Fat Boosting: A Case Study

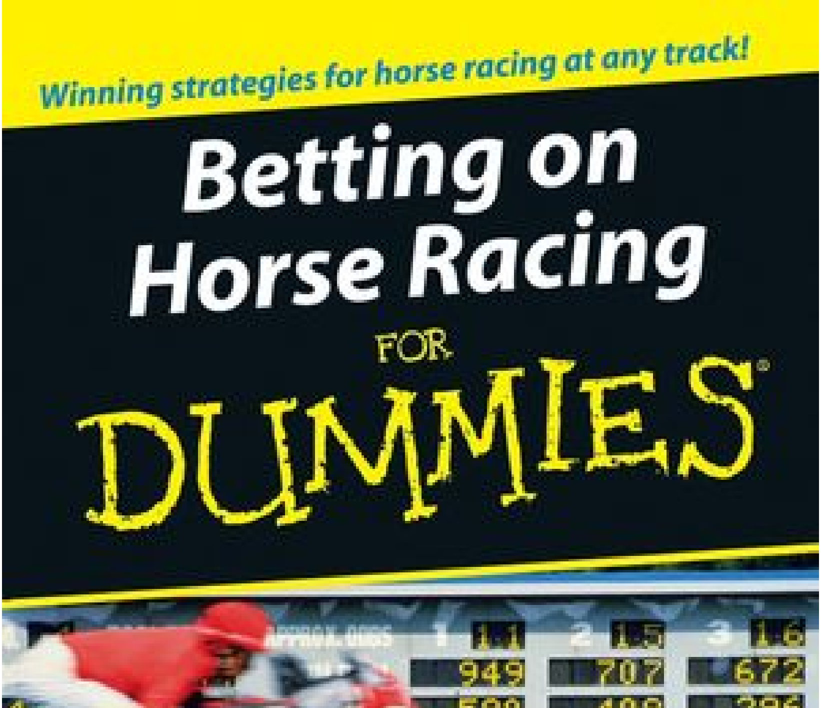
Benjamin Fish, Jeremy Kun, Ádám D. Lelkes

University of Illinois at Chicago
Department of Mathematics, Statistics, and Computer Science

FATML, 2015


WHAT STRATEGY SHOULD I USE TO ALWAYS WIN?
ERM...


WHAT STRATEGY SHOULD I USE TO ALWAYS WIN?
ERM...
FOR THIS RACE... PICK THE HORSE WITH THE MOST WINS IN THE LAST MONTH


Winning strategies for horse racing at any track!
Betting on Horse Racing
FOR
DUMMIES
APPROX. ODDS
949
707
672

The Boosting Problem

How do you combine rules of thumb to form a successful strategy?

Theory Timeline

- 84: Valiant introduces PAC
- 88/89: Kearns and Valiant define weak learnability, pose the boosting question
- 89: Schapire answers in the affirmative
- 90: Better algorithm of Freund
- 90-95: Freund & Schapire team up, design AdaBoost.
- 03: Freund & Schapire win Gödel prize.
- 95 - Present: Lots of great research!

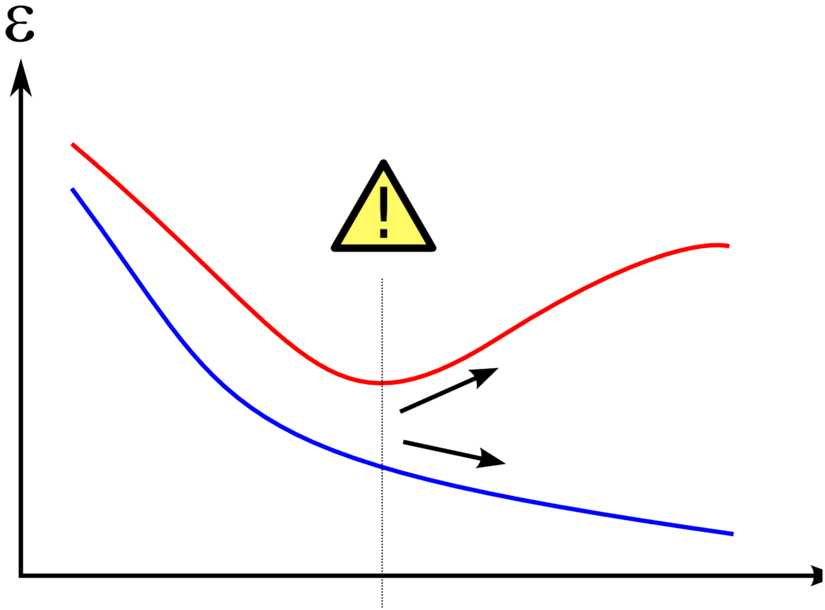
AdaBoost

Input: Weak learner A and data set $D = (x_i, y_i)$.

1. Give each data point x_i a nonnegative weight $\zeta_i = 1$
2. For rounds $t = 1, \ldots, T$:
   - 2.1 Run A on the weighted data, output hypothesis h_t and weighted error rate $\varepsilon_t < 1/2$.
   - 2.2 When A misclassifies x_i, increase ζ_i otherwise decrease ζ_i.
   - 2.3 Give h_t a weight $\alpha_t = \alpha_t(\varepsilon_t)$.
3. Final hypothesis is the sign of $h(x) = \sum_t \alpha_t h_t(x)$

Final h is a **weighted majority vote**.

Boosting does not overfit?!



Margins

Boosting "margin"

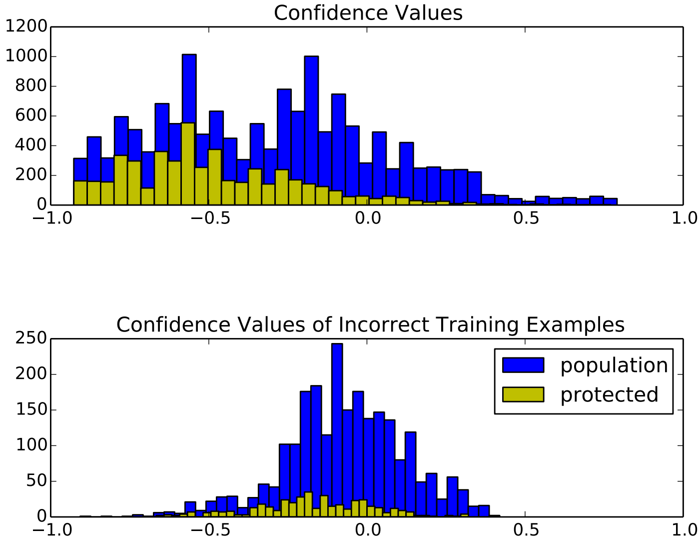
$$\mathrm{conf}(x) = \frac{\sum_{t=1}^{T} \alpha_t h_t(x)}{\sum_t \alpha_t} \in [-1, 1]$$

Theorem (SFBL98)

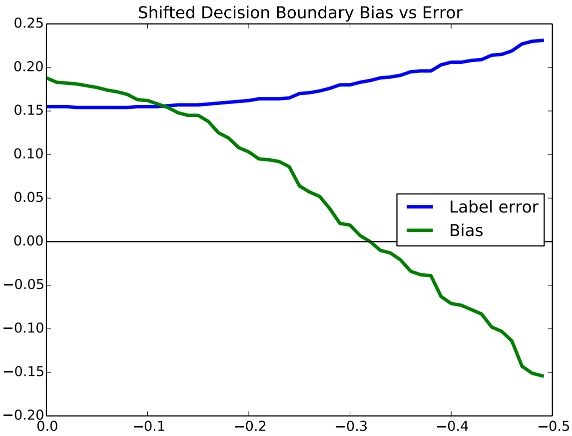
For any $\theta > 0$, the generalization error of AdaBoost is bounded from above by

$$\Pr_{\text{train}} [y_i \, \mathrm{conf}(x_i) \leq \theta] + O\left(\frac{1}{\theta} \cdot \text{ (typical PAC stuff)}\right)$$

Census dataset (UCI adult)



Main idea: relabel protected individuals x with low $|\mathrm{conf}(x)|$ so as to achieve statistical parity.



Algorithm: run boosting and relabel protected individuals x with low $|\mathrm{conf}(x)|$ so as to achieve statistical parity.

Works with any learning algorithm that outputs a "confidence value" or "margin."

Outperforms:

- Obvious baselines
- State of the art (for this dataset)
- Uniform random relabeling
- Boosting a “fair weak learner”
- Massaging data before training
- Variations on these themes . . .

A known distribution for bias?

What if we *knew* how bias was generated?

E.g., labeled dataset $X = (x_i, y_i)$:

1. Add a new (uniformly) random feature x_0.
2. Flip labels of an η-fraction of individuals with $x_0 = 1$ from 1 to -1.
3. See how many labels are corrected by your learning algorithm.

Probability an individual subject to uniform random bias is recovered.

(Our method improves over the baseline)

How resilient are popular algorithms to (known) bias distributions?

Thanks!