Gradient Boosting for Decision Trees (on Scalation)

Problem Statement:

The problem we (Dr. Miller and I) decided to work on was *gradient boosting on decision trees* for classification on the Scalation framework. We started by looking at the loss function for the problem and found the equation on page 347 of the book Elements of Statistical Learning. The equation that fits our problem is termed as "Binomial Deviance" and works for 2 class classification problems. The extension of the same problem for K-class can be done by referring to the equation on page 349 of the same book which goes as follows:

$$\begin{aligned} L(y, p(x)) &= -\sum_{k=1}^{K} I(y = \mathcal{G}_k) \log p_k(x) \\ &= -\sum_{k=1}^{K} I(y = \mathcal{G}_k) f_k(x) + \log \left(\sum_{\ell=1}^{K} e^{f_\ell(x)} \right). \quad (10.22) \end{aligned}$$

[1] Zhu et al. (2005) generalize the exponential loss for K-class problems by:

Ex. 10.5 *Multiclass exponential loss* (Zhu et al., 2005). For a K-class classification problem, consider the coding $Y = (Y_1, \ldots, Y_K)^T$ with

$$Y_k = \begin{cases} 1, & \text{if } G = \mathcal{G}_k \\ -\frac{1}{K-1}, & \text{otherwise.} \end{cases} \quad (10.55)$$

Let $f = (f_1, \ldots, f_K)^T$ with $\sum_{k=1}^{K} f_k = 0$, and define

$$L(Y, f) = \exp\left(-\frac{1}{K} Y^T f\right). \quad (10.56)$$

[3] **Code Execution**:

1. Download one of the following two files from the link http://cobweb.cs.uga.edu/~jam/scalation_1.6/README.html
   scalation_1.6.tar.gz OR scalation_1.6.zip

2. Untar or unzip the file using either of the following commands
   tar xvfz scalation_1.6.tar.gz OR unzip scalation_1.6.zip

3. To build all the modules, change into the ScalaTion base directory and run the build_all.sh shell script
   $ cd scalation_1.6
   $ ./build_all.sh

4. To compile code or run apps, change into one of the module directories, enter sbt and type compile, runMain or exit

```
$ cd scalation_modeling
$ sbt
> compile
> runMain scalation.analytics.classifier.DecisionTreeC45_GBTest
```

[1] **Loss function equations:**

The equation for Binomial Deviance is as follows:
log(1 + exp(–2yf))

* The above equation contains the constant **2** which we thought did not produce identical results the graph in the book is a typo and hence reference to another equation from the same book given on page 427 which drops the 2 from the equation and states the equation as
log(1 + exp(–yf))

TABLE 12.1. *The population minimizers for the different loss functions in Figure 12.4. Logistic regression uses the binomial log-likelihood or deviance. Linear discriminant analysis (Exercise 4.2) uses squared-error loss. The SVM hinge loss estimates the mode of the posterior class probabilities, whereas the others estimate a linear transformation of these probabilities.*

Loss Function	$L[y, f(x)]$	Minimizing Function
Binomial Deviance	$\log[1 + e^{-yf(x)}]$	$f(x) = \log \frac{\Pr(Y = +1\|x)}{\Pr(Y = -1\|x)}$
SVM Hinge Loss	$[1 - yf(x)]_+$	$f(x) = \text{sign}[\Pr(Y = +1\|x) - \frac{1}{2}]$
Squared Error	$[y - f(x)]^2 = [1 - yf(x)]^2$	$f(x) = 2\Pr(Y = +1\|x) - 1$
"Huberised" Square Hinge Loss	$-4yf(x)$, $yf(x) < -1$ $[1 - yf(x)]_+^2$ otherwise	$f(x) = 2\Pr(Y = +1\|x) - 1$

The equation for Exponential is as follows:
exp(–yf)

The equation for Squared Error is as follows:
$(y - f)^2$

The equation for Support Vector is as follows:
(1 – yf)

The equation for Misclassification is as follows:
I(sign(f)/= y)

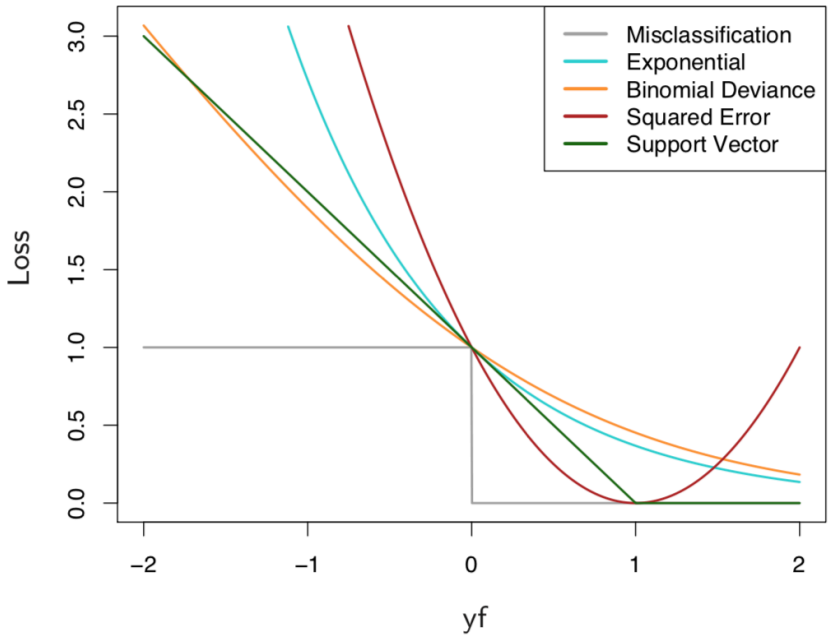
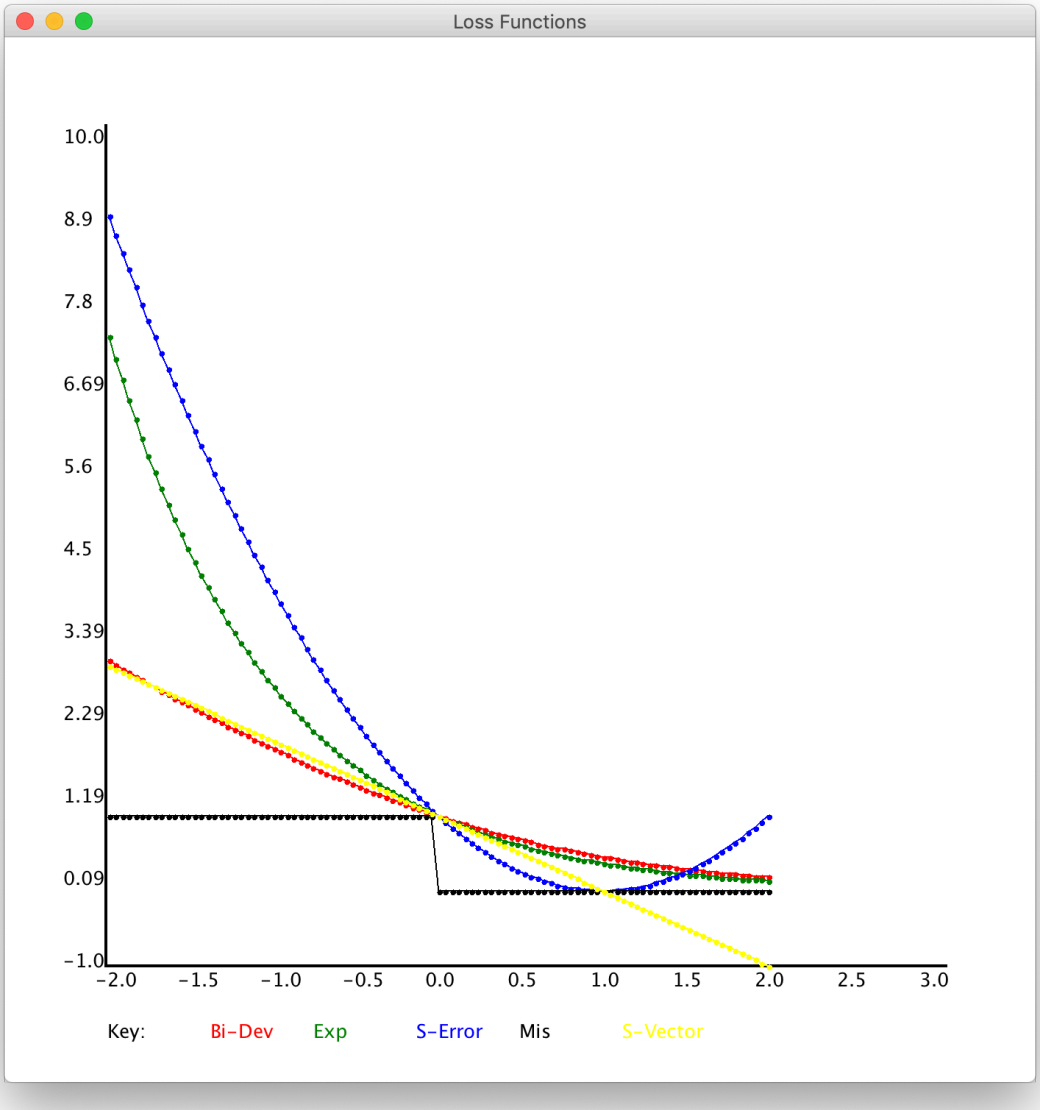


FIGURE 10.4. *Loss functions for two-class classification. The response is* $y = \pm 1$*; the prediction is* f*, with class prediction* $\text{sign}(f)$*. The losses are misclassification:* $I(\text{sign}(f) \neq y)$*; exponential:* $\exp(-yf)$*; binomial deviance:* $\log(1 + \exp(-2yf))$*; squared error:* $(y - f)^2$*; and support vector:* $(1 - yf)_+$ *(see Section 12.3). Each function has been scaled so that it passes through the point* $(0, 1)$*.*

Results:

The verification of the loss function based on the synthetic data produced gave an identical graph as that present in the book. The comparison of the synthetic data points at -1 and 1 and that of the actual Vs. predicted is shown below:

Here's the graph produced using the synthetic data:



References:

[1]. https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf
[2]. https://www.kaggle.com/grroverpr/gradient-boosting-simplified/
[3]. http://cobweb.cs.uga.edu/~jam/scalation_1.6/README.html

Conclusion:

The results produced out of the sampled data are closely accurate and verified. And the function should work with 2 class classification problems where the number of instances are high enough to produce sufficient data points.

Future Scope:

Extending the problem to K-class using multiclass exponential loss for K-class classification problem.