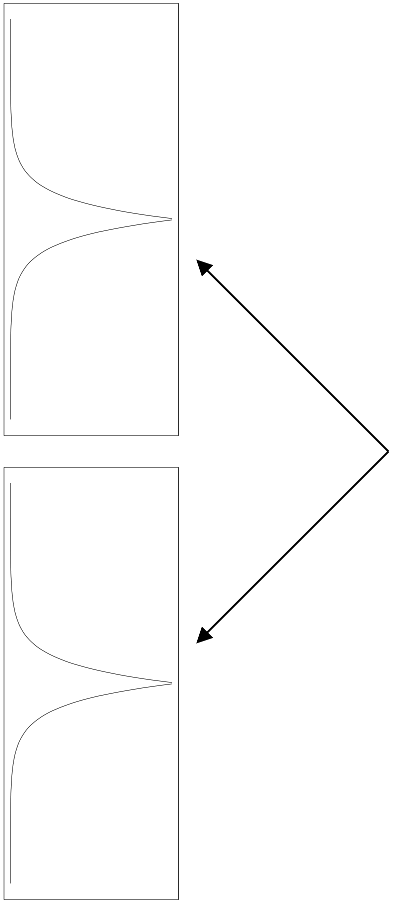
- We constructed an approximation to D^* for some datasets,
- applied AdaBoost,
- observed Laplace-like weight functions peaked around y_i



$$w_i(s) \propto \exp(-|s - y_i| / \sigma)$$

Weighting observations

	X_1	X_2	Y	S	$Y^* = I(S \geq Y)$
Obs. 1	0.6	0.4	0.3	0.00	0
	0.6	0.4	0.3	0.01	0
	0.6	0.4	0.3	¶	0
	0.6	0.4	0.3	0.29	0
	0.6	0.4	0.3	0.30	1
	0.6	0.4	0.3	0.31	1
	0.6	0.4	0.3	¶	1
	0.6	0.4	0.3	0.99	1
	0.6	0.4	0.3	1.00	1
Obs. 2	0.8	0.5	0.9	0.00	0
	0.8	0.5	0.9	0.01	0
	0.8	0.5	0.9	0.02	0
	0.8	0.5	0.9	¶	0
	0.8	0.5	0.9	0.89	0
	0.8	0.5	0.9	0.90	1
	0.8	0.5	0.9	0.91	1
	0.8	0.5	0.9	¶	1
	0.8	0.5	0.9	0.99	1
	0.8	0.5	0.9	1.00	1



Weight functions $w_i(s)$

Estimating a classifier

If we assume that the rows of *D** are independent then

$$L(\theta) = \prod_{i=1}^{N} \mathop{\pi}_{-\infty}^{\infty} P(y_i^*(s), s, \underline{x}_i \mid \theta)^{Nw_i(s)ds}$$

And further make the naïve Bayes assumption to factor the likelihood then

$$= \prod_{i=1}^{N} \mathop{\pi}_{-\infty}^{\infty} \left(P(y_i^*(s) \mid \theta) P(s \mid y_i^*(s), \theta) \prod_{j=1}^{d} P(x_{ji} \mid y_i^*(s), \theta) \right)^{Nw_i(s)ds}$$

The BNB.R algorithm

Initialize: $w_i(y)$ as a Laplace density function with mean y_i and scale σ.

For t = 1, 2, …, T

1. Using $w_i(s)$, estimate the components of the naïve Bayes regression model, $h_t(x)$.

2. $\varepsilon_t = \sum_{i=1}^{N} \left| \int_{y_i}^{h_t(x_i)} w_i(s)ds \right|$ and $\beta_t = \dfrac{\varepsilon_t}{1-\varepsilon_t}$

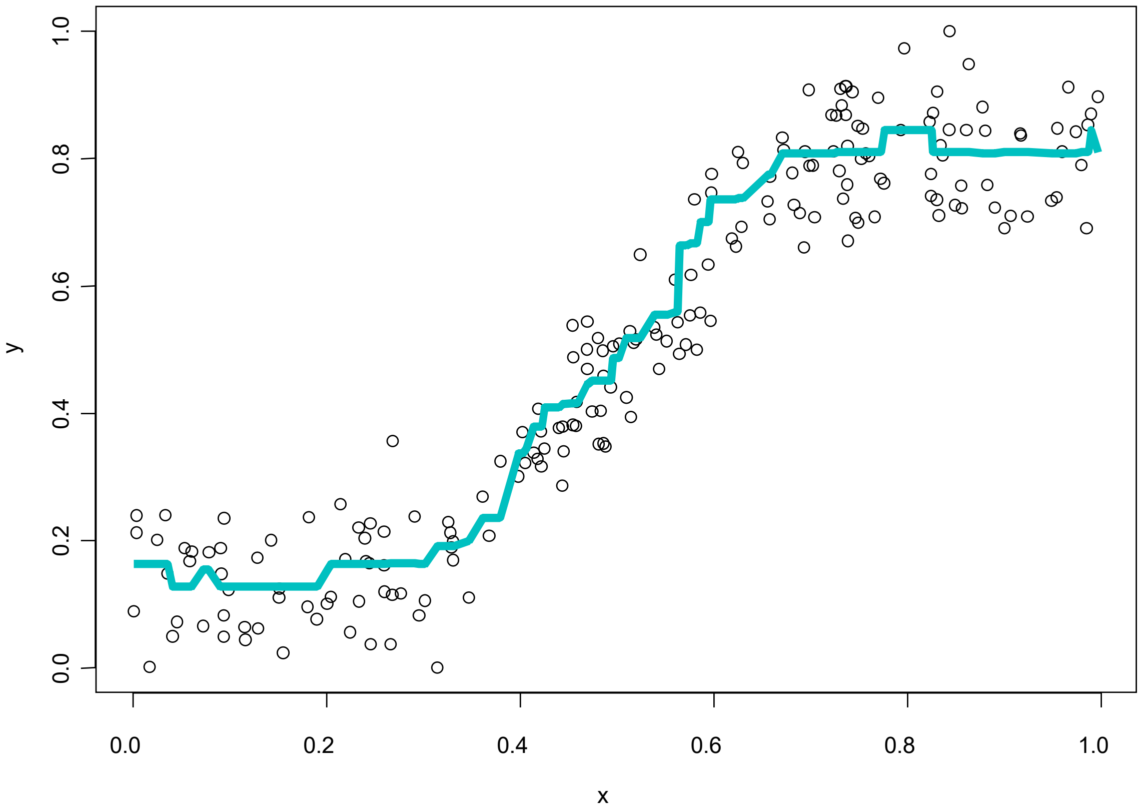
3. $w_i^{t+1}(s) = \begin{cases} w_i^t(s) \cdot \beta_t^{1-P(Y^*=1|X_i,s)} & s \le y_i \\ w_i^t(s) \cdot \beta_t^{P(Y^*=1|X_i,s)} & s > y_i \end{cases}$ and normalize

$$\hat{Y} = \inf_y \left\{ y : \sum_{t=1}^{T} \alpha_t \log \frac{P_S^t(y \mid Y^* = 0)}{P_S^t(y \mid Y^* = 1)} \le \sum_{t=1}^{T} \alpha_t \log \frac{P^t(Y^* = 1)}{P^t(Y^* = 0)} + \sum_{j=1}^{d} \sum_{t=1}^{T} \alpha_t \log \frac{P^t(X_j \mid Y^* = 1)}{P^t(X_j \mid Y^* = 0)} \right\}$$

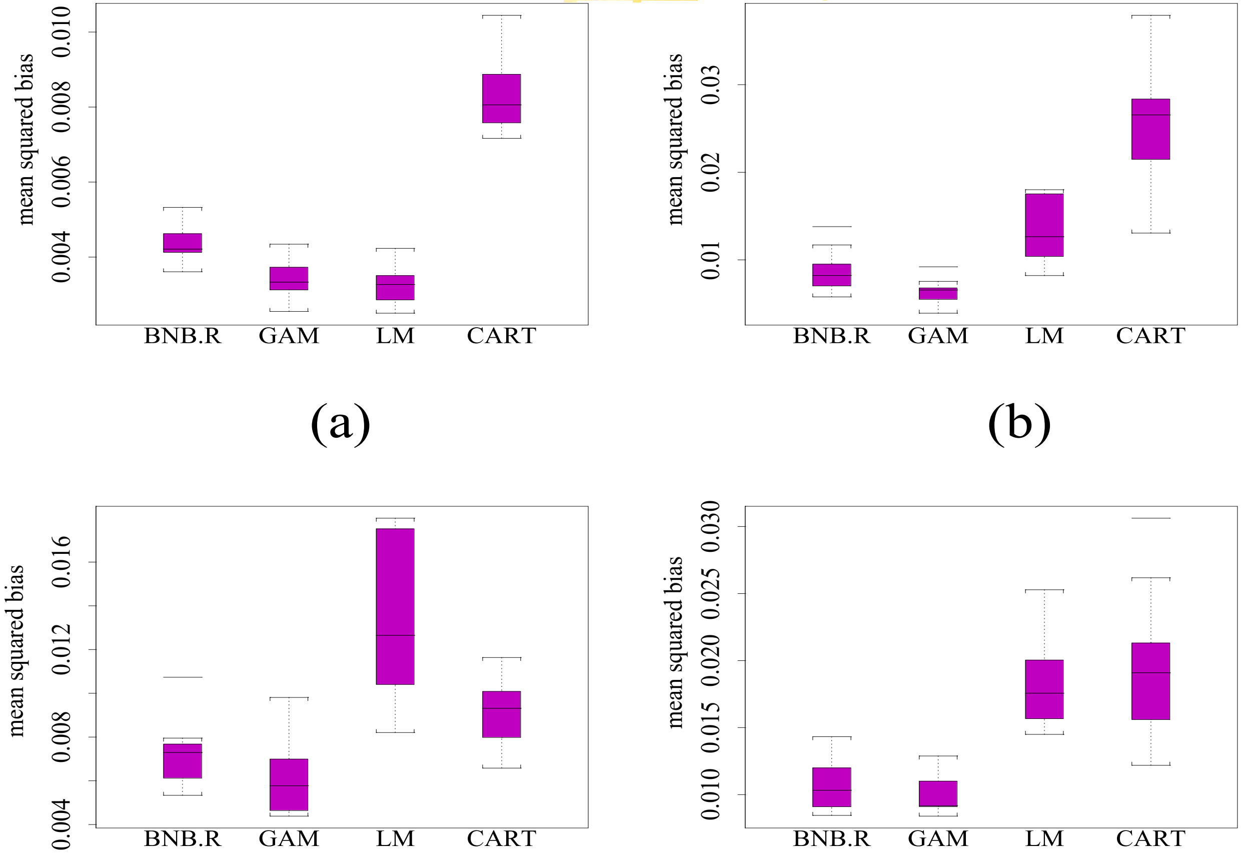
$$\text{where } \alpha_t = (\log \beta_t) / \sum_{t=1}^{T} \log \beta_t$$

Example

BNB.R on a linear threshold/saturation model



Performance results



(a) (b)

(c) (d)

(a) the plane (b) Friedman #1 (c) Friedman #2 and (d) Friedman #3

Conclusions

- Presents a “thought exercise” on using boosting for regression problems.
- Proposes a method for applying classifiers to regression problems.
- Derives estimators for the naïve Bayes regression model.
- Shows that BNB.R does surprisingly well given its unconventional derivation.