LOSS FUNCTIONS FOR BINARY CLASSIFICATION AND CLASS PROBABILITY ESTIMATION

YI SHEN

A DISSERTATION

IN

STATISTICS

For the Graduate Group in Managerial Science and Applied Economics

Presented to the Faculties of the University of Pennsylvania

in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2005

Supervisor of Dissertation

Graduate Group Chairperson

TO

MY PARENTS

ABSTRACT

LOSS FUNCTIONS FOR BINARY CLASSIFICATION AND CLASS PROBABILITY ESTIMATION

YI SHEN

SUPERVISOR: ANDREAS BUJA

What are the natural loss functions for binary class probability estimation? This question has a simple answer: so-called "proper scoring rules". These loss functions, known from subjective probability, measure the discrepancy between true probabilities and estimates thereof. They comprise all commonly used loss functions: log-loss, squared error loss, boosting loss (which we derive from boosting's exponential loss), and cost-weighted misclassification losses. —We also introduce a larger class of possibly uncalibrated loss functions that can be calibrated with a link function. An example is exponential loss, which is related to boosting.

Proper scoring rules are fully characterized by weight functions $\omega(\eta)$ on class probabilities $\eta = P[Y = 1]$. These weight functions give immediate practical insight into loss functions: high mass of $\omega(\eta)$ points to the class probabilities η where the proper scoring rule strives for greatest accuracy. For example, both log-loss and boosting loss have poles near zero and one, hence rely on extreme probabilities.

We show that the freedom of choice among proper scoring rules can be exploited when the two types of misclassification have different costs: one can choose proper scoring rules that focus on the cost c of class 0 misclassification by concentrating $\omega(\eta)$

near c. We also show that cost-weighting uncalibrated loss functions can achieve tailoring. "Tailoring" is often beneficial for classical linear models, whereas non-parametric boosting models show fewer benefits.

We illustrate "tailoring" with artificial and real datasets both for linear models and for non-parametric models based on trees, and compare it with traditional linear logistic regression and one recent version of boosting, called "LogitBoost".

Contents

List of Tables

List of Figures

Chapter 1

Introduction

We consider predictor-response data with a binary response y representing the observation of classes $y = 1$ and $y = 0$. Such data are thought of as realizations of a Bernoulli random variable Y with $\eta = P[Y = 1]$ and $1 - \eta = P[Y = 0]$. The class 1 probability η is interpreted as a function of predictors $\mathbf{x}$: $\eta = \eta(\mathbf{x})$. If the predictors are realizations of a random vector $\mathbf{X}$, then $\eta(\mathbf{x})$ is the conditional probability given $\mathbf{x}$: $\eta(\mathbf{x}) = P[Y = 1|\mathbf{X} = \mathbf{x}]$. Of interest are two types of problems:

- *Classification*: Estimate a region in predictor space in which class 1 is observed with the greatest possible majority. This amounts to estimating a region of the form $\eta(\mathbf{x}) > c$.
- *Class probability estimation*: Approximate $\eta(\mathbf{x})$ as well as possible by fitting a model $q(\mathbf{x}, \mathbf{b})$ ($\mathbf{b}$ = parameters to be estimated).

Of the two problems, classification is prevalent in machine learning (where it is sometimes called "concept learning", betraying its origin in AI), whereas class probability estimation is prevalent in statistics (often in the form of logistic regression).

The classification problem is peculiar in that estimation of a class 1 region requires two kinds of criteria:

- The primary criterion of interest: misclassification rate. This is an intrinsically unstable criterion for estimating models, a fact that necessitates the use of
- auxiliary criteria for estimation, such as the Bernoulli likelihood used in logistic regression, and the exponential loss used in the boosting literature. These are just estimation devices and not of primary interest.

The auxiliary criteria of classification are, however, the primary criteria of class probability estimation.

We describe two universes of loss functions that can be used as auxiliary criteria in classification and as primary criteria in class probability estimation:

- One universe consists of loss functions that estimate probabilities consistently or "properly", whence they are called "proper scoring rules". An example is the negative log-likelihood of the Bernoulli model, also called Kullback-Leibler information, log-loss, or cross-entropy (we use "log-loss" throughout).
- The other universe of loss functions is best characterized as producing estimators that are "distorted" or uncalibrated. Why would one need such loss functions? The reason is the same as in logistic regression where we model a "distortion"

of the probabilities, namely, the logit. We call such "distorted" loss functions "F-losses" because their domains are scales to which functions (models) are fitted. An example is the exponential loss used in boosting.

These two universes of loss functions allow us to address a peculiarity of classification: if a classification region is of the form $q(\mathbf{x}, \mathbf{b}) > c$, it is irrelevant for the performance in terms of misclassification rate whether $q(\mathbf{x}, \mathbf{b})$ fits the true $\eta(\mathbf{x})$ well, as long as $q(\mathbf{x}, \mathbf{b}) > c$ and $\eta(\mathbf{x}) > c$ agree well enough. That is, the classifier does not suffer if $q(\mathbf{x}, \mathbf{b})$ is biased vis-à-vis $\eta(\mathbf{x})$ as long as $q(\mathbf{x}, \mathbf{b})$ and $\eta(\mathbf{x})$ are mostly on the same side of c. It can therefore happen that a fairly inaccurate model yields quite accurate classification. In order to take advantage of this possibility, one should choose a loss function that is closer to misclassification rate than log-loss and exponential loss. Squared error loss is more promising in that regard, but better choices of loss functions can be found, in particular if misclassification cost is not equal for the two classes.

The discrepancy between class probability estimation and classification has recently been illustrated by Hand and Vinciotti (2003) with a striking artificial example. It demonstrates how good classification can occur even in the absence of good class probability estimation. The example, shown in Figure 1.1, consists of a function $\eta(\mathbf{x})$ that is a rule surface, meaning that all the level lines are linear yet the function is nonlinear. If we consider the class of linear functions, possibly passed through a nonlinear link function such as the logistic, we will find that they are insufficient for

fitting this surface globally on its domain because the level lines of linear functions as well as their nonlinear transforms are parallel to each other. Yet, for each level $0 < c < 1$ one can find a linear function that describes the level line $\eta(\mathbf{x}) = c$ perfectly. This linear function will be unable to fit $\eta(\mathbf{x})$ globally, hence in order to find it one has to be able to ignore or at least downweight those training data for which $\eta(\mathbf{x})$ is far from c, the level of interest. Hand and Vinciotti (2003) have shown how such downweighting can be done algorithmically, but their suggestion of a modified likelihood (their Equation (2)) is rather tentative. We show how reweighting follows as a direct consequence of the choice of proper scoring rule, and we also introduce techniques for "tailoring" proper scoring rules for this purpose. —On the side we note that we will show in an exercise of exploratory data analysis that Hand and Vinciotti's (2003) scenario approximately holds in the case of the well-known Pima Indians Diabetes data from the UCI Machine Learning Repository (2003).

As a second application of the freedom of choice of loss function, we show how criteria for tree-based classification can be "tailored" if the interest is in small or large class probabilities. This exercise will produce highly interpretable trees that vastly more expressive than the usual trees grown with the Gini index (CART, Breiman et al. 1984) or entropy (e.g.: C4.5, Quinlan 1993; S language, Clark and Pregibon 1992).

Combining tailored losses with flexible non-parametric fits such as boosting models does not achieve a similarly strong benefit as in linear models. We show empirically that as the complexity of the functional classes increases, the effect of tailoring becomes less significant. Thus we conclude that proper scoring rules are best combined

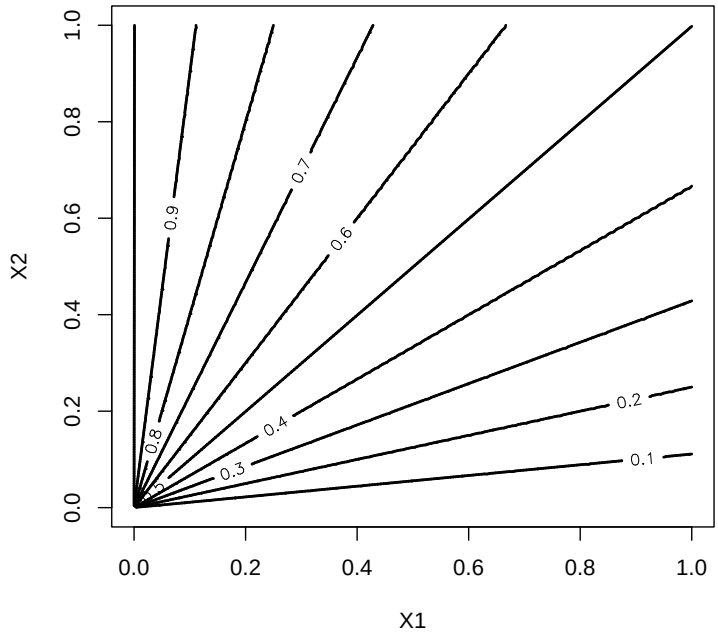

Figure 1.1: *Hand and Vinciotti's (2003) example of a nonlinear response surface* $\eta(\mathbf{x})$ *that permits linear classification boundaries.*

with more rigid models such as linear ones.

For the convenience of the readers, here is a layout of this dissertation:

Chapter 2 introduces the background of cost-weighted binary classification and addresses the need of primary and auxiliary loss functions for this type of problem.

Chapter 3 defines loss functions that are called "*proper scoring rules*" on the probability scale. They are characterized by weight functions $\omega(\eta)$. We then define a family of proper scoring rules that is modeled by the Beta densities with parameters α and β. This family not only is sufficiently rich to encompass most commonly used

losses but can also be tailored for the purposes of cost-weighted classification.

Chapter 4 defines a larger class of loss functions than proper scoring rules that we call "$F - losses$". They are compositions of (inverse) link functions and proper scoring rules. They can also be regarded as "uncalibrated" proper scoring rules in the sense that the estimated values are not probability estimates, but can be calibrated to produce probability estimates by applying a monotone transformation. We then show that F-losses are closed under cost-weighting, thus can also be tailored for the purposes of cost-weighted classification.

Chapters 5 and 6 deal with optimization of proper scoring rules and F-losses. In Chapter 5 we generalize the IRLS algorithm to arbitrary proper scoring rules and F-losses for linear models. In Chapter 6 we generalize stagewise fitting of additive models (="boosting" in Friedman et al.'s (2000) interpretation) to arbitrary proper scoring rules and F-losses. In Chapter 7, we illustrate the application of such algorithms to artificial and real world data sets.

Chapter 2

Two-Class Cost-Weighted Classification

2.1 Cost-Weighting and Quantile Classification

In a two-class classification problem, one observes data $(\mathbf{X}, Y)$, where $\mathbf{X}$ is predictor or feature vector, Y is a binary response which is labeled 0 and 1. The observed Y is regarded as a realization of a binary random variable following a Bernoulli distribution. A problem of interest is to predict or classify future values of Y from the information that the feature $\mathbf{X}$ contains. This amounts to the problem of dividing the predictor space into two regions and classifying a case as 1 or 0 according to which region the associated value of feature $\mathbf{X}$ falls in. Theoretically, the best (Bayes) classification regions for assignment to class 1 are of the form $\{\mathbf{x} \mid \eta(\mathbf{x}) > c\}$, where as before $\eta(\mathbf{x}) = P[Y = 1|\mathbf{x}]$.

The determination of the threshold c is related to how much we weight the relative cost of the two types of misclassification: misclassifying $Y = 1$ as 0 and $Y = 0$ as 1, respectively. In medical contexts, for example, a false negative (missing a disease) is usually more severe than a false positive (falsely detecting a disease). If the cost for class 1 misclassification is c_1 and the cost for class 0 misclassification is c_0, then according to Bayes rule the optimal choice of c is $c = c_0/(c_0 + c_1)$. Without loss of generality, one can always normalize the sum of c_0 and c_1 to 1: $c_0 + c_1 = 1$. The threshold $c = c_0$ is then the cost of class 0 misclassification. In those contexts where costs are not mentioned, equal cost of misclassification is assumed, that is, $c = 0.5$. Bayes rule is optimal in the sense of minimizing the cost of misclassification: the expected costs of classifying as 1 and 0 are, respectively, $c(1 - \eta(\mathbf{x}))$ and $(1 - c)\eta(\mathbf{x})$. Hence optimal classification as 1 is for $c(1 - \eta(\mathbf{x})) < (1 - c)\eta(\mathbf{x})$ or, equivalently, $c < \eta(\mathbf{x})$. Thus, classification with normalized cost weight c is equivalent to "quantile classification" at the quantile c.

In reality the posterior probability $\eta(\mathbf{x})$ is unknown and has to be estimated from the data. By convention, we use $q(\mathbf{x})$ to denote an estimate of $\eta(\mathbf{x})$. Therefore, one uses an estimated classification region $q(\mathbf{x}) > c$ to approximate the true Bayes region $\eta(\mathbf{x}) > c$.

2.2 Cost-Weighting and Change of Baseline

A frequent confusion in classification concerns the roles of cost-weighting and change of baseline frequencies of the two classes. For the convenience of the reader we show how a baseline change affects the cost-weights or quantiles c (see, e.g., Elkan (2001)). We use the following notation:

$$\begin{aligned} \eta(x) &= P[Y = 1 | \, x \,] \, , \\ \pi &= P[Y = 1] \\ f_1(x) &= P[\, x \, | Y = 1] \, , \\ f_0(x) &= P[\, x \, | Y = 0], \end{aligned}$$

which are, respectively, the conditional probability of $Y = 1$ given $X = x$, the unconditional (marginal/prior/baseline) probability of $Y = 1$, and the conditional densities of $X = x$ given $Y = 1$ and $Y = 0$. The densities $f_{1/0}(x)$ describe the distributions of the two classes in predictor space. The joint distribution of (X, Y) is given by

$$\eta(y = 1, x) = f_1(x) \, \pi \ , \qquad \eta(y = 0, x) = f_0(x) \, \pi \ ,$$

and hence the conditional distribution of $Y = 1$ given $X = x$ is

$$\eta(x) \ = \ \frac{f_1(x) \, \pi}{f_1(x) \, \pi + f_0(x) \, (1 - \pi)} \ ,$$

which is just Bayes theorem. Equivalently we have

$$\frac{\eta(x)}{1 - \eta(x)} / \frac{\pi}{1 - \pi} \ = \ \frac{f_1(x)}{f_0(x)} \ . \tag{2.1}$$

We then compare this situation with another one that differs only in the mix of class labels, π^* and $1-\pi^*$, as opposed to π and $1-\pi$. Denote by $\eta^*(x)$ the conditional probability of $Y=1$ given $X=x$ under this new mix. From Equation (2.1) follows that $f_1(x)$ and $f_0(x)$ are the same in both situations, hence

$$\frac{\eta^*(x)}{1-\eta^*(x)}/\frac{\pi^*}{1-\pi^*} \;=\; \frac{\eta(x)}{1-\eta(x)}/\frac{\pi}{1-\pi}\;,$$

or, equivalently,

$$\eta^*(x) \;=\; \frac{k\eta}{k\eta+(1-k)(1-\eta)}\;,$$

where

$$k \;=\; \frac{\frac{\pi*}{1-\pi^*}}{\frac{\pi^*}{1-\pi^*}+\frac{\pi}{1-\pi}} \;=\; \frac{\pi^*(1-\pi)}{\pi^*(1-\pi)+(1-\pi^*)\pi}\;.$$

Obviously thresholds c on $\eta(x)$ and c^* on $\eta^*(x)$ transform the same way:

$$c^* \;=\; \frac{kc}{kc+(1-k)(1-c)}\;,$$

so that $\eta^*(x) > c^*$ iff $\eta(x) > c$.

2.3 Cost-Weighted Misclassification Error Loss

Our goal is to find classification rules whose cost-weighted misclassification error for given costs c and $1-c$ is small. We assume that classification is performed by thresholding an estimate $q(\mathbf{x})$ of $\eta(\mathbf{x})$: the estimated class is $1_{q(\mathbf{x})>c}$. We encounter an error if $Y=1$ and $q(\mathbf{x}) \le c$, or if $Y=0$ and $q(\mathbf{x}) > c$. Hence the cost-weighted misclassification loss at $\mathbf{x}$ can be written as

$$\mathbf{L}(Y|q(\mathbf{x})) \;=\; (1-c)Y1_{[q(\mathbf{x})\le c]} + c(1-Y)1_{[q(\mathbf{x})>c]}\;.$$

When the costs of the two types of misclassification are the same, $c = 1 - c = .5$, this is up to a factor $1/2$ the plain misclassification error:

$$\mathbf{L}(Y|q(\mathbf{x})) \;=\; 0.5 \cdot \big(Y 1_{[q(\mathbf{x})\leq 0.5]} + (1-Y) 1_{[q(\mathbf{x})>0.5]}\big)$$

At the population level, the conditional expected misclassification loss is:

$$E_{Y|\mathbf{X}=\mathbf{x}}\, \mathbf{L}(Y|q(\mathbf{x})) \;=\; (1-c)\, \eta(\mathbf{x}) 1_{[q(\mathbf{x})\leq c]} \;+\; c\,(1-\eta(\mathbf{x})) 1_{[q(\mathbf{x})>c]}$$

This is the expected loss associated with classification for a given value of $\mathbf{X}$. Unconditionally, the cost-weighted misclassification loss or *risk* is:

$$E_{Y,\mathbf{X}}\, (\mathbf{L}(Y|q(\mathbf{X}))) \;=\; (1-c)\; E_{\mathbf{X}}[\, \eta(\mathbf{X}) 1_{[q(\mathbf{X})\leq c]}] \;+\; c\, E_{\mathbf{X}}[(1-\eta(\mathbf{X})) 1_{[q(\mathbf{X})>c]}]$$

One would like to find estimators $q(\mathbf{x})$ that perform well in the sense that they produce a small risk. There is, however, a problem: misclassification loss is a crude measure and does not distinguish between class 1 probability estimates $q(\mathbf{x})$ as long as they are on the same side of c. Hence optimal estimators $q(\mathbf{x})$ are highly non-unique. In many application one can make an argument that one would like $q(\mathbf{x})$ to be a good estimator of $\eta(\mathbf{x})$, not just $q(\mathbf{x}) > c$ to be a good estimator of $\eta(\mathbf{x}) > c$. This is the case when the cost c is not precisely defined, or as in some marketing application when potential customers need to be prioritized for marketing campaigns according to their estimated probability of taking a product.

There is yet another problem with misclassification error: It is possible that two estimators have the same misclassification error yet of them is preferable to the human

expert. For example, consider the following two pairs or misclassification counts:

$$\begin{array}{cc} & \\ \text{Actual} & \begin{array}{c}\text{Class 1}\\ \text{Class 0}\end{array} \end{array} \begin{bmatrix} * & 0 \\ 30 & * \end{bmatrix} \qquad \begin{bmatrix} * & 15 \\ 15 & * \end{bmatrix}$$

Both situations have the same unweighted misclassification error, yet they differ strongly. It may help to consider the cost-weighted loss for each cost ratio: for the first table it is $30 \cdot c + 0 \cdot (1-c) = 30c$, for the second table it is $15 \cdot c + 15 \cdot (1-c) = 15$. Thus, if there is a potential of considering a class-0 cost c less than 0.5, one would clearly prefer the first table, otherwise the second. As we will see, the criteria introduced below will always prefer the first table over the second.

2.4 Some Common Losses

In Section 2.3 we illustrated some problems with misclassification loss. These problems arise from the crude nature of the loss function, one prefers to optimize over loss functions with greater discriminatory properties. We give two standard examples:

Example 1. In logistic regression, one uses log-loss, (equivalently: Kullback-Leibler loss, cross-entropy, or the negative log-likelihood of the Bernoulli model):

$$\mathbf{L}(y|q) = -y \log(q) - (1-y) \log(1-q)$$

with the corresponding expected loss

$$\mathbf{R}(\eta|q) = -\eta \log(q) - (1-\eta) \log(1-q)$$

Example 2. Squared error loss

$$\mathbf{L}(y|q) \;=\; (y-q)^2 \;=\; y(1-q)^2 + (1-y)q^2 \;,$$

where the second equality holds because y takes only the values 0 and 1. The expected squared error loss is

$$\mathbf{R}(\eta|q) \;=\; \eta(1-q)^2 + (1-\eta)q^2 \;.$$

These two loss functions are smooth and they both produce particular solutions of optimal estimates q. Moreover, one can quickly check that the optimal estimate q is actually achieved at the true posterior probability η when minimizing the expected losses. Loss functions $\mathbf{L}(y|q)$ with this property have been known in subjective probability as "*proper scoring rules*". (See Garthwaite, Kadane and O'Hagan (2005, Sec. 4.3) for recent work in this area.) There they are used to judge the quality of probability forecasts by experts, whereas here they are used to judge the fit of class probability estimators. Proper scoring rules form a natural universe of loss functions for use as criteria for class probability estimation and classification.

Chapter 3

Proper Scoring Rules

3.1 Definition and Examples of Proper Scoring Rules

Definition: *If an expected loss* $\mathbf{R}(\eta|q) = \eta L_1(1-q) + (1-\eta)L_0(q)$ *is minimized w.r.t.* q *by* $q = \eta \ \forall \eta \in (0,1)$*, we call the loss function a "proper scoring rule". If moreover the minimum is unique, we call it a "strictly proper scoring rule".*

According to the definition, we can easily see that the two examples in Section 2.4 are actually strictly proper scoring rules.

- For the expected log-loss, we have the first order stationarity condition

$$-\frac{\eta}{q} - \frac{1-\eta}{1-q} = 0$$

  which has the unique solution $q = \eta$.

- For the expected squared error loss, we have the first order stationarity condition

$$-2\eta(1-q) + 2(1-\eta)q = 0$$

which has the unique solution $q = \eta$.

It is also easy to see that misclassification loss is a proper scoring rule yet not strict since the optimal solution q is not unique: any q that falls to the right side of the true classification boundary $\eta = c$ is considered optimal.

3.2 Characterizing Proper Scoring Rules

In this section we give a characterization of proper scoring rules that goes back to Shuford, Albert, Massengill (1966), Savage (1971) and in its most general form Schervish (1989).

We write the stationarity condition under which classification loss functions $\mathbf{L}(y|q)$ form proper scoring rules, and we will characterize them in terms of "weight functions". Writing the loss function as

$$\mathbf{L}(y|q) \;=\; yL_1(1-q) + (1-y)L_0(q)$$

we have:

Proposition 1: *Let $L_1(\cdot)$ and $L_0(\cdot)$ be monotone increasing functions. If $L_1(\cdot)$ and $L_0(\cdot)$ are differentiable, then $\mathbf{L}(y|q)$ forms a proper scoring rule iff*

$$L_1'(1-q) \;=\; \omega(q)(1-q), \qquad L_0'(q) \;=\; \omega(q)q \tag{3.1}$$

If $\omega(q)$ is strictly positive on $(0,1)$, then the proper scoring rule is strict.

The proof follows immediately from the stationarity condition,

$$0 \;=\; \frac{\partial}{\partial q}\bigg|_{q=\eta} \mathbf{L}(\eta|q) \;=\; -\,\eta\, L_1'(1-\eta) + (1-\eta)\, L_0'(\eta)\ .$$

This entails

$$\frac{L_1'(1-q)}{1-q} \;=\; \frac{L_0'(q)}{q}\ ,$$

which defines the weight function $\omega(q)$.

Proposition 1 reveals the possibility of constructing new proper scoring rules by choosing appropriate weight functions $\omega(q)$. Here are two standard examples of such choices:

- Log-loss:

$$\omega(q) = \frac{1}{q(1-q)}$$

- Squared error loss:

$$\omega(q) = 1$$

Misclassification loss does not fit in this framework because its losses $L_0(q) = c\,1_{[q>c]}$ and $L_1(1-q) = (1-c)\,1_{[q\le c]}$ are not smooth. Yet there exists an extension of the above proposition that allows us to write down a "generalized weight function":

$$\omega(q) = \delta_c(dq)$$

We conclude with the observation that in general proper scoring rules do not allow cost-weighting. That is, if $\eta L_1(1-q) + (1-\eta)L_0(q)$ defines a proper scoring rule,

then $(1-c)\eta L_1(1-q) + c(1-\eta)L_0(q)$ in general loses the proper scoring property. Cost-weighted misclassification error is a very special case in this regard.

3.3 The Beta Family of Proper Scoring Rules

We exemplify the idea of defining proper scoring rules with weight functions by proposing a 2-parameter family that is sufficiently rich to encompass most commonly used losses, among them log-loss, squared error loss, and misclassification loss in the limit. This family is modeled after the Beta densities:

$$\omega(q) = q^{\alpha-1}\,(1-q)^{\beta-1}\ .$$

The losses are hence defined by

$$L_1'(1-q) \;=\; q^{\alpha-1}(1-q)^{\beta}\ , \qquad L_0'(q) \;=\; q^{\alpha}(1-q)^{\beta-1}\ ,$$

up to an irrelevant multiplicative constant. Here is a list of special cases:

- $\alpha = \beta = -1/2$: Boosting loss will be introduced in Section 4.1.

$$L_1(1-q) = \left(\frac{1-q}{q}\right)^{1/2}\ , \qquad L_0(q) = \left(\frac{q}{1-q}\right)^{1/2}\ .$$

- $\alpha = \beta = 0$: Log-loss or negative log-likelihood of the Bernoulli model,

$$L_1(1-q) = -\log(q)\ , \qquad L_0(q) = -\log(1-q)\ .$$

- $\alpha = \beta = 1/2$: A new type of loss, intermediate between log-loss and squared error loss,

$$L_1(1-q) = \arcsin((1-q)^{1/2}) - (q(1-q))^{1/2}\ , \quad L_0(q) = \arcsin(q^{1/2}) - (q(1-q))^{1/2}\ .$$

- $\alpha = \beta = 1$: Squared error loss,

$$L_1(1-q) = (1-q)^2 \ , \qquad L_0(q) = q^2 \ .$$

- $\alpha = \beta = 2$: A new loss closer to misclassification than squared error loss,

$$L_1(1-q) = \frac{1}{3}(1-q)^3 - \frac{1}{4}(1-q)^4 \ , \qquad L_0(q) = \frac{1}{3}q^3 - \frac{1}{4}q^4 \ .$$

- $\alpha = \beta \to \infty$: The misclassification error rate,

$$L_1(1-q) = 1_{[1-q>1/2]} \ , \qquad L_0(q) = 1_{[q \geq 1/2]} \ .$$

Values of α and β that are integer multiples of $1/2$ permit closed formulas for L_1 and L_0. For other values one needs a numeric implementation of the incomplete Beta function.

Figure 3.1 and Figures 3.2 show proper scoring rules derived from the Beta family with various parameters α and β for different costs.

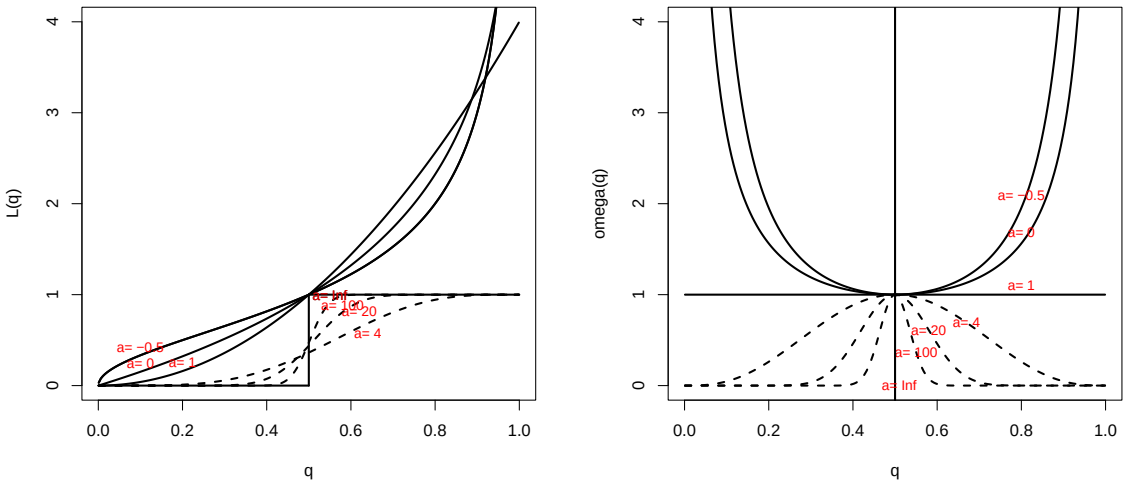

Figure 3.1: *Loss functions $L_0(q)$ and weight functions $\omega(q)$ for various values of $\alpha = \beta$: exponential loss ($\alpha = -1/2$), log-loss ($\alpha = 0$), squared error loss ($\alpha = 1$), misclassification error ($\alpha = \infty$). These are scaled to pass through 1 at $q = 0.5$. Also shown are $\alpha = 4$, 20 and 100 scaled to show convergence to the step function.*

3.4 Tailoring Proper Scoring Rules for Cost-Weighted Classification

Recall that in cost-weighted classification one associates two types of misclassification with different costs. Suppose c is the cost of class 0 misclassification and $1 - c$ is the cost of class 1 misclassification, then the cost-weighted misclassification loss is the measure of accuracy:

$$\mathbf{R}_c(\eta|q) \;=\; (1-c)\,\eta\,1_{[q\le c]} \;+\; c\,(1-\eta)\,1_{[q>c]}$$

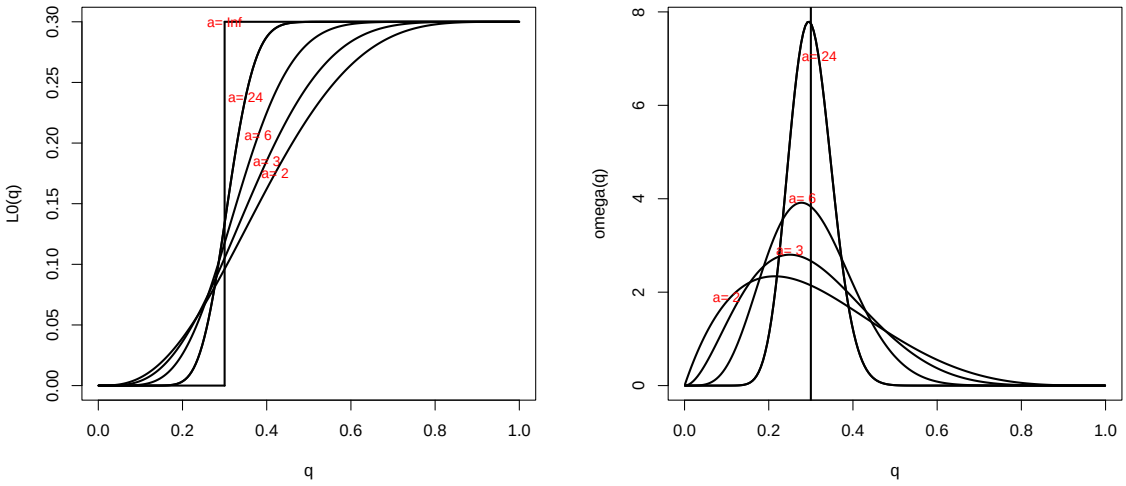

Figure 3.2: *Loss functions $L_0(q)$ and weight functions $\omega(q)$ for various values of $\alpha/\beta = 3/7$, and $c = 0.3$: Shown are $\alpha = 2$, 3 ,6 and 24 scaled to show convergence to the step function.*

Cost-weighted misclassification loss is characterized by the "generalized weight function" $\omega(q) = \delta_c$, a spike at c. The discreteness of this "weight function" is another expression for the problems this loss function has and why it is not successful for forming an empirical risk in estimation from data. Instead, one uses "smoother" criteria such as log-loss or squared error loss which penalized deviations of estimates $q(\mathbf{x})$ from the truth $\eta(\mathbf{x})$ at all levels, not only at c.

Smooth criteria, however, have the problem that they are not particularly adapted to the actual problem at hand: classification at the threshold c. Instead, as for example the weight function $\omega(q) = 1/[q(1-q)]$ of the log-loss shows, common smooth criteria rely on estimated probabilities q near 0 and 1, and not those in the vicinity

of c which is the locus of interest. This consideration motivates our examination of continuous losses other than log-loss and squared error loss, losses that may be better adapted to particular thresholds c of interest. Intuitively, we look for losses with smooth weight functions that approximate the spike δ_c, or at least shift the mass of $\omega(q)$ off-center to reflect the cost-weights c and $1-c$.

With this road map, it is natural to turn to the Beta family of weight functions because Beta distributions are sufficiently rich to approximate point masses δ_c at any location $0 < c < 1$. The Beta densities

$$\omega_{\alpha,\beta}(q) = \frac{1}{\mathrm{B}(\alpha,\beta)} q^{\alpha-1}(1-q)^{\beta-1}$$

converge to $\omega(q) = \delta_c(q)$, for example, when

$$\alpha,\ \beta \to \infty\ , \qquad \text{subject to} \quad \frac{\alpha}{\beta} \ = \ \frac{c}{1-c}\ .$$

For a proof note that the conditions force the expectation to $\mu = c$ while the variance converges to zero:

- The expected value of q under a Beta distribution with parameters α and β is

  $$\mu = \frac{\alpha}{\alpha+\beta}\ ,$$

  which equals c if $c/(1-c) = \alpha/\beta$.

- The variance is

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

  which converges to zero if $\alpha,\ \beta \to \infty$.

In a limiting sense, the ratio of the exponents, α/β, acts as a cost ratio for the classes.

One could also tailor in a slightly different way. Instead of matching the parameters α and β to the mean, one could match to the mode:

$$c \;=\; q_{mode} \;=\; \frac{\alpha - 1}{\alpha + \beta - 2}$$

With either mean or mode matching, we obtain weak convergence $\omega(q)dq \to \delta_c(dq)$.

The above construction has obvious methodological implications for logistic regression and also, as will be shown later, for boosting: log-loss and exponential loss, which they use, respectively, can be replaced with the proper scoring rules generated by the above weight functions $\omega_{\alpha,\beta}(q)$. In doing so we may be able to achieve improved classification for particular cost ratios or class-probability thresholds when the fitted model is biased but adequate for describing classification boundaries individually. The appropriate degree of peakedness of the weight function can be estimated from the data. See Chapter 7 for examples.

3.5 Application: Tree-Based Classification with Tailored Losses

There exists a link between proper scoring rules and information criteria used in tree-based classification, such as entropy and the Gini index. In fact, every proper scoring rule has a unique associated information criterion as follows:

$$\mathbf{H}(\eta) \;=\; \min_q \mathbf{R}(\eta|q) \;=\; \mathbf{R}(\eta|\eta)\ .$$

Tree algorithms agree with each other in that they all estimate local conditional class probabilities with simple proportions, but they differ in how they judge the fit of these proportions in terms of information measures **H**. Here are the usual examples:

- $\alpha = \beta = -1/2$: Boosting loss leads to a semi-circle criterion,

$$\mathbf{H}(q) = 2 \cdot [q(1-q)]^{1/2} \ .$$

- $\alpha = \beta = 0$: Log-loss leads to entropy,

$$\mathbf{H}(q) = -q \log(q) - (1-q) \log(1-q) \ .$$

- $\alpha = \beta = 1$: Squared error loss leads to the Gini index,

$$\mathbf{H}(q) = q(1-q) \ .$$

- $\alpha,\ \beta \to \infty,\ \frac{\alpha}{\beta} \to \frac{c}{1-c}$: Cost-weighted misclassification loss leads to cost-weighted Bayes risk:

$$\mathbf{H}_c(q) = \min((1-c)\, q,\, c\,(1-q)) \ . \tag{3.2}$$

The Gini index is used in CART (Breiman et al. 1984) and entropy in C4.5 (Quinlan 1993) and the original tree functions in the S language (Clark and Pregibon 1992). The semi-circle criterion was proposed by Kearns and Mansour (1996) as a criterion for tree construction. Cost-weighted Bayes risk is not a strict information measure in the same sense that cost-weighted misclassification loss is not a strict proper scoring rule.

The following proposition shows that every information measure determines essentially a unique proper scoring rule, modulo some arbitrary choices for those q for which the function $\mathbf{H}(q)$ does not have a unique tangent, as for cost-weighted misclassification losses at the location $q = c$.

Proposition: *The information measure* $\mathbf{H}(q)$ *is the concave lower envelope of its proper scoring rule* $\mathbf{R}(\eta|q)$*, with upper tangents* $\eta \mapsto \mathbf{R}(\eta|q)$*. Conversely, if* $\mathbf{H}(q)$ *is an arbitrary concave function, it determines a proper scoring rule* $\mathbf{R}(\eta|q)$ *which is unique except at locations where there are multiple upper tangents, in which case any of the tangents can be selected as part of the definition of the proper scoring rule.*

Corollary: *If* $\mathbf{H}(q)$ *is concave and smooth, its associated proper scoring rule is*

$$L_1(1-q) \;=\; \mathbf{H}(q) + \mathbf{H}'(q)\,(1-q)\;, \qquad L_0(q) \;=\; \mathbf{H}(q) - \mathbf{H}'(q)\,q\;. \tag{3.3}$$

For a **proof**, note that $\mathbf{R}(\eta|q)$ is affine in η and $\mathbf{R}(\eta|q) \geq \mathbf{H}(\eta)$, which makes $\eta \mapsto \mathbf{R}(\eta|q)$ an upper tangent for all q and hence $\mathbf{H}(q)$ the concave lower envelope. Conversely, given a concave function $\mathbf{H}(q)$, there exist upper tangents at every q. They are affine functions and can be written $\eta \mapsto \eta\, T_1(q) + (1-\eta)\, T_0(q)$, so one can define $L_1(1-q) = T_1(q)$ and $L_0(q) = T_0(q)$. For the corollary, the upper tangent at q is unique and can be written as an affine function $\eta \mapsto \mathbf{H}'(q)(\eta - q) + \mathbf{H}(q)$. Hence $\mathbf{R}(\eta|q) = \mathbf{H}'(q)(\eta - q) + \mathbf{H}(q)$ defines the proper scoring rule: $L_1(1-q) = \mathbf{R}(1|q)$ and $L_0(q) = \mathbf{R}(0|q)$.

Proposition: *If* $\mathbf{H}(q)$ *is concave and sufficiently smooth, the weight function* $\omega(q)$ *of the associated proper scoring rule is:*

$$\omega(q) \;=\; -\,\mathbf{H}''(q)\;. \tag{3.4}$$

The proof is a simple calculation. The relation links concavity and non-negative weights, as well as strict concavity and positive weights. The proposition indicates that information measures with the same second derivative are equivalent. In other words, information measures that differ only in an affine function are equivalent: $\mathbf{H}(q) + C_1\,q + C_0\,(1-q)$. This fact follows for all information measures, not only smooth ones, from the equivalence of proper scoring rules that differ in two constants only, $L_1(1-q)+C_1$ and $L_0(q)+C_0$.

Tailored classification trees: In light of these facts it is natural to apply the tailoring ideas of Section 3.4 to information measures for tree-based classification. The idea is again to put weight on the class probabilities that are of interest. "Weight" is meant literally in terms of the weight function $\omega(q)$. In practice, areas of interest are often the extremes with highest or lowest class probabilities. In the Pima Indians diabetes data, for example, this may mean focusing on the cases with an estimated probability of diabetes of 0.9 or greater. It would then be reasonable to use an information measure derived from a weight function that puts most of its mass on the right end of the interval (0,1). We will show experiments with weights that are simple power functions of q. In terms of the Beta family of weights this could mean using $\beta = 1$ and $\alpha > 1$: $\omega(q) \sim q^{\alpha-1}$. An associated information measure is

$\mathbf{H}(q) \sim -q^{\alpha+1}$, but taking advantage of the indeterminacy of information measures modulo affine functions, an equivalent but more pleasing choice is $\mathbf{H}(q) = (1 - q^{\alpha})q$, which is normalized such that $\mathbf{H}(0) = \mathbf{H}(1) = 0$.

The result of such an experiment for the Pima Indians Diabetes data (UCI ML Database). Figure 3.5 shows a tree grown with the Gini index. The depth to which it is grown and the quality of the fit is not of concern; instead, we focus on interpretability. This tree shows the usual relative balance whereby most splits are not more lopsided than about 1:2 in bucket size. Overall, interpretation is not easy, at least in comparison to the trees shown in the following two figures. The latter were obtained with information criteria from the Beta family. The tree in Figure 3.5 is based on the parameters $\alpha = 16$ and $\beta = 1$, which means a strong focus on large class 1 probabilities (more correctly: class 1 frequencies). By contrast, Figure 3.5 is based on $\alpha = 1$ and $\beta = 31$, hence a strong focus on small class 1 probabilities.

The focus on the upper and the lower end of probabilities in these two trees really amounts to prioritizing the splits from the bottom up and from the top down, respectively. Accordingly, Figure 3.5 shows a highly unbalanced tree that peels off small terminal leafs with low class 1 probability. As higher level terminal leafs peel off the highest class 1 probability areas, subsequent terminal leafs must consequently have ever lower class 1 probabilities. The result is a cascading tree that layers the data as well as possible from highest to lowest class 1 probabilities. The dual effect is seen in Figure 3.5: a layering of the data according to increasing class 1 probabilities.

While it may be true that the lopsided focus of the criterion is likely to lead to suboptimal trees for standard prediction, cascading trees are vastly more powerful in terms of interpretation: they permit, for example, the expression of dose-response effects and generally of monotone relationships between predictors and response. For example, both trees feature "plasma" as the single most frequent splitting variable which seems to correlate positively with the probability of diabetes: as we descend the trees and as the splitting values on "plasma" decrease in Figure 3.5 and increase in Figure 3.5, the class 1 probabilities decrease and increase, respectively. In Figure 3.5 the variable "b.mass" asserts itself as the second most frequent predictors, and again a positive relation with class 1 probabilities can be gleaned from the tree. Similarly, Figure 3.5 exhibits positive dependences for "age" and "pedigree" as well.

A second aspect that helps the interpretation of cascading trees is the fact that repeat appearances of the same predictors lower the complexity of the description of low hanging leafs. For example, in Figure 3.5 the right most leaf at the bottom with the highest class 1 probability $q = 0.91$ is of depth nine, yet it does not require nine inequalities to describe it. Instead, the following four suffice: "$plasma > 155$", "$pedigree > 0.3$", "$b.mass > 29.9$" and "$age > 24$".

Interestingly the tree of Figure 3.5 that peels off low class 1 probabilities ends up with a crisper high-probability leaf than the tree of Figure 3.5 whose greedy search for high class 1 probabilities results only in an initial leaf with $q = 0.86$ characterized by the single inequality "$plasma > 166$".

We reiterate and summarize: the extreme focus on high or low class 1 probabilities cannot be expected to perform well in terms of conventional prediction, but it may have merit in producing trees with vastly greater interpretability.

The present approach to interpretable trees produces results that are similar to an earlier proposal by Buja and Lee (2001). This latter approach is based on splitting that maximizes the larger of the two class 1 probabilities. The advantage of the tailored trees introduced here is that there actually exists a criterion that is being minimized, and tree-performance can be measured in terms of this criterion.

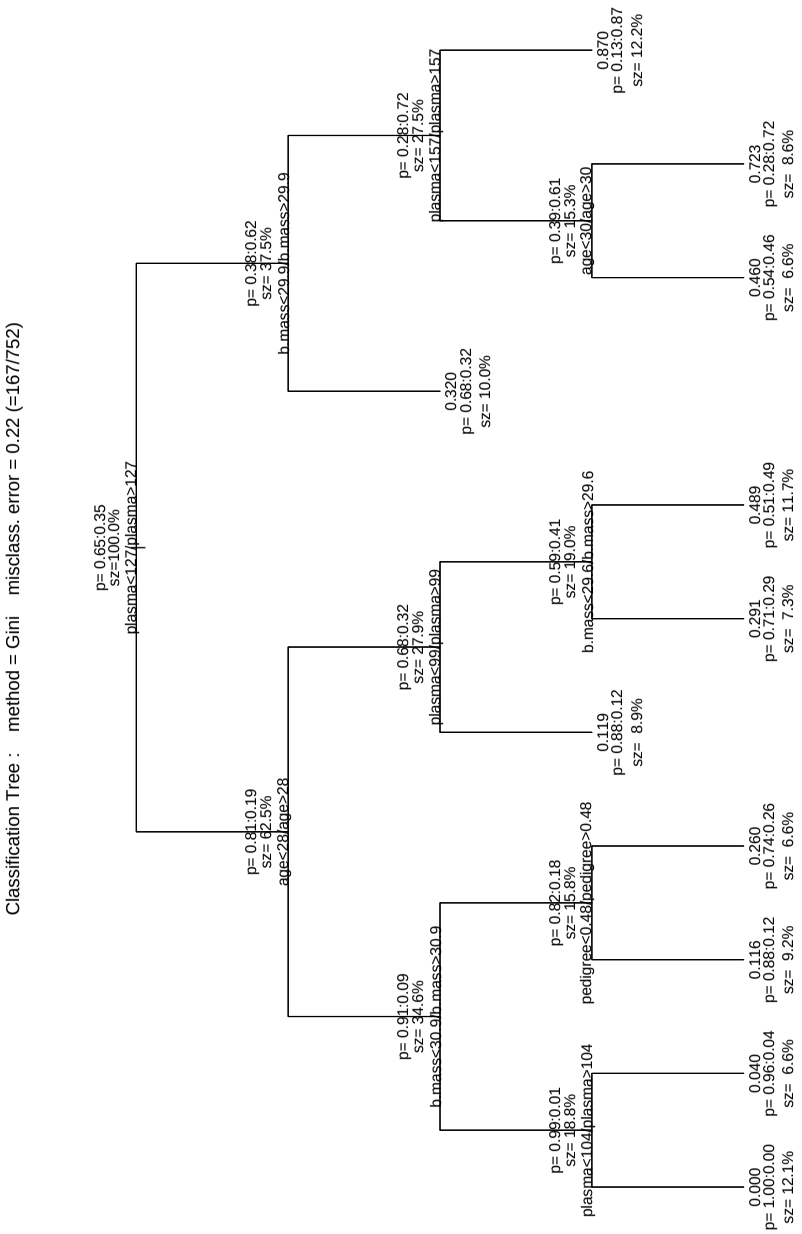

Figure 3.3: *Tree based on the Gini criterion, as in CART. Each split shows the predictor variable and the threshold that were used. Each node shows the fraction of class 0 and class 1 instances (for example: "p=0.65:0.35") and the size of the node (for example: "sz=100%"). The terminal leafs also show the final fitted class 1 probability, which is redundant with the second number shown in the "p=....." clause.*

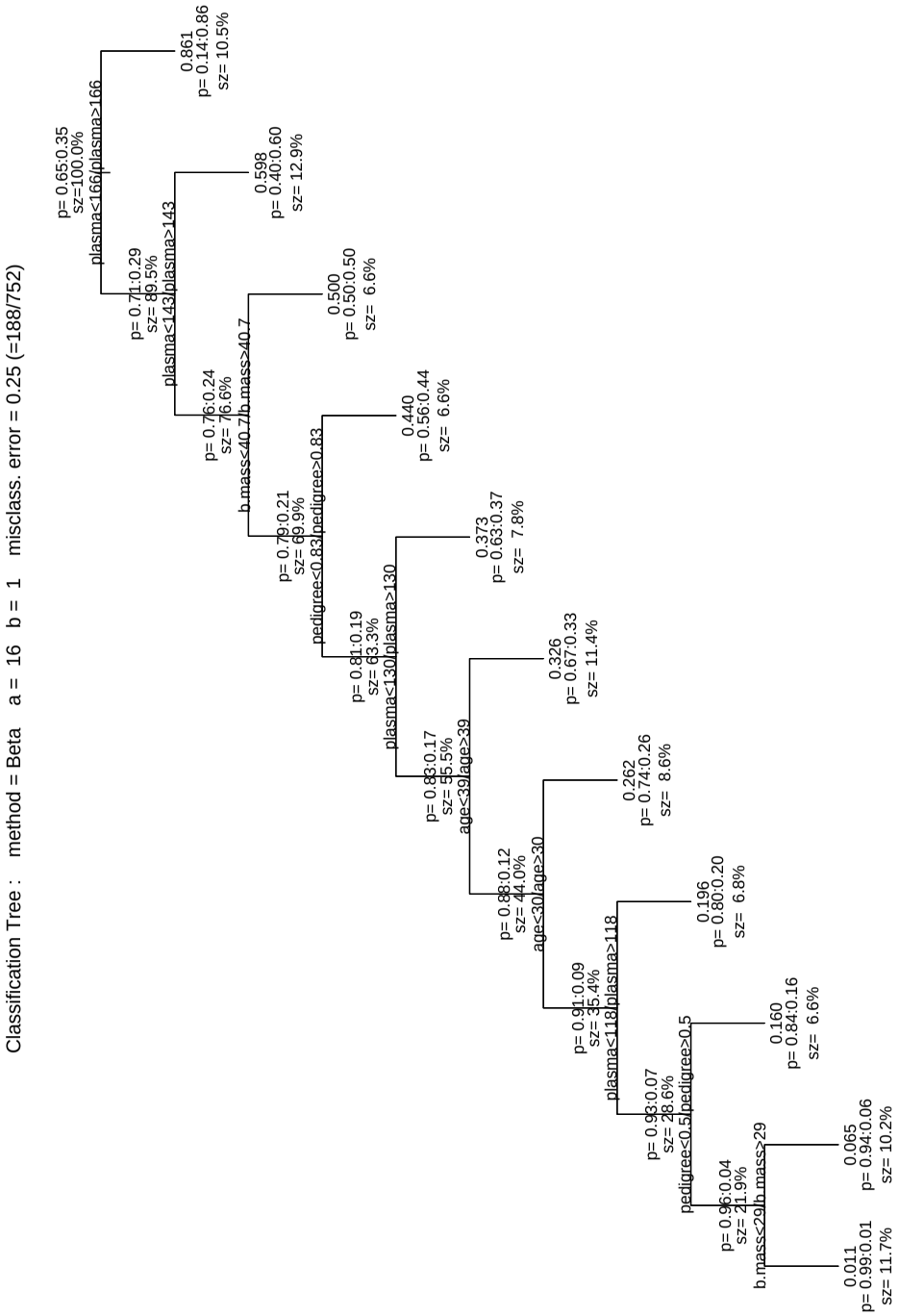

Figure 3.4: *Tree based on the Beta criterion with parameters* $\alpha = 16$ *and beta* $= 1$*. Top to bottom, the tree splits of leafs with decreasing class 1 probabilities.*

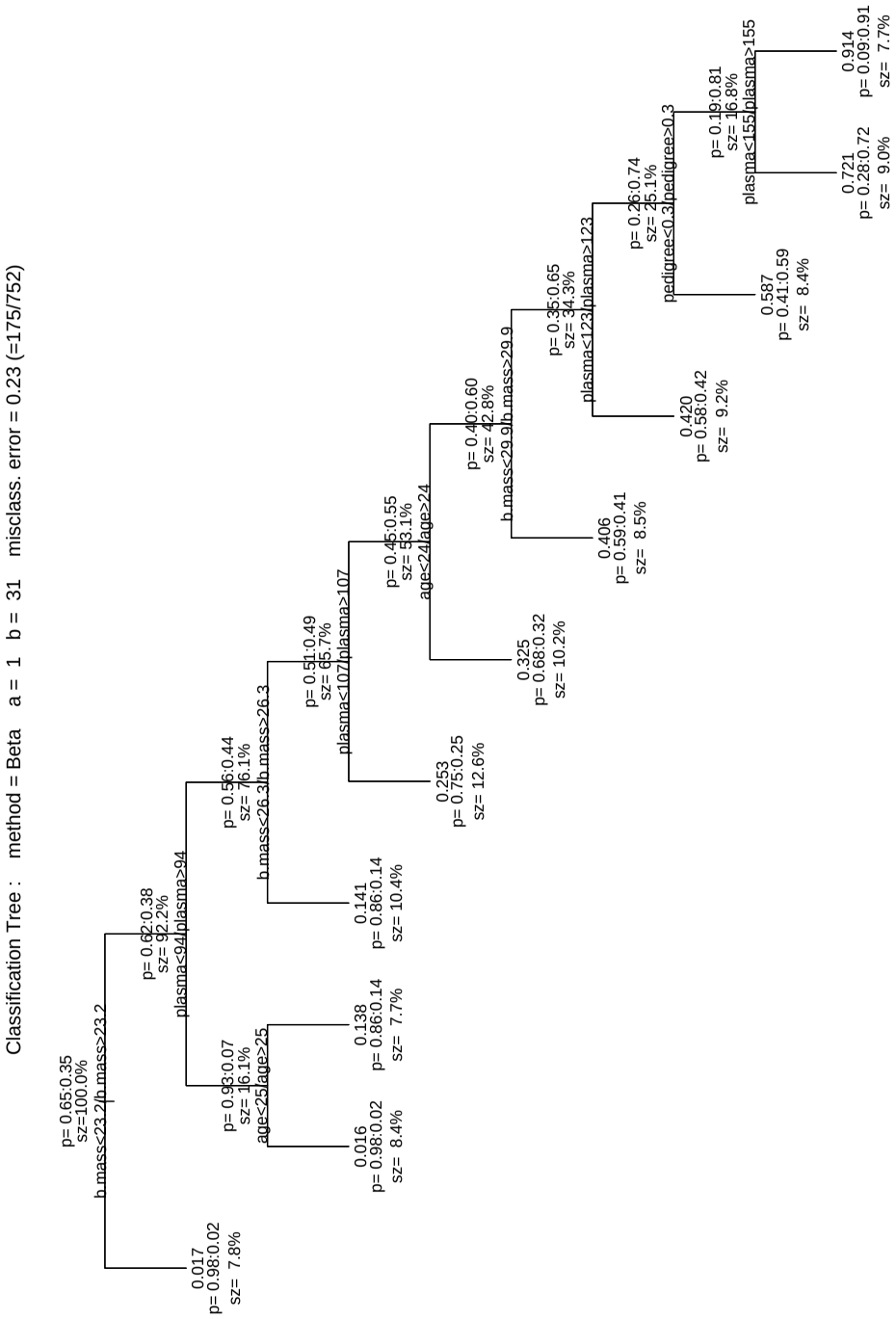

Figure 3.5: *Tree based on the Beta criterion with parameters* $\alpha = 16$ *and beta* $= 1$*. Top to bottom, the tree splits of leafs with increasing class 1 probabilities.*

Chapter 4

F-losses: Compositions of Proper Scoring Rules and Link Functions

We consider in this chapter a larger class of loss functions than proper scoring rules. We call them F-losses for reasons that become clear below. There are two ways of describing F-losses: The first way is to think of them as "uncalibrated" proper scoring rules in the sense that the estimated values are not probability estimates, but they can be calibrated to produce probability estimates by applying a monotone transformation. We can also think of F-losses as the composition of (inverse) link functions and proper scoring rules. This latter description will be our definition and the former a result (Proposition 2).

4.1 Introducing F-losses

The constraint of probabilities to the 0-1 interval is inconvenient for modeling. If one wanted, for example, to directly approximate probabilities with a linear model, it would be difficult to bound the fit for those instances with extreme features. The same problem occurs with any function class that has unbounded functions such as polynomials and splines. The conflict between a desire for unbounded modeling scales and the necessity of estimating probabilities that are bounded between 0 and 1 is overcome by the use of link functions.

We will refer to the "modeling scale" as "F-scale". In a linear model we would have $F(\mathbf{x}) = \sum \beta_j x^{(j)}$, and in an additive model $F(\mathbf{x}) = \sum f_i(x^{(j)})$. The use of the letter F is standard in machine learning and associated statistical literature such as FHT (2000).

Unfortunately, the transformation $F \mapsto q(F)$ that maps $F(\mathbf{x})$ to the probability scale is by convention the *inverse* link function. We therefore have a need to use the notation $q \mapsto F(q)$ for what is conventionally called the link function. This, however, brings us into conflict with the notation for model fits $\mathbf{x} \mapsto F(\mathbf{x})$. We will use both notations and simply trust that the reader will be able to tell which is which from the context. In summary, a model fit $F(\mathbf{x})$ is passed through the inverse $q = q(F)$ of a link function $F = F(q)$ to arrive at a fitted probability $q(F(\mathbf{x}))$.

Definition: *An F-loss is defined as the composition of the inverse $q(F) = F^{-1}(F)$ of a continuous, strictly increasing (hence invertible) link function $F(q)$ and a proper*

scoring rule $\mathbf{R}(\eta|q)$*:*

$$\tilde{\mathbf{R}}(\eta\,|F) \;=\; \mathbf{R}(\eta\,|q(F))$$

An F-loss is called strict *if its proper scoring rule is strict.*

Remarks:

- All proper scoring rules are F-losses because the identity transformation qualifies as a link function. Note that the range of the link function is not required to be all of the real line.

- If $F(q)$ is a link function (continuous and strictly increasing), then by construction $F(\eta)$ minimizes the F-loss: $\inf_F \tilde{\mathbf{R}}(\eta|F) = \tilde{\mathbf{R}}(\eta|F(\eta))$. If the F-loss is strict, this is the unique minimizer.

- There exists a question of non-identifiability between link functions and models because almost all models are closed under addition of constants. Therefore, a model fit $F(\mathbf{x})$ combined with a link function $F(q)$ is equivalent to a model fit $F(\mathbf{x}) + k$ combined with a link function $F(q) + k$. Both map $\mathbf{x}$ to the same probability q. One can therefore fix the translation of the F-scale w.l.o.g. by requiring, for example, $F(1/2) = 0$ or $F(c) = 0$. Such requirements agree with the convention in machine learning that uses 0 as the classification boundary ($\{F(\mathbf{x}) > 0\}$ estimates class 1) and $(2Y - 1)F$ as the "margin" or "signed distance" from the boundary.

- We can leave it to convenience whether the end points 0 and 1 of the unit

intervals are part of the domain of the link function. If they are included in the domain and the link function is unbounded, then $\pm\infty$ will have to be included in the range.

We give some common examples and counter examples of F-losses for classification problems.

Example 1. Logistic loss: log-loss combined with the logistic link

$$F(q) = \log \frac{q}{(1-q)}$$

The link is symmetric about $(1/2, 0)$. The inverse link is

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

The resulting F-loss is

$$\tilde{\mathbf{R}}(\eta|F) \;=\; \eta \log(\frac{1}{1+e^{-F}}) + (1-\eta) \log(\frac{1}{1+e^{F}})$$

Example 2. Probit link and log-loss:

$$F(q) \;=\; \Phi^{-1}(q)$$

where Φ^{-1} is the inverse of the standard normal cumulative distribution function.It is also symmetric about $(1/2, 0)$. Then the inverse link is

$$q(F) \;=\; \Phi(F)$$

The resulting F-loss is

$$\tilde{\mathbf{R}}(\eta|F) \;=\; \eta \log(\Phi(F)) + (1-\eta) \log(1-\Phi(F))$$

Example 3. Log-log link and log-loss:

$$F(q) \;=\; -\log(-\log(q))$$

Then the inverse link is

$$q(F) \;=\; e^{-e^{-F}}$$

The resulting F-loss is

$$\tilde{\mathbf{R}}(\eta|F) \;=\; \eta\, e^{-F} + (1-\eta)\, \log(\frac{1}{1-e^{-e^{-F}}})$$

Notice that by using the Log-log link the F-loss is no longer symmetric since F is mapped to the negative half of the real axis.

Example 4. Exponential loss: Boosting algorithms that will be introduced later are often interpreted as minimizers of exponential loss. It is defined as

$$\tilde{\mathbf{R}}(\eta|F) \;=\; \eta\, e^{-F} + (1-\eta)\, e^{F} \;=\; \mathbf{E}\, e^{-(2Y-1)F}$$

We will show below that exponential loss can be decomposed into a proper scoring rule and a link function.

Example 5. Misclassification loss: Suppose we use a link function that maps the classification boundary $\eta = c$ to $F = 0$, then the transformed cost-weighted misclassification loss with weight c is

$$\tilde{\mathbf{R}}(\eta|F) \;=\; \eta\, 1_{[F<0]} + (1-\eta)\, 1_{[-F<0]}$$

This is an example of a non-strict F-loss. Obviously there exist many link functions that result in the same F-loss.

(Counter)Example 6. SVM loss: The loss function used in classification with support vector machines (SVM) provide a counter example: it is not an F-loss. An SVM judges a classification boundary according to a sum of penalities for training data on the wrong side of the boundary proportionately to their distance from the boundary; no penalty is extracted for data on the correct side of the boundary. We write the empirical SVM loss as

$$\tilde{\mathbf{L}}(Y|F) \;=\; Y\,(1-F)_+ + (1-Y)\,(1+F)_+ \;=\; (1-Y^*F)_+ \;,$$

where as always $Y^* = 2Y - 1$. The expected SVM loss is

$$\tilde{\mathbf{R}}(\eta|F) \;=\; \eta\,(1-F)_+ + (1-\eta)\,(1+F)_+ \;=\; E[(1-Y^*F)_+] \;.$$

This is minimized when $F = -1$ if $\eta < 0.5$ and $F = 1$ otherwise. There does not exist a link function that maps this loss to a proper scoring rule: Recall that $F(\eta)$ must be a minimizer of $\tilde{\mathbf{R}}(\eta|F)$, hence $F(\eta) = -1$ for $\eta < 0.5$ and $F(\eta) = +1$ for $\eta \geq 0.5$, or $F(\eta) = 2\,1_{[\eta \geq 0.5]} - 1$, which is neither a continuous nor a strictly increasing function.

4.2 Characterization of Strict F-losses

In this section we show that F-losses are essentially "uncalibrated" proper scoring rules: the only requirement we need to impose on a loss function is that its estimates are in a monotone relationship to the underlying class 1 probabilities. This monotone relationship is essentially the (inverse) link function that calibrates F-estimates to probability estimates.

Suppose Y is labeled as 0 and 1, then F-losses can be written in the form:

$$\tilde{\mathbf{L}}(Y|F) \;=\; Y\tilde{L}_1(-F) + (1-Y)\tilde{L}_0(F)$$

$$\tilde{\mathbf{R}}(\eta|F) \;=\; \eta\tilde{L}_1(-F) + (1-\eta)\tilde{L}_0(F)$$

where $\tilde{L}_1(\cdot)$ and $\tilde{L}_0(\cdot)$ are monotonically increasing functions on the F-scale. This is achieved by defining

$$\tilde{L}_1(-F) \;=\; L_1(1-q(F))\;, \qquad \tilde{L}_0(F) \;=\; L_0(q(F))\;,$$

where $q(F)$ is the inverse link function and $y\,L_1(1-q) + (1-y)\,L_0(q)$ the proper scoring rule whose composition is the F-loss.

All examples of the previous section can be written in this form. Note that proper scoring rules can also be written in this form by letting

$$\tilde{L}_1(-F) = L_1(1-F)\;, \qquad \tilde{L}_0(F) = L_0(F)\;,$$

where $L_1()$ and $L_0()$ define the proper scoring rule and the F-scale is the same as the q-scale.

We now derive a criterion that determines whether loss functions of the above form are indeed F-losses. We motivate the problem with the following example.

Example 7. Exponential loss: As mentioned in the previous sections, according to some interpretations boosting can be considered as minimization of so-called exponential loss:

$$\tilde{\mathbf{L}}(Y|F) = e^{-Y^*F(\mathbf{x})} = \frac{(1+Y^*)}{2}e^{-F(\mathbf{x})} + \frac{(1-Y^*)}{2}e^{F(\mathbf{x})}$$

where $Y^* = 2Y - 1$ is ± 1 labeling of the classes. Expected exponential loss is

$$\tilde{\mathbf{R}}(\eta|F) \;=\; \mathbf{E}\tilde{\mathbf{L}}(Y|F) \;=\; \eta e^{-F} + (1-\eta)e^{F} \;. \tag{4.1}$$

The question is whether exponential loss is an F-loss, that is, whether it can be decomposed into a link function and a proper scoring rule. FHT (2000) showed part of the way by indicating that the population minimizer of (4.1) is $\frac{1}{2}\log\frac{\eta}{1-\eta}$. The interest in this expression is that it can be used to define a link function,

$$F(\eta) \;=\; \frac{1}{2}\log\frac{\eta}{1-\eta} \;, \tag{4.2}$$

with the effect that minimizing $\tilde{\mathbf{R}}(\eta|F)$ w.r.t. to F is equivalent to minimizing

$$\mathbf{R}(\eta|q) \;=\; \eta e^{-F(q)} + (1-\eta)e^{F(q)}$$

w.r.t q in the sense that the minima are linked by Equation (4.2). FHT (2000) did not answer the other half of our question: whether a proper scoring rule emerges at the other end of the link function in the form of $\mathbf{R}(\eta|q)$. This is indeed the case, by construction: Rewriting the loss by plugging in the link function, $\mathbf{R}(\eta|q) = \tilde{\mathbf{R}}(\eta|F(q))$, we obtain

$$\mathbf{R}(\eta|q) \;=\; \eta\left(\frac{1-q}{q}\right)^{1/2} + (1-\eta)\left(\frac{q}{1-q}\right)^{1/2}$$

so that

$$L_1(1-q) = \left(\frac{1-q}{q}\right)^{1/2}, \qquad L_0(q) = \left(\frac{q}{1-q}\right)^{1/2}$$

The minimum is attained at $q = \eta$ by definition of $F(q)$. Thus a proper scoring rule results, and the exponential loss is shown to be an F-loss.

In light of this example, we can obtain a more general result regarding the characterization of F-losses:

Proposition 2: *Assume that $\tilde{L}_1(-F)$ and $\tilde{L}_0(F)$ are defined and continuously differentiable on an open interval of the real line, that $\tilde{L}_0'(F)/\tilde{L}_1'(-F)$ is strictly increasing, and that its range is the positive half-line:*

$$\inf_F \frac{\tilde{L}_0'(F)}{\tilde{L}_1'(-F)} = 0 \ , \qquad \sup_F \frac{\tilde{L}_0'(F)}{\tilde{L}_1'(-F)} = \infty \ .$$

Then $\tilde{L}_1$ and $\tilde{L}_0$ define a F-loss whose minimizer $F = F(q)$ exists and is the unique solution of

$$\frac{\tilde{L}_0'(F)}{\tilde{L}_1'(-F)} \ = \ \frac{q}{1-q} \ .$$

Proof: The stationarity condition $\frac{d}{dF}(\eta\tilde{L}_1(-F) + (1-\eta)\tilde{L}_0(F)) = 0$ produces the above equation as a necessary condition for the minimizer. The assumptions grant the existence of an inverse of the map $F \to \tilde{L}_0'(F)/\tilde{L}_1'(-F)$. This inverse is defined on the open interval $(0, \infty)$. In addition, this inverse is necessarily strictly monotone, which combines with the strict monotonicity of the odds ratio $q/(1-q)$ to the strict monotonicity of the minimizer $F(q)$. QED

Finally, we note that symmetry of the F-loss, $\tilde{L}_1(F) = \tilde{L}_0(F)$, implies symmetry of the proper scoring rule, $L_1(q) = L_0(q)$, and symmetry of the natural link about $(1/2, 0)$: $F(q) + F(1-q) = 0$. This follows from

$$\begin{aligned} F(q) &= \text{argmin}_F \left[\, q \, \tilde{L}_1(-F) + (1-q) \, \tilde{L}_0(F) \, \right] , \\ F(1-q) &= \text{argmin}_F \left[\, (1-q) \, \tilde{L}_1(-F) + q \, \tilde{L}_0(F) \, \right] . \end{aligned}$$

We give one more example to illustrate the application of the proposition.

Example 8. Power losses: These are losses of the form

$$\tilde{\mathbf{R}}(\eta|F) \;=\; \eta(1-F)^r \;+\; (1-\eta)F^r, \qquad\qquad \text{where} \;\; F \,\epsilon\, [0,1],\; r>1$$

They are strictly convex in F for $r > 1$, hence minima exist and are unique. First order stationarity says

$$\left(\frac{F}{1-F}\right)^{r-1} = \frac{\eta}{1-\eta}\ .$$

Since the left hand side of the equation is a strictly monotonic function of F and the right hand side of the equation is a strictly monotonic function of q, we conclude that there exists a strictly monotonic minimizer $F(\eta)$ which is

$$F(\eta) \;=\; \frac{\eta^{1/(r-1)}}{\eta^{1/(r-1)} + (1-\eta)^{1/(r-1)}}\ .$$

Thus power losses are F-losses and therefore decompose into link functions and proper scoring rules. The corresponding proper scoring rule is given by

$$L_1(1-q) \;=\; \left(\frac{(1-q)^{1/(r-1)}}{q^{1/(r-1)} + (1-q)^{1/(r-1)}}\right)^r\ , \quad L_0(q) \;=\; \left(\frac{q^{1/(r-1)}}{q^{1/(r-1)} + (1-q)^{1/(r-1)}}\right)^r\ .$$

Remarks:

- We restricted the range of F to $[0,1]$ because otherwise the ratio $\tilde{L}_0'(F)/\tilde{L}_1'(-F)$ is not strictly monotone increasing.

- When $0 < r < 1$, the loss $\tilde{\mathbf{R}}(\eta|F)$ is concave in F, and the stationarity condition characterizes maxima instead of minima. The minimum is attained at $F = 0$ if $\eta < .5$, and at $F = 1$ otherwise. It can not be mapped to a proper scoring rule because the minimizers are just two points.

- For $r = 1$ we obtain the SVM loss, except that F is restricted to the unit interval, which it isn't for conventional SVMs.

4.3 Cost-Weighted F-losses and Tailoring

In this section we examine the extension of cost-weighting from misclassification losses to F-losses. Recall from Section 3.2 that in general cost-weighted proper scoring rules are no longer proper scoring rules. We will show, however, that cost-weighted F-losses *are* F-losses as well. In particular, cost-weighted proper scoring rules are F-losses.

The closure of F-losses under cost-weighting is a considerable advantage of this larger universe of loss functions over proper scoring rules. We will put this fact to use for an apparently superior alternative of tailoring losses to costs c.

We first show that any cost-weighted F-loss can be decomposed into a link function and a proper scoring rule, hence is an F-loss also:

Proposition 3: *Assume the F-loss $\tilde{\mathbf{R}}(\eta|F) = \eta\tilde{L}_1(-F)+(1-\eta)\tilde{L}_0(F)$ is decomposed into a proper scoring scoring rule $\mathbf{L}(\eta|q) = \eta L_1(1-q)+(1-\eta)L_0(q)$ and a continuous and strictly monotone inverse link function $q(F) = F^{-1}(F)$: $\tilde{\mathbf{R}}(\eta|F) = \mathbf{L}(\eta|q(F))$. Then the cost-weighted version of the F-loss,*

$$\tilde{\mathbf{R}}^{cw}(\eta|F) \;=\; (1-c)\eta\tilde{L}_1(-F) + c(1-\eta)\tilde{L}_0(F)\;,$$

is also an F-loss. Its link function, proper scoring rule and weight function are obtained as follows:

1. *The link function of the cost-weighted F-loss is*

$$F^{cw}(q) \;=\; F\left(\frac{(1-c)q}{(1-c)q+c(1-q)}\right) \;.$$

2. *The proper scoring rule of the cost-weighted F-loss is given by*

$$\begin{aligned} L_1^{cw}(1-q) &= (1-c)L_1\left(\frac{c(1-q)}{(1-c)q+c(1-q)}\right) \;, \\ L_0^{cw}(q) &= cL_0\left(\frac{(1-c)q}{(1-c)q+c(1-q)}\right) \;. \end{aligned}$$

3. *If the proper scoring rule is differentiable, the weight function for the cost-weighted analog is*

$$\omega^{cw}(q) \;=\; \omega\left(\frac{(1-c)q}{(1-c)q+c(1-q)}\right)\frac{(1-c)^2c^2}{((1-c)q+c(1-q))^3} \;.$$

The proof can be found in the appendix. If we write $q^{cw}(F) = F^{cw-1}(F)$, and if we further abbreviate $q^{cw} = q^{cw}(F)$ and $q = q(F)$, then the first assertion is equivalent to

$$\frac{q^{cw}}{1-q^{cw}} \;=\; \frac{c}{1-c}\cdot\frac{q}{1-q}$$

and

$$q^{cw} \;=\; \frac{cq}{cq+(1-c)(1-q)} \;.$$

It should be kept in mind that $q(F)$ does not provide consistent or calibrated probability estimates; $q^{cw}(F)$ does. The role of equations relating the two is to indicate the functional form that cost-weighted probability estimates $q^{cw}(F)$ take. Because $q = 1/2$ gets mapped to $q^{cw} = c$, the class 1 decision regions $q > 1/2$ and $q^{cw} > c$ are the same.

For tailoring to a specific cost c along the lines of Section 3.4, we need the first and second moments of the normalized cost-weighted weight function $\omega^{cw}(q)$. Recall that the first moment is needed to match the weight function to the cost c, and the second moment is needed as a measure of approximation to the point mass δ_c, which is the "weight function" of cost-weighted misclassification loss.

Proposition 4: *If $\omega(q)$ is a density function on $(0,1)$ with mean μ and variance σ^2, then*

$$\frac{\omega^{cw}(q)}{c\mu + (1-c)(1-\mu)}$$

is also a density function, and its mean is

$$\mu^{cw} \;=\; \frac{c\mu}{c\mu + (1-c)(1-\mu)} \;.$$

Its variance is bounded by

$$\frac{c(1-c)\min(c,1-c)}{(c\mu+(1-c)(1-\mu))^3}\;\sigma^2 \;\le\; \sigma^{cw2} \;\le\; \frac{c(1-c)\max(c,1-c)}{(c\mu+(1-c)(1-\mu))^3}\;\sigma^2 \;.$$

The proof can be found in the appendix. Here is the application that provides the basis for tailoring with cost-weighting:

Corollary: *If $\omega(q)$ is a symmetric density with existing expectation, $\mu = 1/2$, then*

$$\mu^{cw} \;=\; c$$

and

$$8\,c\,(1-c)\min(c,1-c)\,\sigma^2 \;\le\; \sigma^{cw2} \;\le\; 8\,c\,(1-c)\max(c,1-c)\,\sigma^2 \;.$$

Thus cost-weighting a symmetric weight function with costs c and $1-c$ readily achieves tailoring for cost c.

It is somewhat dissatisfying that we are left with mere inequalities for the variance, although they are sufficient for our purposes: if $\omega(q)$ becomes more spiky around $1/2$ in terms of $\sigma \to 0$, then $\omega^{cw}(q)$ becomes more spiky around c in terms of $\sigma^{wc} \to 0$ at the same rate. For a more complete result with an exact formula for σ^{cw} one has to be more specific about $\omega(q)$. This works out for example if $\omega(q)$ is a symmetric Beta weight function, but the formula is not a pretty sight. We start with a general observation about symmetric Beta weights:

Corollary: *If $\omega(q) \sim q^{\alpha-1}(1-q)^{\alpha-1}$ is a symmetric Beta weight function, then the cost-weighted version is*

$$\omega^{cw}(q) \;\sim\; \frac{q^{a-1}(1-q)^{a-1}}{((1-c)q + c(1-q))^{2\alpha+1}}$$

It is merely this denominator that achieves the redistribution of the weight function so as to center it at $\mu^{cw} = c$. An exact expression for the corresponding variance is as follows:

Proposition 5: *If $\omega(q) = q^{\alpha-1}(1-q)^{\alpha-1}/B(\alpha,\alpha)$ is a symmetric Beta density, then the normalized cost-weighted density has the following variance:*

$$\sigma^{cw2} \;=\; c^2 \left(\frac{(\alpha+1)}{(1-c)(2\alpha+1)} \; {}_2F_1\left(1, \alpha+2, 2\alpha+2; \frac{1-2c}{1-c}\right) \;-\; 1 \right) .$$

where ${}_2F_1()$ is the hypergeometric function.

Again, the proof can be found in the appendix. It seems somewhat surprising that mere cost-weighting causes such complications. There may not be much intuition behind the above formulas, but the concept behind them, namely cost-weighting, is simple enough, and the application to tailoring of weights is useful as we will show.

In the remainder of the section we discuss the shape of the cost-weighted weight functions for the symmetric Beta family.

- For $\alpha > 1$ the weight function has a mode at

$$\frac{-(\alpha - 2(3c-1)) + \sqrt{(\alpha - 2(3c-1))^2 + 12c(2c-1)(\alpha-1)}}{6(2c-1)}$$

  In general, the mode and the expected value do not coincide. Thus if one wants to put most of the weight on a certain threshold, one can set the above formula to the desired value and backward calculate the corresponding cost c that is to be applied to the cost-weighted F-loss.

- In the limit as $\alpha \to \infty$, the weight function puts all mass at $q = c$. This follows from the proposition that links α and σ^{cw2}.

- For $\alpha < 1$, the weight function has a U-shape that puts more weight on the two tails.

- For $\alpha = 1$ and $c \neq 1/2$, depending on whether c is greater than $1/2$ or not, the weight function is either increasing or decreasing, thus puts more weight on one tail.

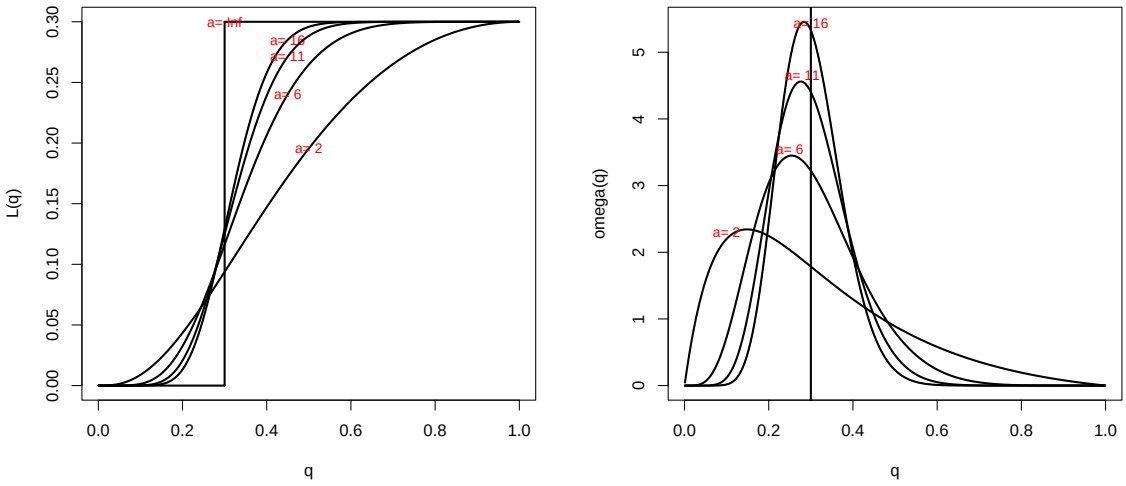

Figure 4.1: *Loss functions $L_0(q)$ and weight functions $\omega(q)$ for various values of α, and $c = 0.3$: Shown are $\alpha = 2$, 6 ,11 and 16 scaled to show convergence to the step function.*

The calculations are given in the Appendix. Figure 4.1 illustrates the cost-weighted weight functions and proper scoring rules for $c = 0.3$.

4.4 Margin-based Loss Functions

In this section we derive the connection between F-losses and margin-based loss functions which have motivated many popular classification methods including boosting and SVM. Yi Lin(2001) gave a nice discussion of margin-based loss functions where he showed that many such loss functions are Fisher consistent in that they generate a classification procedure that is equivalent to Bayes' rule for classification at the

$c = 1/2$ boundary. Here is a formal definition of margin-based losses: A margin-based loss function is any loss function that can be written as

$$\tilde{L}(-Y^*F)$$

where $\tilde{L}(.)$ is a monotone increasing function and $Y^* = 2Y - 1$ maps 0/1 coding of class labels to ± 1 coding. The quantity Y^*F is called "the margin" and can be interpreted as the "signed distance" of F from the boundary $F = 0$. Classification is done according to $\hat{Y}^* = \text{sign}(F) = 2 * 1_{[F \geq 0]}$. A negative sign of Y^*F is hence interpreted as being on the wrong side of the boundary $F = 0$ indicating misclassification.

Expanding the above expression we get

$$\tilde{L}(-Y^*F) \;=\; \tilde{L}(-(2Y-1)F) \;=\; Y\tilde{L}(-F) + (1-Y)\tilde{L}(F)\ .$$

We recognize that such a loss is a special case of F-losses in which one sets both $\tilde{L}_1$ and $\tilde{L}_0$ equal to $\tilde{L}$, assuming that additional conditions are satisfied that permit the decomposition into a proper scoring rule and a link function. We can also see that such a loss treats the two classes symmetrically, thus forcing equal costs on the two types of misclassification error. Therefore, for the purpose of cost weighting, one must allow asymmetry between the classes and use separate $\tilde{L}_1$ and $\tilde{L}_0$:

$$y * \tilde{L}_1(-F) + (1-y) * \tilde{L}_0(F)\ .$$

This in turn allows us to cost-weight the losses,

$$(1-c) * y * \tilde{L}_1(-F) + c * (1-y) * \tilde{L}_0(F)\ ,$$

which means replacing $\tilde{L}_1$ with $(1-c)\tilde{L}_1$ and $\tilde{L}_0$ with $c\,\tilde{L}_0$.

Chapter 5

IRLS for Linear Models

This chapter deals with optimization of F-losses for linear models. We start with fitting a linear model for logistic regression by minimizing the negative log-likelihood and show that the algorithm can be interpreted as an Iteratively Reweighted Least Squares algorithm. Then we generalize the IRLS algorithm from logistic regression to arbitrary proper scoring rules and F-losses, thus permitting us to play with the choice of weights that induce loss functions.

5.1 IRLS for Proper Scoring Rules

Logistic regression is commonly computed with an Iteratively Reweighted Least Squares or IRLS algorithm. We show that IRLS carries over to proper scoring rules in general.

Recall the form of a general proper scoring rule for a Bernoulli model:

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

A linear model is given by $F(q) = \mathbf{x}^T\mathbf{b}$ or, equivalently, $q(F) = q(\mathbf{x}^T\mathbf{b})$ using the inverse link. We assume the data is an i.i.d. sample (x_n, y_n) $(n = 1, ..., N)$ where x_n are drawn from a distribution in predictor space. For brevity we write

$$q_n = q(F(x_n)) \ , \qquad q'_n = q'(F(x_n)) \ , \qquad q''_n = q''(F(x_n)) \ .$$

We minimize the empirical risk, that is, the mean of a proper scoring rule over the observations:

$$\mathbf{R}(\mathbf{b}) \ = \ \frac{1}{N} \sum_{n=1,...,N} [y_n L_1(1 - q_n) + (1 - y_n)L_0(q_n)]$$

For concreteness we recall that this specializes for logistic regression to the well-known form

$$\mathbf{R}(\mathbf{b}) \ = \ -\frac{1}{N} \sum_{n=1,...,N} [y_n \log(q_n) + (1 - y_n) \log(1 - q_n)]$$

For a Newton update we need the gradient and the Hessian. First the gradient:

$$\partial_{\mathbf{b}}\mathbf{R} \ = \ -\frac{1}{N} \sum_{n=1,...,N} (y_n - q_n)\omega(q_n)q'_n\mathbf{x}_n,$$

For logistic regression, $\omega(q) = 1/(q(1 - q))$ and $q(F) = 1/(1 + \exp(-F))$. A simplification peculiar to logistic regression occurs because $q'(F) = q \cdot (1 - q)$, hence $\omega(q)q'(F) = 1$ and therefore

$$\partial_{\mathbf{b}}\mathbf{R} \ = \ -\frac{1}{N} \sum_{n=1,...,N} (y_n - q_n)\mathbf{x}_n \ .$$

Setting the gradient to zero results in the so-called score equation.

The Hessian for the mean of the proper scoring rule is

$$\partial^2_{\mathbf{b}}\mathbf{R} \ = \ \frac{1}{N} \sum_{n=1,...,N} [\omega(q_n)q'^2_n - (y_n - q_n) \left(\omega'(q_n)q'^2_n + \omega(q_n)q''_n\right)]\mathbf{x}_n\mathbf{x}_n^T \ . \qquad (5.1)$$

Specializing again to logistic regression, we observe that $\omega' = -(1-2q)/(q(1-q))^2$ and $q'' = (1-2q)q(1-q)$ and hence the second term of the Hessian disappears and we obtain:

$$\partial^2_{\mathbf{b}}\mathbf{R} \;=\; -\frac{1}{N}\sum_{n=1,\ldots,N} q_n(1-q_n)\mathbf{x}_n\mathbf{x}_n^T$$

A Newton step has the form

$$\mathbf{b}_{new} \;=\; \mathbf{b}_{old} - \left(\partial^2_{\mathbf{b}}\mathbf{R}(\mathbf{b})\right)^{-1}\left(\partial_{\mathbf{b}}\mathbf{R}(\mathbf{b})\right)$$

In direct generalization of IRLS for logistic regression, we can write the update as an IRLS step for general proper scoring rules:

$$\mathbf{b}^{new} \;=\; \mathbf{b}^{old} \;+\; (X^TWX)^{-1}X^Tz$$

where the 'working weights' and the 'working responses' are, respectively,

$$w_n \;=\; \omega(q_n)q_n'^2 - (y_n - q_n)\left(\omega'(q_n)q_n'^2 + \omega'(q_n)q_n''\right) \tag{5.2}$$

$$z_n \;=\; (y_n - q_n)\omega(q_n)q_n'/w_n \tag{5.3}$$

Denote by $\mathbf{y}$ the vector of responses $y_n's$, by $\mathbf{X}$ the $N\times$ *(d+1)* matrix with rows x_n^T, by $\mathbf{q}$ the vector of fitted probabilities with nth element q_n, and by $\mathbf{W}$ a $N \times N$ diagonal matrix with weights w_n in the diagonal. The vectorized equation above shows that the Newton updates can be written as solutions of a weighted least squares problem, with iteration-dependent weights and responses. We wrote the IRLS update in an incremental form whereas in the form usually shown in the literature one absorbs $\mathbf{b}^{old}$ in the working response $\mathbf{z}$. We prefer the incremental form because of our interest in stagewise fitting, which is an incremental procedure.

Specializing once again to logistic regression, we obtain the following familiar expressions:

$$\begin{aligned} w_n &= q_n\,(1-q_n) \\ z_n &= \frac{(y_n - q_n)}{q_n\,(1-q_n)} \end{aligned}$$

5.2 Fisher Scoring

For Fisher scoring one replaces the Hessian $\partial^2_{\mathbf{b}}\mathbf{R}(\mathbf{b})$ with its expectation, assuming the current estimates of the conditional probabilities of the labels to be the true ones: $\mathbf{P}[Y = 1|X = x_n] = q_n$. This means $\mathbf{E}_{y_n}(y_n - q_n) = 0$, hence the weights and the working response simplify as follows:

$$w_n = \omega(q_n)\,{q'_n}^2\,, \tag{5.4}$$

$$z_n = (y_n - q_n)/q'_n\,. \tag{5.5}$$

Some observations:

- The function $\omega(q)$ is the contribution to the working weights w_n due to the proper scoring rule. Logistic regression with $\omega(q) = q^{-1}(1-q)^{-1}$ up-weights extreme probabilities near zero and one.
- The term ${q'}^2$ is the contribution to the working weights w_n due to the link function.
- The losses do not enter in the working response z_n.

- The observed responses y_n do not enter in the current weights w_n.

It is well known that for logistic regression with log-loss and the logistic link, Fisher scoring and Newton iterations coincide. From a machine learning perspective, it is curious to learn that there exist loss functions for which the dependence of the weights on the responses disappears for structural reasons: this is the case for so-called "canonical links", which are well-known in statistics from generalized linear models. They have a generalization to certain combinations of proper scoring rules and link functions as we will show now.

5.3 Canonical Links: Equality of Observed and Expected Hessians

We examine under what condition on the combination of proper scoring rules and link functions, Newton steps and Fisher scoring steps are identical. The requirement is

$$\mathbf{R}''(\mathbf{b}) \;=\; \mathbf{E}_y\,[\,\mathbf{R}''(\mathbf{b})\,]\;.$$

This condition is equivalent to requiring the weights w_n in Equation (5.1) to be independent of y_n, which in turn is equivalent to

$$\omega'(q)\,{q'}^{\,2} \;+\; \omega(q)\,q'' \;=\; 0\;.$$

This differential equation can be solved immediately: The left side equals $(\,\omega(q)\,q'\,)'$, hence $\omega(q)\,q'$ is a constant, which we may choose to be 1:

$$\omega(q)\,q' \;=\; 1\;.$$

Since $q = q(F)$ is the inverse of the link function $F = F(q)$, we have $F'(q) = 1/q'(F(q))$, and hence

$$F'(q) \;=\; \omega(q) \;=\; L_1'(1-q) + L_0'(q)\;, \tag{5.6}$$

where the second equality follows from $L_1'(1-q) + L_0'(q) = (1-q)\omega(q) + q\omega(q)$, which is the sum of the Equations (3.1). The solution, unique up to an irrelevant additive constant, is therefore

$$F(q) \;=\; \int^q \omega(q)dq \;=\; L_0(q) - L_1(1-q)\;. \tag{5.7}$$

We summarize:

Proposition 6: *For any proper scoring rule there exists a canonical link function $F(q)$. The link function is unique up to a constant.*

Here are the usual examples in the Beta family of proper scoring rules for $\alpha = \beta =$ 1, 0, -1/2 (Section 3.4):

- *Squared error loss*: $L_1(1-q) = (1-q)^2$ and $L_0(q) = q^2$, hence the inverse link is

  $$F(q) \;=\; 2q - 1\;.$$

  The canonical link is essentially the identity transform.

- *Log-loss*: $L_1(1-q) = -\log(1-q)$ and $L_0(q) = -\log(q)$, hence the natural link is

$$F(q) \;=\; \log\frac{q}{1-q}\;.$$

  Its inverse is the logistic function.

- *Boosting loss*: $L_1(1-q) = ((1-q)/q)^{1/2}$ and $L_0(q) = (q/(1-q))^{1/2}$, hence the natural link is

$$F(q) \;=\; \left(\frac{q}{1-q}\right)^{1/2} - \left(\frac{1-q}{q}\right)^{1/2}\;.$$

  Its inverse is

$$q(F) \;=\; \frac{1}{2} + \left(\frac{F}{2}\right)\left(\left(\frac{F}{2}\right)^2 + 1\right)^{-1/2}\;.$$

  The F-loss is given by

$$\tilde{L}_{0/1}(F) \;=\; \frac{F}{2} + \left(\left(\frac{F}{2}\right)^2 + 1\right)^{1/2}\;.$$

We see that the decomposition of the exponential loss into a proper scoring rule and a link function does *not* result in a canonical pairing in which the observed and the expected Hessian are idential. For exponential loss Newton iterations are not identical with Fisher scoring iteration.

We conclude by noting that the converse of the above proposition holds also: prescribing a link function as canonical essentially determines the proper scoring rule.

Proposition 7: *For any link function $F(q)$ there exists a proper scoring rule for which $F(q)$ is canonical. The proper scoring rule is unique up to an additive constant.*

This is immediately seen from Equation (5.6): $F'(q) = \omega(q)$, and the fact that, by Equations (3.1), $\omega(q)$ determines $L_1(1-q)$ and $L_0(q)$ up to a constant.

5.4 Newton Updates and IRLS for F-losses

For Newton updates based on *general* F-losses, the ingredients are

$$\begin{aligned}
\mathbf{R}(\mathbf{b}) &= \frac{1}{N}\sum_{n=1..N}\Big[y_n\,\tilde{L}_1(-\mathbf{x}_n^T\mathbf{b}) + (1-y_n)\,\tilde{L}_0(y_n\,\mathbf{x}_n^T\mathbf{b})\Big]\ , \\
\partial_{\mathbf{b}}\mathbf{R}(\mathbf{b}) &= \frac{1}{N}\sum_{n=1..N}\Big[-y_n\,\tilde{L}_1'(-\mathbf{x}_n^T\mathbf{b}) + (1-y_n)\,\tilde{L}_0'(\mathbf{x}_n^T\mathbf{b})\Big]\,\mathbf{x}_n\ , \\
\partial_{\mathbf{b}}^2\mathbf{R}(\mathbf{b}) &= \frac{1}{N}\sum_{n=1..N}\Big[y_n\,\tilde{L}_1''(-\mathbf{x}_n^T\mathbf{b}) + (1-y_n)\,\tilde{L}_0''(\mathbf{x}_n^T\mathbf{b})\Big]\,\mathbf{x}_n\mathbf{x}_n^T\ .
\end{aligned}$$

The current IRLS weights and responses are therefore

$$\begin{aligned}
w_n &= y_n\,\tilde{L}_1''(-\mathbf{x}_n^T\mathbf{b}) + (1-y_n)\,\tilde{L}_0''(\mathbf{x}_n^T\mathbf{b})\ , \\
z_n &= y_n\frac{\tilde{L}_1'(-\mathbf{x}_n^T\mathbf{b})}{\tilde{L}_1''(-\mathbf{x}_n^T\mathbf{b})} - (1-y_n)\frac{\tilde{L}_0'(\mathbf{x}_n^T\mathbf{b})}{\tilde{L}_0''(\mathbf{x}_n^T\mathbf{b})}\ .
\end{aligned}$$

Specializing to *symmetric* F-losses with $\tilde{L} = \tilde{L}_1 = \tilde{L}_0$, the ingredients of Newton updates are

$$\begin{aligned}
\mathbf{R}(\mathbf{b}) &= \frac{1}{N}\sum_{n=1..N}\tilde{L}(-y_n^*\,\mathbf{x}_n^T\mathbf{b})\ , \\
\partial_{\mathbf{b}}\mathbf{R}(\mathbf{b}) &= \frac{1}{N}\sum_{n=1..N}\tilde{L}'(-y_n^*\,\mathbf{x}_n^T\mathbf{b})\,(-y_n^*)\mathbf{x}_n\ , \\
\partial_{\mathbf{b}}^2\mathbf{R}(\mathbf{b}) &= \frac{1}{N}\sum_{n=1..N}\tilde{L}''(-y_n^*\,\mathbf{x}_n^T\mathbf{b})\,\mathbf{x}_n\mathbf{x}_n^T\ ,
\end{aligned}$$

where we made use of $(y_n^*)^2 = 1$. A Newton increment $-\left(\partial_{\mathbf{b}}^2\mathbf{R}(\mathbf{b})\right)^{-1}\left(\partial_{\mathbf{b}}\mathbf{R}(\mathbf{b})\right)$ can be

written as a weighted LS solution with the following current weights and responses:

$$\begin{aligned} w_n &= \tilde{L}''(-y_n^* \mathbf{x}_n^T \mathbf{b}) \ , \\ z_n &= \frac{\tilde{L}'(-y_n^* \mathbf{x}_n^T \mathbf{b})}{\tilde{L}''(-y_n^* \mathbf{x}_n^T \mathbf{b})} \, y_n^* \ . \end{aligned}$$

For exponential loss $\tilde{L}(F) = \exp(F)$, some simplifications occur:

$$w_n = \exp(-y_n^* \mathbf{x}_n^T \mathbf{b}) \ , \qquad z_n = y_n^* \ .$$

The weights are exactly those of the boosting algorithm in the next section with $F_n = \mathbf{x}_n^T \mathbf{b}$.

These algorithms are worthwhile keeping in mind for the common cases where the losses $\tilde{L}_i$ and hence $\mathbf{R}(\mathbf{b})$ are convex, as is the case for exponential loss. Convexity and the ensuing stability of minimization may get obscured when an F-loss is decomposed into a link function and a proper scoring rule. On the other hand, in those cases where the weight function $\omega(q)$ is tailored for classification with unequal cost of misclassification, the F-loss is unlikely to be convex, and stability may be gained by using Fisher scoring instead, whose expected Hessian is always positive definite.

5.5 Convexity of F-losses

Convexity of F-losses is of interest for two reasons:

- When the F-loss is convex, Newton steps may be more efficient than Fisher scoring steps.

- Convexity of F-losses is assumed in theoretical work on consistency of classification schemes, as for example in Lugosi and and Vayatis (2004) and Zhang (2004).

A special case of convex F-loss arises when a proper scoring rule is matched with its canonical link function. Convexity follows because for canonical links the empirical Hessian and the expected Hessian coincide, and the expected Hessian is always non-negative definite. Convexity is more general than the case of canonical links, though. For example, we saw above that the link function of exponential loss is not canonical, that is, observed and expected Hessians do not coincide, yet exponential loss as a F-loss is convex.

We give a general condition under which a pairing of a link function and a proper scoring rule results in a convex F-loss. The condition is derived by requiring the weights in the Hessian of Equation (5.2) to be non-negative: $w_n \geq 0$, that is,

$$\omega(q)\, q'^2 \;-\; (y-q)\,\Big(\omega'(q)\, q'^2 \;+\; \omega(q)\, q''\Big) \;\geq\; 0\;.$$

This requirement results in two redundant inequalities, one for $y = 1$ and one for $y = 0$. We summarize:

Proposition 8: *A combination of proper scoring rule and link function results in a convex loss* $\mathbf{R}(\mathbf{b})$ *if* $\omega(q)$ *and* $q(F)$ *satisfy*

$$-\frac{1}{q} \;\leq\; \frac{\omega'}{\omega} + \frac{q''}{q'^2} \;\leq\; \frac{1}{1-q}\;.$$

As a corollary, the combination of proper scoring rules in the Beta family with equal exponents,

$$\omega(q) \;=\; q^{\alpha-1}(1-q)^{\beta-1} \;,$$

and scaled logistic links,

$$q(F) \;=\; 1/(1+e^{-F/s}) \;, \qquad F(q) \;=\; s\,\log\frac{q}{1-q} \;,$$

result in convex F-losses iff

$$-1 \le \alpha,\ \beta \le 0 \;, \qquad s > 0 \;.$$

These proper scoring rules result in convex F-losses for any scaled logistic link. Special cases are exponential loss ($\alpha = \beta = -1/2$, $s = 1/2$) and log-loss with the logit scale ($\alpha = \beta = 0$, $s = 1$). But squared error loss ($\alpha = \beta = 1$) does not result in convex F-losses with any scaled logistic link.

For the combination of the cost-weighted Beta family,

$$\omega^{cw}(q) \;\sim\; \frac{q^{a-1}(1-q)^{a-1}}{((1-c)q + c(1-q))^{2\alpha+1}} \;,$$

and the cost-weighted logistic link,

$$q^{cw}(F) \;=\; \frac{c}{c+(1-c)e^{-F/s}} \;, \qquad F^{cw}(q) \;=\; s\,\log\frac{q}{1-q} - s\,\log\frac{c}{1-c} \;, \qquad (5.8)$$

convexity of the resulting cost-weighted F-losses holds iff

$$-1 \le \alpha \le 0 \;, \qquad s > 0 \;.$$

This includes exponential loss for $\alpha = -1/2$ and $s = 1/2$ as well as logistic loss for $\alpha = 0$ and $s = 1$.

5.6 Some Peculiarities of Exponential Loss

Exponential loss has some special properties under cost-weighting. We see that for $\alpha = \beta = -1/2$ the denominator of the cost-weighted weight function is constant and hence:

$$\omega^{cw}(q) \;\sim\; q^{-3/2}(1-q)^{-3/2} \;.$$

That is, the weight function does not depend on the cost c. The link function, however, depends on c according to Equation (5.8). In effect, cost-weighting exponential loss is achieved by shifting the modeling scale alone:

$$F(q) \;=\; s \log \frac{q}{1-q} - s \log \frac{c}{1-c}$$

Because for cost-weighted link functions one still has $q'(F) = q(1-q)/s$, the working weights look superficially the same as without cost-weighting: $w_n = 1/s^2 \cdot [q(1-q)]^{1/2}$. However, the meaning of $q = q(F)$ has changed because this is now the shifted logistic link.

Chapter 6

Stagewise Fitting of Additive Models

6.1 Boosting

Boosting is an algorithm that has achieved huge success in the field of machine learning. It combines a group of weak classifiers a weighted voting scheme so that a strong learner results. In the following we describe the most common version of the boosting algorithm, *discrete Adaboost*, which was proposed by Freund and Schapire (1996):

Suppose the response y_n^* is labeled as -1 or 1,

1. Start with weights $w_n = 1/N$, $n = 1, ..., N$.

2. For $m = 1$ to M:

(a) Fit a classifier $f_m(\mathbf{x})$ to the training data using weights w_n.

(b) Compute the weighted misclassification error of $f_m(\mathbf{x})$:

$$\text{err}_m \;=\; \frac{\sum_{n=1,\ldots,N} w_n I(y_n^* \neq f_m(\mathbf{x}_n))}{\sum_{n=1,\ldots,N} w_n}$$

(c) Compute the logit of the weighted misclassification error:

$$\alpha_m = \log\left(\frac{1-\text{err}_m}{\text{err}_m}\right)$$

(d) Set $w_n \leftarrow w_n \cdot \exp[\alpha_m \cdot I(y_n^* \neq f_m(\mathbf{x}_n))]$, $n = 1, ..., N$.

3. Output sign$[\sum_{m=1,\ldots,M} \alpha_m f_m(\mathbf{x})]$ as the ± 1-labeled classifier.

Adaboost up-weights those points that are misclassified according to how well the classifier performs at the current iteration. Thus intuitively, the classifier would focus more on the hard cases in the next round.

In FHT(2000), Adaboost was viewed as stagewise additive fitting which exploits Newton steps to optimize exponential loss, which we write as follows with non-parametric fits in mind:

$$\mathbf{R} \;=\; \frac{1}{N}\sum_{n=1\ldots N} \tilde{L}(-y_n^*\, F_n)\;, \qquad \tilde{L}(F) = \exp(F)\;.$$

The Newton updates are based on the following components:

$$w_n \;=\; \exp(-y_n F_n^{old})\;, \qquad\qquad z_n \;=\; y_n\;,$$

where F has the additive form

$$F(\mathbf{x}) \;=\; \sum_{t=1,\ldots,T} f_t(\mathbf{x})$$

In their paper, FHT also created LogitBoost which optimizes log-loss by fitting a stagewise additive model.

6.2 Stagewise Fitting of Additive Models by Optimizing General Losses

In this section we adopt FHT's view of boosting and adapt the IRLS iterations to stagewise forward additive modeling.

For additive modeling one needs a set $\mathcal{F}$ of possible basis functions $f(\mathbf{x})$. An additive model is a linear combination of elements of $\mathcal{F}$: $F(\mathbf{x}) = \sum_k b_k f_k(\mathbf{x})$. One does not assume that $\mathcal{F}$ is a linear space because in a typical application of boosting $\mathcal{F}$ would be a set of trees with a certain structure, such as stumps (trees of depth one), that are not closed under linear combinations.

Stagewise fitting is the successive acquisition of terms $f^{(K)} \in \mathcal{F}$ given that terms $f^{(1)}, \ldots, f^{(K-1)}$ for a model $F^{(K-1)} = \sum_{k=1..K-1} b_k f^{(k)}$ have already been acquired. In what follows we write F^{old} for $F^{(K-1)}$, f for $f^{(K)}$, b for b_K, and finally $F_{new} = F^{old} + b f$. For a loss function

$$\mathbf{R}(F) \;=\; \mathbf{E}_{\mathbf{x},y}\, \mathbf{L}(y|q(F(\mathbf{x}))) \;,$$

the conceptual goal is to find

$$(f,\, b) \;=\; \mathrm{argmin}_{f\in\mathcal{F},\; b\in\mathbb{R}}\, \mathbf{R}(\, F^{old} + b\, f) \;.$$

Given a training sample $\{(\mathbf{x}_n,\, y_n)\}_{n=1..N}$, the loss is estimated by

$$\hat{\mathbf{R}}(F^{old} + b\, f) \;=\; \frac{1}{N} \sum_{n=1..N} \mathbf{L}(y_n \,|\, q(\, F^{old}(\mathbf{x}_n) + b\, f(\mathbf{x}_n)\,)) \;.$$

With these modifications stagewise fitting can proceed according to the IRLS scheme in such a way that each IRLS step produces a new term f, a coefficient b, and an

update $F^{new} \;=\; F^{old} + b\,f$ based on current weights w_n, current estimated class 1 probabilities q_n, and current responses z_n, as follows:

$$\begin{aligned}
F_n &= F^{old}(\mathbf{x}_n)\ , \\
q_n &= q(F_n)\ , \qquad q'_n \;=\; q'(F_n)\ , \qquad q''_n \;=\; q''(F_n)\ , \\
w_n &= \omega(q_n)\,{q'_n}^2 \;-\; (y_n - q_n)\,\left(\omega'(q_n)\,{q'_n}^2 \;+\; \omega(q_n)\,q''_n\right)\ , \\
z_n &= (y_n - q_n)\,\omega(q_n)\,q'_n/w_n\ .
\end{aligned}$$

In practice, f is the result of some heuristic search procedure such as greedy tree construction or, in the simplest case, search for the best fitting stump (a tree with only one split). Whatever this may be, we denote it by

$$f \;\leftarrow\; \text{feature-proposer}(X,\, W,\, \mathbf{z})\ .$$

We can think of f as a proposal for a derived predictor vector $\mathbf{f} = (f(\mathbf{x}_1), ..., f(\mathbf{x}_N))^T$ which is to be used instead of X in the IRLS step at the current stage. The remaining task is to optimize the coefficient b for $\mathbf{f}$. Sometimes, the coefficient b is absorbed in f because the feature-proposer may find b at the same time as it performs the search, so no optimization is necessary. At other times, the coefficient b has to be optimized explicitly, for example through a line search:

$$b \;=\; \text{argmin}_b \,\frac{1}{N}\sum_{n=1..N} \mathbf{L}(\, y_n \,|\, q(F_n + b f_n)\,)\ ,$$

or approximately by a simple weighted LS regression of $\mathbf{z}$ on $\mathbf{f}$:

$$b \;=\; \frac{\mathbf{z}^T\, W\, \mathbf{f}}{\mathbf{f}^T\, W\, \mathbf{f}}\ .$$

The two standard examples for feature-proposers are the following:

- In "real Adaboost", the class $\mathcal{F}$ is the set of trees of a fixed depth, which is often only depth one and called "stumps".
- In "discrete Adaboost", the class $\mathcal{F}$ is the set of indicator functions of decision regions obtained by thresholding trees.

Discrete Adaboost follows then the same heuristic tree construction as in real Adaboost, followed by thresholding. When $\mathcal{F}$ is the set of stumps, the search for an optimal stump can be performed exhaustively. In this case real and discrete Adaboost are almost identical, except for the treatment of the intercept: real Adaboost can adjust the intercept at each step due to the fitting of two constants (one on either side of the split), whereas discrete Adaboost can fit only one constant, on the area where the indicator is 1.

In general the role of the feature-proposer is to propose descent directions from a stock of possible choices, the set $\mathcal{F}$. The behavior of the feature-proposer and the nature of the set $\mathcal{F}$ are of course crucial for the performance of the boosting scheme based on them.

Chapter 7

Experiments

7.1 Examples of Biased Linear Fits with Successful Classification

We illustrate the fact that linear models can be unsuitable as global models and yet successful for classification at specific thresholds or, equivalently, for cost-weighted misclassification loss. To this end we recreate an artificial example of Hand and Vinciotti (2003) and show that our methods for tailoring loss functions to specific classification thresholds allow us to estimate boundaries. We then show that the Hand and Vinciotti (2003) scenario is a fair approximation to some well-known real data, the Pima Indian diabetes data (UCI Machine Learning database). Finally we apply our model to two more real data: German credit data and adult income data from the UCI Machine learning database.

Example 1. Artificial data: We start with a linear model, that is, a composition $q(\mathbf{b}^T x)$ of a linear function, $\mathbf{b}^T x$, and a link function, $q()$, as in logistic regression. Note that the level curves, $\{\, \mathbf{x} \,|\, q(\mathbf{b}^T \mathbf{x}) = c \,\}$, are parallel lines. This fact can be used to construct scenarios $\eta(\mathbf{x})$ for which the model $q(\mathbf{b}^T \mathbf{x})$ is globally inadequate and yet can produce excellent classification boundaries. Such a construction was demonstrated by Hand and Vinciotti (2003): they designed $\eta(\mathbf{x})$ as a "rule surface" with the shape of a smooth spiral ramp on the unit square, as depicted in Figure 7.1. The critical feature of the surface is that the optimal classification boundaries $\eta(\mathbf{x}) = c$ are all linear but not parallel. The absence of parallelism renders any linear model $q(\mathbf{b}^T \mathbf{x})$ unsuitable as a global fit, but the linearity of the classification boundaries allows linear models to describe these boundaries, albeit every level requires a different linear model. The point of Hand and Vinciotti's (2003) and our schemes is to home in on these level-specific models.

In recreating Hand and Vinciotti's (2003) example, we simulated 4,000 data points whose two predictors were uniformly distributed in the unit square, and whose class labels had a conditional class 1 probability $\eta(x_1, x_2) = x2/(x1+x2)$. We fitted a linear model with the logistic link function $q(t) = 1/(1+\exp(t))$, using a proper scoring rule in the Beta family with $\alpha = 6$ and $\alpha/\beta = 0.3/0.7$ in order to home in on the $c = 0.3$ boundary (Section 3.4). We also fitted a linear model with the cost-weighted logistic link $q(t) = c/(c + (1-c)\exp(t))$, where $c = .3$, using a proper scoring rule in the cost-weighted Beta family with $\alpha = \beta = 4$ (Section 4.3). Figure 7.1, which is similar to Hand and Vinciotti's (2003) Figure 1 shows the success: the estimated boundary is

close to the true 0.3-boundary. The figure also shows that the 0.3-boundary estimated with the log-loss of logistic regression is essentially parallel to the 0.5-boundary, which is sensible because logistic regression is bound to find a compromise model which, for reasons of symmetry between the two labels, should be a linear model with level lines roughly parallel to the true 0.5-boundary.

We also created a model which was based on a variation of Hand and Vinciotti's (2003) example. We simulated 4,000 data points whose two predictors were uniformly distributed in the unit square, and whose class labels had a conditional class 1 probability $\eta(x_1, x_2) = kx_2/(x_1 + kx_2)$, where k is constant. The parameter k was used to adjust the priors of the two classes, when $k > 1$, there were more class 1 than class 0, otherwise there were more class 0 than class 1. In our simulated data, we set $k = 5$, so class 1 was roughly 78% of the data. We would like to show that under an unbalanced design such as this, even when estimating a boundary $\eta(x) = 0.5$, one might want to use a tailored loss other than log-loss. This is illustrated in Figure 7.2 which shows the 0.3-boundary, 0.5-boundary and 0.7-boundary estimated with log-loss and tailored losses. Indeed, the imbalance of the classes makes the tailored solutions superior to log-loss even for the 0.5-boundary.

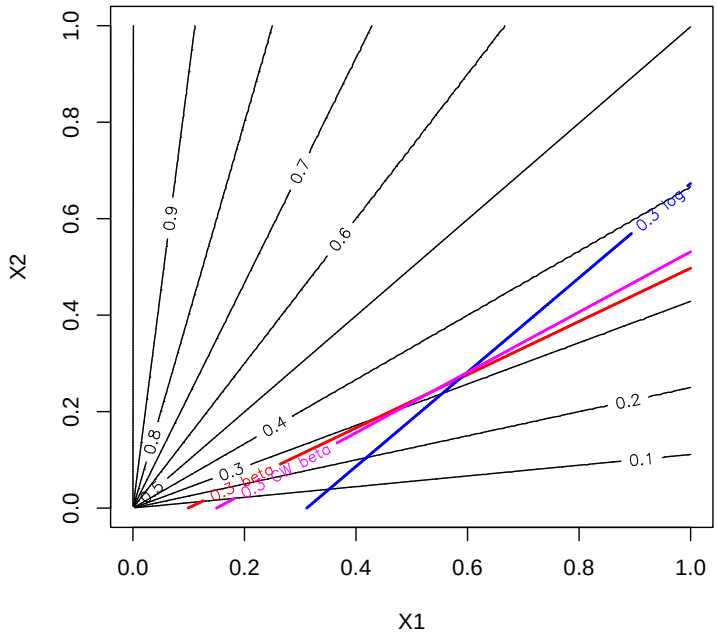

Figure 7.1: *Hand and Vinciotti's Artificial Data: The class probability function* $\eta(\mathbf{x})$ *has the shape of a smooth spiral ramp on the unit square with axis at the origin. The bold line marked "0.3 log" shows a linear logistic fit thresholded at* $c = 0.3$*. The second bold line, marked "0.3 beta", shows a thresholded linear fit corresponding to a proper scoring rule in the Beta family with parameters* $\alpha = 6$ *and* $\beta = 14$*. The last bold line, marked "0.3 cw beta", shows thresholded linear fit corresponding to an cost-weighted* F*-loss derived from the Beta family with parameters* $\alpha = 4$ *and* $\beta = 4$*.*

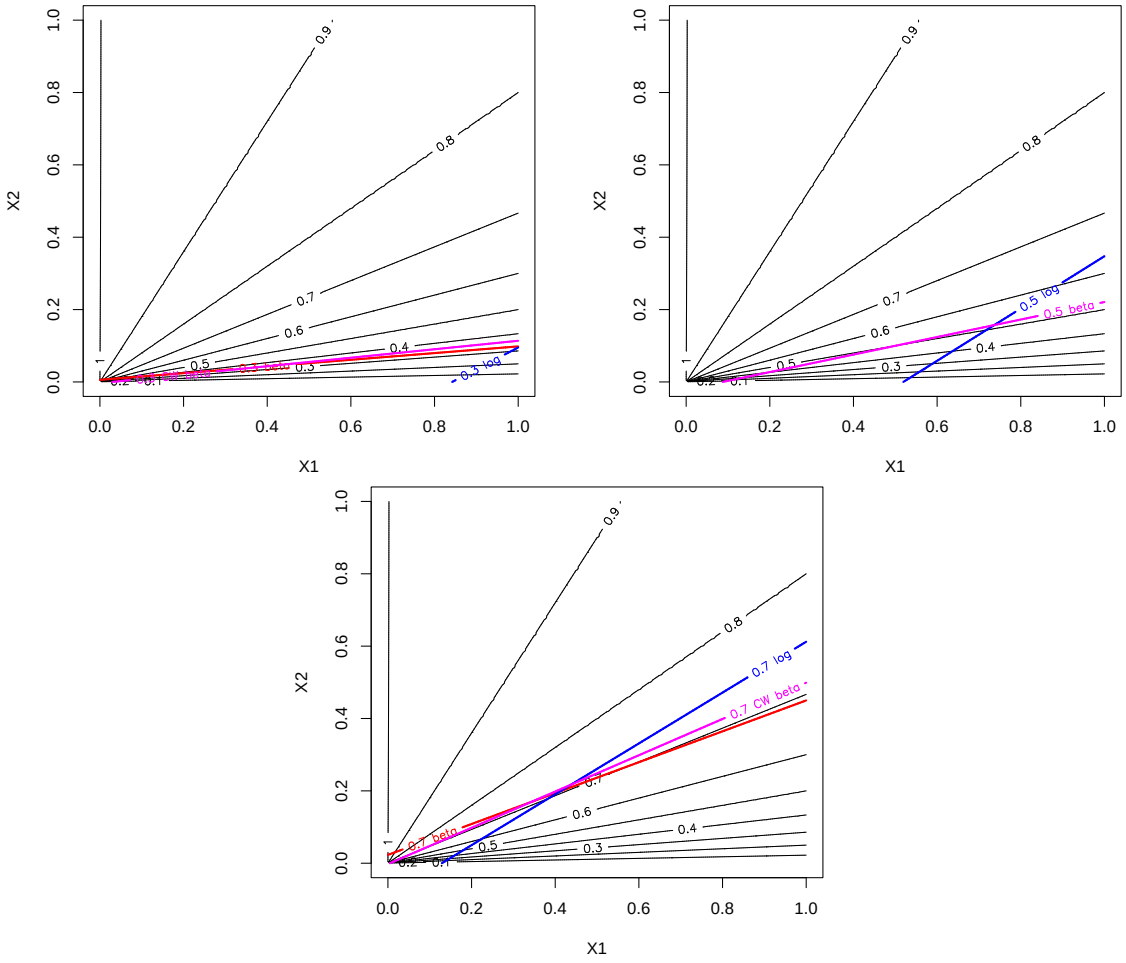

Figure 7.2: *A variation of Hand and Vinciotti's Artificial Data with unbalanced classes: The class 1 probability $\eta(\mathbf{x})$ has the shape of a smooth spiral ramp with axis at the origin. Shown are estimated contours of the levels 0.3, 0.5 and 0.7. The blue lines show log-loss estimates (logistic regression), the red lines Beta tailored estimates, and the magenta lines cost-weighted tailored estimated, with tailoring for the respective levels.*

Example 2. Pima data: Our first real data example shows that Hand and Vinciotti's stylized scenario is met in the well-known Pima Indians diabetes data when restricting attention to the two major predictors, here labeled "PLASMA" and "BODY". Figure 7.3 shows scatterplots of these two predictors with glyph-coding of the classes. We illustrate the estimation for $c = .1$, $c = .5$ and $c = .8$. Cost-weighted tailoring seems to indicate the spiral shape the best.

But is there really a spiral ramp? We give independent evidence in Figure 7.4 which shows thresholded class probabilities that are estimated with a completely different method, namely, the nearest-neighbor method. The figure clearly shows the sweeping motion of the level curves as the level is raised, lending credence to the idea that the level curves may be well-approximated by a set of non-parallel lines.

For comparison, we report in Tables 7.1, 7.2 and 7.3 the mean of 10-fold cross-validated cost-weighted misclassification loss (error) for different costs c. The tailored losses often yield slightly better and never worse results than log-loss. If the numbers are to be believed, cost-weighted Beta tailoring may hold a slight edge over plain Beta tailoring.

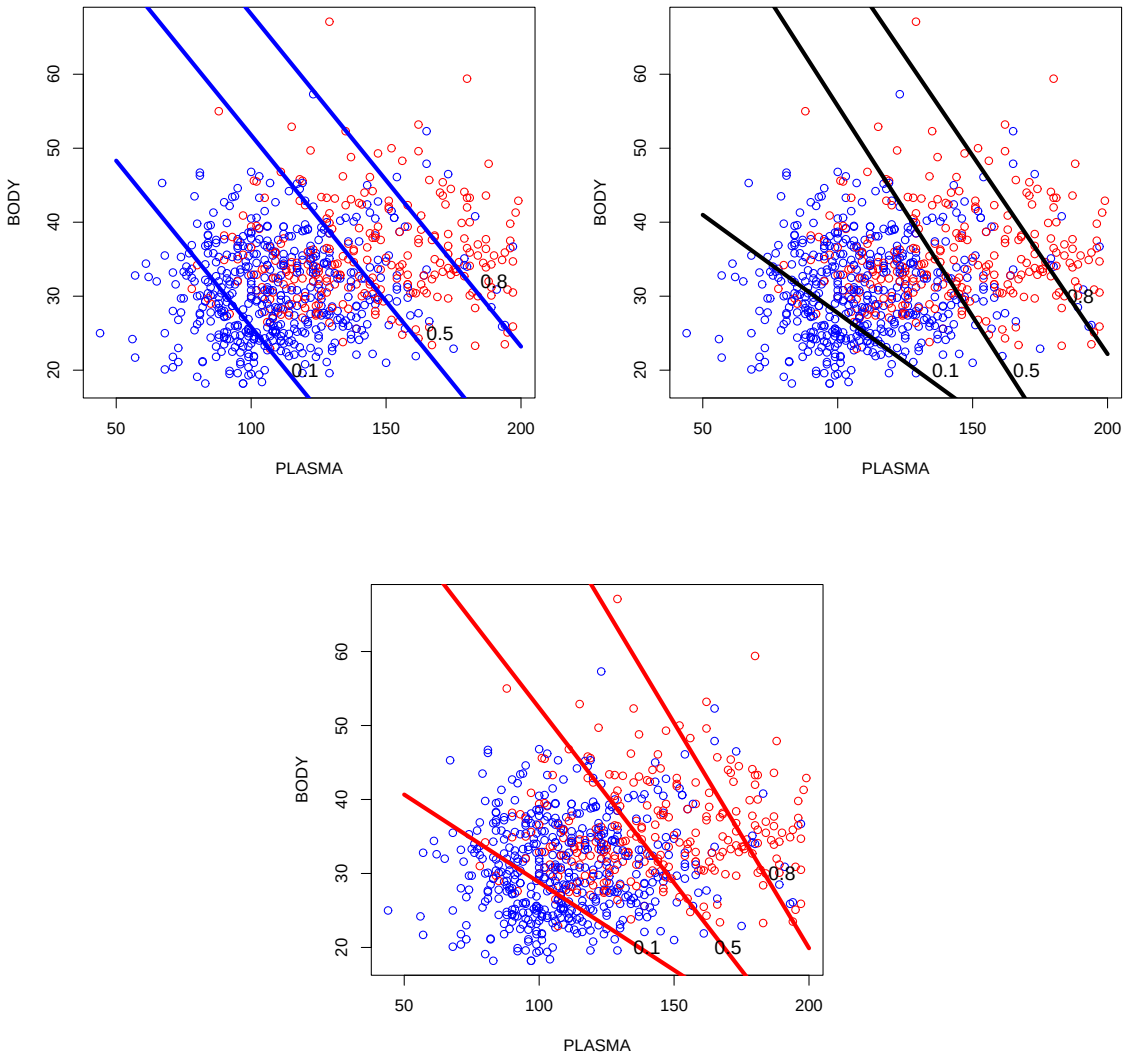

Figure 7.3: *The Pima Indian Diabetes Data, BODY against PLASMA. The colored lines show the probability contours estimated with logistic regression (blue), Beta tailoring (black), and cost-weighted tailoring (red).*

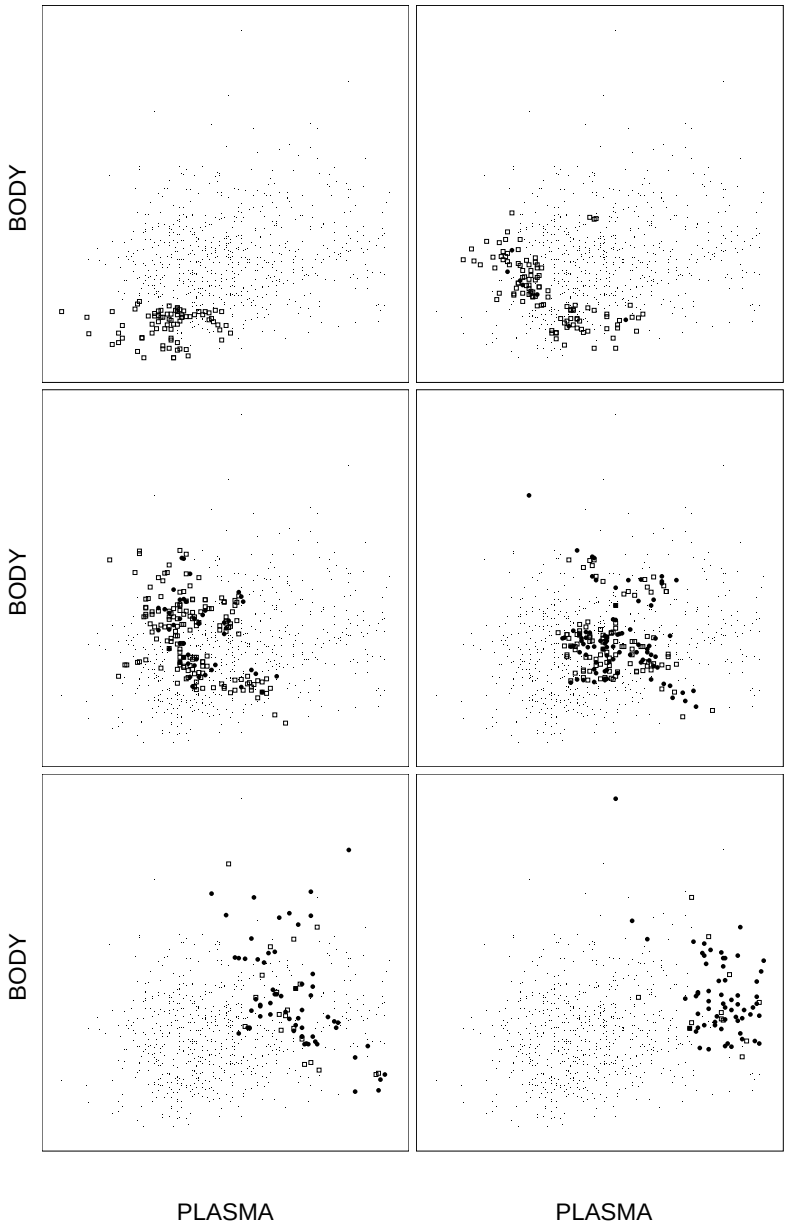

Figure 7.4: *The Pima Indian Diabetes Data,* BODY *against* PLASMA*. The highlights represent slices with near-constant* q *(*$c-\epsilon \leq q^1 \leq c+\epsilon$*). The values of* q *in the slices increase left to right and top to bottom. Open squares: no diabetes (class 0), filled circles: diabetes (class 1).*

Table 7.1: Pima Indian Diabetes: $c = 0.1$

	class 1 error	class 0 error	cost-weighted error	parameters
log-loss	0.5	36.7	0.055 ± 0.003	$\alpha = \beta = 0$
Beta	0.7	34.4	0.054 ± 0.003	$\sigma^{opt} = 0.08$
cost-weighted	0.7	32.7	0.052 ± 0.004	$\alpha^{opt} = 2$

Table 7.2: Pima Indian Diabetes: $c = 0.5$

	class 1 error	class 0 error	cost-weighted error	parameters
log-loss	12.2	5.8	0.120 ± 0.008	$\alpha = \beta = 0$
Beta	11.8	5.8	0.117 ± 0.009	$\alpha^{opt} = 2$
cost-weighted	11.8	5.8	0.117 ± 0.009	$\alpha^{opt} = 2$

Example 3. German credit data: This dataset has 1000 observations and 20 attributes of creditors of a bank. The goal is to predict the likelihood of good credit of a customer. The data comes with a cost matrix that assigns a cost of 5 if a customer is misclassified as good credit when he has actually bad credit, and a cost of 1 if a customer is misclassified as bad credit when he actually has good credit. We labeled a good credit as class 1 and a bad credit as class 0. Thus, to minimize the cost of misclassification, we classify a customer as class 1 if $\eta(x) > 1/6$, otherwise class 0.

We compared again log-loss with Beta tailoring and cost-weighted tailoring. The target threshold is $c = 1/6$ based on the given cost information. When evaluating the models, we estimated α by fitting models on randomly selected 640 cases, optimiz-

Table 7.3: Pima Indian Diabetes: $c = 0.8$

	class 1 error	class 0 error	cost-weighted error	parameters
log-loss	20.7	1.4	0.070 ± 0.006	$\alpha = \beta = 0$
Beta	21.6	1.3	0.072 ± 0.006	$\sigma^{opt} = 0.18$
cost-weighted	21.6	1.2	0.070 ± 0.006	$\alpha^{opt} = 2$

ing cost-weighted misclassification loss on 160 cross-validation cases, and measuring performance on 200 hold-out cases, —repeating this whole procedure 100 times (= 20 times 5-fold holdouts). Thus each method is now characterized by 100 values of honestly estimated cost-weighted misclassification errors.

We now compare these three sets of values with a graphical version of a 2-sample permutation test. The upper panel of Figure 7.5 shows the result of the comparison of log-loss and Beta-tailored loss: The blue line represents the empirical quantile-quantile curve with the sorted Beta loss values on the horizontal axis and the sorted log-loss values on the vertical axis. The red band is a spaghetti graph that overplots many (100) empirical quantil-quantile curves when the labels "log-loss" and "Beta loss" are randomly permuted against the (200) values of the estimated misclassification losses. The fact that the blue curve stays within the red band shows that the distributions of the log-loss and the Beta loss error values are indistinguishable modulo randomness.

Similarly, the lower panel of Figure 7.5 shows the comparison of log-loss with cost-weighted F-loss. In this case the blue curve is consistently above the red band,

Table 7.4: German Credit: $c = 1/6$

	class 1 error	class 0 error	cost-weighted error
log-loss	13.43	46.92	0.095 ± 0.001
Beta	10.64	59.44	0.094 ± 0.001
cost-weighted	8.75	56.32	0.083 ± 0.001

indicating a statistically significant difference simultaneously for all quantiles of the error distributions.

For comparison, we report in Table 7.4 the mean of 100 cross-validated cost-weighted errors for costs $c = 1/6$. The tailored losses yield better results than log-loss, and cost-weighted tailoring performs best.

Figure 7.6 shows the distribution of the estimated σ^{opt} for Beta tailoring and α^{opt} for cost-weighted tailoring. In spite of its slightly better performance, cost-weighted tailoring has a more erratic distribution of estimated parameter values in that a majority is $\alpha^{opt} = 2$, but the tail straggles up to $\alpha^{opt} = 12$. This behavior does not seem to impinge on performance, though.

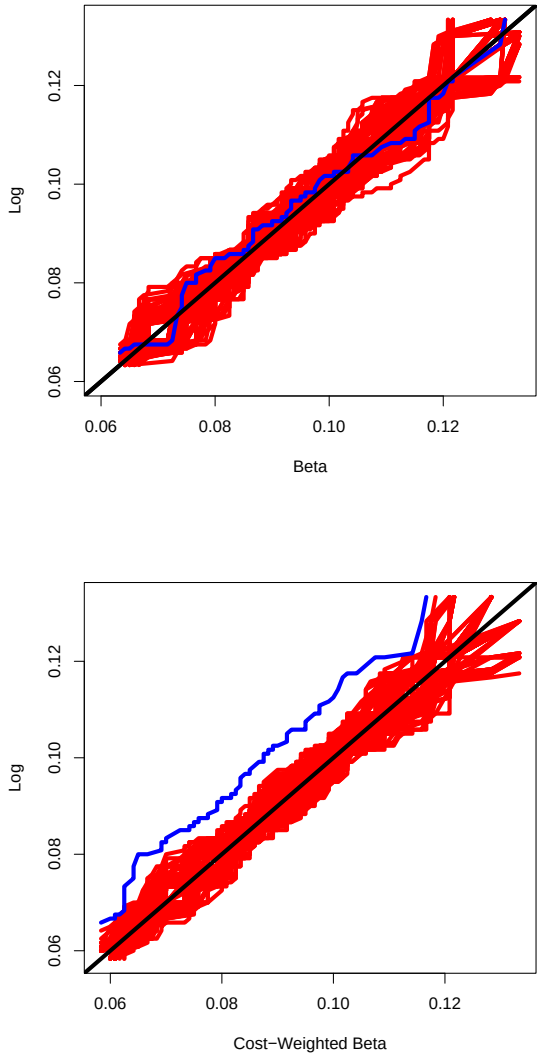

Figure 7.5: *German credit data: Graphical permutation tests for comparing the test errors based on log-loss, Beta loss with* $\alpha/\beta = c/(1-c)$*, and cost-weighted* F*-loss with* $c = 1/6$.

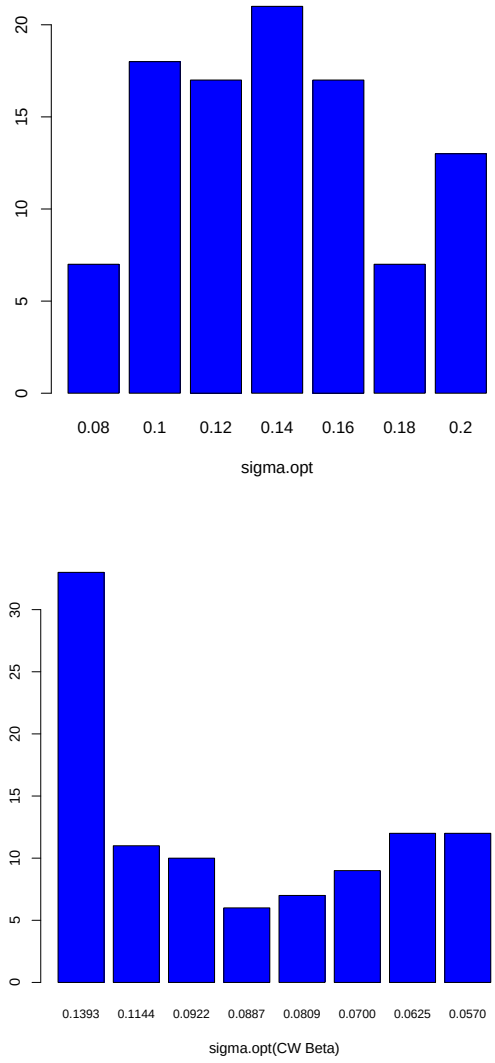

Figure 7.6: *Histogram of the estimated σ for Beta tailoring (upper panel) and estimated α for cost-weighted tailoring (lower panel) for the German credit data with $c = 1/6$. In the estimation of σ and α with optimization of cross-validated cost-weighted error we constrained σ between 0.06 and 0.2 in 0.02 increments and α to $\{2, 3, 4, 5, 6, 8, 10, 12\}$ with the corresponding σ to* $\{0.139,0.114,0.092,0.0887,0.081,0.070,0.063,0.057\}$.

Table 7.5: Adult income: $c = 0.2$

	class 1 error	class 0 error	cost-weighted error	parameters
log-loss	751	1950	0.066	$\alpha = \beta = 0$
Beta	668	2190	0.065	$\sigma^{opt} = 0.1$
cost-weighted	556	2452	0.062	$\sigma^{cw^{opt}} = 0.065$

Table 7.6: Adult income: $c = 0.5$

	class 1 error	class 0 error	cost-weighted error	parameters
log-loss	1934	461	0.080	$\alpha = \beta = 0$
Beta	1830	538	0.079	$\sigma^{opt} = 0.07$
cost-weighted	1830	538	0.079	$\sigma^{cw^{opt}} = 0.07$

Example 4. Adult income data: These are census data of adult annual income with 14 attributes. We are interested in predicting whether a person's income exceeds 50k or not. There are a training set of size $30,162$ and a test set of size $15,060$. We hold out one-fifth of the training set as the validation set to choose the optimal parameters of the weight functions. The cost-weighted misclassification errors on the test set and the optimal parameters for $c = .2$, $c = .5$ and $c = .8$ are shown in Tables 7.5, 7.6 and 7.7. Again, we see a slight benefit of tailored losses over log-loss and slight superiority of cost-weighted tailoring over Beta tailoring.

Table 7.7: Adult income: $c = 0.8$

	class 1 error	class 0 error	cost-weighted error	parameters
log-loss	2979	42	0.042	$\alpha = \beta = 0$
Beta	3003	31	0.042	$\sigma^{opt} = 0.18$
cost-weighted	2850	67	0.041	$\sigma^{cw^{opt}} = 0.065$

Example 5. More real world data from UCI.

We illustrate tailoring on a few more datasets from the UCI ML database. For details about the datasets we refer to documentation given at this website. The number of predictors (features), number of observations and the baseline classification (proportion of 1's) are shown in Table 7.8. It should be said that these datasets are mostly not suitable for tailoring because of the large number of features relative to the number of observations, which makes these cases of overly great flexibility and hence likely overfit. Recall that tailoring is promising for biased models, and bias is detectable only if the fitted function class is relatively rigid and the sample size is relatively large. Tailoring would more likely be of help if model selection were applied first to reduce the number of predictors.

Be this as it may, Tables 7.9, 7.10, 7.11, 7.12 and 7.13 show the results. In a majority of cases tailoring does not contribute, but in a few situations it seems to improve classification. Examples are the levels $c = 0.8$ for the Cleve data (Table 7.9) where CW Beta tailoring is beneficial, but curiously Beta tailoring is not. In the Ionosphere data (Table 7.10) both modes of tailoring improve classification at $c = 0.5$,

Table 7.8: Description of More UCI Data

	cleve	ionosphere	spect	wdbc	wpbc
number of features	13	34	22	30	33
number of observations	296	351	267	569	194
proportions of 1's	54%	64%	79%	37%	24%

Table 7.9: CW-Error for Data Cleve

	c = 0.2	c = 0.5	c = 0.8
log-loss	0.055 ± 0.009	0.086 ± 0.011	0.065 ± 0.007
Beta	0.064 ± 0.010	0.082 ± 0.010	0.071 ± 0.008
cost-weighted	0.062 ± 0.011	0.082 ± 0.010	0.057 ± 0.007

but neither at $c = 0.2$ nor $c = 0.8$. For the next two datasets, Spect and Wdbc (Tables 7.11 and 7.12), no benefit is discernable. The last dataset, Wpbc (Table 7.13) shows an improvement for $c = 0.5$ but not for $c = 0.2$ and $c = 0.8$.

If there is a message, it is that the benefits of tailoring cannot be predicted without trying. Essentially, one has to try a range of values α as we did in these data examples where α was selected with cross-validation. [In detail: We form a training set of size 80%; we partition the training set again into a subset of 80% and and a hold-out set of 20% to determine α; then we apply the model with the estimated α to the whole training set and evaluate out-of-sample misclassication on the test set.]

Table 7.10: CW-Error for Data Ionosphere

	c = 0.2	c = 0.5	c = 0.8
log-loss	0.030 ± 0.007	0.060 ± 0.008	0.070 ± 0.013
Beta	0.038 ± 0.005	0.054 ± 0.007	0.088 ± 0.012
cost-weighted	0.030 ± 0.007	0.054 ± 0.007	0.086 ± 0.011

Table 7.11: CW-Error for Data Spect

	c = 0.2	c = 0.5	c = 0.8
log-loss	0.043 ± 0.006	0.097 ± 0.007	0.077 ± 0.015
Beta	0.041 ± 0.005	0.100 ± 0.008	0.081 ± 0.010
cost-weighted	0.042 ± 0.006	0.100 ± 0.008	0.080 ± 0.011

7.2 Examples of Cost-Weighted Boosting

We apply tailored losses to more flexible nonparametric models, in particular, boosting with stumps or shallow trees. We will find evidence that the effect of weighting with tailored losses becomes less important as the complexity of the fitted functional class increases. We illustrate this with an artificial example fitted with stumps and trees of depth three: while tailoring has a slight edge over log-loss when using stumps, they perform equally when using trees of depth three. On the German credit data, tailoring is inferior to log-loss, and on the adult income data tailoring is very slightly superior to log-loss. Thus a mixed picture emerges: tailoring combined with boosting can be beneficial, in particular when used with "very weak learners" such as stumps,

Table 7.12: CW-Error for Data Wdbc

	c = 0.2	c = 0.5	c = 0.8
log-loss	0.018 ± 0.004	0.018 ± 0.006	0.013 ± 0.003
Beta	0.021 ± 0.005	0.017 ± 0.004	0.016 ± 0.004
cost-weighted	0.022 ± 0.005	0.017 ± 0.004	0.016 ± 0.004

Table 7.13: CW-Error for Data Wpbc

	c = 0.2	c = 0.5	c = 0.8
log-loss	0.088 ± 0.021	0.118 ± 0.016	0.071 ± 0.014
Beta	0.106 ± 0.026	0.108 ± 0.008	0.111 ± 0.012
cost-weighted	0.102 ± 0.023	0.108 ± 0.008	0.104 ± 0.015

but it all depends on the data.

Example 1. Artificial data: We create a variation of Hand's example in the previous section. There are 2000 observations in the training set. Instead of letting variable X_1 uniformly distributed in $[0, 1]$, we let it uniformly distributed in $[-1, 1]$. If $X_1 < 0$, we let the posterior probability $\eta(x) = x_2^2/(x_1^2 + x_2^2)$, otherwise $\eta(x) = x_1^2/(x_1^2 + x_2^2)$. Thus the posterior probability is a double spiral ramp around the origin which, in a counter-clockwise sweep, ascends in the first quadrant and descends in the second quadrant (see, for example, Figure 7.7). We illustrate estimation for cost 0.3 by fitting an additive model with stumps using Beta and cost-weighted tailored losses. For the Beta tailored loss we choose $\sigma = .1$ fixed, while for the cost-weighted

tailored loss we choose $\alpha = 4$ ($\sigma = 0.10$)fixed. (In particular, we do not estimate these parameters from the data.) We then compare the results tailoring with log-loss, which is FHT's (2000) LogitBoost.

Figure 7.7 shows contour plots for the estimation of the .3 boundary when using log-loss, tailored Beta loss and cost-weighted tailored loss at iteration 50. None of the models generates accurate estimation in terms of geometrically correct boundaries. Clearly log-loss performs the poorest. Yet, by using the tailored loss, one can recover one side of the boundary. After 50 iterations, the stumps start to overfit. From Figure 7.8, one can see that at iteration 300 stumps fit many small windows on one side of the boundary yet they are never able to better approximate the boundary on the other side. Log-loss shows a more symmetric attempt at capturing both sides of the boundary.

Figure 7.9 shows the average cost-weighted misclassification losses for 10 simulations on a test set of size $10,000$. In this artificial example, tailoring wins over log-loss by a considerable margin of about 2%! We also notice a mild tendency to overfit after about 50 iterations, an indication that boosting is in fact not immune to overfitting, as is now common knowledge.

However, if we fit trees with depth three to this data, the advantage of tailoring disappears. Figure 7.10 shows the estimate of the probability contour for $c = .3$ at iteration 10. There is no significant difference between the tailored loss and log-loss. Figure 7.11 shows the cost-weighted test error of log-loss and tailored losses. If one is willing to fit a more complex model in this case, then it is overfitting that one should

worry about instead of the tailoring.

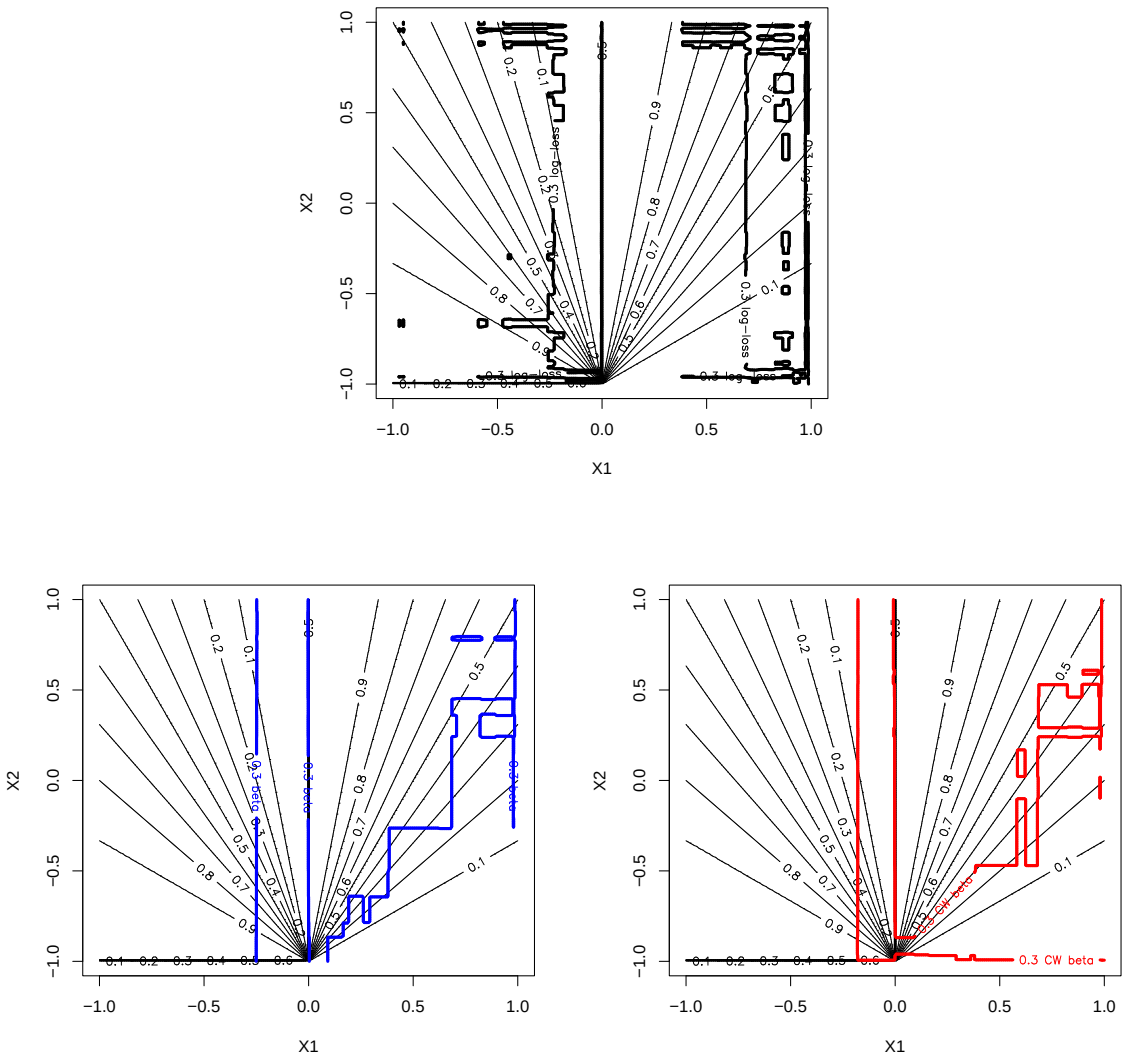

Figure 7.7: *Log-loss, Beta loss and CW-Beta loss with stumps after 50 iterations: estimated class boundaries at* $c = 0.3$. *The data are a variation of Hand and Vinciotti's Artificial Data, where the class probability function* $p(\mathbf{x})$ *has the shape of an ascending-descending smooth spiral ramp.*

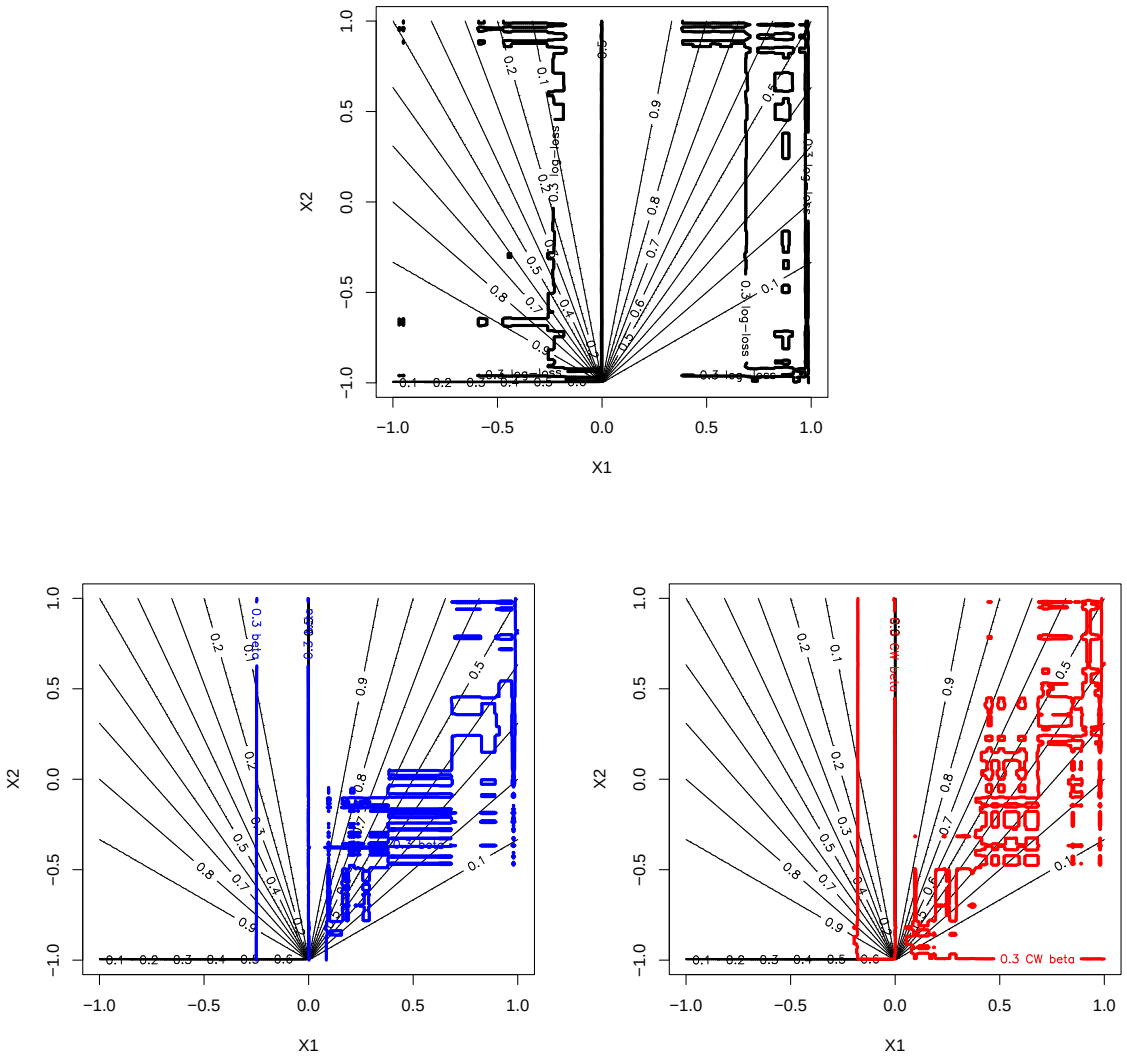

Figure 7.8: *Similar to Figure 7.7: Log-loss, Beta tailored and cost-weighted tailored loss after 300 iterations.*

Examples 2 and 3. German credit data and adult income data: Boosting trees is more complex than fitting linear models because an essential parameter of boosting is the number of iterations used for fitting. The problem is that for every number of iterations, the estimated optimal choice of the tailoring parameter σ may be different. We therefore pre-specify a number of iterations, then randomly sample a proportion of data from the training set and use it to estimate the optimal σ for the specified number of iterations. We then apply the tailored loss with the optimal σ to a test set.

For the German credit data, we estimate the optimal parameter σ both for 20 and 200 iterations. Figure 7.12 shows the mean cost-weighted misclassification loss on holdout sets for $c = 1/6$ averaged over 100 simulations (20 runs of 5-fold holdouts). Figure 7.12 shows a general tendency to overfit the data after about 30 iterations. Tailoring is not beneficial on these data: log-loss dominates both types of tailoring after about 20 iterations.

For the adult income data, we estimate the optimal parameter σ for 300 iterations. Figure 7.13 shows the cost-weighted misclassification loss for three thresholds $c = .2,\ .5$ and .8. The loss is computed from the given test set of these data. Figure 7.13 shows that tailoring is mildly beneficial. This holds when using stumps, but additional experiments (not shown here) indicated that if one fits more complex models with trees of larger depth, the advantage of tailoring disappears.

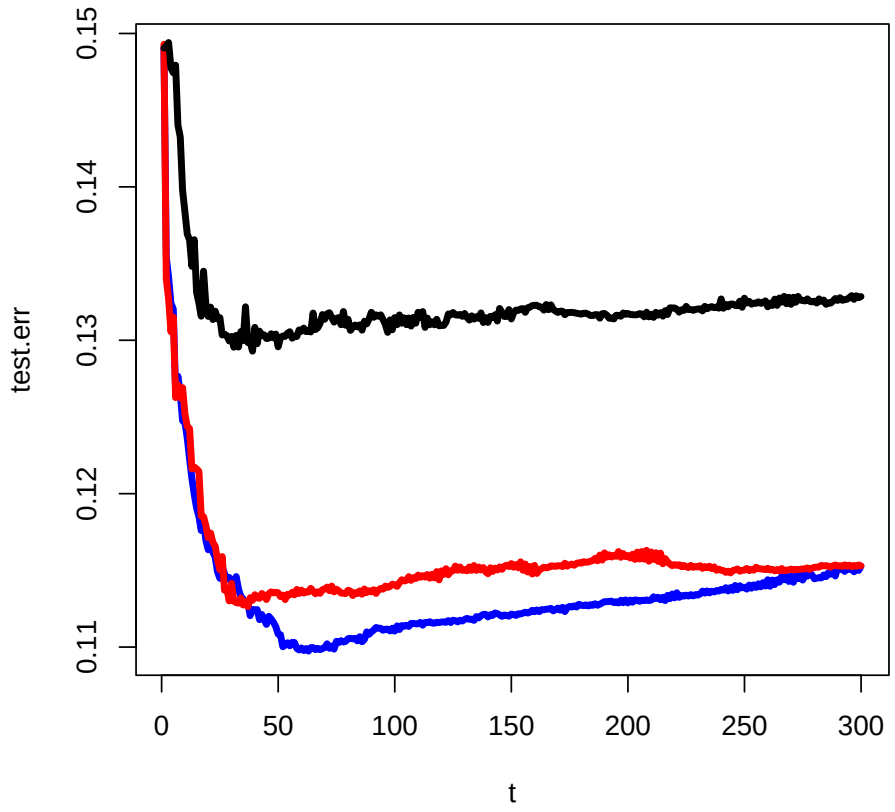

Figure 7.9: *Cost-weighted misclassification test error for $c = .3$. Tailored losses perform better than log-loss by about 2%, but there is a slight tendency to overfit after about 50 iterations.*

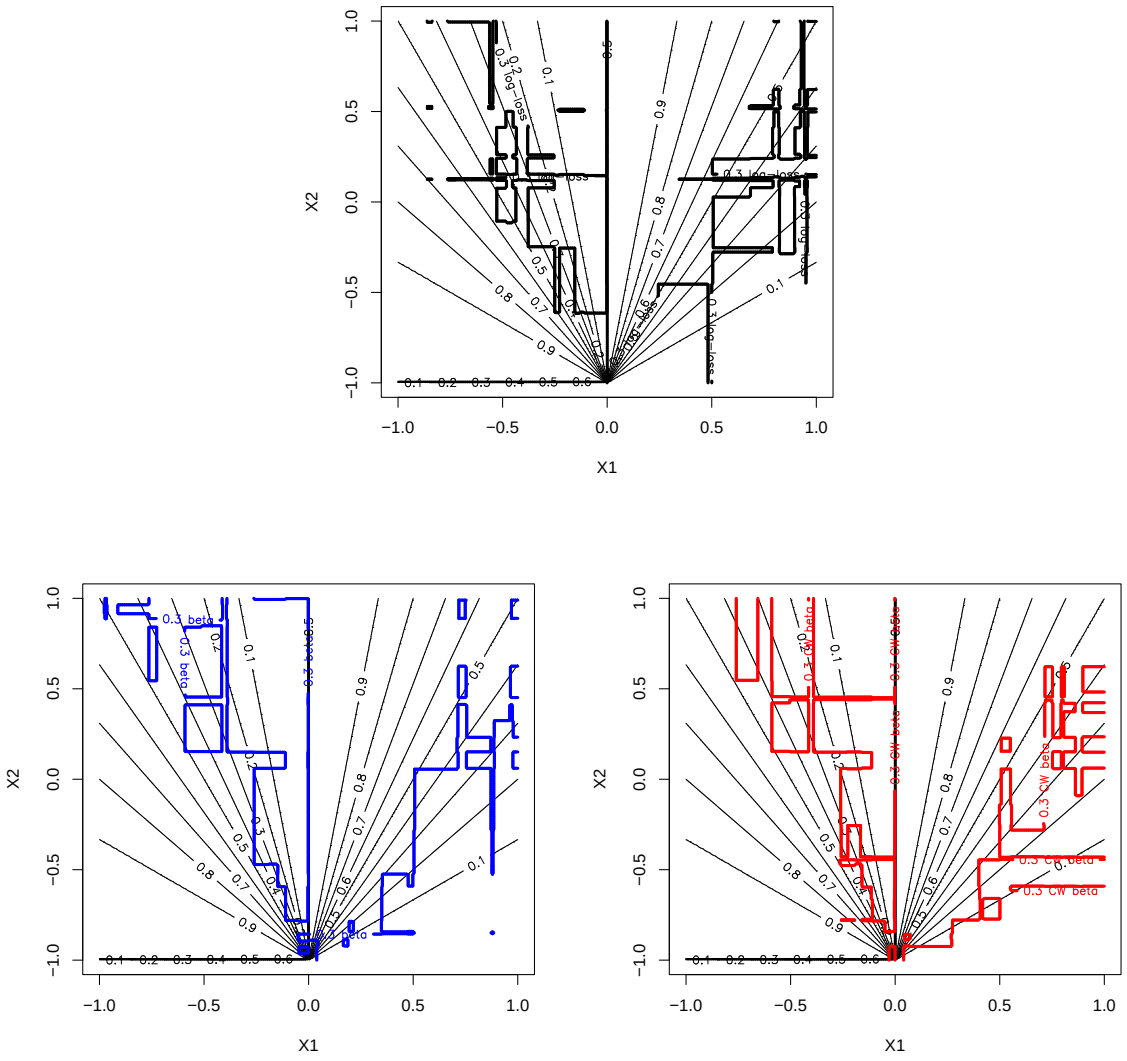

Figure 7.10: *Similar to Figure 7.7: Log-loss, Beta tailored and cost-weighted tailored loss with* **trees of depth three** *after 10 iterations.*

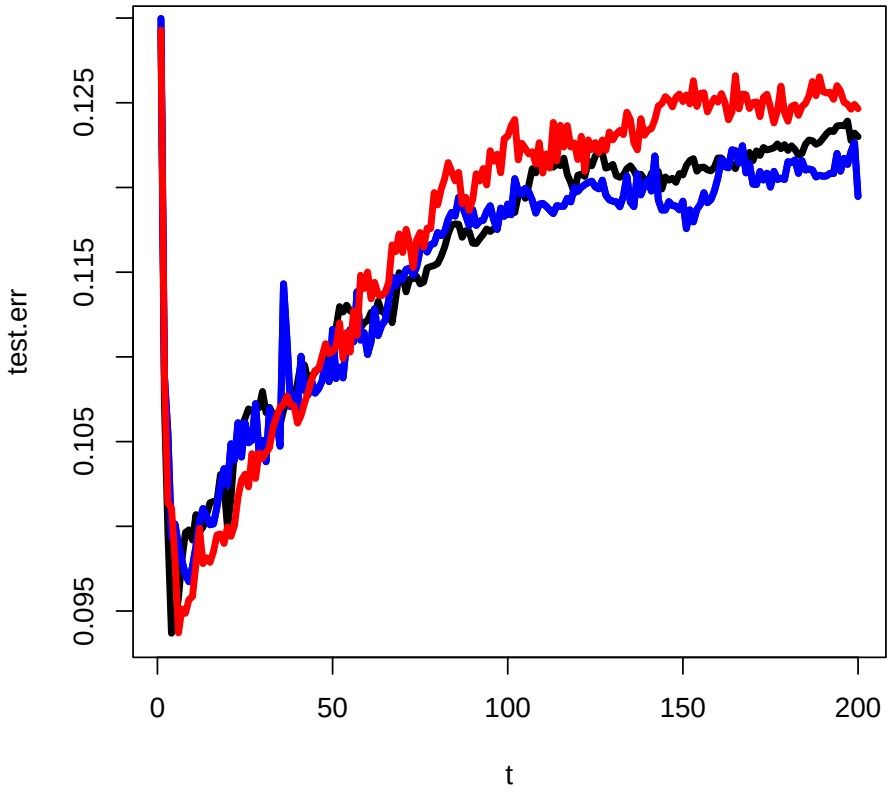

Figure 7.11: *Boosting with trees of depth three to the artificial data: The picture shows the cost-weighted misclassification loss for* $c = .3$ *on test sets. There is no significant difference between log-loss and tailored losses, and they all tend to overfit quickly.*

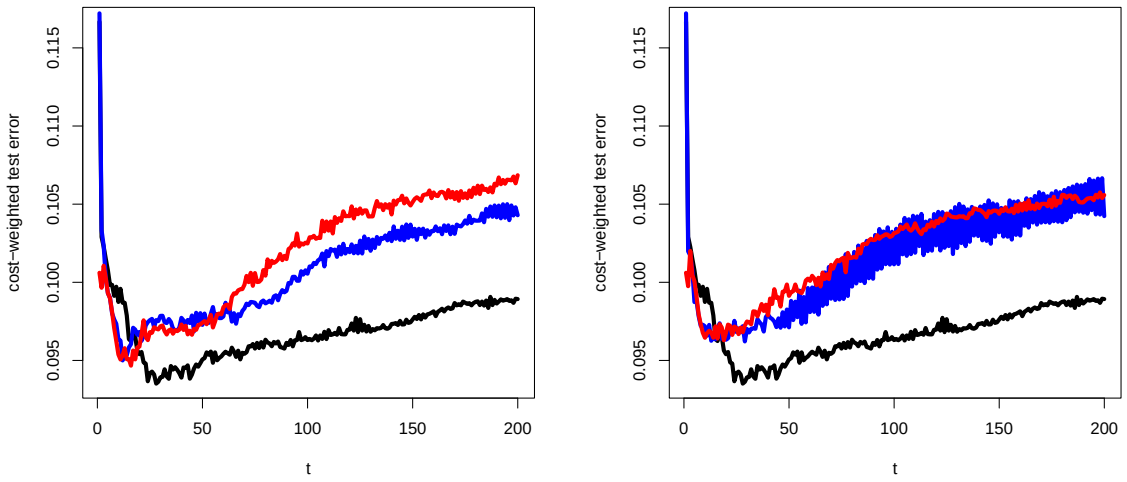

Figure 7.12: *German Credit Data: The frames show the cost-weighted misclassification loss on holdout sets for* $c = 1/6$.

Left panel: optimal choice of σ *at 20 iterations; Right panel: at 200 iterations.*

Black: log-loss; Blue: Beta tailored loss; Red: cost-weighted tailored loss.

Recall, however, that linear regression with tailored loss performs better than boosting (Table 7.4).

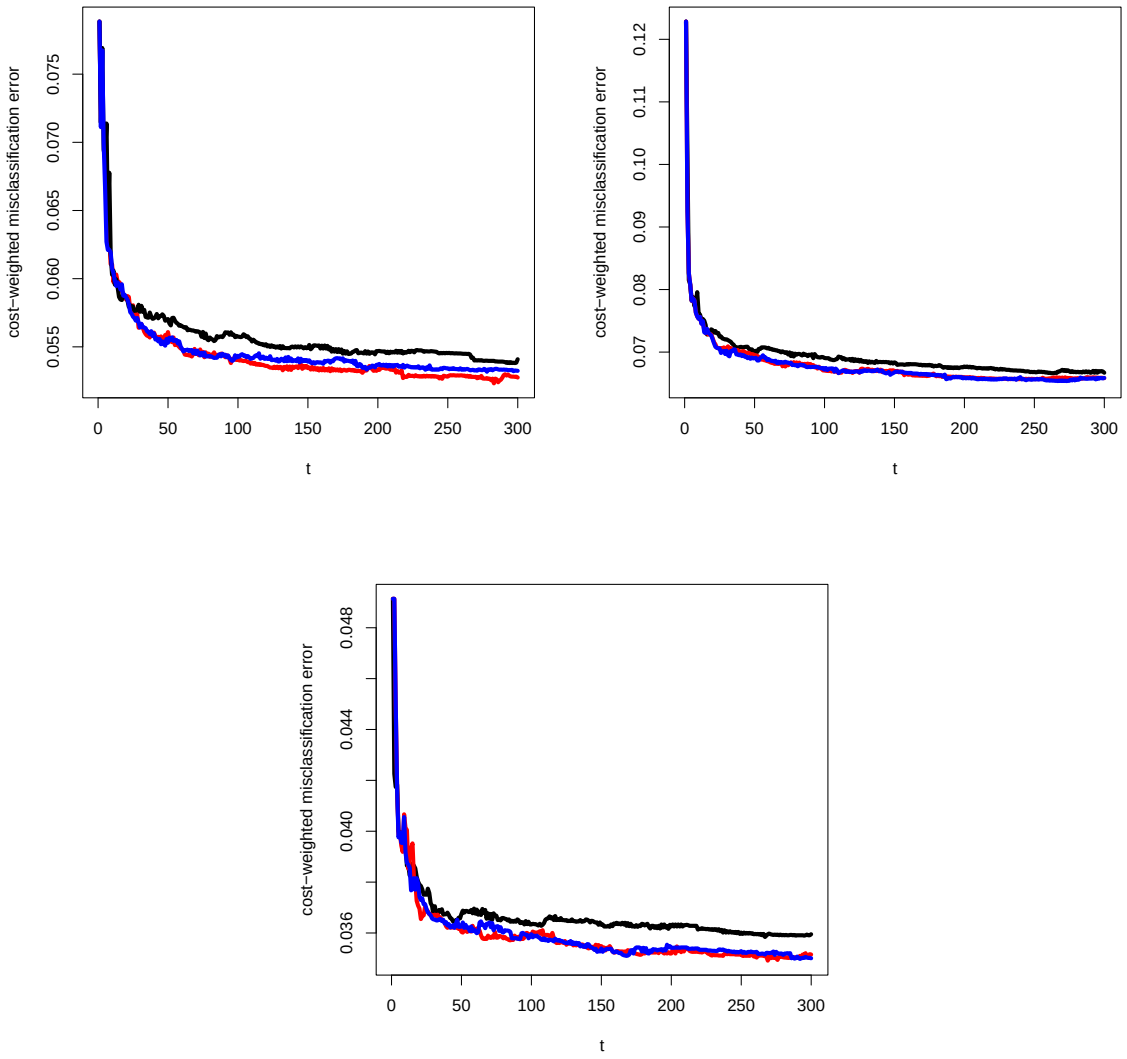

Figure 7.13: *Adult Income Data: The three frames show the cost-weighted misclassification losses on the test set for* $c = .2,\ .5$ *and* $.8$*, respectively. Tailored losses achieve better misclassification loss than log-loss during 300 iterations. The two types of tailoring are virtually indistinguishable.*

Black: log-loss; Blue: Beta tailored loss; Red: cost-weighted tailored loss.

Chapter 8

Conclusions

This work is concerned with binary classification and class probability estimation. We described a very general class of loss functions, called F-losses, that can be used for estimating "uncalibrated probabilities". An important subset is formed by those losses that produce calibrated probability estimates; they are known as "proper scoring rules". General F-losses are compositions of (inverse) link functions and proper scoring rules. We showed that F-losses are closed under cost-weighting, that is, the assignment of different weights to the two types of misclassification losses.

We described the known structure of proper scoring rules and showed how the freedom of choosing among them can be used to tailor fitted models in such a way that good performance is achieved at a prescribed ratio of misclassification costs. Similarly, we showed how cost-weighting F-losses can achieve tailoring at given costs. We also indicated that appropriate degree of tailoring can be estimated from the data with cross-validation.

We gave evidence that tailoring can be of interest in two situations:

1. Tailoring tree-based classifiers to the high or the low end of class 1 probabilities, thus producing trees with vastly greater interpretability than standard trees based on the Gini index or entropy.

2. Tailoring to specific classification thresholds when the models are biased for class probability estimation, yet are capable of describing good classification boundaries. This case can occur, for example, in linear logistic models when the sample size is relatively large compared to the number of fitted degrees of freedom. In such situations it may be beneficial to downweight data points whose class probabilities are far from the intended threshold. In linear models, the heavy reliance of log-loss and exponential loss on the most predictable instances can be a drawback for classification where one cares most about a specific classification boundary rather than global probability estimation.

Application of tailored losses to more flexible nonparametric models such as boosting does not achieve similar improvements as in linear models or surprising interpretability as in trees.

Bibliography

[1] Bregman, L. M. (1967). The Relaxation Method of Finding the Common Point of Convex Sets and its Applications to the Solution of Problems in Convex Programming. *U.S.S.R. Computational Mathematics and Mathematical Physics* **7** (1), 200–217.

[2] Breiman, L. (1996). Bias, variance and arcing classifiers. Technical Report 460, Statistics Department, University of California at Berkeley, Berkeley, CA.

[3] Breiman, L., Friedman, J. H., Olshen, R., and Stone, C. J. (1984). *Classification and Regression Trees*, Belmont, California: Wadsworth.

[4] Buja, A., and Lee, Y.-S. (2001). *Data Mining Criteria for Tree-Based Regression and Classification*, Proceedings of KDD 2001, 27–36.

[5] Clark, L. A., and Pregibon, D. (1992). Tree-Based Models, in *Statistical Models in S*, edited by J. M.Chambers and T. J. Hastie, Pacific Grove, CA: Wadsworth & Brooks/Cole, 377–419.

[6] Collins, M., Schapire, R. E., Singer, Y. (2002). Logistic Regression, AdaBoost and Bregman Distances, *Machine Learning*, **48** (1/2/3).

[7] Cressie, N. and Read, T. R. C. (1984). Multinomial Goodness-of-fit Tests, *J. R. Statist. Soc.* B, **46**, No. 3, 440–464.

[8] Domingos, P. (2000). A Unified Bias-Variance Decomposition for Zero-One and Squared Loss. In: *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, 564–569. Austin, TX: AAAI Press.

[9] Elkan, C. (2001). The Foundations of Cost-Sensitive Learning. In: *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence* (IJCAI), 973–978.

[10] Freund, Y., and Schapire, R. (1996). Experiments with a New Boosting Algorithm. In: *Machine Learning: Proceedings of the Thirteenth International Conference*, 148–156.

[11] Friedman, J. H. (1997). On bias, variance, 0/1 - loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, **1**, 55–77.

[12] Friedman, J. H., Hastie, T., and Tibshirani, R. (2000). Additive Logistic Regression: A Statistical View of Boosting, *The Annals of Statistics* **28**, No. 2, 337–407.

[13] Garthwaite, P. H., Kadane, J. B., and O'Hagan, A. 2005). Statistical Methods for Eliciting Probability Distributions, *J. of the American Statistical Association*, **100**, No. 470, 680-700.

[14] Hand, D. J., and Vinciotti, V. (2003). Local Versus Global Models for Classification Problems: Fitting Models Where it Matters. *The American Statistician* **57**, No. 2, 124–131.

[15] Kearns, M., and Mansour, Y. (1996). On the boosting ability of top-down decision tree learning algorithms. *Proceedings of the Annual ACM Symposium on the Theory of Computing*, 459–468.

[16] Kohavi, R., and Wolpert, D. H. (1996). Bias plus variance decomposition for zero-one loss functions. In: *Proceedings of the Thirteenth International Conference on Machine Learning*, 275–283. Bari, Italy: Morgan Kaufmann.

[17] Kong, E.B., and Dietterich, T.G. (1995). Error-correcting output coding corrects bias and variance. In: *Proceedings of the Twelfth International Conference on Machine Learning*, 313–321. Tahoe City, CA: Morgan Kaufmann.

[18] Lafferty, J. D., Della Pietra S., Della Pietra V. (1997). Statistical Learning Algorithms based on Bregman Distances, in: *Proceedings of the Canadian Workshop on Information Theory 1997.*

[19] Lin, Y. (2001) A Note on Margin-based Loss Functions in Classification in *Statistics and Probability Letters* **68/1**. 73-82.

[20] Lugosi, G., and Vayatis, N. (2004). On the Bayes-Risk Consistency of Regularized Boosting Methods, *The Annals of Statistics* **32**, 30-55.

[21] Murphy, A.H. and Daan, H. (1985). Forecast Evaluation, in: *Probability, Statistics and Decision Making in the Atmospheric Sciences*, eds. Murphy, A.H. and Katz, P.W.

[22] Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning.* San Francisco, CA: Morgan Kaufmann Publishers.

[23] Savage, L.J. (1973). Elicitation of Personal Probabilities and Expectations, *J. of the American Statistical Association* **66**, No. 336, 783–801.

[24] Schervish, M.J. (1989). A General Method for Comparing Probability Assessors, *The Annals of Statistics* **17**, No. 4, 1856–1879.

[25] Shuford, E.H., Albert, A., Massengill, H.E. (1966). Admissible Probability Measurement Procedures, *Psychometrika* **31**, 125–145.

[26] Winkler, R.L, (1993). Evaluating Probabilities: Asymmetric Scoring Rules, *Management Science* **40**, No. 11, 1395–1405.

[27] Schapire, R.E., Singer, Y. (1998). Improved Boosting Algorithms Using Confidence-Rated Predictions, in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory.*

[28] Schapire, R.E. (2002). The Boosting Approach to Machine Learning: An Overview. In: *MSRI Workshop on Nonlinear Estimation and Classification.*

[29] Tibshirani, R. (1996). Bias, variance and prediction error for classification rules. Technical report, Department of Preventive Medicine and Biostatistics and Department of Statistics, University of Toronto, Toronto, Canada.

[30] UC Irvine Machine Learning Repository (2003):
http://www.ics.uci.edu/˜mlearn/MLRepository.html

[31] Weisstein, E. W. (2005). Hypergeometric Function. From MathWorld–A Wolfram Web Resource.
http://mathworld.wolfram.com/HypergeometricFunction.html

[32] Zhang, T. (2004). Statistical Behavior and Consistency of Classification Methods based on Convex Risk Minimization. *The Annals of Statistics* **32**, 56-134.

Appendix: Proofs

Link function: Recall that $F(p)$ is a minimizer of $\tilde{\mathbf{R}}(p|F)$. Now modify the cost-weighted F-loss such that the combined coefficients involving c and p add up to one:

$$\begin{aligned}\tilde{\mathbf{R}}^{cw}(p|F) &\sim \frac{(1-c)p}{(1-c)p+c(1-p)}\tilde{L}_1(-F)+\frac{c(1-p)}{(1-c)p+c(1-p)}\tilde{L}_0(F)\\ &= \tilde{\mathbf{R}}\left(\frac{(1-c)p}{(1-c)p+c(1-p)}\middle| F\right)\end{aligned}$$

It follows that the minimizer is

$$F\left(\frac{(1-c)p}{(1-c)p+c(1-p)}\right) ,$$

hence the link function is

$$F^{cw}(q) = F\left(\frac{(1-c)q}{(1-c)q+c(1-q)}\right) ,$$

which proves the assertion about the cost-weighted link function.

Proper scoring rule. Plug the new link function into the cost-weighted F-loss and obtain

$$\begin{aligned}L_1^{cw}(1-p) &= (1-c)\tilde{L}_1(-F\left(\frac{(1-c)p}{(1-c)p+c(1-p)}\right)) ,\\ L_0^{cw}(p) &= c\tilde{L}_0(F\left(\frac{(1-c)p}{(1-c)p+c(1-p)}\right)) .\end{aligned}$$

Recalling that the original proper scoring rule was $L_1(1-q)=\tilde{L}_1(-F(q))$ and $L_0(q)=\tilde{L}_0(F(q))$ and comparing with the above equations, the assertion about the proper scoring rule follows.

Weight function: The third part is obtained from the following calculation:

$$\begin{aligned}
\omega^{cw}(q) &= \frac{L^{cw\prime}{}_0(q)}{q} \\
&= \frac{c}{q}\, L_0'\left(\frac{(1-c)q}{(1-c)q+c(1-q)}\right)\, \frac{(1-c)c}{((1-c)q+c(1-q))^2} \\
&= \frac{L_0'\left(\frac{(1-c)q}{(1-c)q+c(1-q)}\right)}{\frac{(1-c)q}{(1-c)q+c(1-q)}} \frac{(1-c)^2c^2}{((1-c)q+c(1-q))^3} \\
&= \omega\left(\frac{(1-c)q}{(1-c)q+c(1-q)}\right) \frac{(1-c)^2c^2}{((1-c)q+c(1-q))^3}
\end{aligned}$$

Normalizing constant: We assume that $\omega(q)$ is a density with expectation μ:

$$\int_0^1 \omega(q)\, dq \;=\; 1\;, \qquad \mu \;=\; \int_0^1 q\, \omega(q)\, dq\;.$$

The normalizing constant of $\omega^{cw}(q)$ is

$$\int_0^1 \omega^{cw}(q)\, dq \;=\; \int_0^1 \omega\left(\frac{(1-c)q}{(1-c)q+c(1-q)}\right) \frac{(1-c)^2c^2}{((1-c)q+c(1-q))^3}\, dq\;.$$

The following variable transformation will be used repeatedly in what follows:

$$t \;=\; \frac{(1-c)q}{(1-c)q+c(1-q)}\;, \qquad q \;=\; \frac{ct}{ct+(1-c)(1-t)}\;, \tag{8.1}$$

$$\omega^{cw}(q)\, dq \;=\; [ct+(1-c)(1-t)]\, \omega(t)\, dt$$

Thus the normalizing constant is

$$\int_0^1 \omega^{cw}(q) dq \;=\; \int_0^1 \omega(t)\, [ct+(1-c)(1-t)]\, dt \;=\; c\mu+(1-c)(1-\mu)\;,$$

and the following is hence a density:

$$\frac{\omega^{cw}(q)}{c\mu+(1-c)(1-\mu)}\;. \tag{8.2}$$

Expectation: We start with the non-normalized weight function $\omega^{cw}(q)$. With the same variable transformation as above we have

$$\begin{aligned}\int_0^1 q\,\omega^{cw}(q)\,dq &= \int_0^1 \frac{ct}{ct+(1-c)(1-t)}(ct+(1-c)(1-t))\,\omega(t)\,dt \\ &= \int_0^1 c\,t\,\omega(t)\,dt \\ &= c\,\mu\end{aligned}$$

Normalizing according to (8.2), we obtain the desired result.

Variance inequalities: To derive the upper and lower bounds for the variance of the cost-weighted density using the variable transformation (8.1):

$$\begin{aligned}\sigma^{cw2} &= \frac{1}{c\mu+(1-c)(1-\mu)}\int_0^1 (q-\mu^{cw})^2\,\omega^{cw}(q)\,dq \\ = &\int_0^1 \left(\frac{ct}{ct+(1-c)(1-t)}-\frac{c\mu}{c\mu+(1-c)(1-\mu)}\right)^2 \frac{ct+(1-c)(1-t)}{c\mu+(1-c)(1-\mu)}\,\omega(t)\,dt \\ = &\int_0^1 \left(\frac{c(1-c)(t-\mu)}{(c\mu+(1-c)(1-\mu))(ct+(1-c)(1-t))}\right)^2 \frac{ct+(1-c)(1-t)}{c\mu+(1-c)(1-\mu)}\,\omega(t)\,dt\end{aligned}$$

Thus:

$$\sigma^{cw2} = \frac{c^2(1-c)^2}{(c\mu+(1-c)(1-\mu))^3}\int_0^1 \frac{(t-\mu)^2}{ct+(1-c)(1-t)}\,\omega(t)\,dt \tag{8.3}$$

Trivially, the integral (without the preceding factor) is upper bounded by

$$\frac{1}{\min(c,1-c)}\int_0^1 (t-\mu)^2\,\omega(t)\,dt$$

and lower bounded by

$$\frac{1}{\max(c,1-c)}\int_0^1 (t-\mu)^2\,\omega(t)\,dt$$

With $\sigma^2 = \int_0^1 (t-\mu)^2 \omega(t) dt$ we have:

$$\begin{aligned} \sigma^{cw2} &\leq \frac{c^2(1-c)^2}{(c\mu + (1-c)(1-\mu))^3} \frac{1}{\min(c, 1-c)} \sigma^2 \\ \sigma^{cw2} &\geq \frac{c^2(1-c)^2}{(c\mu + (1-c)(1-\mu))^3} \frac{1}{\max(c, 1-c)} \sigma^2 \end{aligned}$$

Noting $c(1-c)/\min(c, 1-c) = \max(c, 1-c)$ and its dual, the assertion follows.

Variance for the cost-weighted symmetric Beta density: A calculation similar to the one that lead to Equation (8.3) shows, using $\mathrm{V}[X] = \mathrm{E}[X^2] - [\mathrm{E}X]^2$:

$$\sigma^{cw2} \;=\; \frac{c^2}{c\mu + (1-c)(1-\mu)} \int_0^1 \frac{t^2}{ct + (1-c)(1-t)} \, \omega(t) \, dt \;-\; \mu^{cw2} \tag{8.4}$$

This simplifies somewhat with $\mu = 1/2$:

$$\sigma^{cw2} \;=\; 2\,c^2 \int_0^1 \frac{t^2}{ct + (1-c)(1-t)} \, \omega(t) \, dt \;-\; c^2 \tag{8.5}$$

Specializing ω to the symmetric Beta density with parameter α, we get:

$$\sigma^{cw2} \;=\; \frac{2\,c^2}{\mathrm{B}(\alpha, \alpha)} \int_0^1 \frac{t^{\alpha+1}(1-t)^{\alpha-1}}{ct + (1-c)(1-t)} \, dt \;-\; c^2 \tag{8.6}$$

This integral can be expressed in terms of the hypergeometric function ${}_2\mathrm{F}_1(a, b; c; z)$ using the following fact:

$${}_2\mathrm{F}_1(a, b\,; c\,; z) \;=\; \frac{1}{\mathrm{B}(b, c-b)} \int_0^1 \frac{t^{b-1}(1-t)^{c-b-1}}{(1-tz)^a} \, dt \;.$$

(See, for example, Weisstein 2005.) Thus with $a = 1$, $b = \alpha + 2$, $c = 2\alpha + 2$ and $z = (1-2c)/(1-c)$ we arrive at:

$$\sigma^{cw2} \;=\; c^2 \left(\frac{(\alpha+1)}{(1-c)(2\alpha+1)} \, {}_2\mathrm{F}_1 \left(1, \alpha+2, 2\alpha+2; \frac{1-2c}{1-c} \right) \;-\; 1 \right) \;. \tag{8.7}$$

Shape of the cost-weighted symmetric Beta: For the symmetric Beta family, $\omega^{cw}(q)$ is

$$\omega^{cw}(q) \ \sim \ \frac{q^{\alpha-1}(1-q)^{\alpha-1}}{((1-c)q+c(1-q))^{2\alpha+1}}$$

We derive the stationary point of the function as follows: Set $\omega^{cw'}(q) \ = \ 0$ and note that

$$\begin{aligned}\left(q^{\alpha-1}(1-q)^{\alpha-1}\right)' &= (\alpha-1)(1-2q)(q(1-q))^{\alpha-2},\\ \left(((1-c)q+c(1-q))^{2\alpha+1}\right)' &= (2\alpha+1)(1-2c)((1-c)q+c(1-q))^{2\alpha}\end{aligned}$$

Thus the stationarity condition becomes

$$\begin{aligned}&(\alpha-1)(1-2q)(q(1-q))^{\alpha-2}((1-c)q+c(1-q))^{2\alpha+1}\\ -\ &(2\alpha+1)(1-2c)(q(1-q))^{\alpha-1}((1-c)q+c(1-q))^{2\alpha} \ = \ 0\end{aligned}$$

which can be simplified to

$$3(1-2c)q^2-(\alpha-(6c-2))q+(\alpha-c)c \ = \ 0 \ .$$

Solving this equation we get the stationary points

$$\frac{-(\alpha-2(3c-1))\pm\sqrt{(\alpha-2(3c-1))^2+12(\alpha-1)c(2c-1)}}{6(2c-1)}$$

Rewrite this expression as

$$\begin{aligned}&\frac{-((\alpha-1)-3(2c-1))\pm\sqrt{(\alpha-1)^2+6(2c-1)^2(\alpha-1)+9(2c-1)^2}}{6(2c-1)}\\ &\quad=\frac{-((\alpha-1)-3(2c-1))\pm\sqrt{(\alpha-1)^2+3(2\alpha+1)(2c-1)^2}}{6(2c-1)}\end{aligned}$$

Notice for $\alpha > 1$, the weight function has a unique mode at

$$\frac{-(\alpha - 2(3c-1)) + \sqrt{(\alpha - 2(3c-1))^2 + 12(\alpha-1)c(2c-1)}}{6(2c-1)}$$

The other stationary point is outside the $[0, 1]$ interval.

For $\alpha < 1$, the weight function has a unique minimum at

$$\frac{-(\alpha - 2(3c-1)) - \sqrt{(\alpha - 2(3c-1))^2 + 12(\alpha-1)c(2c-1)}}{6(2c-1)}$$

The other stationary point is out of bound.

For $\alpha = 1$ and $c \neq 0.5$, the weight function has a unique mode and maximum at 0 or 1.