Boosting

UBCO MDS — DATA 571

Boosting

- Now to a completely different option (as compared to bagging/RF) for improving the performance of a simple model.
- Boosting can be summarized as recursively fitting a model to residuals, weighted specifically to increase focus on larger residuals.
- It requires several tuning parameters.

Boosting Tuning Parameters

- B — the number of fits (trees for todays example)
- λ — learning rate for algorithm (usually a small value, 0.01 or 0.001 are common)
- d — complexity/interaction depth parameter (for boosting trees, this is the the number of conditions...so $d + 1$ is the number of terminal nodes)

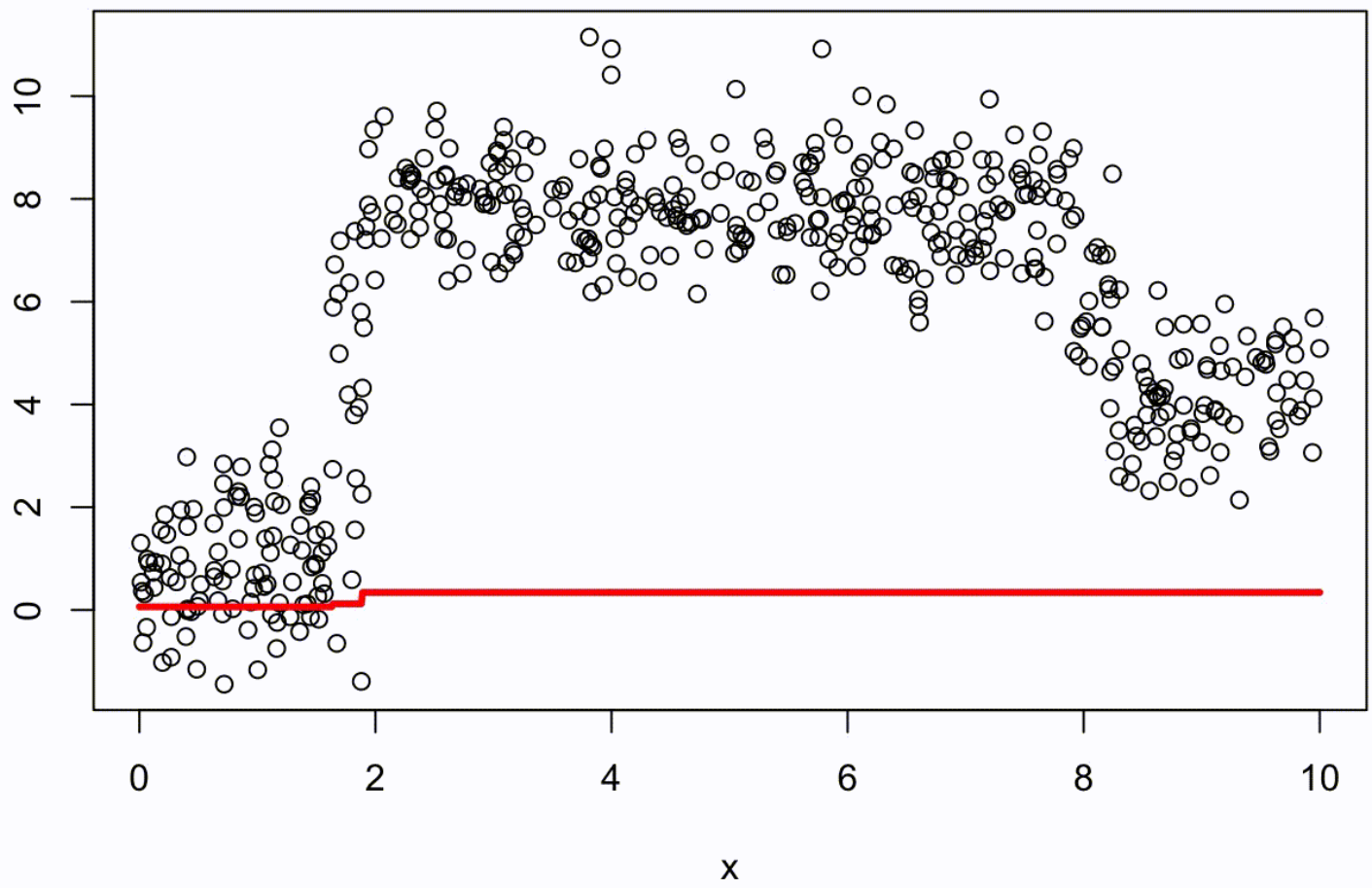
Boosting Regression Trees

1. Start with the full data set, call it (y, x), set $r_i = y_i$, index $b = 1$, and $\hat{f}(x) = 0$
2. Fit a tree $\hat{f}_b(x)$ with d splits to training data x with response r
3. Update $\hat{f}(x) = \hat{f}(x) + \lambda \hat{f}_b(x)$
4. Update response $r_i = r_i - \lambda \hat{f}_b(x_i)$
5. Set $b = b + 1$, if $b \leq B$, return to step 2, else report model $\hat{f}(x)$

Boosting Comments

- Boosting is philosophically a very different idea as compared to bagging/RF, or any of the standard models (lm, kNN) you've seen thus far.
- It is often described as 'learning slowly'
- Very simple models are often preferable. Specifically for trees, small depths are utilized (even $d = 1$, or 'stumps').

Boosting model-fitting example



Boosting Results

```
> table(wine$Class, cvboost)
   cvboost
     1  2  3
  1 58  1  0
  2  1 69  1
  3  0  0 48
```

- Cross validated error rate of approx 1.6% (equivalent to randomForests) on wine classification

- Cross validated RSS of 101.6 for the beer price regression (worst of the options we've tried, highly overfits as training RSS is 4.0!)

Boosting Additional Comments

- Two important things to note...
- 1.) Boosted models are incredibly powerful — which is certainly not clear from my two brief examples.
- Breiman (of trees fame) once declared boosted trees as the 'best off-the-shelf classifier in the world'.
- Properly tuned neural nets (we'll scratch the surface here soon) or gradient boosted algorithms (more generalized boosting) almost universally win any pure predictive modelling competition (such as Kaggle).
- ...which brings us back around to tuning...

Boosting Additional Comments

- 2.) The number of trees, learning rate, and tree complexity/depth are interlinked. How do we tune these?
- The easiest answer, is through cross-validation. But a complete grid-search becomes problematic with three parameters, so...
- If learning rate $\downarrow$, then number of trees $\uparrow$ and/or tree complexity $\uparrow$.
- Elith, Leathwick, & Hastie (2008) suggest approximate inverse relationship between tree complexity and learning rate. AKA, if you double complexity, halve the learning rate and keep the number of trees constant.
- They also recommend at least 1000 trees when dealing with small sample sizes ($n = 250$ in their example).


THE UNIVERSITY OF BRITISH COLUMBIA