LECTURES ON STATISTICS AND DATA ANALYSIS

Columbia University, June 10-19, 2009

Andreas Buja (*Statistics Dept, The Wharton School, UPenn*)

This series of eight lectures will cover a loose collection of topics in statistics, machine learning, data exploration, and applications. Some background for each topic will be provided, and while the technical level varies there will be take-home messages from each lecture for Ph.D. students in statistics and related fields.

* "Trees that speak": classification and regression trees for interpretation (as opposed to prediction)
* "Bagging", its bias-variance properties and a correspondence between subsampling and bootstrap sampling
* "Boosting" for classification and class probability estimation
* "It's the metric, stupid": a principle for multivariate analysis methods that use eigen- or singular value decompositions
* "Flattening warps and cobwebs": non-linear dimension reduction and graph drawing
* "On a scale from 1 to 3...": an exercise in survey data analysis
* "Tuna fishing -- the movie": dynamic graphics for space-time data
* "Seeing is believing": statistical inference for exploratory data analysis

(Additional topics: k-means clustering, calibration for simultaneity)

Some Bio

- PhD 1980 from ETH (Zurich, Switzerland) in Statistics/Math
- -1981 Children's Hospital (Zurich) & ETH
- -1982 Visiting Asst Prof Stanford U & SLAC
- -1985 Asst Prof, U of Wash, Seattle
- 1986 Visiting Bellcore (J. Kettenring, R. Gnanadesikan)
- 1987 Salomon Brothers (4 months)
- -1994 Bellcore
- -1995 AT&T Bell Labs (D. Pregibon, D. Lambert)
- -2001 AT&T Labs
- -present: The Wharton School, UPenn, Philadelphia

FIRST TOPIC: EXPLORING THE UNIVERSE OF LOSS FUNCTIONS FOR CLASS PROBABILITY ESTIMATION

JoinL Work with

Werner Stuetzle (*Statistics Dept, University of Washington*)

Yi Shen (*then at Wharton*)

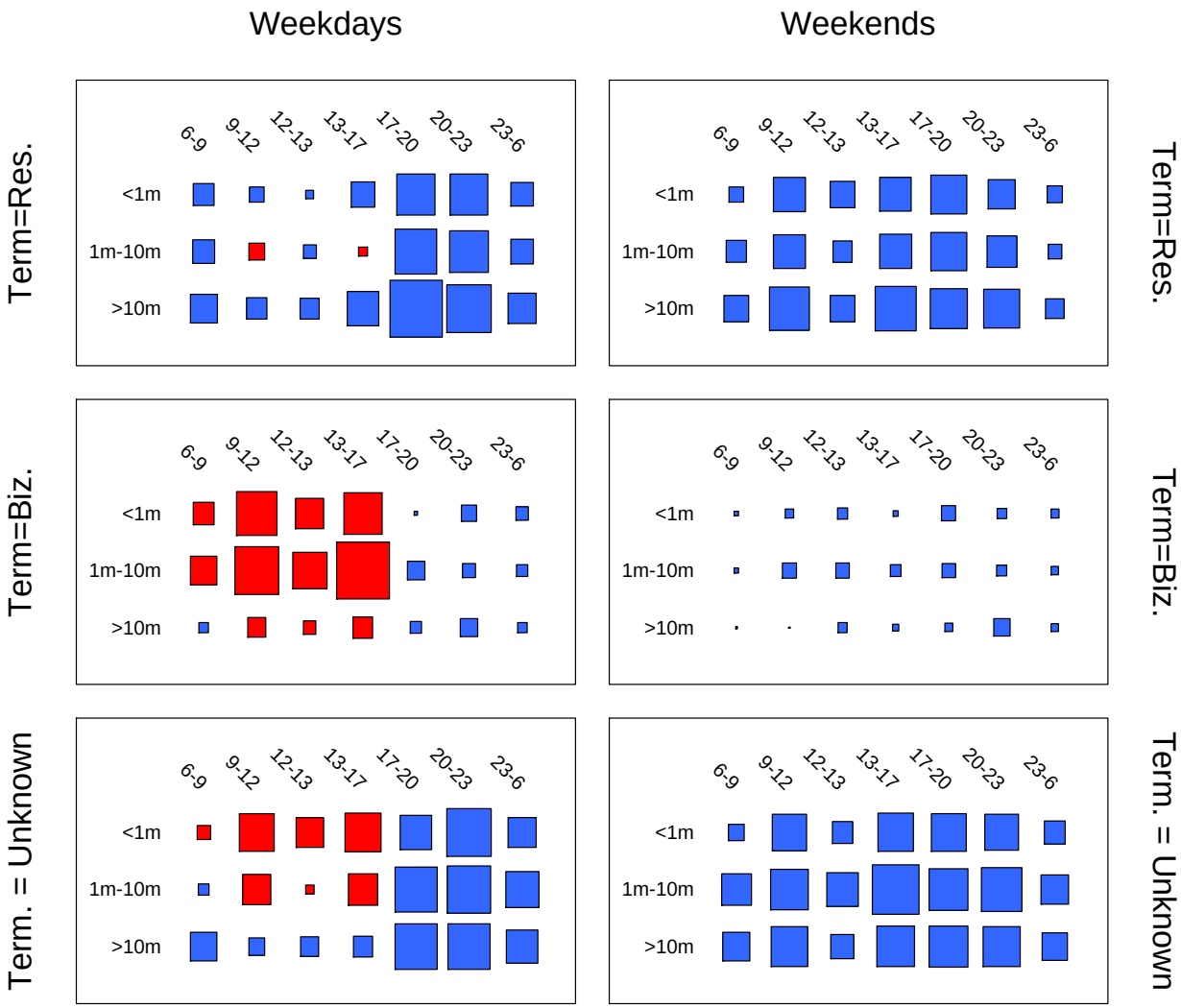
(Part of the work done while AB and WS were with AT&T Labs)

Example

- Data: AT&T Labs' store of call detail records
- Problem: Find residences with home businesses
- Idea: Look for phone numbers with calling patterns that resemble those of small businesses
- Training data: Several months of calls of 50K small businesses and 50K residences
- Feature extraction: $>$100 counts such as
  #{calls: weekdays, 9am$<$begin$<$11am, 1min$<$dur$<$10min}
- Techniques: Boosting vs. logistic ridge regression
- Use: Scoring of $>$50,000,000 residences
  score $= \hat{\mathrm{P}}$(small business)

Coefficients of Logistic Ridge Regression

Red => business-likeness, Blue => residence-likeness



Conclusions from the Example:

- *Classification* is sometimes not sufficient.
- Real interest: *Class Probability Estimation*
- "Labeled data" can be available if looked at the right way
- Rich bag of tools: Discriminant analysis, logistic regression, boosting, SVMs, CART, random forests, ...
- ... but class probability estimation takes a back seat to classification.

Basics 1: Learning/Classification

- *Supervised* vs unsupervised classification
- *Binary* vs multi-class classification
- Training data:

$$(\mathbf{x}_n, y_n) \ , \qquad n = 1...N$$

- $\mathbf{x}_n \in \mathbb{R}^K$: features, predictors
- $y_n \in \{0, 1\}$: class labels, responses

Basics 2: Stochastics

- Assumption intuitively: sampling
- Assumption, technically: $(\mathbf{x}_n, y_n)$ i.i.d. realizations of $(\mathbf{X}, Y)$
  - Marginal distribution of predictors:

$$f(\mathbf{x}) \;=\; P[d\mathbf{x}]/d\mathbf{x}$$

  - Conditional distribution of labels:

$$\eta(\mathbf{x}) \;=\; P[Y = 1|\mathbf{X} = \mathbf{x}] \;=\; E[Y|\mathbf{X} = \mathbf{x}]$$

Together they describe the joint distribution of $\mathbf{X}$ and Y:

$$P[Y = 1, d\mathbf{x}] \;=\; P[Y = 1|\mathbf{X} = \mathbf{x}]P[d\mathbf{x}] \;=\; \eta(\mathbf{x})f(\mathbf{x})d\mathbf{x}$$

Basics 3: Classification vs Class Prob Estimation

- Classifier $\mathrm{cl}(\mathbf{x})$:

  $\mathrm{cl}(\mathbf{x}) \;=\; \mathrm{cl}(\mathbf{x}; (\mathbf{x}_n, y_n)_{n=1\ldots N}) \;\in\; \{0,1\}$

- Class probability estimator $p(\mathbf{x})$:

  $p(\mathbf{x}) \;=\; p(\mathbf{x}; (\mathbf{x}_n, y_n)_{n=1\ldots N}) \;\in\; [0,1]$

- Class probability estimators define classifiers: $p(\mathbf{x}) \mapsto \mathrm{cl}(\mathbf{x})$

  $\mathrm{cl}(\mathbf{x}) \;=\; 1_{[p(\mathbf{x})>t]}$ (e.g. $t = 0.5$)

- Estimation: $p(\mathbf{x})$ estimates $\eta(\mathbf{x})$, $\mathrm{cl}(\mathbf{x})$ estimates $1_{\eta(\mathbf{x})>t}$.

- (Note on ML history:

  Early ML assumed classes to be perfectly separable: $\eta(\mathbf{x}) = 1_A(\mathbf{x})$.

  $\Rightarrow$ No distinction between classification and class prob estimation.

  $\Rightarrow$ Classification is a purely geometric problem of finding boundaries.)

Basics 4: Differences in Conventions between ML and Stats

- Notation: Relabeling of classes $\boxed{\{-1, +1\} \leftrightarrow \{0, 1\}}$
- ± 1 response vs 0-1 response: $y^* = 2\,y - 1$
- ± 1 classifier vs 0-1 classifier: $\mathrm{cl}^*(\mathbf{x}) = 2\,\mathrm{cl}(\mathbf{x}) - 1$
- $(\mathbf{x},\, y)$ is correctly classified iff: $y^* \mathrm{cl}^*(\mathbf{x}) = 1$

Product	$\mathrm{cl}^*(\mathbf{x}) = +1$	$\mathrm{cl}^*(\mathbf{x}) = -1$
$y^* = +1$	$+1$	-1
$y^* = -1$	-1	$+1$

- Misclassification rate := $P[y \neq \mathrm{cl}] = P[y^* \mathrm{cl}^* = -1]$

  What assumption was made in this definition? (Diabetics ...)

Basics 5: Quantile Classification and Unequal Cost Classification

- Common in older AI/ML work: Equal misclassification cost for

  $y = 0,\ \mathrm{cl} = 1 \quad \Rightarrow \quad$ false positive

  $y = 1,\ \mathrm{cl} = 0 \quad \Rightarrow \quad$ false negative

- Assume cost $c \in (0, 1)$ for misclassifying $y = 0$ as $\mathrm{cl} = 1$ (false positive)

  and cost $\quad 1 - c \quad$ for misclassifying $y = 1$ as $\mathrm{cl} = 0$ (false negative)

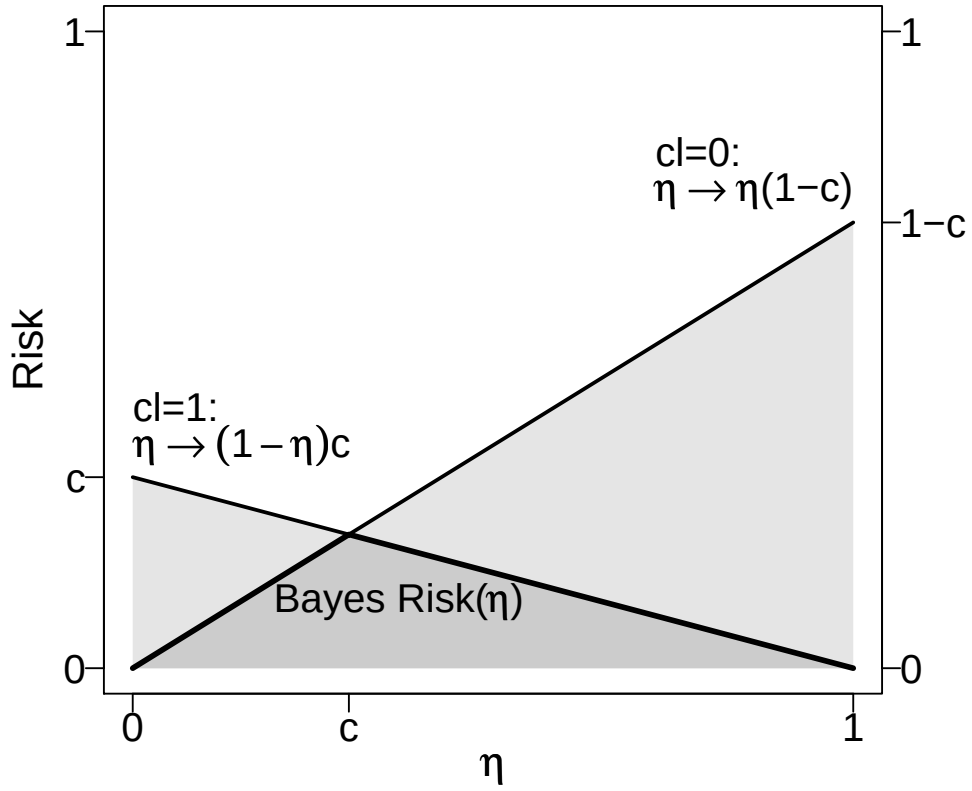
$$\mathbf{L}(y|\mathrm{cl}) \ = \ \begin{pmatrix} c \quad \text{when } y = 0,\ \mathrm{cl} = 1 \\ 1 - c \text{ when } y = 1,\ \mathrm{cl} = 0 \end{pmatrix} \ = \ y\,(1 - c)1_{\mathrm{cl}=0} + (1 - y)\,c\,1_{\mathrm{cl}=1}$$

- Local/pointwise Risk $= E[\mathbf{L}(Y|\mathrm{cl})] \ =: \ \mathbf{L}(\eta|\mathrm{cl})$ when $P[Y = 1] = \eta$:

$$\mathbf{L}(\eta|\mathrm{cl}) \ = \ \begin{pmatrix} (1 - \eta)\,c \text{ when } \mathrm{cl} = 1 \\ \eta\,(1 - c) \text{ when } \mathrm{cl} = 0 \end{pmatrix} \ = \ \eta\,(1 - c)1_{\mathrm{cl}=0} + (1 - \eta)\,c\,1_{\mathrm{cl}=1}$$

Basics 5 (contd.): Quantile Classification and Unequal Cost Classification

- Bayes Risk $= \min_{\mathrm{cl}\in\{0,1\}} \mathbf{L}(\eta|\mathrm{cl}) = \min(\ (1-\eta)\,c,\ \eta\,(1-c)\ )$
- Minimizer: Classify $\boxed{\mathrm{cl}=1 \text{ when } \eta > c}$



Basics 5 (contd.): Quantile Classification and Unequal Cost Classification

- Equivalence: - classification at quantile c and
  - classification with costs $c/(1-c)$ for false positives/negatives

  In particular: Median classification = Equal-cost classification

- Population Bayes risk: If we knew $\eta(\mathbf{X})$, the average Bayes risk would be

$$E[\,\min(\,(1-\eta(\mathbf{X}))\,c,\ \eta(\mathbf{X})\,(1-c)\,)]$$

$$= \text{unavoidable average misclassification cost}$$

- Baseline misclassification rate:

  If $\eta_1 = P[Y=1] = E[\eta(\mathbf{X})]$ is the marginal class 1 probability,
  then the trivial classifier that ignores $\mathbf{X}$ is

$$\mathrm{cl} = 1 \ \text{ if } \ \eta_1 > c \quad \text{and} \quad \mathrm{cl} = 0 \ \text{ otherwise.}$$

  Any classifier that uses predictors $\mathbf{X}$ must beat the baseline classifier
  with risk $\quad \min(\,(1-\eta_1)\,c,\ \eta_1\,(1-c)\,)$.

Basics 6: Statisticians' True and Trusted Tools

- **Logistic regression**: a conditional model of Y given $\mathbf{X}$

$$\eta(\mathbf{x}) \;=\; \psi(\mathbf{x}'\boldsymbol{\beta}) \;, \qquad \psi(F) = 1/(1+\exp(-F))$$

  Idea: Estimate a linear model and map the values to the range (0,1).

- **Linear discriminant analysis** (LDA): a conditional model of $\mathbf{X}$ given Y

$$f(\mathbf{X}|Y=1) \sim \mathcal{N}(\boldsymbol{\mu}_1, \Sigma) \;, \qquad f(\mathbf{X}|Y=0) \sim \mathcal{N}(\boldsymbol{\mu}_0, \Sigma)$$

  Actually, this is equivalent to LS regression of the 0-1 response Y on $\mathbf{X}$.

- Extensions to more than two classes exist:
  multinomial logistic regression and multi-class discriminant analysis.

- Non-parametric extensions exist:
  . logistic regression with polynomial or spline bases, ...
  . LDA based on non-linear transformations of $\mathbf{X}$: FDA (Hastie et al. 94)

Basics 7: Recap of Logistic Regression

- Logistic link and linear model: $\eta(\mathbf{x}) \;=\; \psi(F(\mathbf{x}))\;, \quad F(\mathbf{x}) \;=\; \mathbf{x}'\boldsymbol{\beta}$

  $\text{Logit}(\eta) = \log(\eta/(1-\eta))\;, \quad \psi(F) = 1/(1+e^{-F})\;, \quad 1-\psi(F) = \psi(-F)$

- Loss from one observation when observing $y \in \{0,1\}$ and guessing $\hat{\eta} = p$:

  $\mathbf{L}(y|p) \;=\;$ $-$log likelihood of a Bernoulli variable

  $= \; -\log(p^y(1-p)^{1-y}) \;=\; \boxed{-y\log(p) - (1-y)\log(1-p)}$

  $= \; \begin{pmatrix} -\log(p) & \text{when } y{=}1 \\ -\log(1{-}p) & \text{when } y{=}0 \end{pmatrix} \;\geq 0$

- Composed for one observation $(\mathbf{x}, y)$: $F = \mathbf{x}'\boldsymbol{\beta}$

  $$\mathbf{L}(y|\psi(F)) \;=\; -\log(\psi(y^*F)) \;=\; \boxed{\log(1+e^{-y^*F})}$$

- Composed for a sample $(\mathbf{x}_n, y_n),\ n = 1...N$: $F_n = \mathbf{x}_n'\boldsymbol{\beta}, \;\; p_n = \psi(F_n)$

  $$\textstyle\sum_{n=1,\ldots,N} \mathbf{L}(y_n|\psi(\mathbf{x}_n'\boldsymbol{\beta})) \;=\; \sum_{n=1,\ldots,N} \log(1+e^{-y_n^*F_n})$$

Basics 8: Properties of the Bernoulli Likelihood

- The $-$log likelihood of the Bernoulli model is also called **log-loss**:

$$\mathbf{L}(y|p) \;=\; -\,y\log(p) - (1-y)\log(1-p)$$

- **Risk** = average loss: Assume $\eta = P[Y=1]$, and estimate $\hat{\eta} = p$. Then:

$$E[\mathbf{L}(Y|p)] \;=\; -\,\eta\log(p) - (1-\eta)\log(1-p) \;=:\; \boxed{\mathbf{L}(\eta|p)}$$

- Relation to entropy: measure of "dis-information"

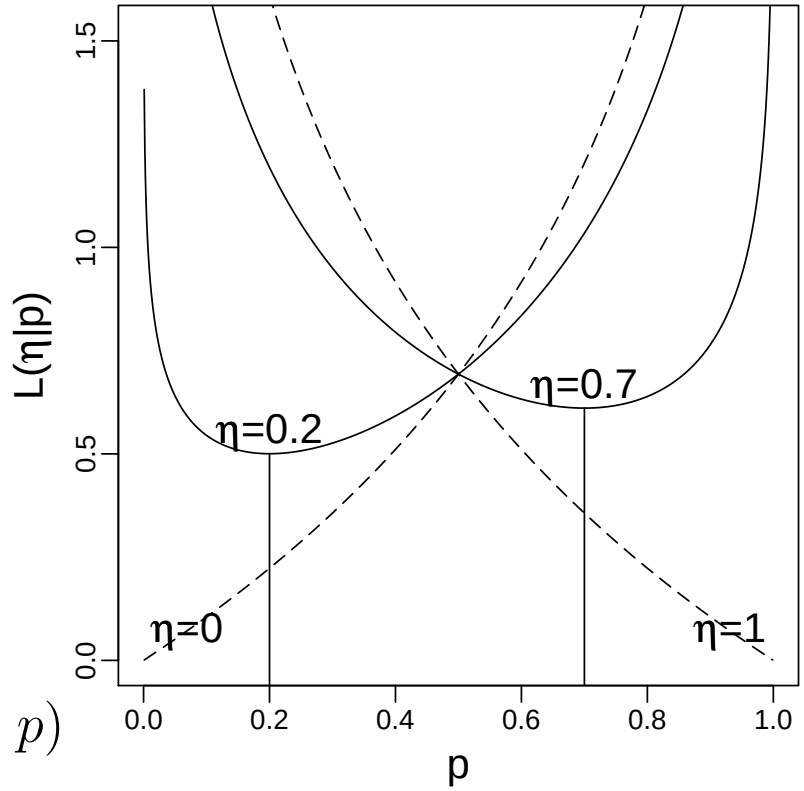
$$\mathbf{L}(\eta|\eta)] \;=\; -\,\eta\log(\eta) - (1-\eta)\log(1-\eta) \;=\; \text{Entropy}(\eta)$$
$$= \text{ unavoidable average loss/risk if } \eta \text{ is true}$$

- **Fisher Consistency**: At "$N=\infty$", log-loss should estimate correctly.

$$\boxed{\text{argmin}_p\, \mathbf{L}(\eta|p) \;=\; \eta}$$

Another term for the same: log-loss is a **Proper Scoring Rule**.

Basics 8 (contd.): Log-Loss a Proper Scoring Rule



$$\mathbf{L}(0|p) = -\log(1-p) \qquad \mathbf{L}(1|p) = -\log(p)$$

$$p \;\mapsto\; \mathbf{L}(\eta|p) \;=\; \eta\,\mathbf{L}(1|p) + (1-\eta)\,\mathbf{L}(0|p)$$

(A more playful version of the above in the form of a "Rapplet" is in this R source file.)

Basics 9: Scores and Margins — Comments on ML

- In ML classifiers are obtained by thresholding a score:

  $\mathrm{cl}(\mathbf{x}) \;=\; 1 \quad$ iff $\quad F(\mathbf{x}) > 0$

  Example of logistic regression: Score = Logit

  $F(\mathbf{x}) = \log(\frac{p(\mathbf{x})}{1-p(\mathbf{x})}) = \boldsymbol{\beta}'\mathbf{x}, \qquad F(\mathbf{x}) > 0$ iff $p(\mathbf{x}) > 1/2$

  Comment: The habit of thresholding F at 0 (equivalent to thresholding p at $1/2$) locked the ML literature into equal-cost classification for a long time.

- Margin := $y^* F(\mathbf{x}) =$ "Degree of correct classification"

  The greater $y^* F(\mathbf{x})$, "the correcter" the classification.

  Comment:

  High margins $\Rightarrow$ Low out-of-sample misclassification error. — Wrong!

  Reason: Overfitting can create high margins in-sample.

  Then again: Low hold-out misclassification rate can be achieved with very strange "over-fitting" boundaries.

Famous AdaBoost: Where does this come from?

Given a class $\mathcal{C}$ of "weak" classifiers $\mathrm{cl}(\mathbf{x})$ (e.g., stumps)

1. Initialize weights $W_n = 1/N$
2. Repeat for $t = 1$ to T:
   (a) Fit classifier $\mathrm{cl}_t(\mathbf{x})$ to $(\mathbf{x}_n,\, y_n,\, W_n)_{n=1..N}$ $\quad (\mathrm{cl}_t(\mathbf{x}) \in \mathcal{C})$
   (b) $e_t \;=\; \sum_n W_n \; 1_{[y_n \neq \mathrm{cl}_t(\mathbf{x}_n)]} \;/ \sum_n W_n$
   (c) $\beta_t \;=\; \frac{1}{2} \log \frac{1-e_t}{e_t}$ $\qquad (> 0$ if $e_t < 1/2)$
   (d) $W_n \;\leftarrow \begin{cases} W_n \cdot e^{\beta_t} & [y_n \neq \mathrm{cl}_t(\mathbf{x}_n),\ \text{up-weighting}] \\ W_n \cdot e^{-\beta_t} & [y_n = \mathrm{cl}_t(\mathbf{x}_n),\ \text{down-weighting}] \end{cases}$ (normalize)
3. Classifier: $\mathrm{cl}(\mathbf{x}) = 1_{\left[\sum_{t=1..T} \beta_t \, \mathrm{cl}_t^*(\mathbf{x}) > 0\right]}$ ("majority vote")

Discrete AdaBoost: Some Reverse Engineering

- Weights: $W_n^{(t+1)} \sim W_n^{(t)} \cdot \exp(-\beta_t \; y_n^* \; \mathrm{cl}_t^*)$

$$
\begin{aligned}
W_n^{(1)} &\sim 1 \\
W_n^{(2)} &\sim \exp(-\, y_n^* \; \beta_1 \; \mathrm{cl}_1^*(\mathbf{x}_n)\,) \\
W_n^{(3)} &\sim \exp(-\, y_n^* \; (\, \beta_1 \; \mathrm{cl}_1^*(\mathbf{x}_n) + \beta_2 \; \mathrm{cl}_2^*(\mathbf{x}_n)\,)\,) \\
W_n^{(4)} &\sim \exp(-\, y_n^* \; (\, \beta_1 \; \mathrm{cl}_1^*(\mathbf{x}_n) + \beta_2 \; \mathrm{cl}_2^*(\mathbf{x}_n) + \beta_3 \; \mathrm{cl}_3^*(\mathbf{x}_n)\,)\,) \\
&\ldots \\
W_n &\sim \exp(-\, y_n^* \; F(\mathbf{x}_n)\,)
\end{aligned}
$$

- Coefficients β_t:

$$
\frac{1}{2} \log \frac{1 - e_t}{e_t} \;=\; \mathsf{argmin}_\beta \sum_n W_n \; \exp\,(\, -\beta \cdot y_n^* \, \mathrm{cl}_t^*(\mathbf{x}_n)\,)
$$

AdaBoost as Minimizer of Exponential Loss

1. Initialize $F(\mathbf{x}) = 0$
2. Repeat for $t = 1$ to T:

$$(\beta_t,\, \mathrm{cl}_t) \;\leftarrow\; \mathsf{argmin}_{\beta\in\mathbb{R},\ \mathrm{cl}(\mathbf{x})\in\mathcal{C}} \sum_n e^{-\,y_n^*\cdot(F(\mathbf{x}_n)+\beta\,\mathrm{cl}^*(\mathbf{x}_n))}$$

$$F(\mathbf{x}) \;\leftarrow\; F(\mathbf{x}) + \beta_t\,\mathrm{cl}_t^*(\mathbf{x})$$

3. Classifier: $\mathrm{cl}(\mathbf{x}) \;=\; 1_{[\,F(\mathbf{x})>0\,]}$

Note exponential loss!

Real AdaBoost with Exponential Loss, Barebones

Main difference between Discrete and Real AdaBoost:

. Discrete AdaBoost: base learner = classifier (± 1=valued)

. Real AdaBoost: base learner = real function with arbitrary values

1. Let $\mathcal{F} = \{ f : \mathbb{R}^K \to \mathbb{R} \,|\ldots\}$
2. Initialize $F(\mathbf{x}) = 0$
3. Repeat for $t = 1$ to T:

$$f_t \leftarrow \text{argmin}_{f \in \mathcal{F}} \sum_n e^{-y_n^* \cdot (F(\mathbf{x}_n) + f(\mathbf{x}_n))}$$

$$F(\mathbf{x}) \leftarrow F(\mathbf{x}) + f_t(\mathbf{x})$$

4. Classifier: $\text{cl}(\mathbf{x}) = 1_{[\,F(\mathbf{x}) > 0\,]}$

Exponential Loss for Linear Models Anyone?

- Could use exponential loss to fit linear models:

$$\hat{\boldsymbol{\beta}} \;=\; \mathsf{argmin}_{\boldsymbol{\beta}} \sum_{n=1\ldots N} e^{-\,y_n^*\,\mathbf{x}_n'\boldsymbol{\beta}}$$

- Logistic regression:

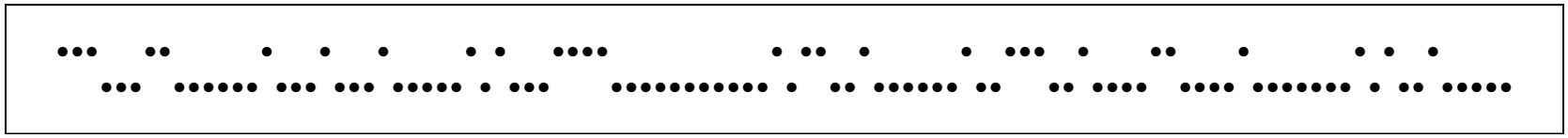
$$\hat{\boldsymbol{\beta}} \;=\; \mathsf{argmin}_{\boldsymbol{\beta}} \sum_{n=1\ldots N} \log(1+e^{-\,y_n^*\,\mathbf{x}_n'\boldsymbol{\beta}})$$

- Comparison:
$$\begin{aligned} e^{-F} &\geq \log(1+e^{-F}) && (\forall F)\\ e^{-F} &\approx \log(1+e^{-F}) && (F\to\infty)\\ e^{-F} &\gg \log(1+e^{-F}) \approx -F && (F\to-\infty) \end{aligned}$$

- Exponential loss penalizes negative scores more drastically than logistic loss.

Pointwise Examination of Exponential Loss

- Stylized, as in the examination of properties of log-loss:
  - Assume for given $\mathbf{x}$ we have multiple observations.
  - Assume $Y \sim \text{Bernoulli}(\eta)$ where $\eta = \eta(\mathbf{x})$.



- Exponential loss at that point:

$$E[\, e^{-Y^* F}\,] \;=\; \eta\, e^{-F} + (1-\eta)\, e^{F}$$

- The minimizing score is:

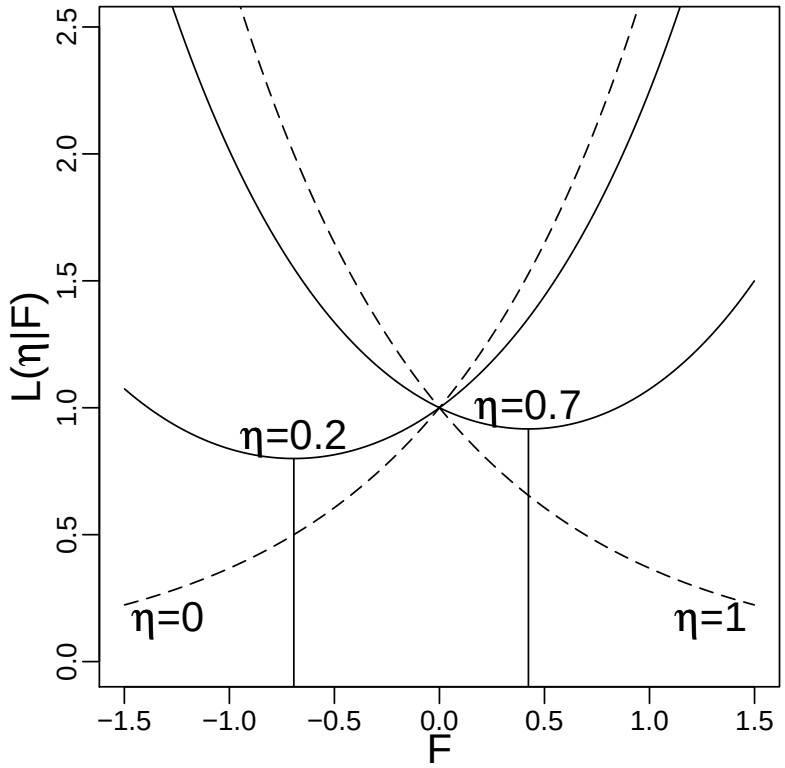
$$F_{\min}(\eta) \;=\; \tfrac{1}{2} \log \tfrac{\eta}{1-\eta}$$

- Conclusion: Boosting tries to estimate half the logit at each $\mathbf{x}$.

  (Friedman, Hastie, Tibshirani 2000)

Graph of Exponential Loss

- $\mathbf{L}(\eta|F) \;:=\; E[\,e^{-Y^*F}\,] \;=\; \eta\, e^{-F} + (1-\eta)\, e^{F}$



- $\mathsf{argmin}_F\; \mathbf{L}(\eta|F) \;=\; \frac{1}{2}\log\frac{\eta}{1-\eta}$

A Link Function for Boosting

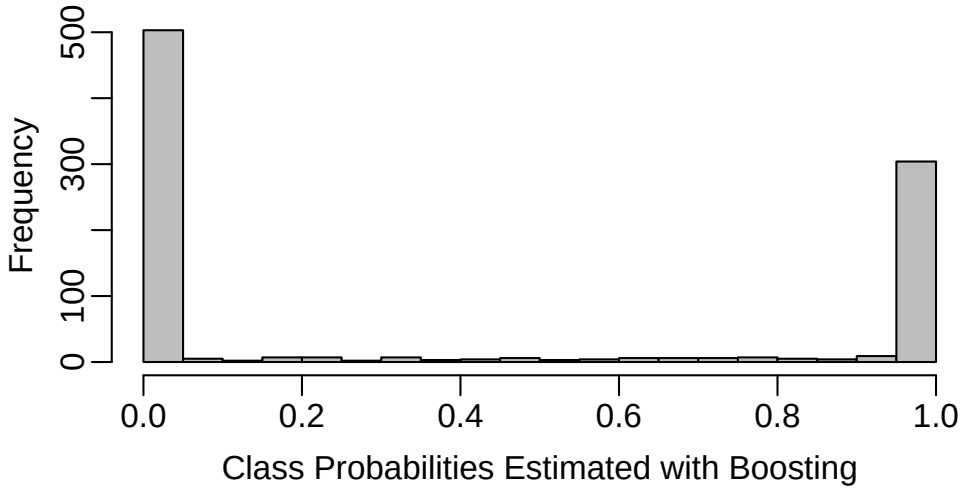
- Natural link for boosting:

$$\eta \;=\; \psi(2F) \;=\; 1/(1+e^{-2F})\,, \qquad F(\eta) \;=\; \frac{1}{2}\log\frac{\eta}{1-\eta}$$

- After estimating $F(\mathbf{x})$ with boosting, one obtains class prob estimates by

$$p(\mathbf{x}) \;=\; \psi(2F(\mathbf{x}))$$

- Just kidding!!!! Such class prob estimates tend to look like this ...



$\Rightarrow$ Total overfit of the class probabilities, yet good classification ...

Where is Boosting's Beef?

(1) Not in the exponential loss function?

Logistic loss produces MLEs $\Rightarrow$ Statistical efficiency

(Take that with a grain of salt: No model is correct.)

But: Exponential loss has interesting invariance properties (see below).

(2) In the stagewise fitting?

Stagewise fitting used to be a no-no in statistics:

"If you add a predictor, you must update the previous predictors!"

But: Stagewise fitting has a gradient descent interpretation.

(FHT, Breiman, ..., Bühlmann)

Where is Boosting's Beef? (Contd.)

(3) In the particular use of "base learners" such as stumps?

Total confusion here:

The "Weak Learner" paradigm asks for simple base learners, e.g., stumps.

$\Rightarrow$ Boosting is supposed to remove bias.

Yet some of the most successful applications used C4.5.

$\Rightarrow$ Boosting actually removes variance!

Can boosting remove both bias and variance?

$\Rightarrow$ Yes, but it may need some randomization.

(Friedman's Stochastic Gradient Boosting)

(4) Why would overfitted class probs produce good classification?

Misclassification rate counts only being on the correct side of c:

$$p(\mathbf{x}) > c \quad \text{or} \quad p(\mathbf{x}) < c$$

$\Rightarrow p(\mathbf{x})$ can be **high variance** or biased for low misclassification rate.

How to Avoid Over-Fitted Class Probability Estimates

- Performance of classification is measured with hold-out misclassification rate.
- This, however, produces over-fitted class probability estimates.
- If interested in class probability estimation, do the following:
  Use log-loss or exponential loss or any other "surrogate loss"
  for cross-validation.
- Example: "Regularize" boosting by stopping boosting iterations
  when hold-out exponential loss starts to increase.
- Now the mapping of boosting scores $F(\mathbf{x})$ to probability estimates with
  $$p(\mathbf{x}) = \psi(2F(\mathbf{x}))$$
  should produce more reasonable results.
- However, classifying by thresholding $p(\mathbf{x}) > 1/2$ will not produce results as good as boosting with overfitted class probs — a quandary.

Classification versus Class Prob Estimation, Again

- Let's focus now on good class prob estimation — it's more difficult.
- Classification is idiosyncratic:

  Use a smooth "surrogate" loss function (logistic, exponential),
  but measure performance with crude misclassification loss.
- In class prob estimation, the surrogate loss is the primary loss.
- In what follows, we will examine the universe of prob loss functions.
- We will disentangle class prob loss functions and link functions.

Analysis of Exponential Loss: A Proper Scoring Rule for Boosting

- Recall logistic regression: Compose
  the log-loss $\mathbf{L}(y|p) = -y\log(p) - (1-y)\log(1-p)$ and
  the link function $p = \psi(F)$
  $\Rightarrow$ Logistic loss $F \mapsto \mathbf{L}(y|p = \psi(F)) = \log(1 + e^{-y^* F})$

- For boosting:
  We have the composition loss $F \mapsto \mathbf{L}(y|p = \psi(F)) = e^{-y^* F}$,
  and we have the link $p = \psi(2F)$.
  $\Rightarrow$ What is the loss on the prob scale p rather than F?

(Comment: Loss functions for class probability estimation are defined on p, not F.)

A Proper Scoring Rule for Boosting (Contd.)

- The answer: Start from $\mathbf{L}(y|F) \;=\; y\,e^{-F} + (1-y)\,e^{F}$ and
  substitute the inverse link $F = \log(p/(1-p))/2$.

$$\boxed{\mathbf{L}(y|p) \;=\; y\,\left(\tfrac{p}{1-p}\right)^{-1/2} + (1-y)\,\left(\tfrac{p}{1-p}\right)^{1/2}}$$

  (Comment: Note the similarity/dissimilarity to log-loss.)

- The associated risk/average loss when $P[Y=1] = \eta$ and $\hat{\eta} = p$:

$$\mathbf{L}(\eta|p) \;=\; E[\,\mathbf{L}(Y|p)\,] \;=\; \eta\,\left(\frac{p}{1-p}\right)^{-1/2} + (1-\eta)\,\left(\frac{p}{1-p}\right)^{1/2}$$

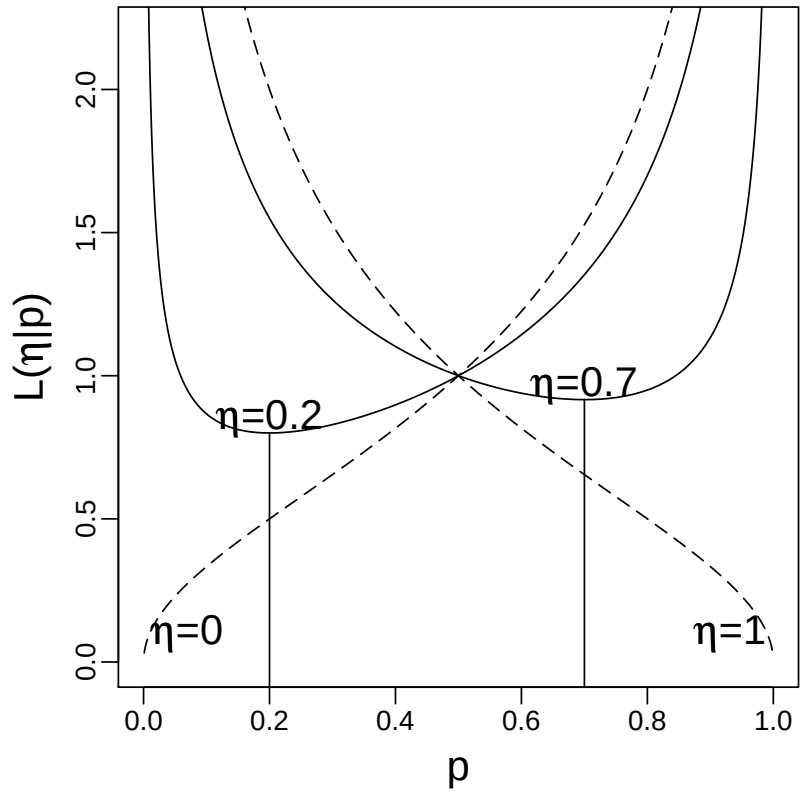
  Another **Proper Scoring Rule**: $\text{argmin}_p\, \mathbf{L}(\eta|p) \;=\; \eta$

- The associated entropy: the semi-circle function (modulo factor 2)

$$\mathbf{L}(\eta|\eta) \;=\; 2\,(\eta(1-\eta))^{1/2}$$

A Proper Scoring Rule for Boosting (Contd.)

$$\text{argmin}_p \, \mathbf{L}(\eta|p) \;=\; \eta$$



(Again, a more playful version in the form of a "Rapplet" is in this R source file; hit 'b'.)

A Proper Scoring Rule from Squared Error Loss

- The general form of a class prob loss fct is:

$$\mathbf{L}(y|p) \quad \mapsto \quad L_0(p) = \mathbf{L}(0|p), \quad L_1(1-p) = \mathbf{L}(1|p)$$

- Squared error loss:

$$\mathbf{L}(y|p) \;=\; y(1-p)^2 + (1-y)p^2 \qquad\qquad [= (y-p)^2]$$

- The associated risk/average loss when $P[Y=1] = \eta$ and $\hat{\eta} = p$:

$$\mathbf{L}(\eta|p) \;=\; E[\,\mathbf{L}(Y|p)\,] \;=\; \eta(1-p)^2 + (1-\eta)p^2 \qquad [\neq (\eta-p)^2]$$

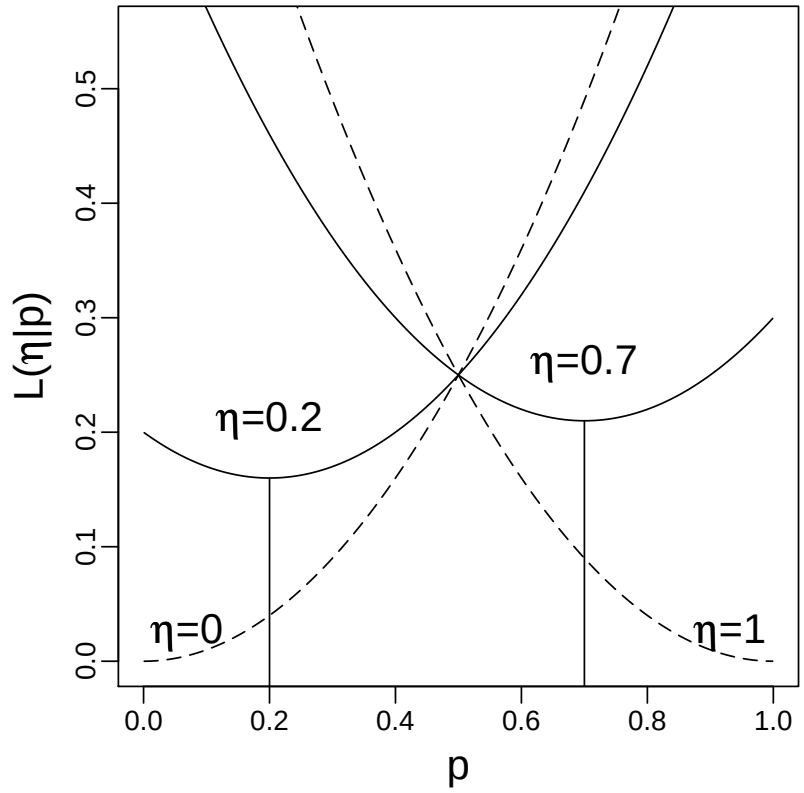
  Another **Proper Scoring Rule**: $\text{argmin}_p\, \mathbf{L}(\eta|p) \;=\; \eta$

- The associated entropy: the Gini index

$$\mathbf{L}(\eta|\eta) \;=\; \eta(1-\eta)$$

A Proper Scoring Rule for Squared Error Loss (Contd.)

$$\text{argmin}_p \, \mathbf{L}(\eta|p) \;=\; \eta$$



(Once again, play the "Rapplet" in this R source file; hit 's'.)

Counter-Examples for Proper Scoring Rules

- Power losses are **not** proper scoring rules for $r \neq 2$ (> 1):

$$L_0(p) = p^r, \qquad L_1(p) = (1-p)^r$$

$$\text{argmin}_p \mathbf{L}(\eta|p) \;=\; \frac{\eta^{1/(r-1)}}{\eta^{1/(r-1)} + (1-\eta)^{1/(r-1)}} \;\neq\; \eta$$

- Special case $r = 1$: $\quad \mathbf{L}(\eta|p) = \eta(1-p) + (1-\eta)p = \eta + (1-2\eta)p$

  This is minimized by $\quad p = 1$ if $1 - 2\eta < 0$, i.e., $\eta > 1/2$,

  and it is minimized by $\;p = 0$ if $1 - 2\eta > 0$, i.e., $\eta < 1/2$.

  $\Rightarrow$ Minimization attempts classification, not class prob estimation.

  Equivalent to the SVM hockey stick loss function.

- Why is $(\eta - p)^2$ not a proper scoring rule?

  It is a loss function in the sense of Wald's decision theory.

Class Probability Loss Functions in General

- General form of observed loss:

$$\begin{aligned}\mathbf{L}(y|p) &= yL_1(1-p) + (1-y)L_0(p) \\ &= \begin{cases} L_1(1-p) & \text{when } y = 1 \\ L_0(p) & \text{when } y = 0 \end{cases}\end{aligned}$$

  where: $L_1(t),\ L_0(t) \quad \uparrow \quad$ on $(0,1)$

- The associated risk/average loss when $P[Y=1] = \eta$ and $\hat{\eta} = p$:

$$\begin{aligned}\mathbf{L}(\eta|p) &= E[\,\mathbf{L}(Y|p)\,] \\ &= \eta\, L_1(1-p) + (1-\eta)\, L_0(p)\end{aligned}$$

- Associated entropy/unavoidable minimum average loss:

$$\mathbf{L}(\eta|\eta) = \eta\, L_1(1-\eta) + (1-\eta)\, L_0(\eta)$$

Class Probability Loss Functions in General (Contd.)

- Interpretation: $L_0(p)$ = loss for distance of p from 0
  $L_1(1-p)$ = loss for distance of p from 1
- Irrelevance of additive constants:
  $L_0(t) + k_0,\ L_1(t) + k_1$ are equivalent to $L_0(t),\ L_1(t)$.
- Irrelevance of shared factors:
  $k\,L_0(t),\ k\,L_1(t)$ are equivalent to $L_0(t),\ L_1(t))$.
  $k_0\,L_0(t),\ k_1\,L_1(t)$, however, are not equivalent.
- On the probability scale, the notion of "margin" is no longer natural, or else it has to be translated using "distance from $1/2$".
- So far we have only considered symmetric cases: $L_1(t) = L_0(t)$.
  Later we will want unequal losses.

Which Loss Fcts are Proper Scoring Rules?

- When do $L_1(t)$, $L_0(t)$ combine to satisfy $\text{argmin}_p \mathbf{L}(\eta|p) = \eta$?

  (Recall: "Proper scoring rule" = "Fisher consistency")

- Answer: Assuming $L_1(t)$, $L_0(t)$ differentiable, stationarity at $p = \eta$ yields

$$\frac{d}{dp}|_{p=\eta} \mathbf{L}(\eta|p) \;=\; 0 \;=\; -\eta L_1'(1-\eta) + (1-\eta) L_0'(\eta)$$

$$\rightarrow \quad \frac{L_1'(1-\eta)}{1-\eta} \;=\; \frac{L_0'(\eta)}{\eta} \;=:\; \omega(\eta) \;>\; 0$$

- The **weight function** $\omega(.)$ essentially determines $\mathbf{L}(\eta|p)$:

$$\boxed{L_1'(1-p) \;=\; (1-p)\,\omega(p)\;, \qquad L_0'(p) \;=\; p\,\omega(p)}$$

(This goes back to theories of subjective probability:

Shuford, Albert, Massengill (1966), Savage (1971), Schervish (1989))

Concocting Prob Loss Fcts that are Proper Scoring Rules

- Recipe: Make up any non-negative weight function $\omega(p)$ on $[0,1]$.
  (No need to normalize to a density; can allow $\int \omega(p)\mathrm{d}p = \infty$ near 0, 1.)
- Define $L_0(p)$ to be any indefinite integral of $p\,\omega(p)$.
- Define $L_1(1-p)$ to be any indefinite integral of $(1-p)\,\omega(p)$.
  (Note that $p \mapsto L_1(1-p)$ is descending.)
- One could also prescribe $L_0(p)$, say, derive $\omega(p)$ from it,
  then determine the $L_1(1-p)$ to go with it, and vice versa.
  In general, if one of the three functions is given, the others are determined.

What Does the Weight Function $\omega(p)$ Mean?

- It describes the emphasis with which the loss function penalizes getting specific ranges of p wrong.
- Example: If $\omega(p)$ has asymptotes near 0 and 1, it really wants you to to get extreme probabilities right.
- Example: If $\omega(p)$ puts a lot of mass near 0.3, it really wants you to be right about guessing whether $\eta > 0.3$ or $\eta < 0.3$.

(Below we will see a technical result that makes the above very precise.)

Constructing a Family of Proper Scoring Rules with Beta Weights

- Introduce a large family of weight functions:

$$\begin{aligned} \omega(p) &= p^{\alpha-1}\,(1-p)^{\beta-1} \\ \alpha,\ \beta &\qquad \text{unrestricted} \end{aligned}$$

- All common loss functions are in this family:

(1) $\alpha = \beta = -1/2$: boosting loss, $\omega(p) = p^{-3/2}(1-p)^{-3/2}$

(2) $\alpha = \beta = 0$: log-loss, $\omega(p) = p^{-1}(1-p)^{-1}$

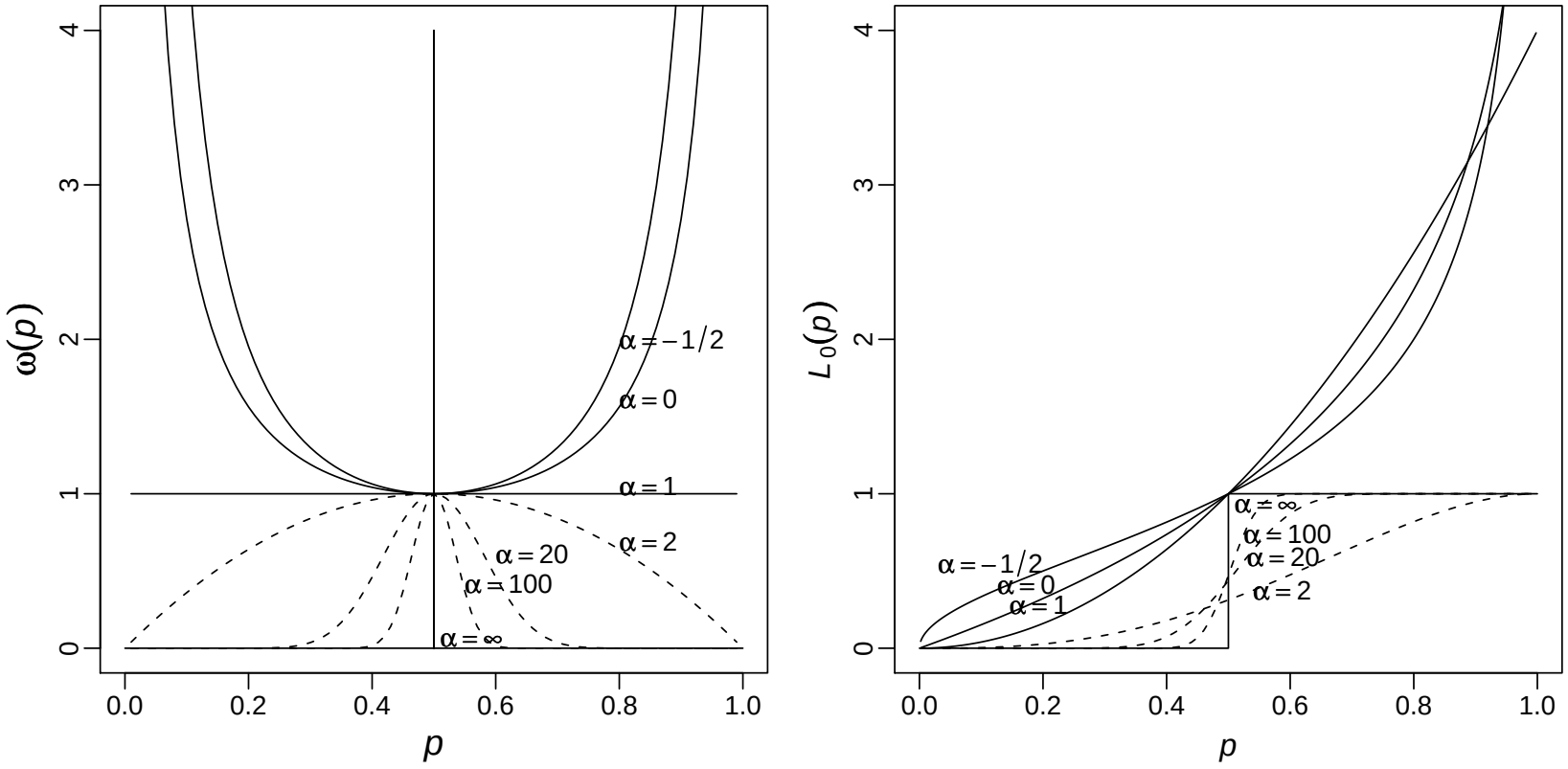
(3) $\alpha = \beta = 1$: squared error loss, $\omega(p) = 1$

(4) $\alpha = \beta \to \infty$: misclassification loss, $\omega(p) = \delta_{1/2}(p)$

- For $\alpha,\ \beta > 0$ these are densities with

$$\mu = \frac{\alpha}{\alpha+\beta} \qquad \sigma^2 = \frac{\mu(1-\mu)}{\alpha+\beta+1}$$

Graphs of Some Beta Weights and their Loss Fcts $\omega(q)$



All examples are symmetric: $\alpha = \beta = -1/2, 0, 1, 2, 20, 100, \infty$

(The first three correspond to boosting loss, log-loss, squared error loss, resp.)

Cost-Weighted Misclassification as a Limiting Case

- What do $L_0(p)$ and $L_1(1-p)$ look like if $\omega(p) \to \delta_c$? Step functions!

$$L_0(p) \;=\; c\,1_{[p \geq c]}\,, \qquad L_1(1-p) \;=\; (1-c)\,1_{[p<c]}$$

- The observed loss for an observation y and an estimate p is:

$$\mathbf{L}(y|p) \;=\; y\,(1-c)\,1_{[p<c]} + (1-y)c\,1_{[p \geq c]}$$

- The associated risk/average loss when $P[Y=1] = \eta$ and $\hat{\eta} = p$:

$$\mathbf{L}(\eta|p) \;=\; \eta\,(1-c)\,1_{[p<c]} + (1-\eta)c\,1_{[p \geq c]}$$

- The associated entropy: cost-weighted Bayes risk! (Cost $= c$)

$$\mathbf{L}(\eta|\eta) \;=\; \eta\,(1-c)\,1_{[\eta<c]} + (1-\eta)\,c\,1_{[\eta \geq c]} \;=\; \min(\,\eta\,(1-c),\,(1-\eta)\,c\,)$$

A Theorem Relating Classification Losses and Class Prob Losses

- Parametrize cost-weighted classification losses with the cost c:

$$\mathbf{L}_c(y|p) \;=\; y\,(1-c)\,1_{[p<c]} + (1-y)c\,1_{[p\geq c]}$$

- If $\omega(p)$ is the weight fct of a proper scoring rule $\mathbf{L}(\eta|p)$, then:

$$\boxed{\mathbf{L}(y|p) \;=\; \int \mathbf{L}_c(y|p)\,\omega(c)\,\mathrm{d}c}$$

- The same "loss mixing" over cost weights c holds ...

$$\text{... for risks :} \qquad \mathbf{L}(\eta|p) \;=\; \int \mathbf{L}_c(\eta|p)\,\omega(c)\,\mathrm{d}c$$

$$\text{... for entropies :} \qquad \mathbf{L}(\eta|\eta) \;=\; \int \mathbf{L}_c(\eta|\eta)\,\omega(c)\,\mathrm{d}c$$

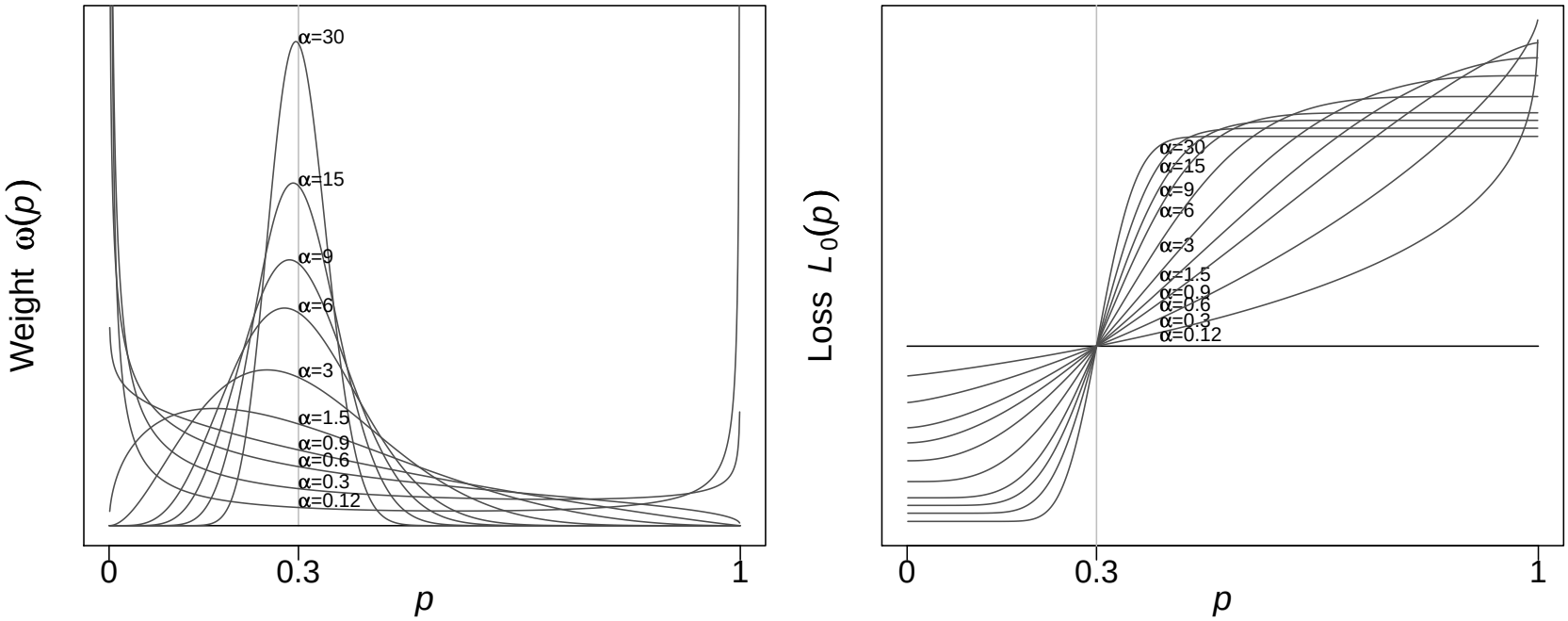
- This result explains in what sense $\omega(p)$ emphasizes certain p-ranges:

  Where $\omega(p)$ is large, $\mathbf{L}(y|p)$ attempts good quantile classification.

  In particular: log-loss and boosting try to get extreme probs classified correctly.

Making Use of New-Found Freedom: Tailoring Losses

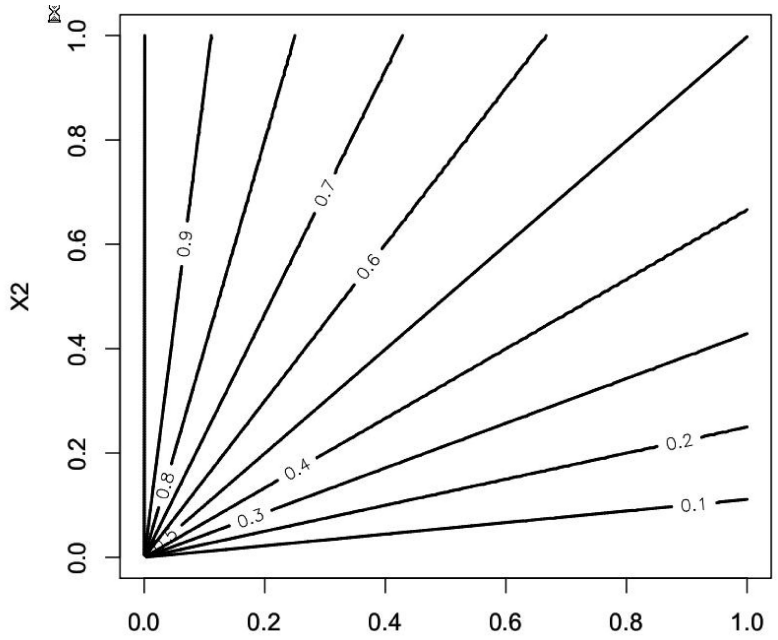
- Playing with α and β gives new ideas...



$\Rightarrow$ Weights (left) and Losses (right) that emphasize $p \approx 0.3$.

Reasons for Wanting to Tailor Losses

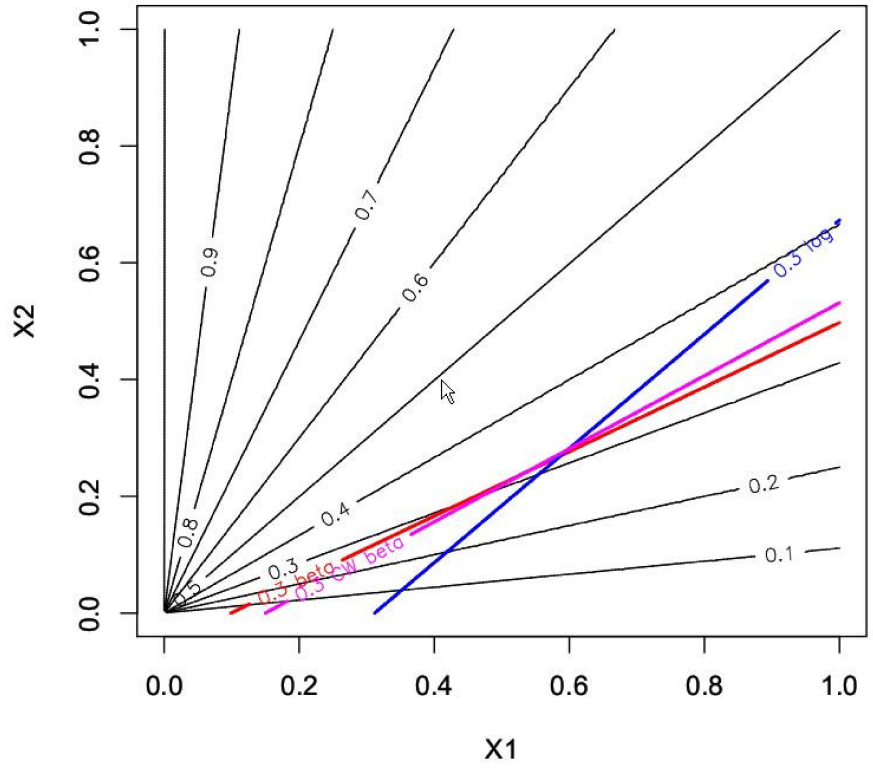
- Hand and Vinciotti (2003) suggest the following example:



- Problem: Each boundary can be described by a linear model,
  but no single linear model can describe all boundaries.

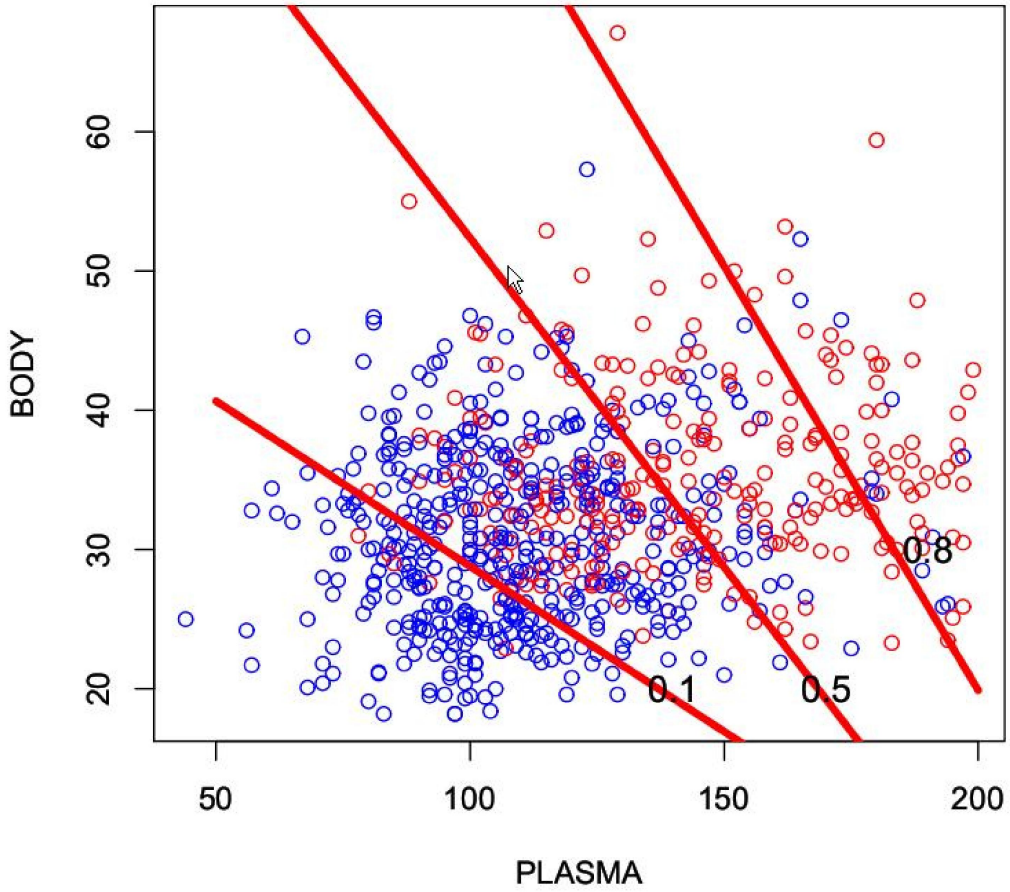
Estimating the Hand-Vinciotti Example with Tailored Losses

- By emphasizing probabilities near 0.3 with $\omega(q)$, one can adapt the linear model to the 0.3 level:



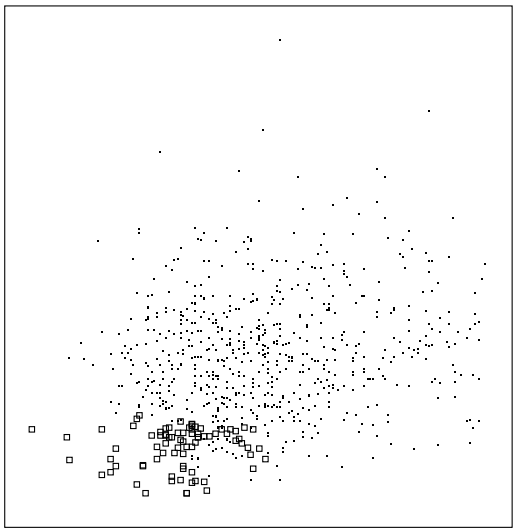
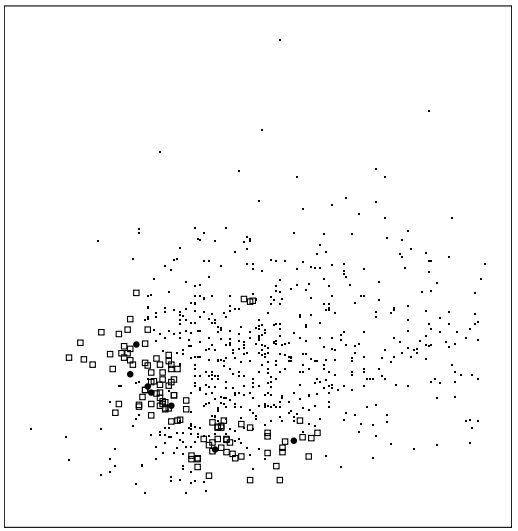
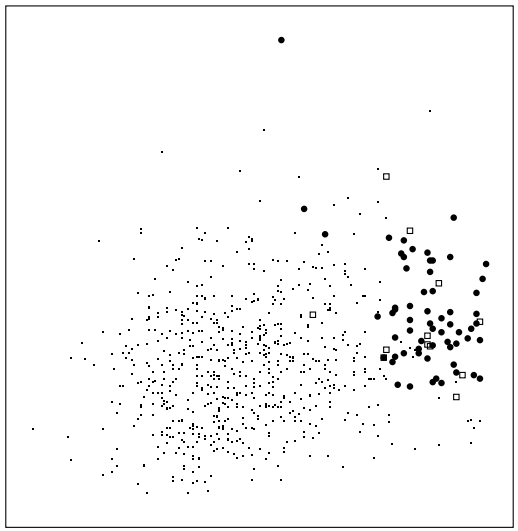
A Real Example of the Hand-Vinciotti Scenario

- Pima Indians diabetes data (UCI ML-DB):
  Predictors 'PLASMA' and 'BODY'



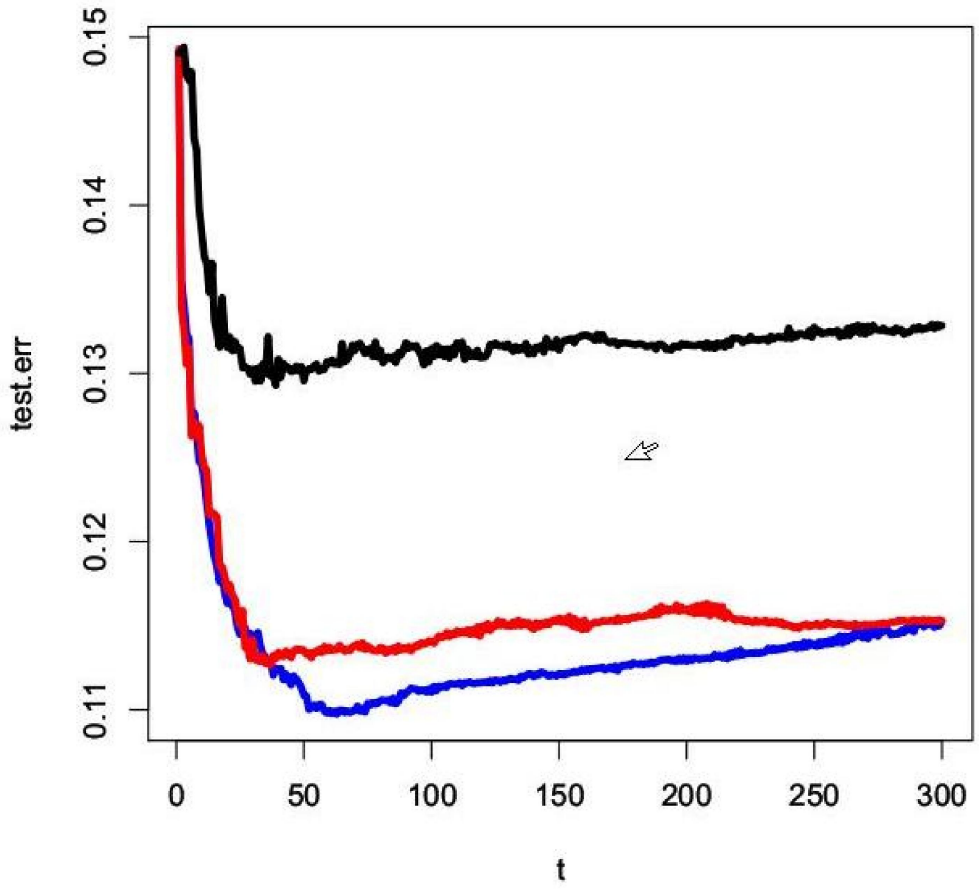
Independent Verification of the Hand-Vinciotti Scenario

- Estimate class probabilities non-parametrically with 20-nearest neighbors, then slice the probability surface at $p = 0.1,\ 0.2,\ 0.9$:

The Hand-Vinciotti Scenario with Boosting

- Tailored boosting to $p = 0.3$ shows considerable improvement over standard boosting when using stumps:



Estimation of Linear Models with Proper Scoring Rules

- Data: $(\mathbf{x}_n, y_n),\ n = 1...N$
- Linear model scores: $F_n = \mathbf{x}_n'\boldsymbol{\beta}$
- Inverse Link: $\psi(F)$ (arbitrary smooth cdf)
- Estimated class probs: $p_n = \psi(F_n), \quad p_n' = \psi'(F_n)$
  (e.g., $\psi'(F) = \psi(F)(1-\psi(F))$ when $\psi() =$ logistic)
- Weight fct: $\omega(p)$
- Proper scoring rule: $\mathbf{L}(y|p) = y\, L_1(1-p) + (1-y)L_0(p)$
- Sample loss: $\mathcal{L}(\boldsymbol{\beta}) \ = \ \sum_{n=1,...,N} \mathbf{L}(y_n|\psi(\mathbf{x}_n'\boldsymbol{\beta}))$
- Coefficient estimate: $\hat{\boldsymbol{\beta}} \ = \ \text{argmin}_{\mathbf{b}}\, \mathcal{L}(\mathbf{b})$

Fisher Scoring: A Reweighting Scheme

- Minimization: Newton iterations $\rightarrow$ Fisher Scoring (using E[Hessian])
- Fisher scoring as '**I**teratively **R**eweighted **L**east **S**quares' (IRLS):
  - Iteratively perform LS regression of the "working response,"
    $$z_n \;=\; (y_n - p_n)/p'_n \,,$$
  - on predictors $\mathbf{x}_n$
  - with weights $\boxed{W_{nn} \;=\; {p'_n}^2 \; \omega(p_n)}$ and
  - class prob estimates $p_n \;=\; \psi(\mathbf{x}'_n\mathbf{b}) \;, \quad p'_n \;=\; \psi'(\mathbf{x}'_n\mathbf{b}).$
- The weight function $\omega(p)$ drives the IRLS weights.
- Curiosity: By boosting logic, misclassified cases would have to be up-weighted.
  Instead, IRLS weights only depend on the current p_n and p'_n.
  $\Rightarrow$ Difference between AdaBoost and LogitBoost

Messages for Classification and Class Prob Estimation

- In practice, first decide: class prob estimation or classification?
  - Class prob estimation allows delaying decisions about cost weights/quantiles.
  - Classification requires up front decisions about cost weights/quantiles.
  - Cost of false positives/negatives requires thought.
- Class prob estimation is more difficult, requires global model $p(\mathbf{x}|\boldsymbol{\beta})$ for $\eta(\mathbf{x})$.

  (Perform cross-validation on the "surrogate loss", e.g., logistic loss or exp loss.)
- Classification at cost/quantile c only requires a model $p(\mathbf{x}|\boldsymbol{\beta})$ that can match $\eta(\mathbf{x}) > c$ well with $p(\mathbf{x}|\boldsymbol{\beta}) > c$ for some $\boldsymbol{\beta}$.

  (Perform cross-validation on cost-weighted classification loss.)
- For good classification the model $p(\mathbf{x}|\boldsymbol{\beta})$ can totally overfit $\eta(\mathbf{x})$.

  (However, such over-fitted models are generally uninterpretable.)
- (Finally, use of predictors must beat the baseline obtained w/o predictors!)

PS: More Basics — Differing Baseline Probs

- Problem: In the initial real example we trained on a sample of 50K small businesses and 50K residences to estimate class probs $p(\mathbf{x})$.
  Q: In reality, the true marginal probs might be more like 1 small business in 20 phone numbers. How should the estimates $p(\mathbf{x})$ be interpreted?
- A: Reweight $p(\mathbf{x})$ from baseline 50:50 to baseline 1:19.
  - Marginal class prob: $\pi = P[Y = 1]$
  - Class-conditionals: $f_1(\mathbf{x}) = P(\mathrm{d}\mathbf{x}|Y = 1)/\mathrm{d}\mathbf{x}\,, \quad f_0(\mathbf{x}) = P(\mathrm{d}\mathbf{x}|Y = 0)/\mathrm{d}\mathbf{x}$
  - Predictor-conditionals: $\eta(\mathbf{x}) = \frac{f_1(\mathbf{x})\pi}{f_0(\mathbf{x})(1-\pi)+f_1(\mathbf{x})\pi}$ (Bayes theorem)
  - In terms of odds: $\frac{\eta(\mathbf{x})}{1-\eta(\mathbf{x})} = \frac{f_1(\mathbf{x})}{f_0(\mathbf{x})}\frac{\pi}{1-\pi}$

  $\Rightarrow$ Reweight class odds estimates from training π to actual π^*:

$$\frac{p^*(\mathbf{x})}{1-p^*(\mathbf{x})} = \frac{p(\mathbf{x})}{1-p(\mathbf{x})} \cdot \frac{1-\pi}{\pi} \cdot \frac{\pi^*}{1-\pi^*}$$

  (This only matters for numeric interpretation of $p(\mathbf{x})$, not ranking.)

PPS: Application to Tree-Based Classification

- Construction of classification trees:
  - Search all predictor variables and split locations on them
  - for the best split as judged by a measure of "impurity".
- Measure of "impurity" in a bucket "B":

  Estimated ave loss of the fitted prob $\hat{\eta} = \sum_{i \in B} y_i / |B|$

$$\frac{1}{|B|} \min_p \sum_{i \in B} \mathbf{L}(y_i|p) \;=\; \min_p \mathbf{L}(\hat{\eta}|p) \;=\; \mathbf{L}(\hat{\eta}|\hat{\eta}) \;=\; \text{Entropy}(\hat{\eta})$$

- Note that **any** measure of entropy can be used:
  - CART uses the Gini index (squared error entropy)
  - S ('tree' package), C4.5 use log-loss entropy
  - We will use tailored entropies.