Statistical Learning - Milan, Fall2019

Homework problem 9

AdaBoost and ϵ-Adaboost

This problem refers to the AdaBoost algorithm (Freund and Schapire, 1997), which is used for binary classification with labels $y_i \in \{\pm 1\}$. Adaboost initializes $\hat{f}_0 = 0, w_i \equiv 1$, then for $t = 1...T$ updates:

1. Fit a classification tree with response y and weights w on the observations, getting tree h_t
2. Denote by Err_t the (weighted) misclassification error of h_t
3. Set $\alpha_t = 0.5 \log\left((1 - Err_t)/Err_t\right)$
4. Update weights: $w_i \leftarrow w_i \exp\left(-\alpha_t(y_i h_t(x_i))\right)$

The model after step t is $f_t(x) = \sum_{u=1}^{t} \alpha_u h_u(x)$, the final model is f_T, and classification if according to the sign of $f_T(x)$.

As we said in class, from the coordinate descent perspective of boosting, AdaBoost can be viewed as gradient boosting with loss function $L(y, \hat{y}) = \exp(-y\hat{y})$ and line-search steps, explicitly:

- Fitting a classification tree is minimizing $\sum_i w_i y_i h_k(x_i) = \langle wy, h_k(X) \rangle$ (so $w_i y_i$ is the actual gradient)
- The calculated α_t is the solution to the line search problem: $\alpha_t = \arg\min_\alpha \sum_i L\left(y_i, (f_{T-1} + \alpha h_t)(x_i)\right)$
- The updated w_i is indeed (proportional to) the absolute value of the gradient:

$$w_i \propto \left| \frac{dL\left(y_i, l\right)}{dl}|_{l=\hat{f}_t(x_i)} \right|$$

1. Choose one of the three properties above and prove explicitly that it holds (for example, if you choose the first one, show how fitting h_k classification tree minimizing weighted miclassification error is equivalent to choosing a coordinate descent direction in the exponential loss function).
2. The code in www.tau.ac.il/∼saharon/StatsLearn2019-Milan2/AdaBoost.r implements AdaBoost on the competition data for the problem of whether $y > 3$ or $y \leq 3$. Read the code carefully to make sure you understand the details. Note especially the parameters "method" and "weights" that rpart takes.
   (a) With 8000-2000 training-test division as in the code, run the algorithm for 1000 iterations and draw a plot of training and test misclassification as a function of iterations. Explain its form.
   (b) Change the algorithm from line-search boosting to ϵ-boosting with $\epsilon = 0.01$, not changing the other parameters. Run this version for 1000 iterations and draw the same plot. Discuss the results and compare to the line-search version.
   (c) Now change the loss function from the exponential loss to squared error loss, and the approach to the regular gradient boosting with regression tree. Explain briefly in writing what you did, and run with the same parameters ($\epsilon = 0.01, T = 1000$). Classify according to sign of $\hat{y}$ and repeat the same analysis again. Discuss the relative results.