Classification with TreeNet

Letting Y be a binary response variable having possible values 1 and -1,

$$Y' = (Y+1)/2$$

(so that Y' is a Bernoulli r.v.), and

$$p(x) = P(Y'=1 \mid x) = P(Y=1 \mid x),$$

the likelihood for an observed y' assoc. w/ x is

$$[\,p(x)\,]^{y'}\,[\,1-p(x)\,]^{1-y'},$$

and the log-likelihood is

$$y' \log[\,p(x)] + (1-y')\log[\,1-p(x)].$$

We can elect to express $p(x)$ in terms of a "modeling function," $f(x)$, using

$$p(x) = 1/(1 + \exp(-2f(x))),$$

which insures that $p(x)$ will not be outside of $(0,1)$ no matter what $f(x)$ is and no matter how extreme x is. In terms of $f(x)$, the log-likelihood is

$$y' \log\left(\frac{1}{1 + \exp(-2f(x))}\right) + (1-y')\log\left(\frac{\exp(-2f(x))}{1 + \exp(-2f(x))}\right),$$

which, after a bit of careful work, can be shown to be equal to

$$-\log(1 + e^{-2yf(x)}).$$

Equivalent to maximizing the log-likelihood (and also the likelihood) is minimizing

$$\log(1 + e^{-2yf(x)}),$$

which is the *deviance* (aka *cross-entropy* (and one could also refer to it as the negative *Bernoulli* log-likelihood)). Given data, we could seek $f(x)$ to make

$$\sum_{i=1}^{n} \log(1 + e^{-y_i f(x_i)})$$

as small as possible. However, since $E(e^{-Yf(x)} | x)$ has the same minimizer as the neg. log-likelihood, we can instead seek $f(x)$ to minimize

$$\sum_{i=1}^{n} e^{-y_i f(x_i)}.$$

For the *exponential loss function* we have

$$L(y, f(x)) = \exp(-yf(x)).$$

FSAM w/ exponential loss is an iterative method to approximate the f'n, $f(x)$, which minimizes

$$\sum_{i=1}^{n} e^{-y_i f(x_i)}.$$

Since

$$f(x) = \frac{1}{2} \log\left(\frac{P(Y=1\,|\,x)}{P(Y=-1\,|\,x)} \right),$$

FSAM w/ exp. loss can be thought of as a way to fit a logistic regression model. (Note that we don't guess what the form of $f(x)$ is and estimate its parameters — we approximate $f(x)$ nonparametrically.) The approximation of $f(x)$ which results can be used to do classification, and this way of doing classification can be shown (with a bit of effort) to be equivalent to doing two-class classification using AdaBoost.M1. If a shrinkage factor is used with the FSAM procedure, we no longer have the equivalence with AdaBoost.M1, but the incorporation of a shrinkage factor can improve performance.

With *TreeNet*, one can fit this type of classifier by specifying that a *Regression* model be built using *Logistic likelihood* as the Regression Loss Criterion. (Note: All of the response values in the data need to be either 1 or -1.) If there are more than two classes possible for the response, then one can specify that a *Classification* model be built (and I think the algorithm given on p. 345 of HTF is used).

Note: The *TreeNet* manual gives contradictory information concerning the default prior probabilities. P. 43 specifies that the default is DATA (which I think is a reasonable choice), while p. 68 indicates that it is EQUAL (which may not be what you want to use). With DATA, the class proportions of the available data are assumed to be what one expects to encounter when classification is done using the classifier which is built.