Statistical Machine Learning GR5241
Spring 2021

Homework 4

Due: Monday, April 11th, 2021

Homework submission: Please submit your homework electronically through Canvas by 11:59pm on the due date. You need to submit both the pdf file and your code (either in R or Python).

Problem 1 (Boosting, 50 points)

The objective of this problem is to implement the AdaBoost algorithm. We will test the algorithm on handwritten digits from the USPS data set.

AdaBoost: Assume we are given a training sample $(\mathbf{x}^{(i)}, y_i), i = 1, ..., n$, where $\mathbf{x}^{(i)}$ are data values in $\mathbb{R}^d$ and $y_i \in \{-1, +1\}$ are class labels. Along with the training data, we provide the algorithm with a training routine for some classifier c (the "weak learner"). Here is the AdaBoost algorithm for the two-class problem:

1. Initialize weights: $w_i = \frac{1}{n}$
2. for $b = 1, ..., B$
   (a) Train a weak learner c_b on the weighted training data. (**See the note below**)
   Note: step 2(a) can be completed using two different methods: (1) use the weight vector directly in the training of the weak learner, or (2) use the weight vector to sample data points with replacement from the original data, then train the weak learner on the sampled data. Either way will guarantee full credit.
   (b) Compute error: $\epsilon_b := \frac{\sum_{i=1}^n w_i \mathbb{I}\{y_i \neq c_b(\mathbf{x}^{(i)})\}}{\sum_{i=1}^n w_i}$
   (c) Compute voting weights: $\alpha_b = \log\left(\frac{1-\epsilon_b}{\epsilon_b}\right)$ or $\alpha_b = \frac{1}{2}\log\left(\frac{1-\epsilon_b}{\epsilon_b}\right)$
   (d) Recompute weights: $w_i = w_i \exp\left(\alpha_b \mathbb{I}\{y_i \neq c_b(\mathbf{x}^{(i)})\}\right)$
3. Return classifier $\hat{c}_B(\mathbf{x}^{(i)}) = \underline{\operatorname{sgn}\left(\sum_{b=1}^B \alpha_b c_b(\mathbf{x}^{(i)})\right)}$

Decision stumps: Recall that a stump classifier c is defined by

$$c(\mathbf{x}|j, \theta, m) := \begin{cases} +m & x_j > \theta \\ -m & \text{otherwise.} \end{cases} \tag{1}$$

Since the stump ignores all entries of $\mathbf{x}$ except x_j, it is equivalent to a linear classifier defined by an affine hyperplane. The plane is orthogonal to the jth axis, with which it intersects at $x_j = \theta$. The orientation of the hyperplane is determined by $m \in \{-1, +1\}$. We will employ stumps as weak learners in our boosting algorithm. To train stumps on weighted data, use the learning rule

$$(j^*, \theta^*) := \arg\min_{j,\theta} \frac{\sum_{i=1}^n w_i \mathbb{I}\{y_i \neq c(\mathbf{x}^{(i)}|j, \theta, m)\}}{\sum_{i=1}^n w_i} . \tag{2}$$

In the implementation of your training routine, first determine an optimal parameter θ_j^* for each dimension $j = 1, ..., d$, and then select the j^* for which the cost term in (2) is minimal.

Note: If the data is sampled using weights w_i, then the decision stump can be trained using a loss function other than *weighted 0-1 loss.* Using other loss functions to train the weak learner is technically not correct; however, the results are similar.

Homework problems:

1. Implement the AdaBoost algorithm in R.
2. Run your algorithm on the USPS data (use the training and test data for the 3s and 8s) and evaluate your results using cross validation.

   More precisely: Your AdaBoost algorithm returns a classifier that is a combination of B weak learners. Since it is an incremental algorithm, we can evaluate the AdaBoost at every iteration b by considering the sum up to the b-th weak learner. At each iteration, perform 5-fold cross validation to estimate the training and test error of the current classifier (that is, the errors measured on the cross validation training and test sets, respectively).
3. Plot the training error and the test error as a function of b.

Submission. Please make sure your solution contains the following:

- Your implementation of `AdaBoost`.
- Plots of your results (training error and cross-validated test error).

Problem 2 (Basic Theory Related to the Lasso [30 points])

2.i Consider the **univariate lasso** objective function with no bias:

$$Q(\beta) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - x_i\beta)^2 + \lambda|\beta|$$

Also suppose $\mathbf{x}$ is *scaled* using the formula

$$x_i := \frac{x_i}{\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}}, \quad i = 1, 2, \ldots, n.$$

Derive a closed form expression for the lasso solution, i.e., show $Q(\beta)$ is minimized at

$$\hat{\beta} = \begin{cases} \frac{1}{n}\sum_i x_i y_i - \lambda & \text{if} \quad \frac{1}{n}\sum_i x_i y_i > \lambda \\ 0 & \text{if} \quad -\lambda \le \frac{1}{n}\sum_i x_i y_i \le \lambda \\ \frac{1}{n}\sum_i x_i y_i + \lambda & \text{if} \quad \frac{1}{n}\sum_i x_i y_i < -\lambda \end{cases}$$

2.ii Consider the **multivariate lasso** objective function with no bias:

$$Q(\boldsymbol{\beta}) = Q(\beta_1, \beta_2, \ldots, \beta_p) = \frac{1}{2n}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

Also suppose that the j^{th} feature $\mathbf{x}_j$ is *scaled* using the formula

$$x_{ij} := \frac{x_{ij}}{\frac{1}{n}\sum_{i=1}^{n} x_{ij}^2}, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, p.$$

Solve the expression $\frac{\partial Q}{\partial \beta_j} = 0$ for β_j. Your final answer should be

$$\beta_j = S_\lambda\left(\frac{1}{n}\mathbf{x}_j^T \mathbf{r}_i^{(j)}\right),$$

where $\mathbf{x}_j$ is the j^{th} feature, $\mathbf{r}^{(j)}$ is the *partial residual*, and S_λ is the *soft thresholding operator*. See the lecture slides for further details.

Problem 3 (Regression Trees, ISL 8.4 [20 points])

3.i Sketch the tree corresponding to the partition of the predictor space illustrated in the left-hand panel of Figure 1. The numbers inside the boxes indicate the mean of Y within each region.

3.ii Create a diagram similar to the left-hand panel of Figure 1, using the tree illustrated in the right-hand panel of the same figure. You should divide up the predictor space into the correct regions, and indicate the mean for each region.

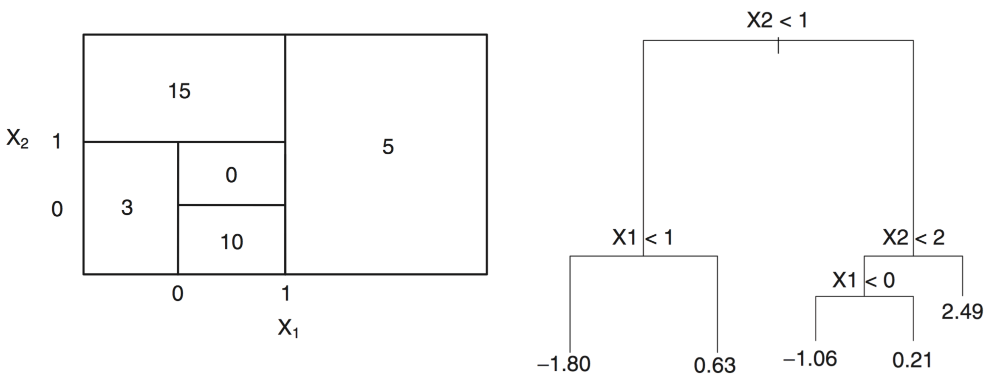

Figure 1: Left: *A partition of the predictor space corresponding to part a*. Right: *A tree corresponding to part b.*