http://poloclub.gatech.edu/cse6242

CSE6242 / CX4242: Data & Visual Analytics

Ensemble Methods

(Model Combination)

Duen Horng (Polo) Chau
Associate Professor
Associate Director, MS Analytics
Machine Learning Area Leader, College of Computing
Georgia Tech

Partly based on materials by
Professors Guy Lebanon, Jeffrey Heer, John Stasko, Christos Faloutsos, Parishit Ram (GT PhD alum; IBM), Alex Gray

Numerous Possible Classifiers!

Classifier	Training time	Cross validation	Testing time	Accuracy
kNN classifier	None	Can be slow	Slow	??
Decision trees	Slow	Very slow	Very fast	??
Naive Bayes classifier	Fast	None	Fast	??
…	…	…	…	…

Which Classifier/Model to Choose?

Possible strategies:

- Go from simplest model to more complex model until you obtain desired accuracy
- Discover a new model if the existing ones do not work for you
- Combine all (simple) models

Common Strategy: Bagging

(**B**ootstrap **Agg**regat**ing**)

Originally designed for combining multiple models, to improve classification "stability" [Leo Breiman, 94]

Uses random training datasets
(sampled from one dataset)

http://statistics.about.com/od/Applications/a/What-Is-Bootstrapping.htm

Common Strategy: Bagging

(**B**ootstrap **Agg**regat**ing**)

Consider the data set $S = \{(x_i, y_i)\}_{i=1,..,n}$

- Pick a sample S^* with replacement of size n
  (S^* called a "bootstrap sample")
- Train on S^* to get a classifier f^*
- Repeat above steps B times to get $f_1, f_2,...,f_B$
- Final classifier $f(x) = \text{majority}\{f_b(x)\}_{j=1,...,B}$

http://statistics.about.com/od/Applications/a/What-Is-Bootstrapping.htm

Bagging decision trees

Consider the data set S

- Pick a sample S^* with replacement of size n
- Grow a decision tree T_b
- Repeat B times to get $T_1,\ldots,T_B$
- The final classifier will be

$$f(x) = \text{majority}\{f_{T_b}(x)\}_{b=1,\ldots,B}$$

Random Forests

Almost identical to bagging decision trees, except we introduce some randomness:

- Randomly pick *m* of the *d* available attributes, at every split when growing the tree (i.e., d - m attributes ignored)

Bagged **random** decision trees

= **Random forests**

Explicit CV not necessary

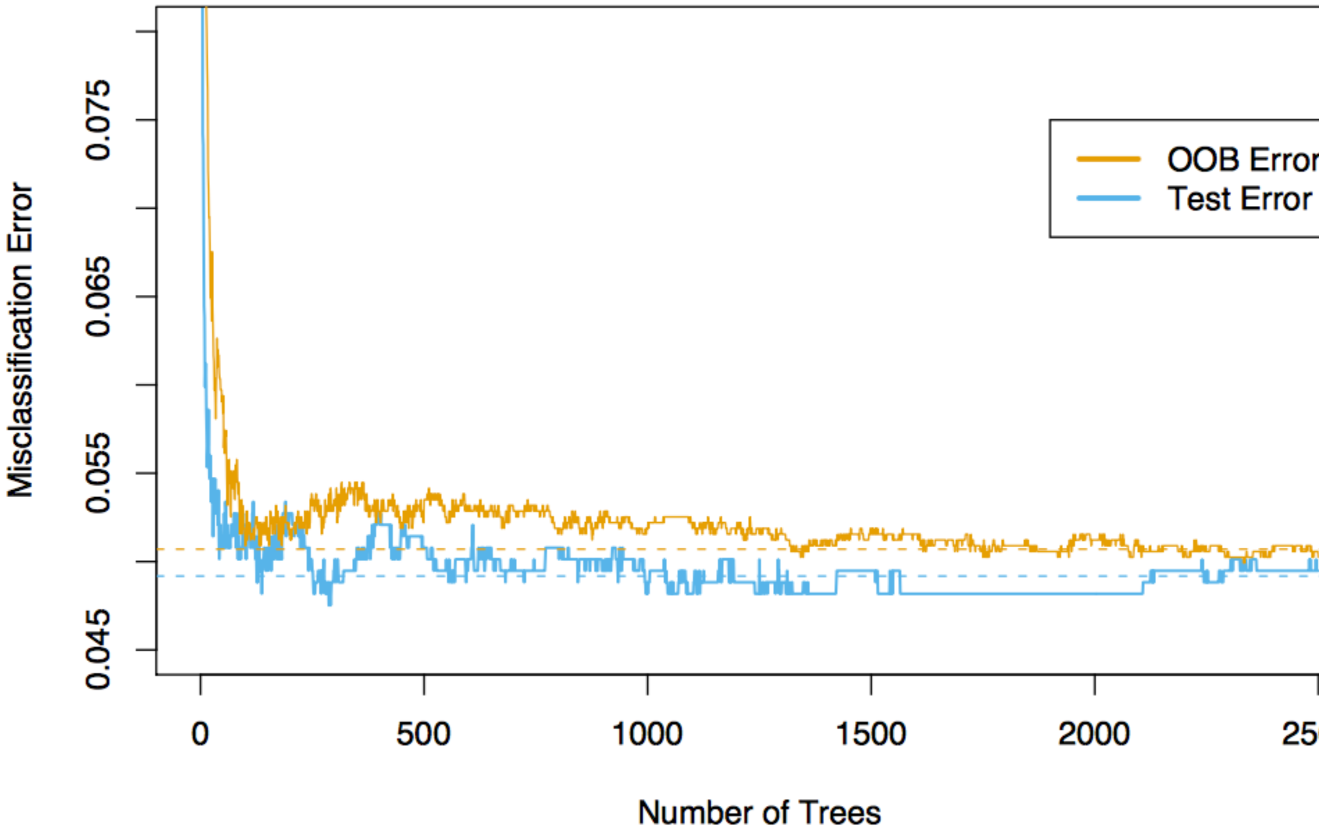
- Unbiased test error can be estimated using out-of-bag data points (OOB error estimate)
- You can still do CV explicitly, but that's not necessary, since research shows that OOB estimate is as accurate

Section 15.3.1 of http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf
https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#ooberr
http://stackoverflow.com/questions/18541923/what-is-out-of-bag-error-in-random-forests

Important points about random forests

Algorithm parameters

- Usual values for *m:* $\sqrt{d}, 1, 10$
- Usual value for *B*: keep adding trees until training error stabilizes



Important points about random forests

Algorithm (hyper) parameters

- Size/#nodes of each tree
  - as in when building a decision tree
- May randomly pick an attribute, and may even randomly pick the split point!
  - Significantly simplifies implementation and increases training speed
  - PERT - Perfect Random Tree Ensembles
    http://www.interfacesymposia.org/I01/I2001Proceedings/ACutler/ACutler.pdf
  - Extremely randomized trees
    http://orbi.ulg.be/bitstream/2268/9357/1/geurts-mlj-advance.pdf

Advantages

- Efficient and simple training
- Allows you to work with simple classifiers
- Random-forests generally useful and accurate in practice (one of the best classifiers)
  - The other is *gradient-boosted tree* *http://fastml.com/what-is-better-gradient-boosted-trees-or-random-forest/*
- Embarrassingly parallelizable

Final words

Reading material

- Bagging: ESL Chapter 8.7
- Random forests: ESL Chapter 15
  http://www-stat.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf