Lecture 17. Boosting¶

CS 109A/AC 209A/STAT 121A Data Science:

Harvard University

Fall 2016

Instructors: P. Protopapas, K. Rader, W. Pan

Announcements

Pavlos office hours: Wed 4-5 will be covered by Hari.

HW6 grading will not be out until we are sure of the consistency of the grades.

Final homework is HW8

No quiz

Outline

Boosting

Miscellaneous other issues

Examples

Outline

Boosting

Miscellaneous other issues

Examples

Boosting

Boosting is a general approach that can be applied to many statistical learning methods for regression or classification.

Bagging: Generate multiple trees from bootstrapped data and average the trees.

Recall bagging results in i.d. trees and not i.i.d.

RF produces i.i.d (or more independent) trees by randomly selecting a subset of predictors at each step

Boosting

"Boosting is one of the most powerful learning ideas introduced in the last twenty years."

Hastie-Tibshirani-Friedman,The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer (2009)

Boosting

Boosting works very differently.

1. Boosting does not involve bootstrap sampling
2. Trees are grown sequentially: each tree is grown using information from previously grown trees

3. Like bagging, boosting involves combining a large number of decision trees, $f^1, \ldots, f^B$

Sequential fitting (Regression)

Given the current model,

- We fit a decision tree to the **residuals** from the model. Response variable now is the residuals and not *Y*
- We then add this new decision tree into the fitted function in order to update the residuals
- The learning rate has to be controlled

Boosting for regression

1. Set *f(x)=0* and $r_i = y_i$ for all *i* in the training set.

2. For *b=1,2,…,B*, repeat:

   a. Fit a tree with *d* splits(+1 terminal nodes) to the training data (X, r).

   b. Update the tree by adding in a shrunken version of the new tree:

   $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x)$$

   c. Update the residuals,

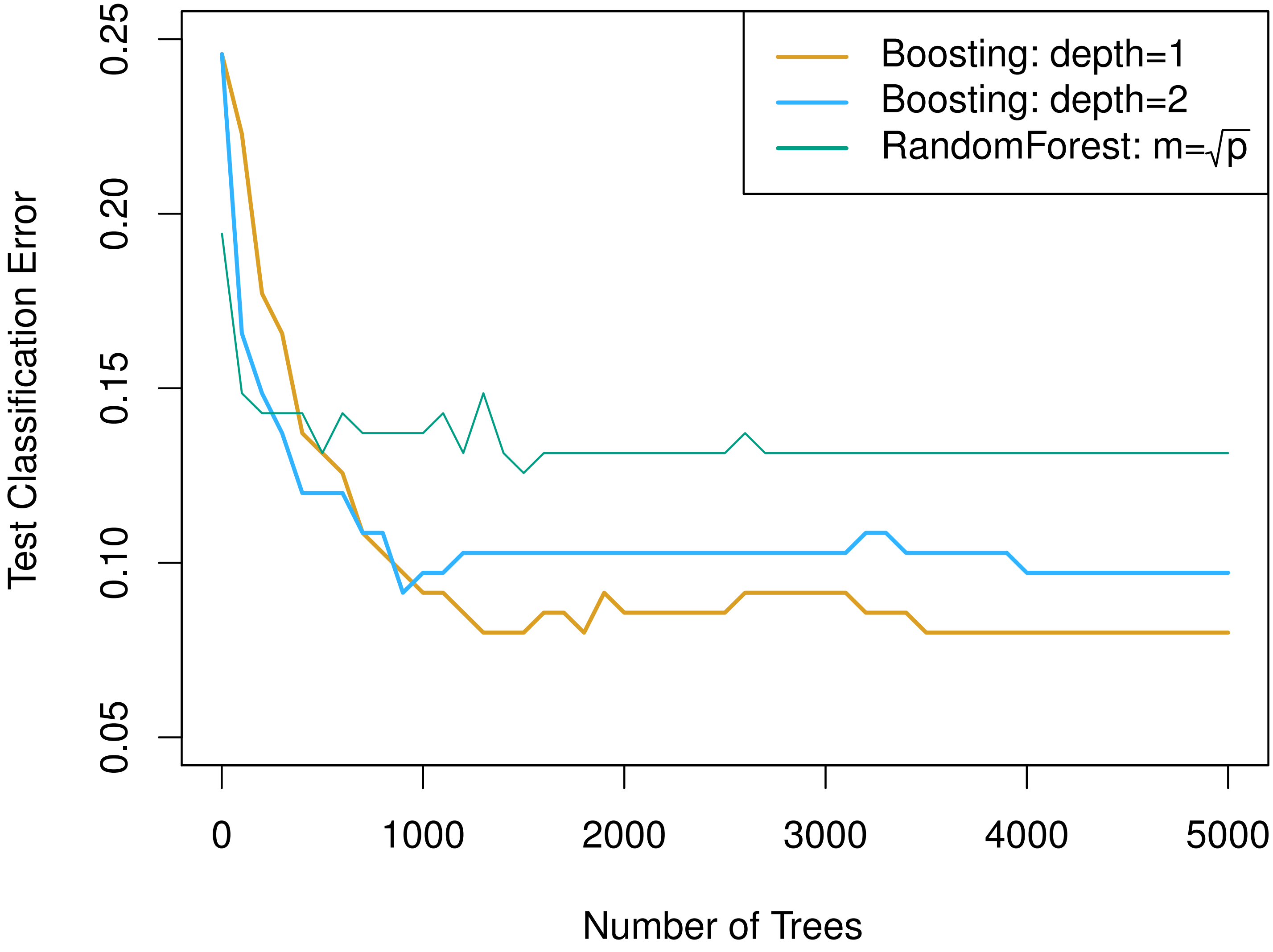
   $$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i)$$

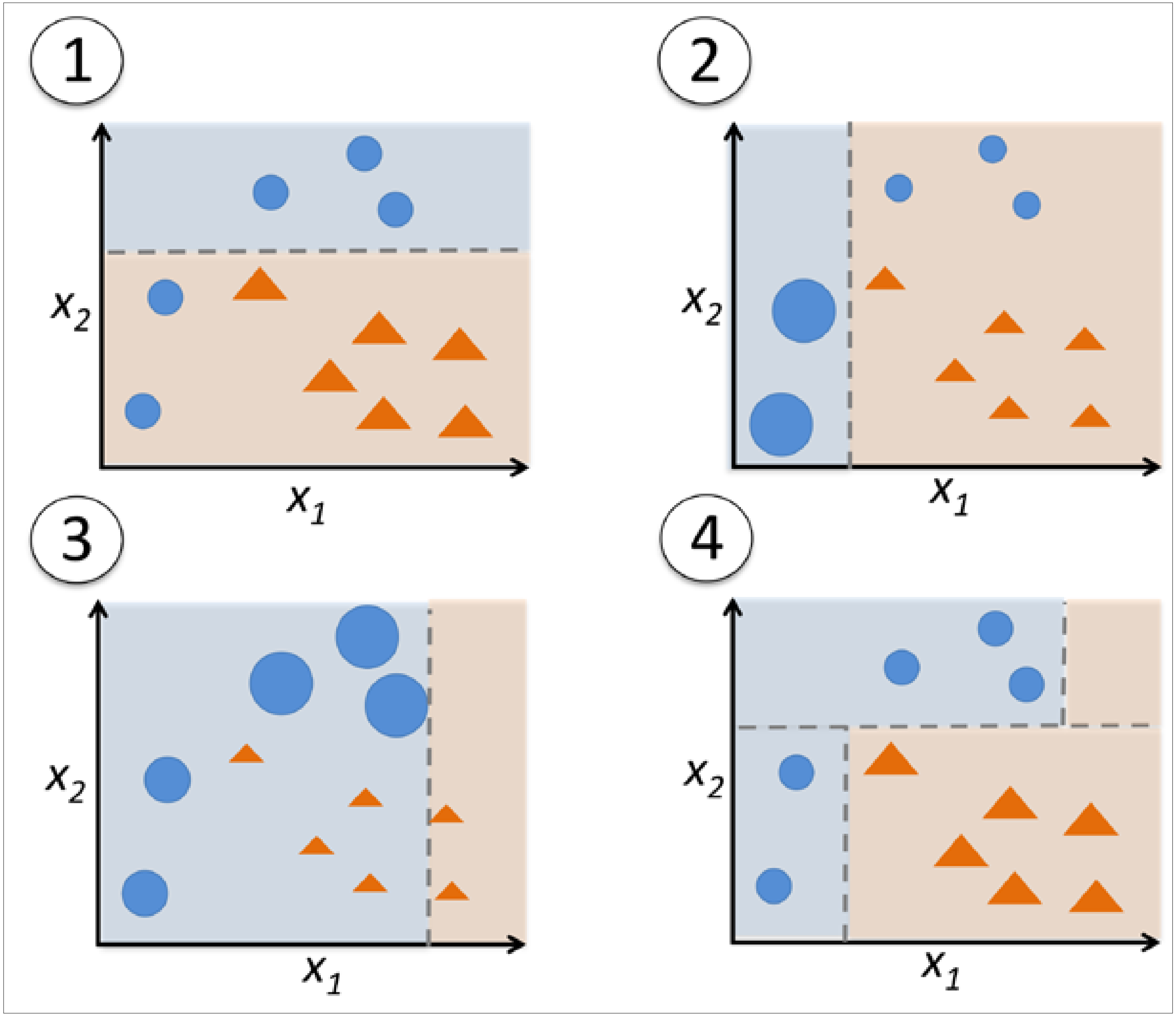
3. Output the boosted model,

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$$

Boosting tuning parameters

- The number of trees B. RF and Bagging do not overfit as B increases. Boosting can overfit! **Cross Validation**
- The shrinkage parameter λ, a small positive number. Typical values are 0.01 or 0.001 but it depends on the problem. λ only controls the learning rate
- The number d of splits in each tree, which controls the complexity of the boosted ensemble. Stumpy trees, $d = 1$ works well.


Boosting: depth=1
Boosting: depth=2
RandomForest: m=$\sqrt{p}$
Test Classification Error
0.05
0.10
0.15
0.20
0.25
0
1000
2000
3000
4000
5000
Number of Trees



Raschka, Python Machine Learning

Boosting for classification

Challenge question for HW7

Different flavors

- ID3, or alternative Dichotomizer, was the first of three Decision Tree implementations developed by Ross Quinlan (Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81-106.) Only categorical predictors and no pruning.
- C4.5, Quinlan's next iteration. The new features (versus ID3) are: (i) accepts both continuous and discrete features; (ii) handles incomplete data points; (iii) solves over-fitting problem by (very clever) bottom-up technique usually known as "pruning"; and (iv) different weights can be applied the features that comprise the training data.

Used in orange http://orange.biolab.si/

Different flavors

- C5.0, The most significant feature unique to C5.0 is a scheme for deriving rule sets. After a tree is grown, the splitting rules that define the terminal nodes can sometimes be simplified: that is, one or more condition can be dropped without changing the subset of observations that fall in the node.

- CART or Classification And Regression Trees is often used as a generic acronym for the term Decision Tree, though it apparently has a more specific meaning. In sum, the CART implementation is very similar to C4.5. **Used in sklearn**

Missing data

- What if we miss predictor values?
  - Remove those examples => depletion of the training set
  - Impute the values either with mean, knn, from the marginal or joint distributions
- Trees have a nicer way of doing this
  - Categorical, create a “missing” category
  - Surrogate predictors

Missing data

- A surrogate is a substitute for the primary splitter.
- The ideal surrogate splits the data in exactly the same way as the primary split, in other words, we are looking for something else in the data that can do the same work that the primary splitter accomplished.

- First, the tree is split in order to find the primary splitter. Then, surrogates are searched for, even when there is no missing data

Missing data

- The primary splitter may never have been missing in the training data. However, it may be missing on future data.
- When it is missing, then the surrogates will be able to take over and take on the work that the primary splitter accomplished during the initial building of the tree

Further reading

- Pattern Recognition and Machine Learning, Christopher M. Bishop
- The Elements of Statistical Learning

  Trevor Hastie, Robert Tibshirani, Jerome Friedman

  http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf

  Gradient Boosting: https://en.wikipedia.org/wiki/Gradient_boosting)

  Boosting: https://en.wikipedia.org/wiki/Boosting_(machine_learning))

  Ensemble Learning(https://en.wikipedia.org/wiki/Ensemble_learning)

Random best practices

Cross Validation

🟩	🟨	🟥
training data	validation data	test data

- Training data: train classifier
- Validation data: estimate hyper parameters
- Test data: estimate performance

- Be mindful of validation and test set, validation set might refer to test set in some papers.