The State of Boosting

Greg Ridgeway
Department of Statistics
University of Washington

Work with David Madigan and Thomas Richardson
http://www.stat.washington.edu/greg

Outline

1. Origins of boosting
2. Boosting as an optimization problem
3. Generalized boosted models
4. Extension to survival models
5. Discussion

Origin of Boosting

Classification problems

$$\{\underline{X}, Y\}_i \, , \, i = 1,\ldots,n$$

$$Y \in \{0, 1\}$$

The task - construct a function,

$$h(\underline{X}) : \underline{X} \rightarrow \{0, 1\}$$

so that h minimizes misclassification error.

Combining multiple classifiers

Generally, combining several classifiers into one results in a more accurate classifier.

- Bagging (and adaptive bagging)
- Bumping
- Bayesian Model Averaging
- Bundling
- Boosting

Boosting

Equally weight the observations $(\underline{X},Y)_i$

For t in $1,\ldots,T$

- Using the weights, fit a classifier $h_t(\underline{X}) \rightarrow Y$
- Upweight the poorly predicted observations
- Downweight the well-predicted observations

Merge $h_1,\ldots,h_T$ to form the boosted classifier

AdaBoost's Performance

Freund & Schapire [1996]

- Leo Breiman - AdaBoost with trees is the "best off-the-shelf classifier in the world."
- Performs well with many base classifiers and in a variety of problem domains.
- AdaBoost is generally slow to overfit.
- Boosted naïve Bayes tied for first place in the 1997 KDD Cup. (Elkan [1997])
- Boosted naïve Bayes is a scalable, interpretable classifier (Ridgeway, *et al* [1998]).

Boosting as optimization

- Friedman, Hastie, Tibshirani [1998] - AdaBoost is an optimization method for finding a classifier.
- Let $y \in \{-1,1\}$, $F(x) \in (-\infty,\infty)$

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

Criterion

- $E(e^{-yF(x)})$ bounds the misclassification rate.

$$I(yF(x) < 0) < e^{-yF(x)}$$

- The minimizer of $E(e^{-yF(x)})$ coincides with the maximizer of the expected Bernoulli likelihood.

$$E\big(\ell(p(x), y)\big) = -E\log(1 + e^{-2yF(x)})$$

Optimization step

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

- Select f to minimize J…

$$F^{(t+1)}\leftarrow F^{(t)}+\tfrac{1}{2}\log\frac{E_w[I(y=1)\mid x]}{1-E_w[I(y=1)\mid x]}$$

$$w(x,y)=e^{-yF^{(t)}(x)}$$

LogitBoost

Friedman, Hastie, Tibshirani [1998]

- Logistic regression

$$y = \begin{cases} 1 & \text{with probability } p(x) \\ 0 & \text{with probability } 1 - p(x) \end{cases}$$

$$p(x) = \frac{1}{1 + e^{-F(x)}}$$

- Expected log-likelihood of a regressor, $F(x)$

$$\mathrm{E}\,\ell(F) = \mathrm{E}\left(yF(x) - \log(1 + e^{F(x)}) \mid x \right)$$

Newton steps

$$J(F+f) = E\left(y(F(x)+f(x)) - \log(1+e^{F(x)+f(x)}) \mid x\right)$$

- Iterate to optimize expected log-likelihood.

$$F^{(t+1)}(x) \leftarrow F^{(t)}(x) - \frac{\frac{\partial}{\partial f} J(F^{(t)}+f)\Big|_{f=0}}{\frac{\partial^2}{\partial f^2} J(F^{(t)}+f)\Big|_{f=0}}$$

LogitBoost, continued

- Newton steps for Bernoulli likelihood

$$F(x) \leftarrow F(x) + E_w\left(\frac{y - p(x)}{p(x)(1-p(x))} \middle| x \right)$$

$$w(x) = p(x)(1-p(x))$$

- In practice the $E_w(\bullet|x)$ can be any regressor - trees, smoothers, etc.
- Trees are adaptive and work well for high dimensional data.

Classification results

Friedman, Hastie, Tibshirani [1998]

	CART	AdaBoost	LogitBoost
Breast	4.5%	4.0%	2.9%
Ion	7.6%	6.8%	7.1%
Glass	40.0%	25.7%	26.6%
Sonar	59.6%	20.2%	20.2%
Waveform	36.4%	19.5%	20.6%

Generalized Boosted Models

G. Ridgeway, D. Madigan, T. Richardson [in progress]

- Exponential family

$$f_Y(y \mid \theta, \phi) = \exp\left(\frac{y\theta(\mu) - b(\theta(\mu))}{a(\phi)} + c(y, \phi) \right)$$

- Model the conditional mean

$$\mu_i = E(Y \mid x_i)$$

- Link the mean to the covariates

$$g(\mu_i) = F(x_i)$$

The GBM algorithm

With $b'(F^{(0)}) = \bar{y}$, for $t = 1,\ldots,T$ update F as

$$F^{(t+1)}(x) \leftarrow F^{(t)}(x) + \lambda_t E_w\left(\frac{y - b'(F^{(t)}(x))}{b''(F^{(t)}(x))}\middle| x\right)$$

$$w(x) = b''(F(x))$$

Proposition *The GBM algorithm is a Newton method, with learning schedule λ_t, for maximizing a canonical exponential family regression model.*

GBM…

- encompasses a large set of standard statistical models,
- allows for non-linear predictors (prediction bias-reduction),
- trees work well for high-dimensional feature extraction,
- can incorporate robust regression, and
- reduces variance through a “bagging” step.

Alternate paths

$$F^{(t+1)}(x) \leftarrow F^{(t)}(x) + \lambda E_w\left(z(y,x)\middle|x\right)$$

- Sub-sample a fraction of the data at each step when computing the expectation.
- “Robustify” the expectation.
- Trim observations with small weights.

Survival prediction

- Exponential survival model

$$t_i \sim \mathrm{Exp}\left(\lambda e^{F(x_i)}\right)$$

- Cox model

$$\mathrm{PL}(F) = \prod_{h=1}^{k} \frac{e^{F(x_h)}}{\sum_i I(t_i \geq t_h) e^{F(x_i)}}$$

Gradient ascent

- Consider maximizing the N dimensional function

$$\log \mathrm{PL}(F) = \sum_{i=1}^{N} \delta_i \left[F_i - \log \sum_j I(t_j \geq t_i) e^{F_j} \right]$$

- The gradient gives the direction of the largest (local) increase in log PL.

$$g_i = \frac{\partial}{\partial F_i} \log \mathrm{PL}(F)$$

Compress the gradient

- Optimization would have us modify F as

$$F_i^{(t+1)} = F_i^{(t)} + \rho g_i$$

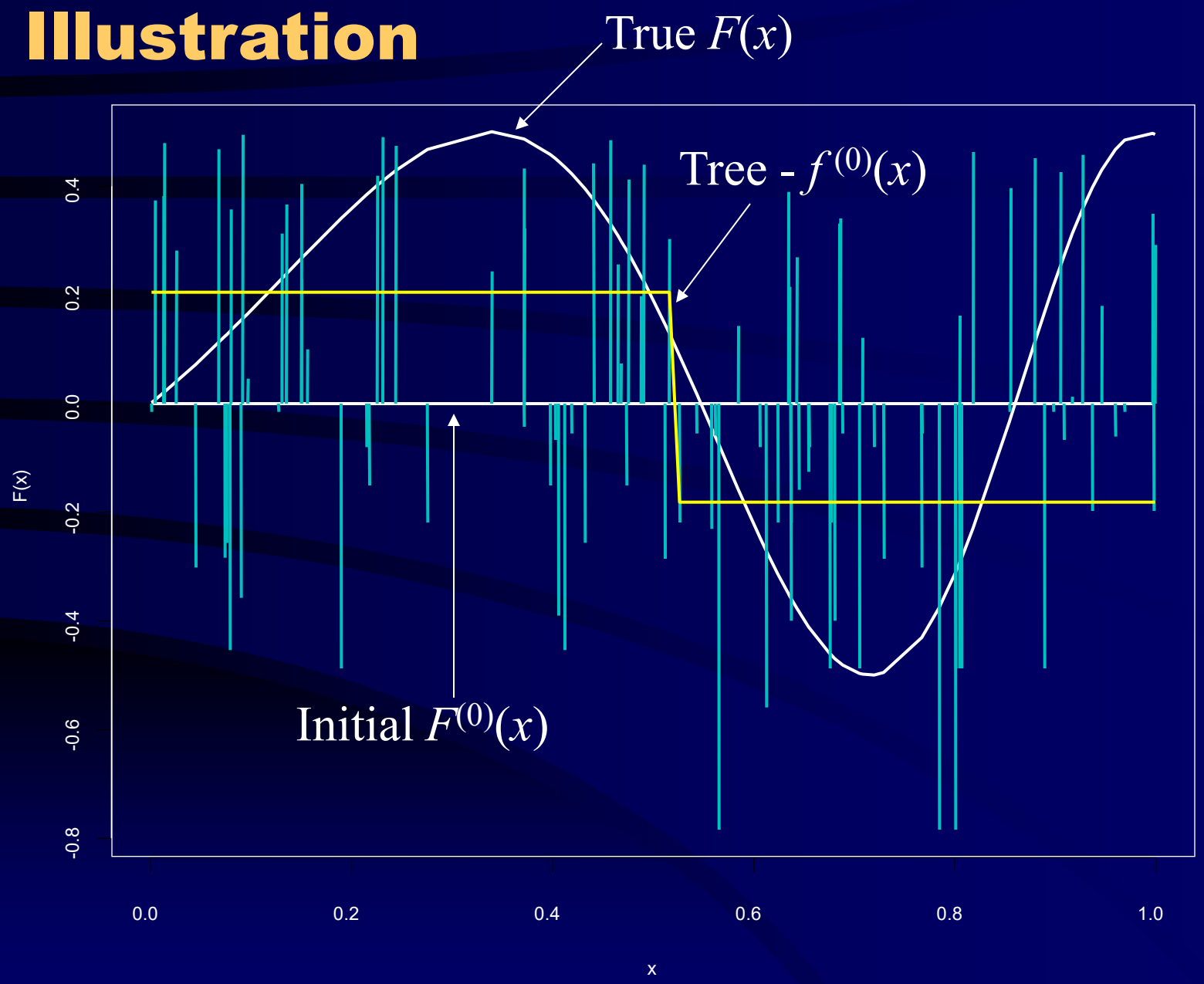
- Approximate the gradient using x.

$$\arg\min_f \sum_{i=1}^{N} \left(g_i - f^{(t)}(x_i) \right)^2$$

- Modify F as

$$F^{(t+1)}(x) = F^{(t)}(x) + \rho f^{(t)}(x)$$

Illustration



Computing the step size

$$F^{(t+1)}(x) = F^{(t)}(x) + \rho \cdot f^{(t)}(x)$$

- Choose ρ to maximize log PL.

$$\sum_{i=1}^{N} \delta_i \left[F(x_i) + \rho f(x_i) - \log \sum_j I(t_j \geq t_i) e^{F(x_j) + \rho f(x_j)} \right]$$

- This is just a linear Cox model!

Boosted Cox model

Initialize $F^{(0)}(x) = 0$ and for t in 1 to T

1. Compute the working response

$$g_i = \frac{\partial}{\partial F_i} \log \mathrm{PL}(F)$$

2. Predict g from the covariates, x.

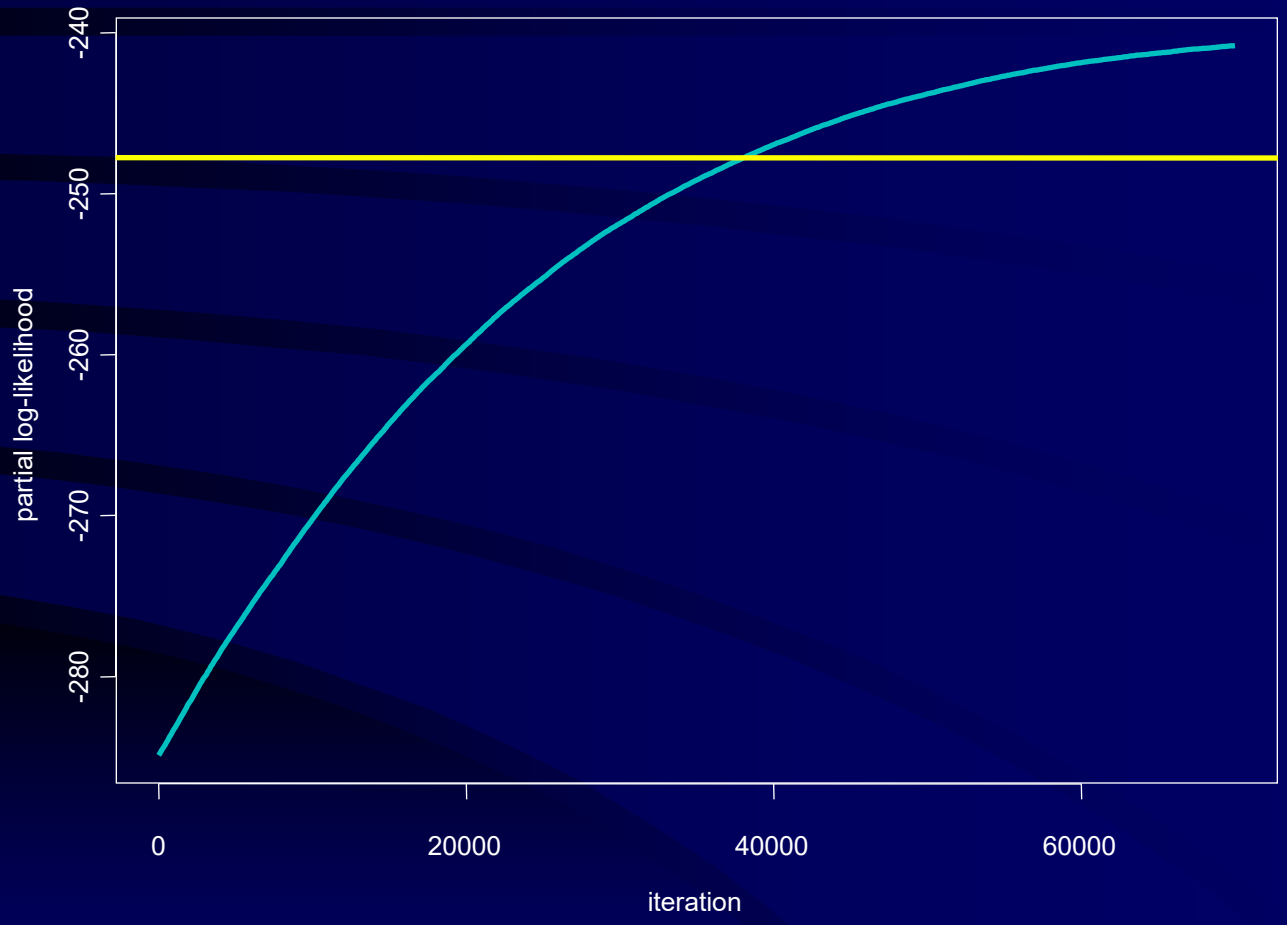
3. Fit a Cox model of the form

$$(t, \delta) \sim offset(F^{(t)}(x)) + \rho f^{(t)}(x)$$

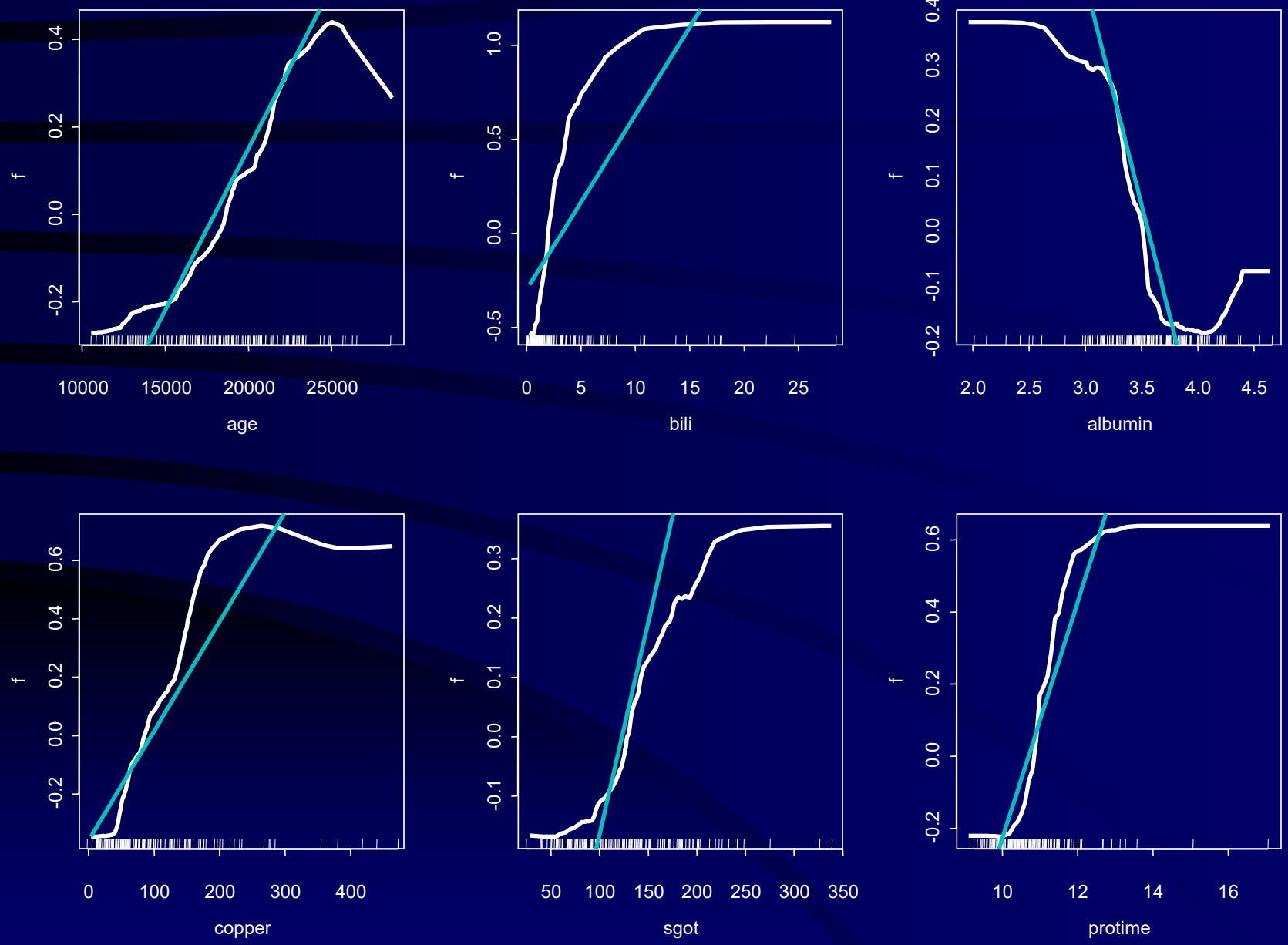
4. The new regression function is

$$F^{(t+1)}(x) = F^{(t)}(x) + \hat{\rho} f^{(t)}(x)$$

Performance



Main effects



Discussion

- Exponential family models
- Bias reduction - non-linear fitting
- Massive datasets - bagging, trimming
- Variance reduction - bagging
- Interpretability - additive models
- High-dimensional regression - trees
- Robust regression

Detail slides

AdaBoost

Freund & Schapire 1996

$(X,Y)_i$ where $Y_i \in \{0,1\}$, $w_i^{(1)} = \frac{1}{N}$

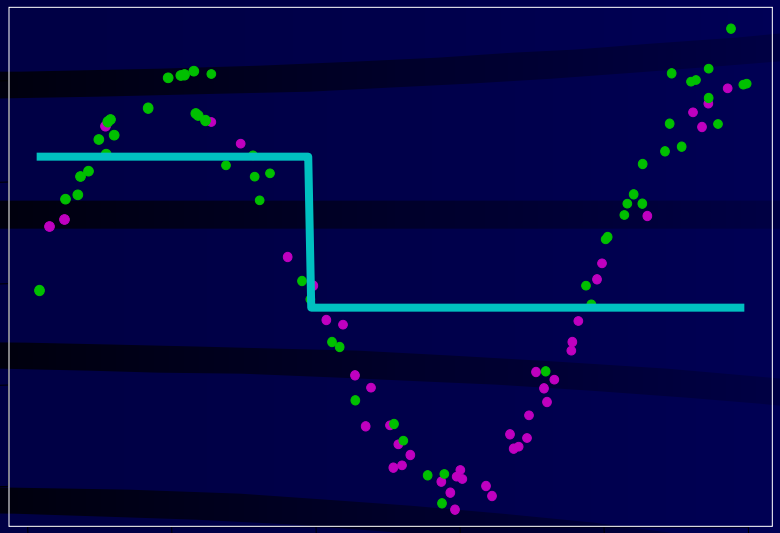
- With weights, fit the model $H_t(x_i) : X^o$ [0,1].
- Compute the error $\varepsilon_t = \sum_{i=1}^{N} w_i^{(t)} \left| y_i - H_t(x_i) \right|$
- Reweight

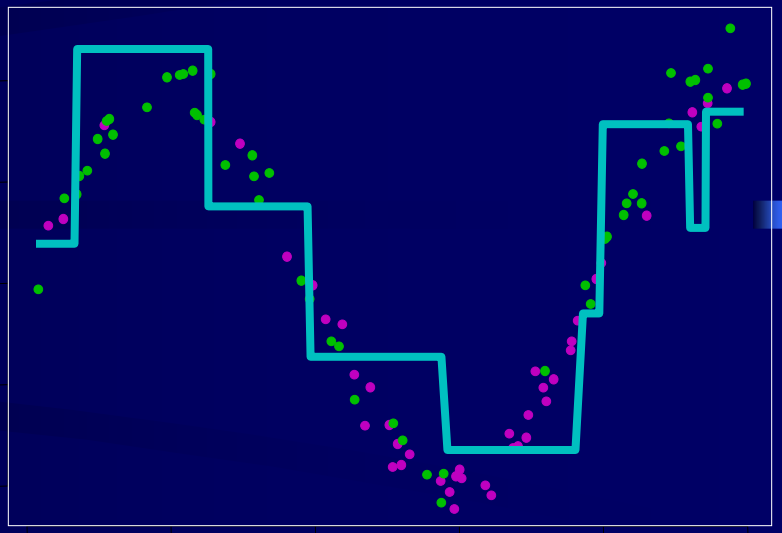
$$w_i^{(t+1)} = w_i^{(t)} \beta_t^{1-\left|y_i - H_t(x_i)\right|} \qquad \beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$$

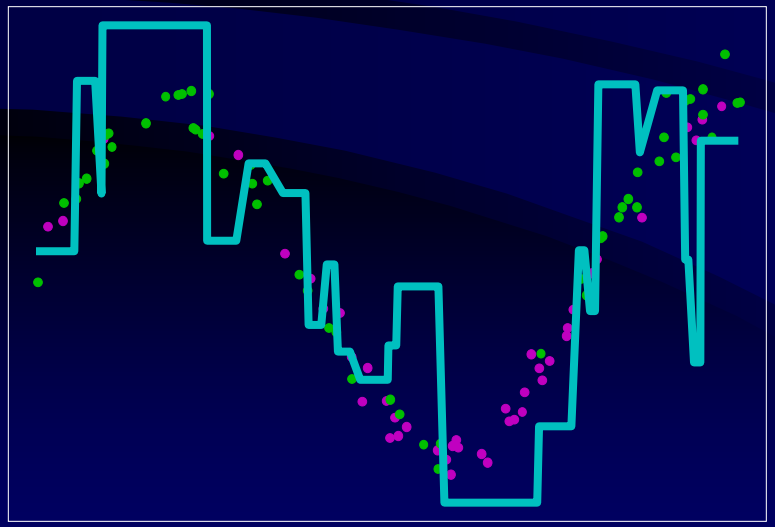
Lastly, predict

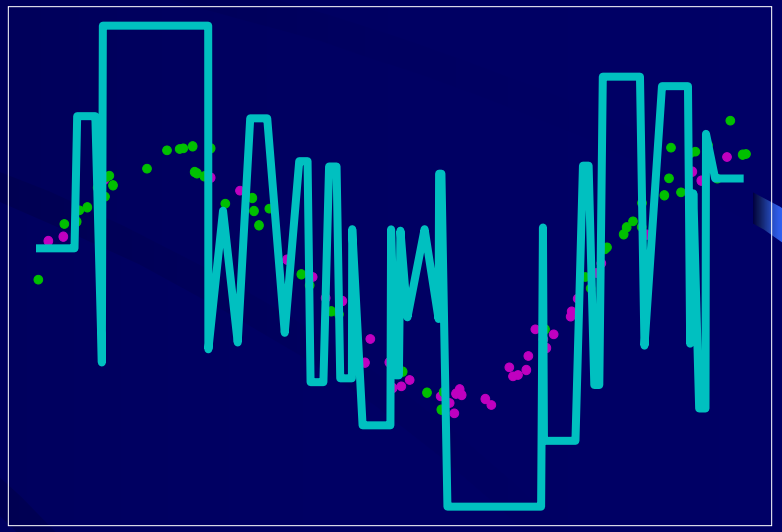
$$H(x) = \frac{1}{1 + \prod_{t=1}^{T} \beta_t^{2r(x)-1}} \qquad r(x) = \frac{\sum_{t=1}^{T} (\log \frac{1}{\beta_t}) H_t(x)}{\sum_{t=1}^{T} (\log \frac{1}{\beta_t})}$$

LogitBoost performance









Learning rate

G. Ridgeway, D. Madigan, T. Richardson [in progress]

- Aggressive maximization causes overfitting.
- Slow down the "learning" rate, $\lambda \in (0,1)$.

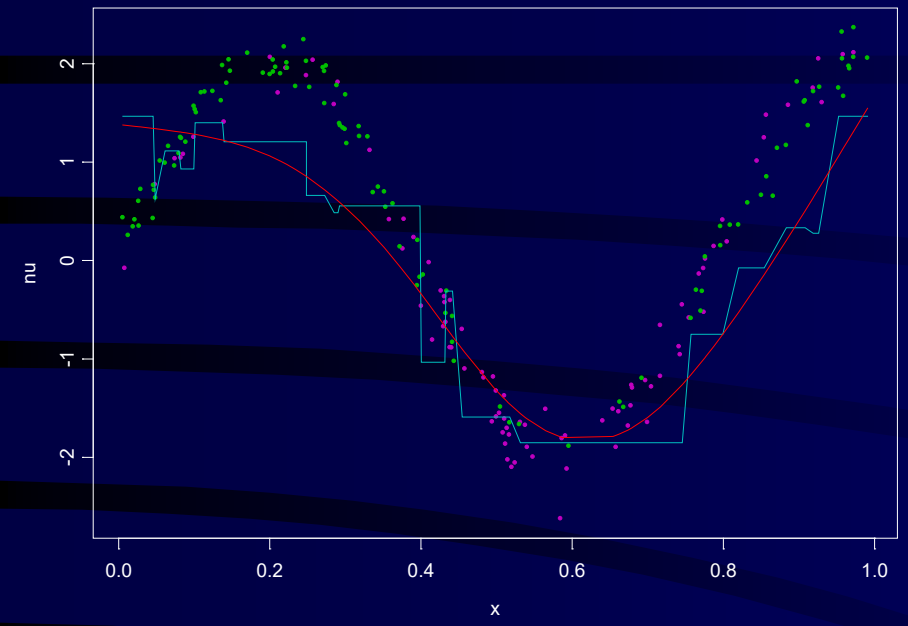
$$F(x) \leftarrow F(x) + \lambda_t E_w\left(\frac{y - p(x)}{p(x)(1-p(x))} \middle| x \right)$$

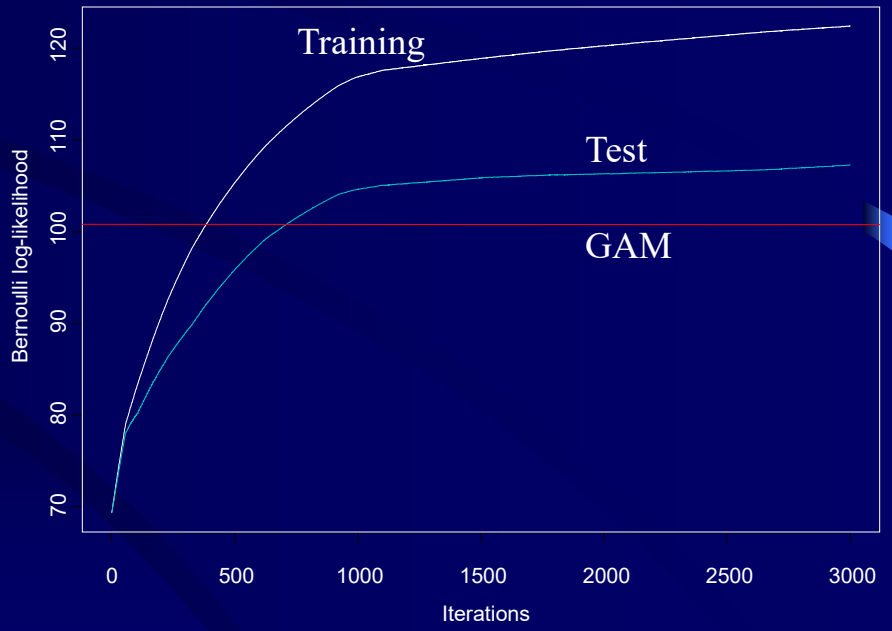
- Related to Copas' proportional shrinkage

$$\bar{y} + K(\hat{y}(x) - \bar{y})$$

LogitBoost performance

$\lambda_t = 0.01$

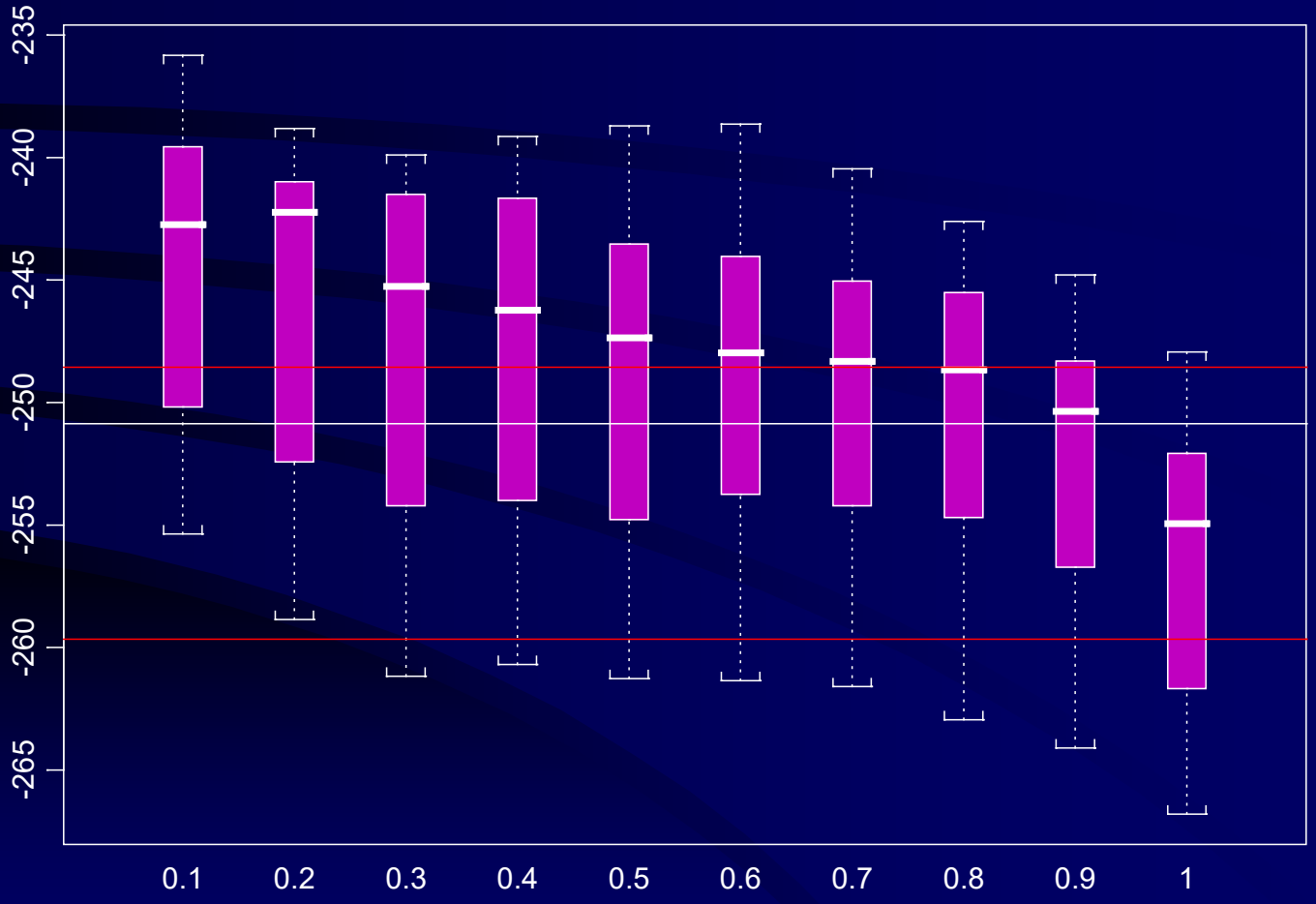




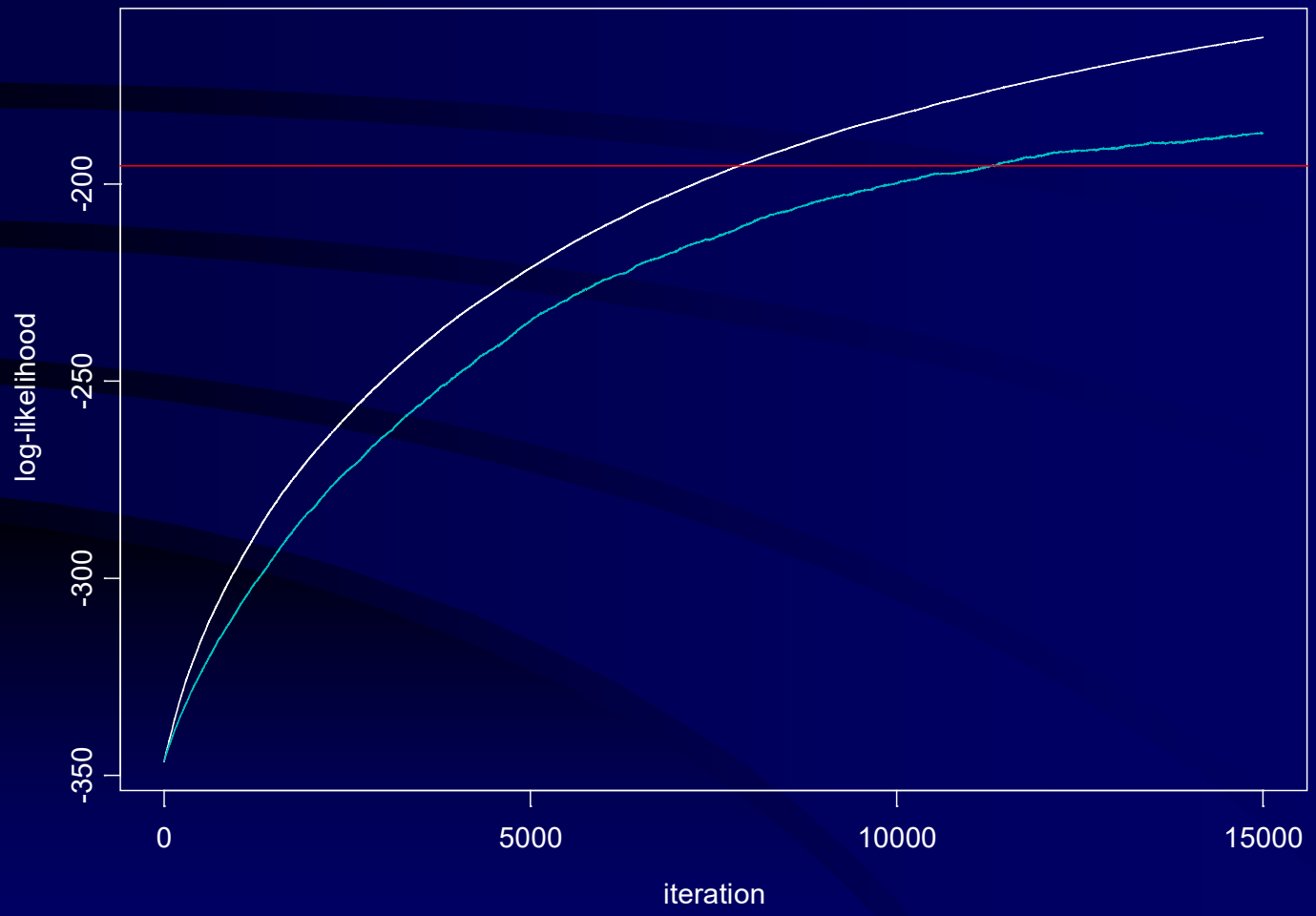
Bagging stage

$F(x_1, x_2) = f(x_1) + f(x_2)$

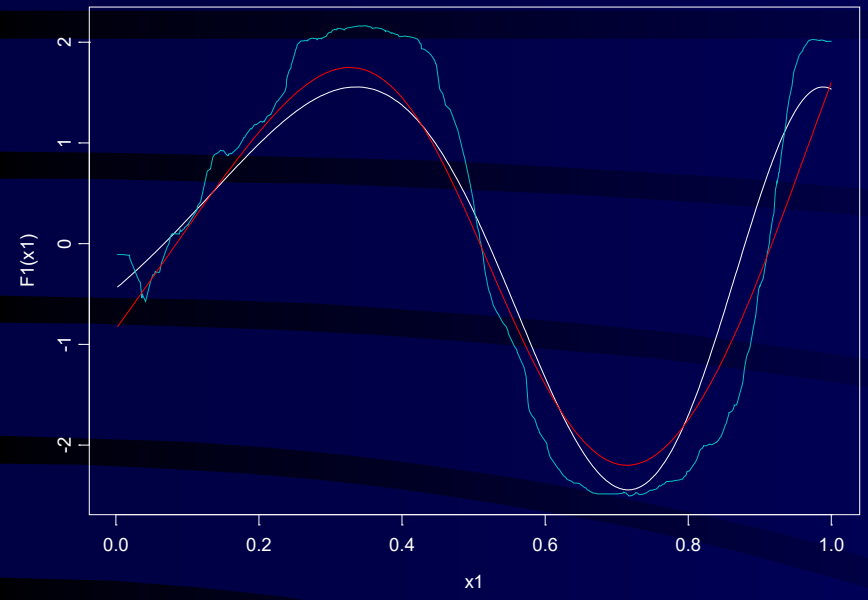
Bagging fraction vs. Validation log-likelihood

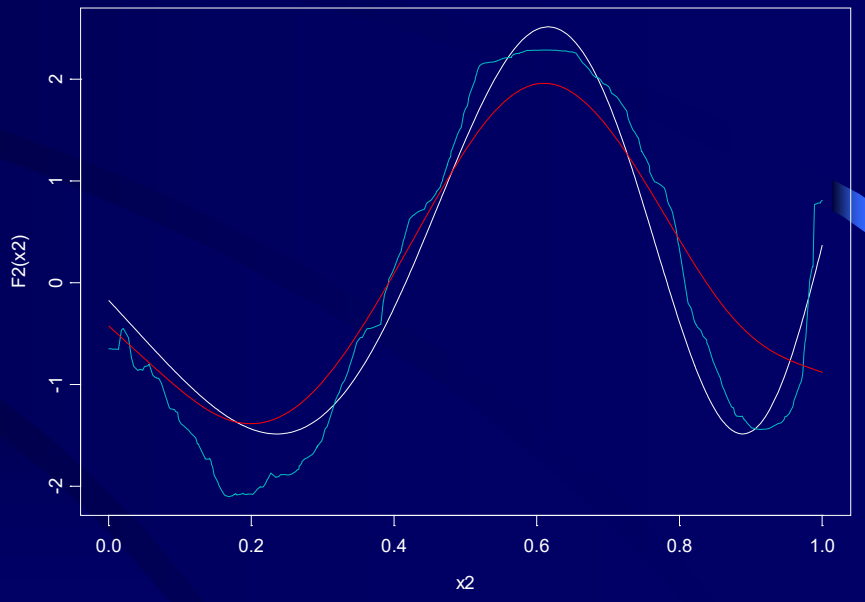


Log-likelihood, bagged=0.1



Interpretation





Censored exponential regression

τ - survival time, δ - death indicator

$$\ell(F \mid \delta, \tau) = \delta(\log \tau + F(x)) - e^{\log\tau + F(x)}$$

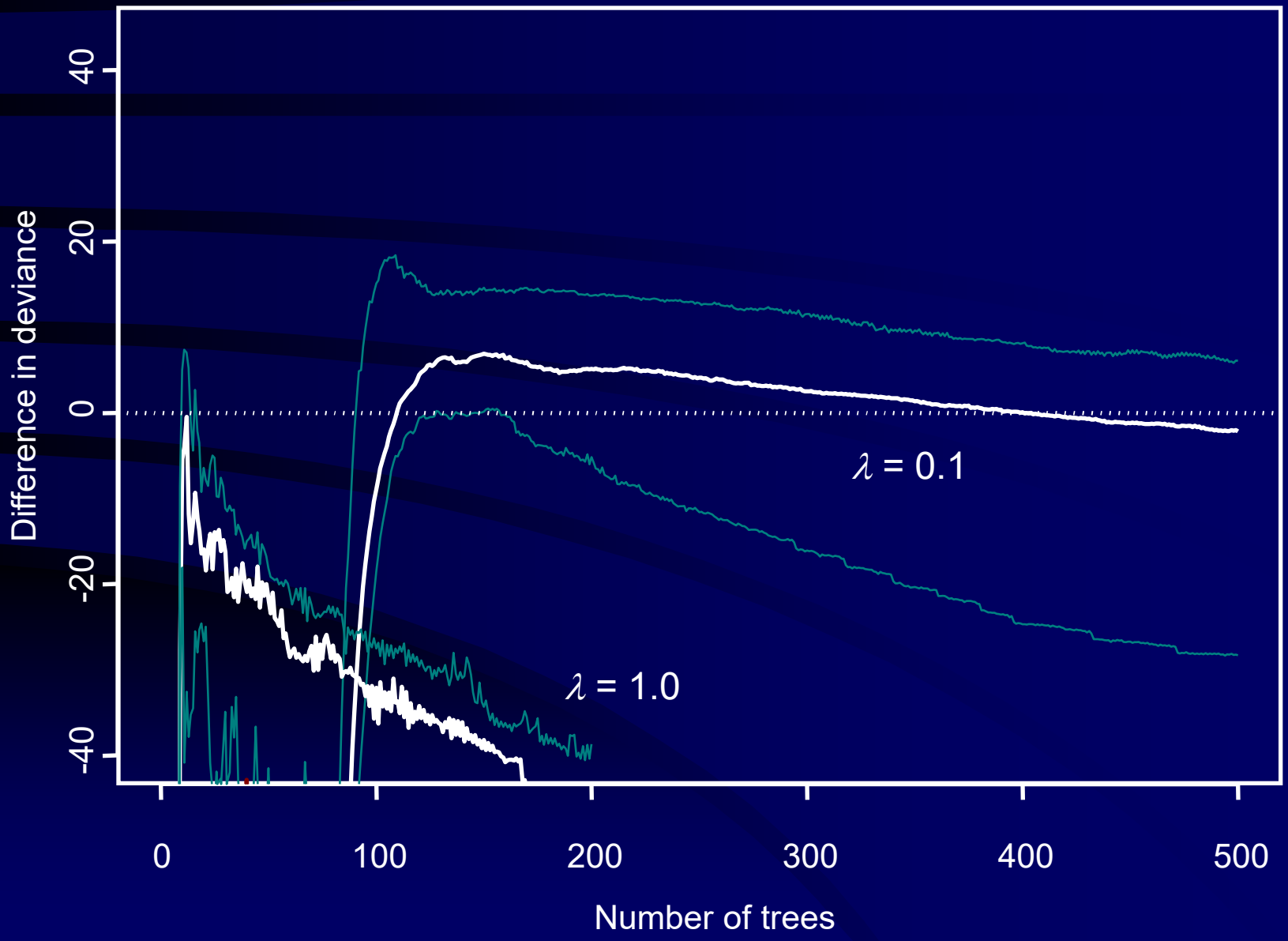
$$b(F) = e^{\log\tau + F(x)},\ a(\phi) = 1$$

$$F(x) \leftarrow F(x) + \lambda_t E_w\left(\frac{\delta}{\tau e^{F(x)}} - 1 \,\middle|\, x \right),\quad w(x) = \tau e^{F(x)}$$

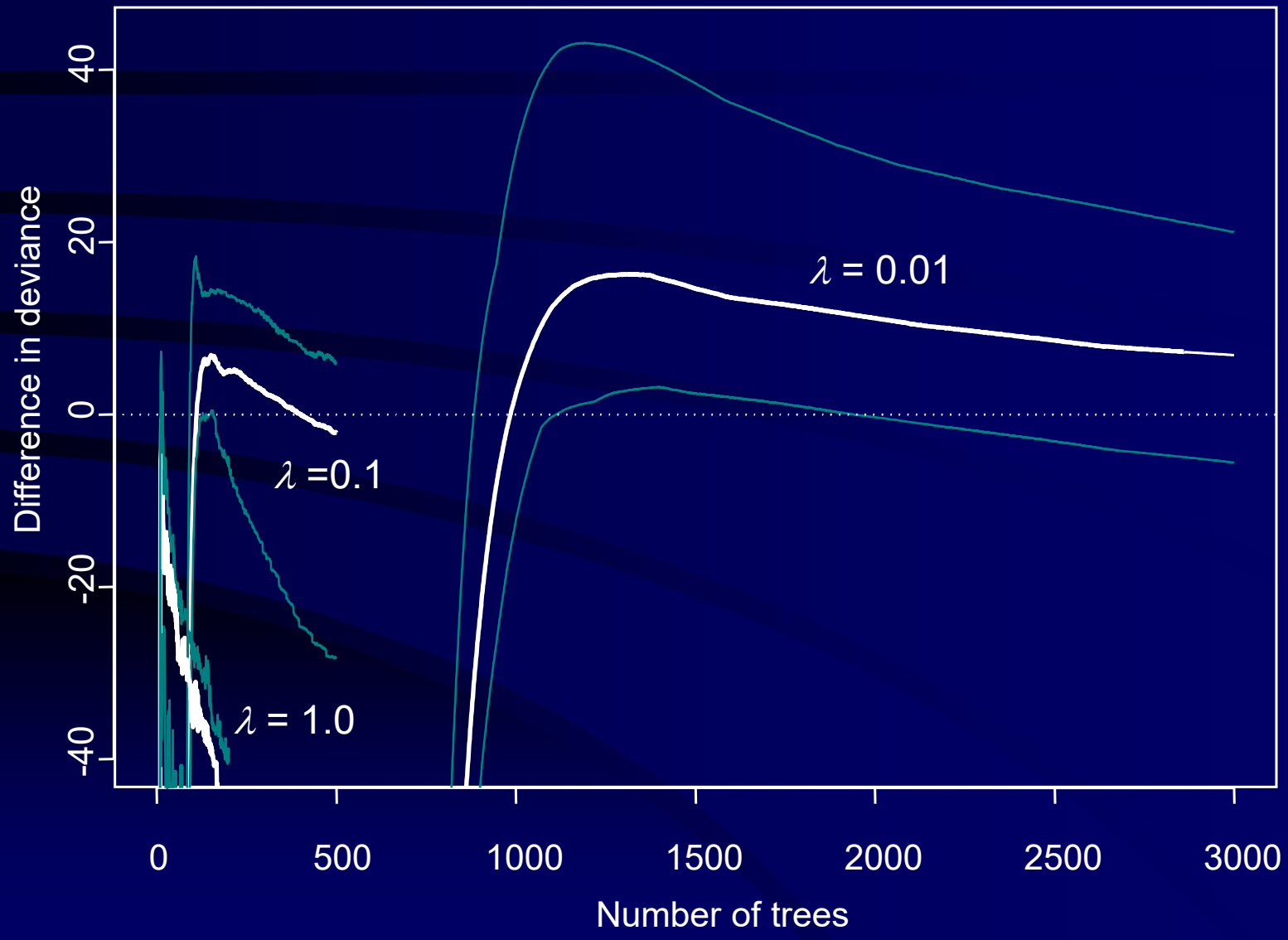
Results for primary biliary cirrhosis

- GBM for survival time given treatment, sex, age, and clinical measurements.
- 310 observations, split in half for model fitting and model validation
- Compared with linear censored exponential regression
- 10 replicates, 3 learning rates, using regression stumps

PBC Results


Difference in deviance
40
20
0
-20
-40
0
100
200
300
400
500
Number of trees
λ = 0.1
λ = 1.0

PBC Results


Difference in deviance
40
20
0
-20
-40
λ = 0.01
λ =0.1
λ = 1.0
0
500
1000
1500
2000
2500
3000
Number of trees