Ensembles & Random Forest

CS 534: Machine Learning

Slides adapted from Lee Cooper, Ryan Tibshirani, Yubin Park, and Joydeep Ghosh

Review: Trees & Boosting

Review: Trees

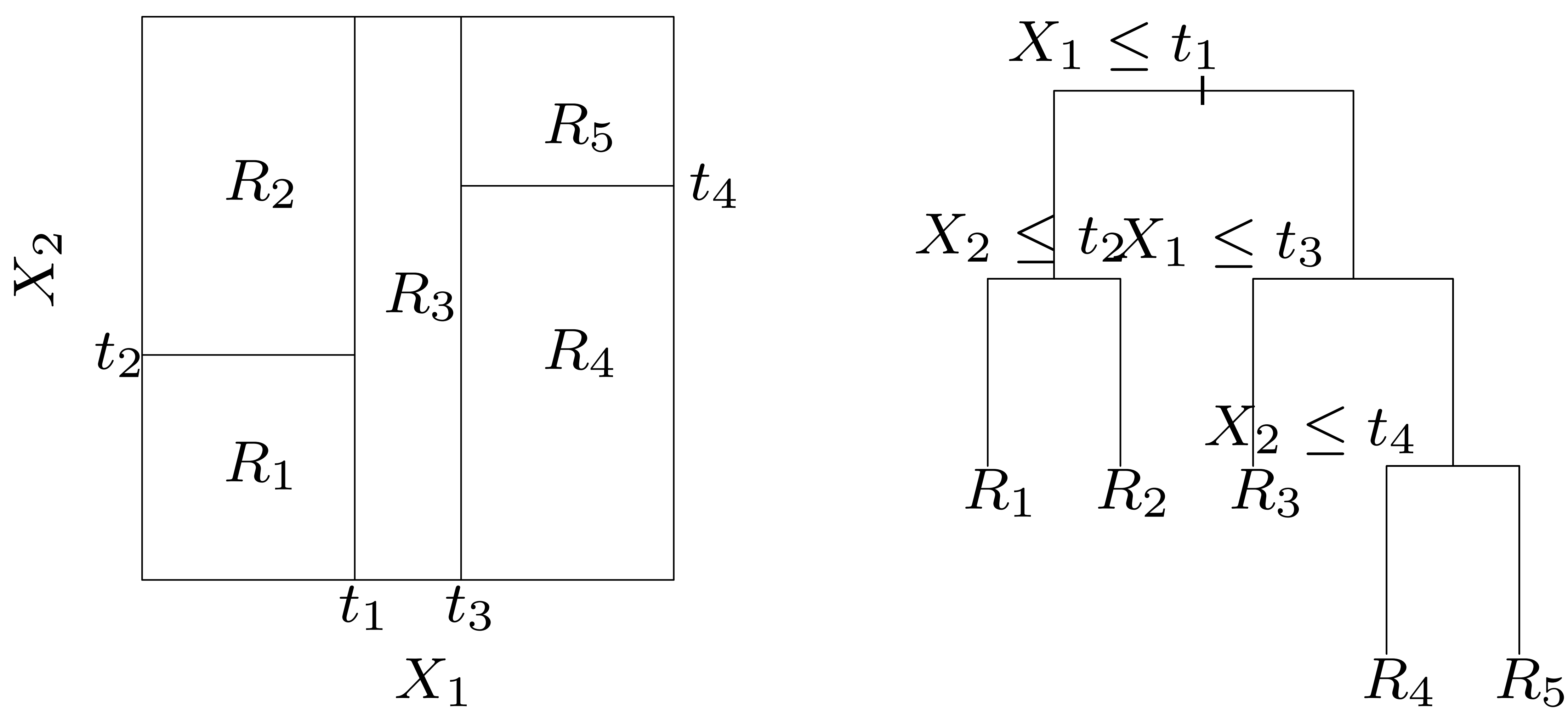


Figure 9.2 (Hastie et al.)

Review: Trees

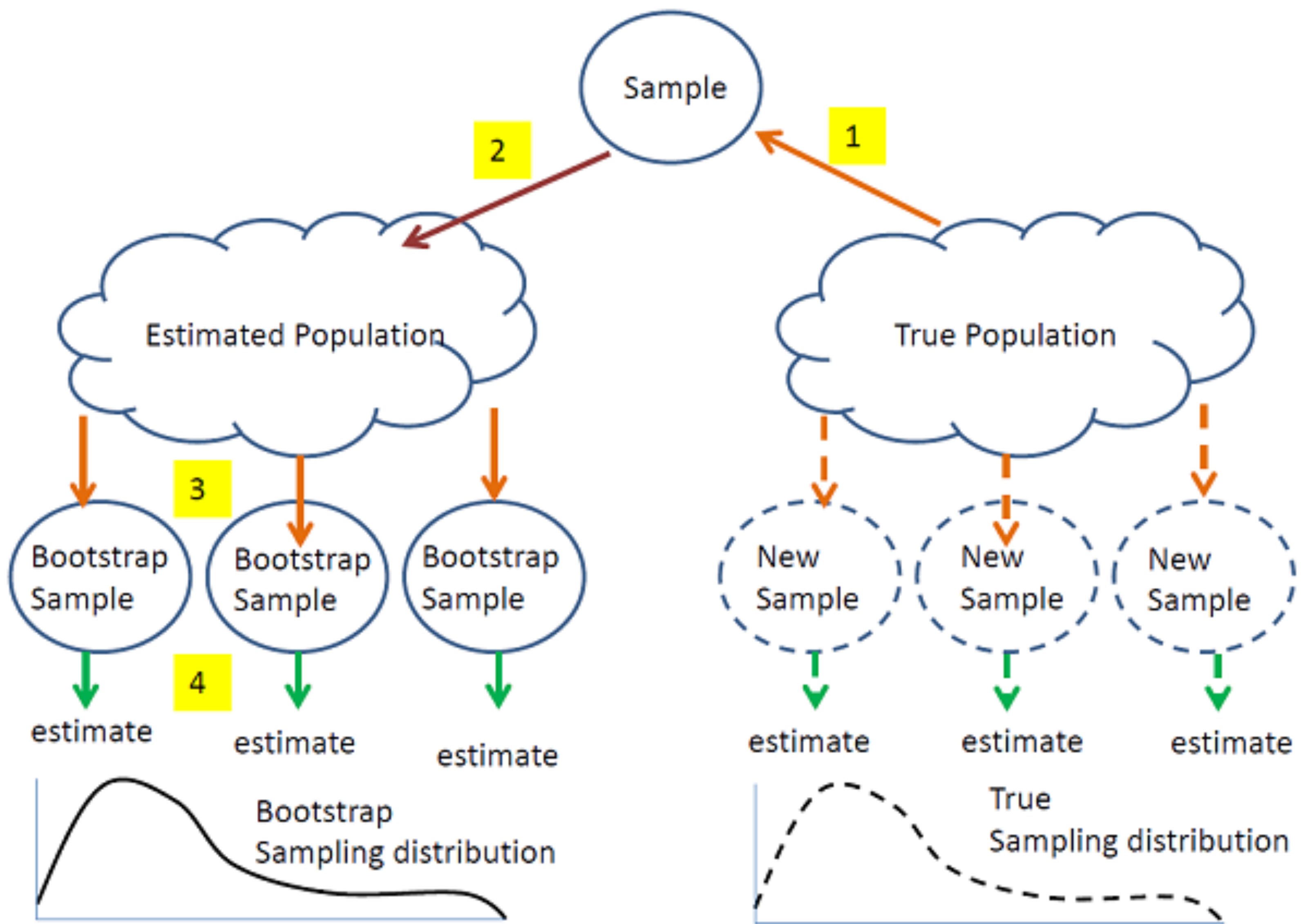
- Divide up the feature space into regions
- Greedy split the features based on some criterion
- Grow a large tree and prune back using cross-validation
- Each leaf then predicts class based on majority class and probability is the proportion of points of that class k

Review: Boosting

- Idea: Combine output of many weak classifiers to produce powerful committee
- Method: Sequentially fit weak learners with later models compensating the shortcomings of the existing learners
- Also shown to be an additive model fit using forward stage-wise manner
- Gradient boosting & Adaboost identify shortcomings differently

Review: Bootstrap

"The population is to the sample as the sample is to the bootstrap samples"



https://onlinecourses.science.psu.edu/stat555/node/119

Review: Tree Properties

- (Pro) Popular since they are highly interpretable
- (Pro) Model-free (don't assume an underlying distribution)
- (Con) Prediction accuracy is not that great — inherently high variance

We controlled variance and stabilized predictions using boosting — is there an other way?

Bagging

Bagging

- Bootstrap Aggregating: variance reduction technique introduced by Breiman in 1992
- Method: Average predictions over collection of bootstrap samples
  - Create B bootstrap replicates
  - Fits model to each replicate
  - Combines predictions via averaging or voting

$$\hat{f}^{\text{bag}}(\mathbf{x}) = \text{argmax}_G \sum_b \mathbb{1}_{\{\hat{f}_b^{\text{tree}}(\mathbf{x})=g\}}$$

Bagging Strategies

- Simple strategy: Grow fairly large trees on each sampled data set with no pruning
- More involved strategy: Prune back each tree but use original training data as validation set instead of performing cross-validation

Example: Bagging

Simulated data with n=30, two classes, and 5 features

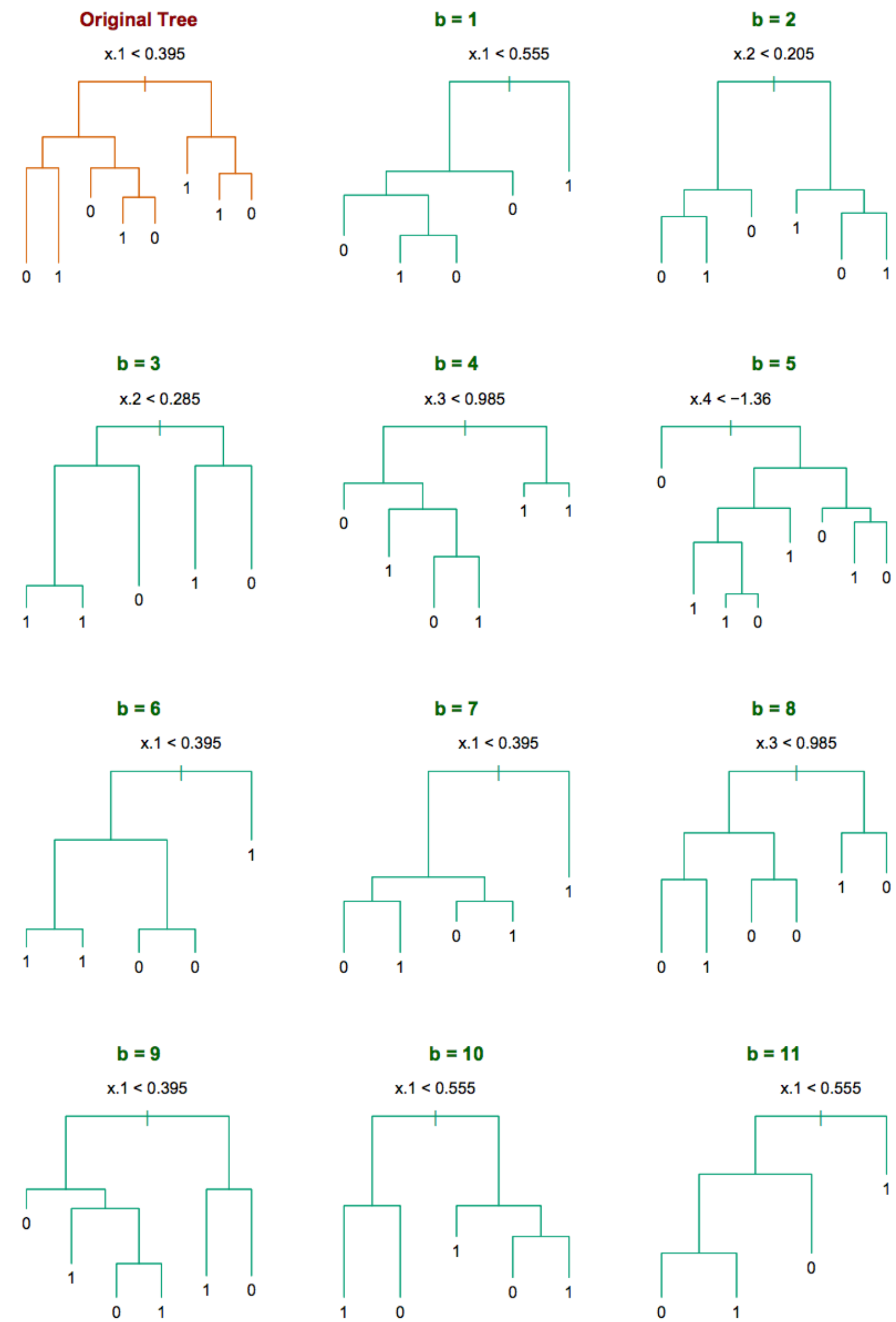


Figure 8.9 (Hastie et al.)

Example: Breiman's Experiment

Data Set	$\bar{e}_S$	$\bar{e}_B$	Decrease
waveform	29.1	19.3	34%
heart	4.9	2.8	43%
breast cancer	5.9	3.7	37%
ionosphere	11.2	7.9	29%
diabetes	25.3	23.9	6%
glass	30.4	23.6	22%
soybean	8.6	6.8	21%

Comparison of misclassification error between CART tree (pruned via cross-validation) and bagging (B = 50)

Bagging: Estimated Probability

- What if we were use to the proportion of votes for class g?

$$\hat{p}_g^{\text{bag}}(\mathbf{x}) = \frac{1}{B} \sum_b \mathbb{1}_{\{\hat{f}_b^{\text{tree}}(\mathbf{x}) = g\}}$$

- Why would this not be a good estimate?

Bagging: Estimated Probability

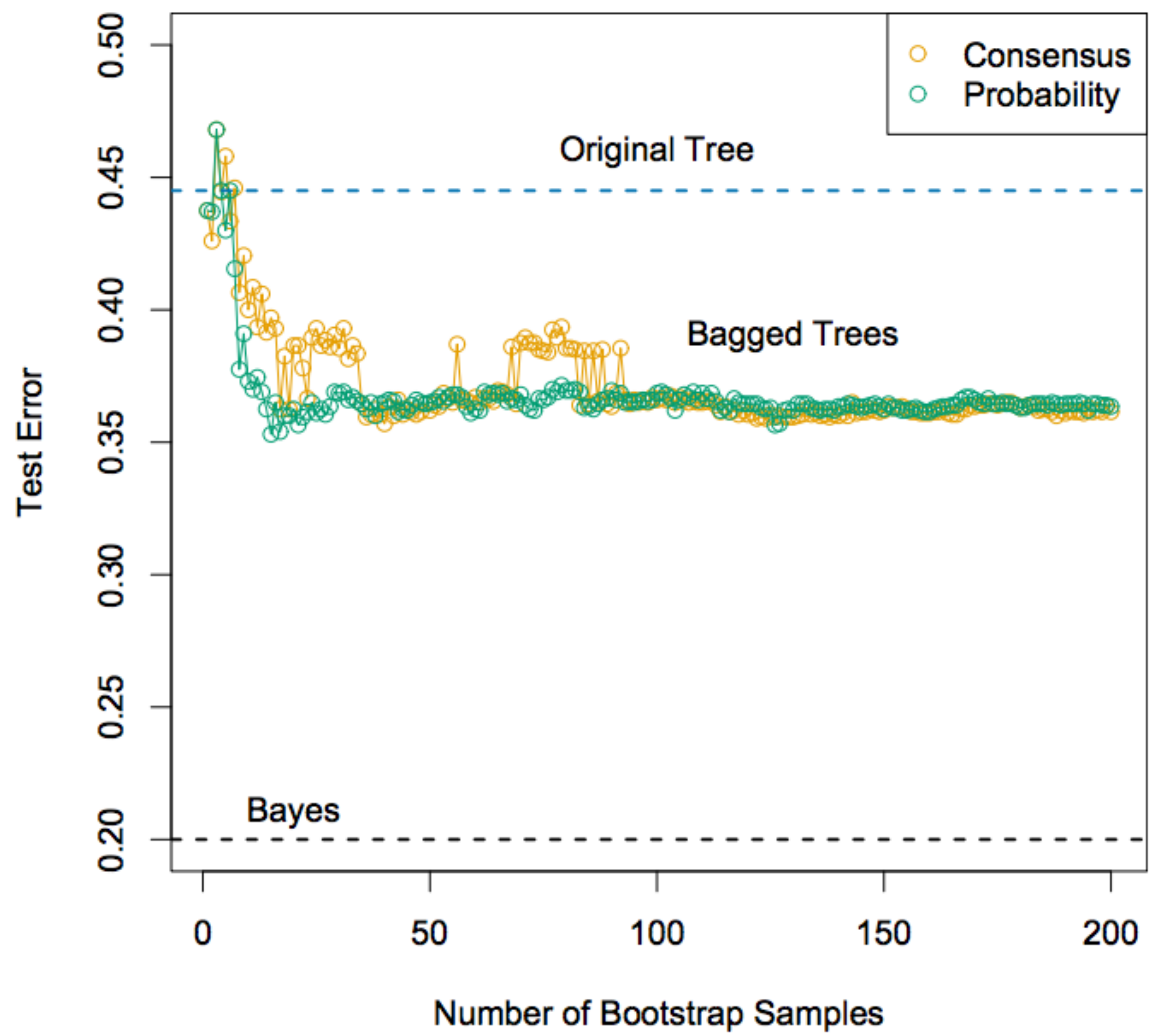
- Alternative form using predicted class probabilities from each tree

$$\hat{p}^{\text{bag}}(y = g|\mathbf{x}) = \frac{1}{B}\sum_b \hat{p}_b^{\text{tree}}(y = g|\mathbf{x})$$

- Final bagged classifier chooses class with highest probability

- Preferable for estimates of class probabilities and can help overall prediction accuracy

Example: Bagging



Bagging helps decrease the misclassification rate of the classifier (evaluated on large independent test set)

Figure 8.10 (Hastie et al.)

Why Does Bagging Work?

- Suppose that for a given input x in a binary classification problem where we have B independent classifiers and each as a misclassification rate e = 0.4

- Assume without loss of generality that the true class is 1

$$\Pr(\hat{f}_b(\mathbf{x}) = -1) = 0.4$$

- Our bagged classifier:

- $$\hat{f}(\mathbf{x}) = \text{argmax}_G \sum_b \mathbb{1}_{\{\hat{f}_b^{\text{tree}}(\mathbf{x}) = g\}}$$

Why Does Bagging Work?

- Let B_{-1} be the number of votes for class -1, a binomial variable with p=0.4
- Misclassification rate of the bagged classifier:

$$B_{-1} \sim \text{Binom}(B, 0.4)$$
$$\Pr(\hat{f}_b^{\text{bag}}(\mathbf{x}) = -1) = \Pr(B_{-1} \geq B/2)$$

- As B grows larger, our classifier should be perfect in theory
  - This is not the case as this assumes independence and our classifiers are not independent

When Does Bagging Fail?

- Assume misclassification rate is higher than 0.5

$$\Pr(\hat{f}_b(\mathbf{x}) = -1) = 0.6$$

- Bag misclassification rate

$$B_{-1} \sim \text{Binom}(B, 0.6)$$
$$\Pr(\hat{f}_b^{\text{bag}}(\mathbf{x}) = -1) = \Pr(B_{-1} \geq B/2)$$
$$B \to \infty, \Pr(B_{-1} \geq B/2) \to 1$$

- If the misclassification rate is high, the bagged classifier is perfectly inaccurate as B approaches infinity (degradation in predictive accuracy)

Example: Wisdom of Crowds

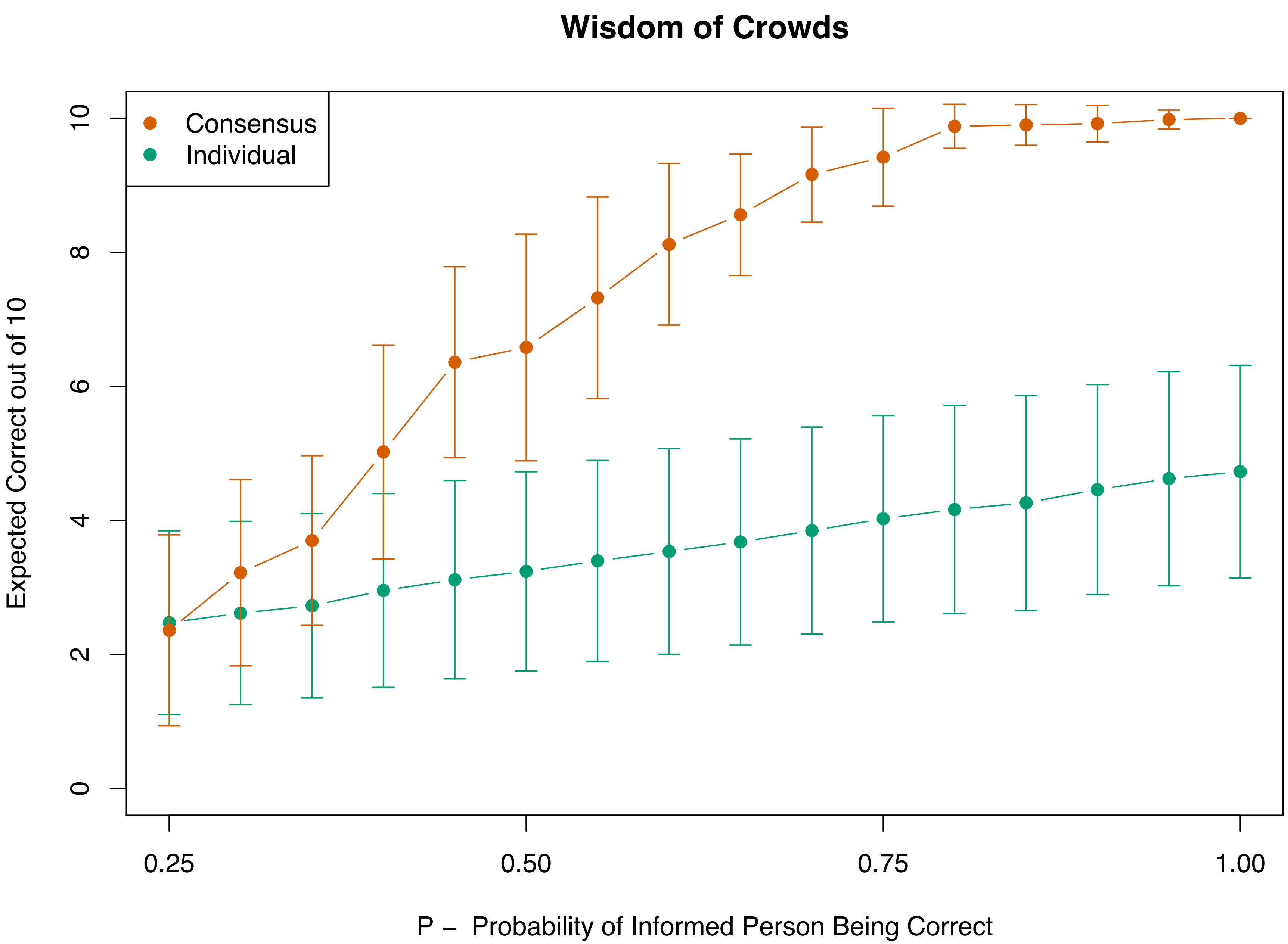


Figure 8.11 (Hastie et al.)

Bagging: Properties

- (Pros) Stabilizes unstable models
- (Pros) Easily parallelizable
- (Cons) Loss of interpretability
- (Cons) Computational complexity
- (Cons) Limited model space — bagging can still not easily represent certain boundaries

Bagging & Trees

- Bagging — average noise but approximately unbiased models to reduce variance
- Trees are ideal candidates for bagging
    - Capture complex interactions
    - Relatively low bias (with sufficient depth)
- Each tree grown in bagging is i.i.d — expectation of average is same as expectation of one of them

Boosting vs Bagging

- Boosting fits the entire training set whereas bagging is just bootstrap samples
- Boosting adaptively adjusts the weight of the observations to encourage better predictions for misclassified points — bagging uses equal weights for all observations
- Boosting tends to have greater accuracy compared to bagging but also risks overfitting
- Boosting reduces bias while bagging does not

Boosting vs Bagging

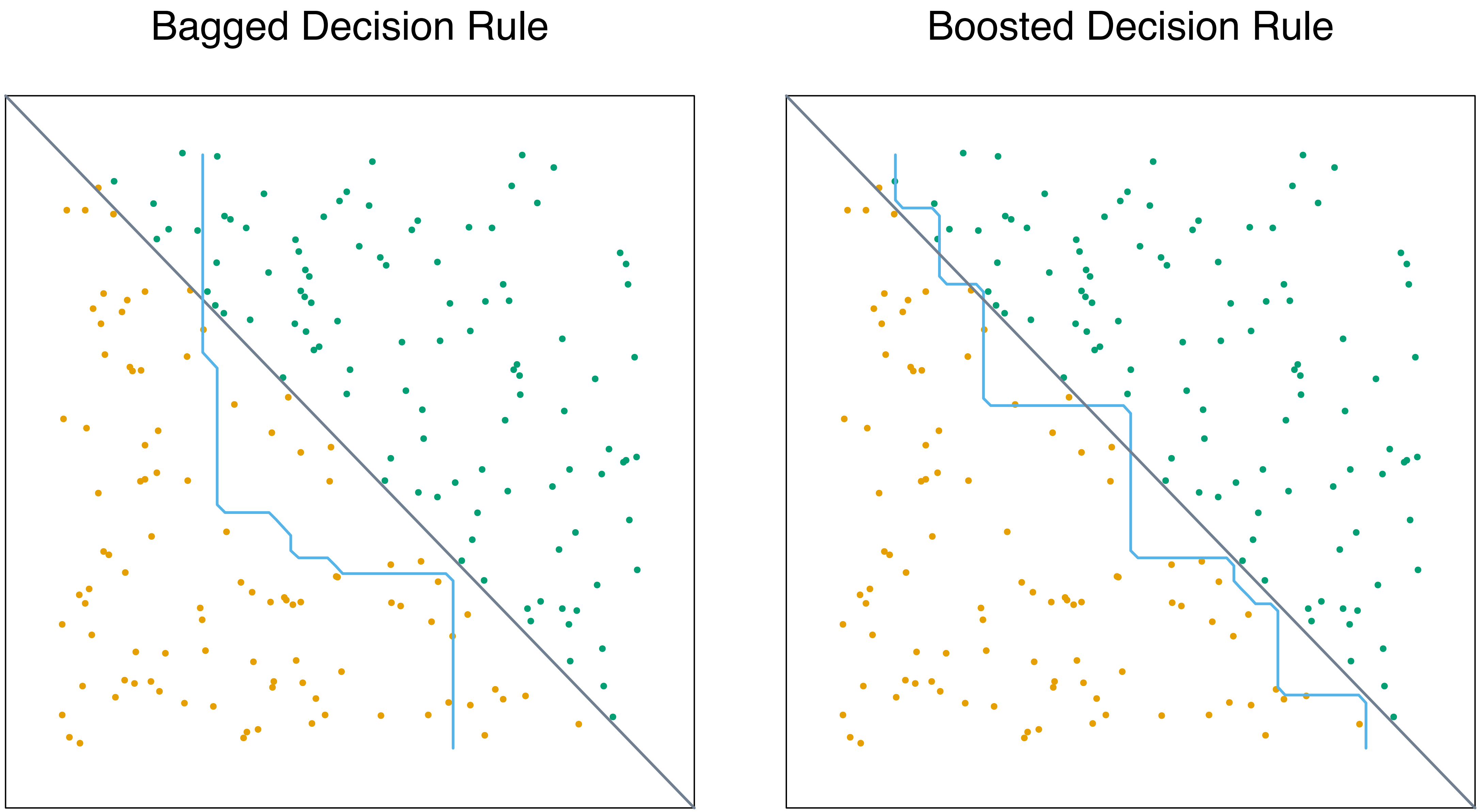


Figure 8.12 (Hastie et al.)

Random Forest

Random Forest: Motivation

- Average of B i.i.d random variables with variance σ^2 has variance σ^2/B

- Average of B i.d. random variables with positive pairwise correlation has a variance

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

- Size of the correlation of bagged trees limits benefits of averaging —> reduce correlation between trees without increasing variance too much

Random Forests (Breiman, 2001)

- Bagged classifier using decision trees
  - Each split only considers a random group of features
  - Tree is grown to maximum size without pruning
  - Final predictions obtained by aggregating over the B trees

$$\hat{f}_{\mathrm{rf}}^{B}(\mathbf{x}) = \frac{1}{B}\sum_{b} T(\mathbf{x}; \theta_b)$$

Random Forest: Algorithm

Algorithm 15.1 *Random Forest for Regression or Classification.*

1. For $b = 1$ to B:
   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.
   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i. Select m variables at random from the p variables.
      ii. Pick the best variable/split-point among the m.
      iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote } \{\hat{C}_b(x)\}_1^B$.

Example: Spam Data

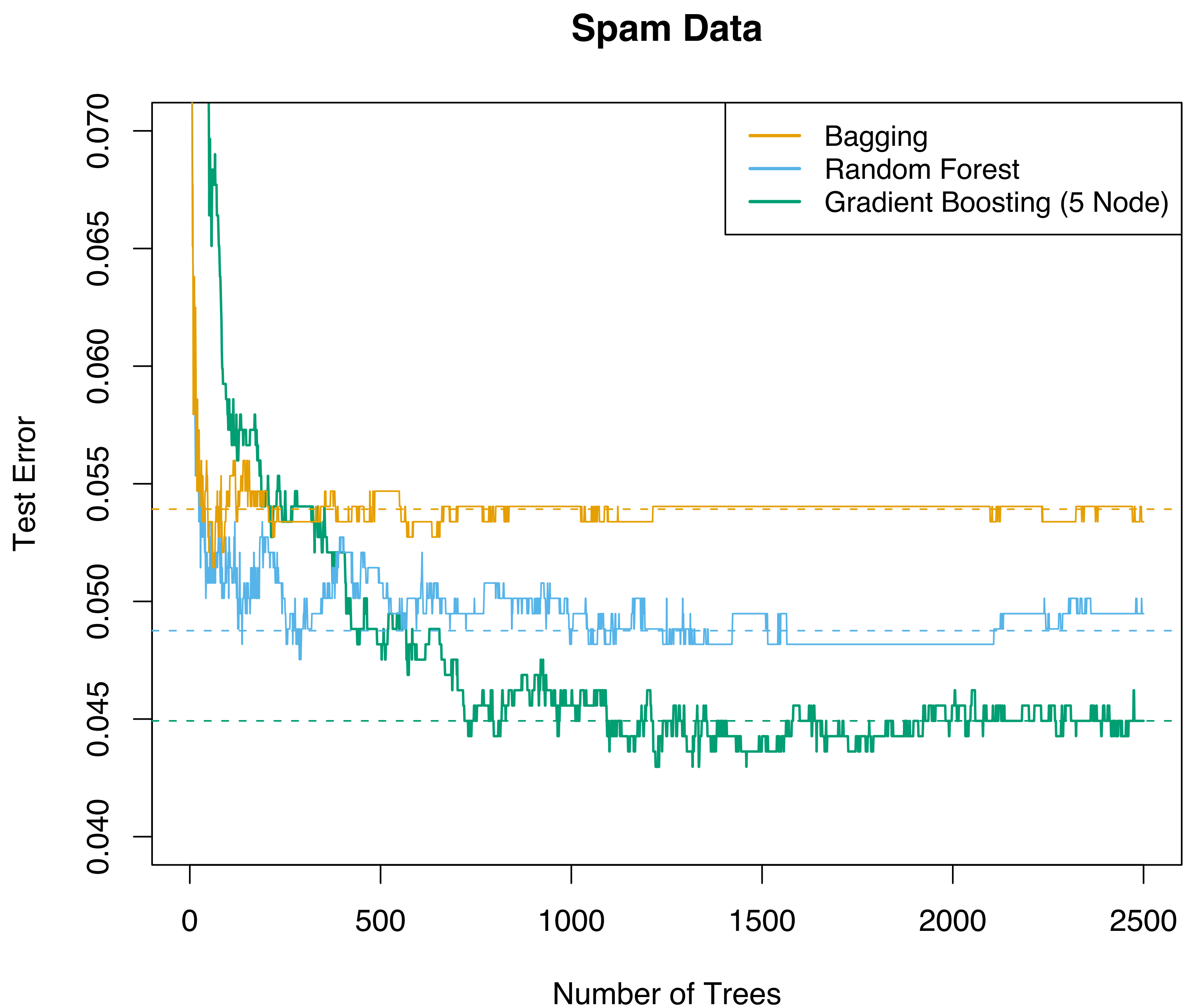


Figure 15.1 (Hastie et al.)

Out of Bag (OOB) Samples

- For each observation, construct its random forest predictor by averaging only those trees corresponding to bootstrap samples in which observation does not appear
- OOB error estimates almost identical to N-fold cross-validation — means can be fit in one sequence
- Once OOB stabilizes, training can be stopped

Example: OOB Error

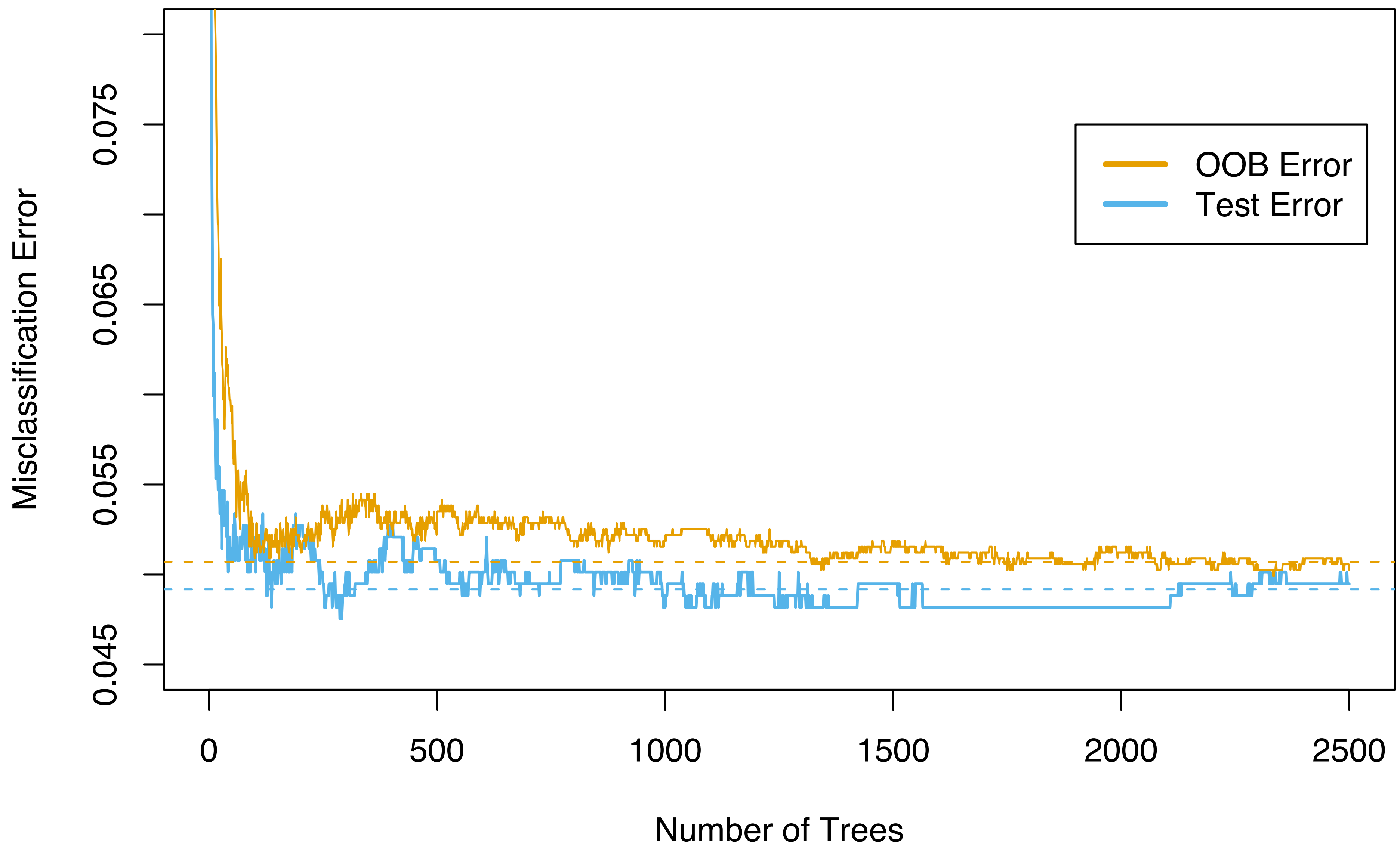


Figure 15.4 (Hastie et al.)

Variable Importance

- Option 1: Same way as gradient-boosted models
- Option 2: OOB samples to measure prediction strength
  - For bth tree, OOB samples are passed down tree and accuracy recorded
  - Values for jth variable are randomly permuted in OOB samples and accuracy again computed
  - Decrease in accuracy is used as measure of importance

Example: Variable Importance

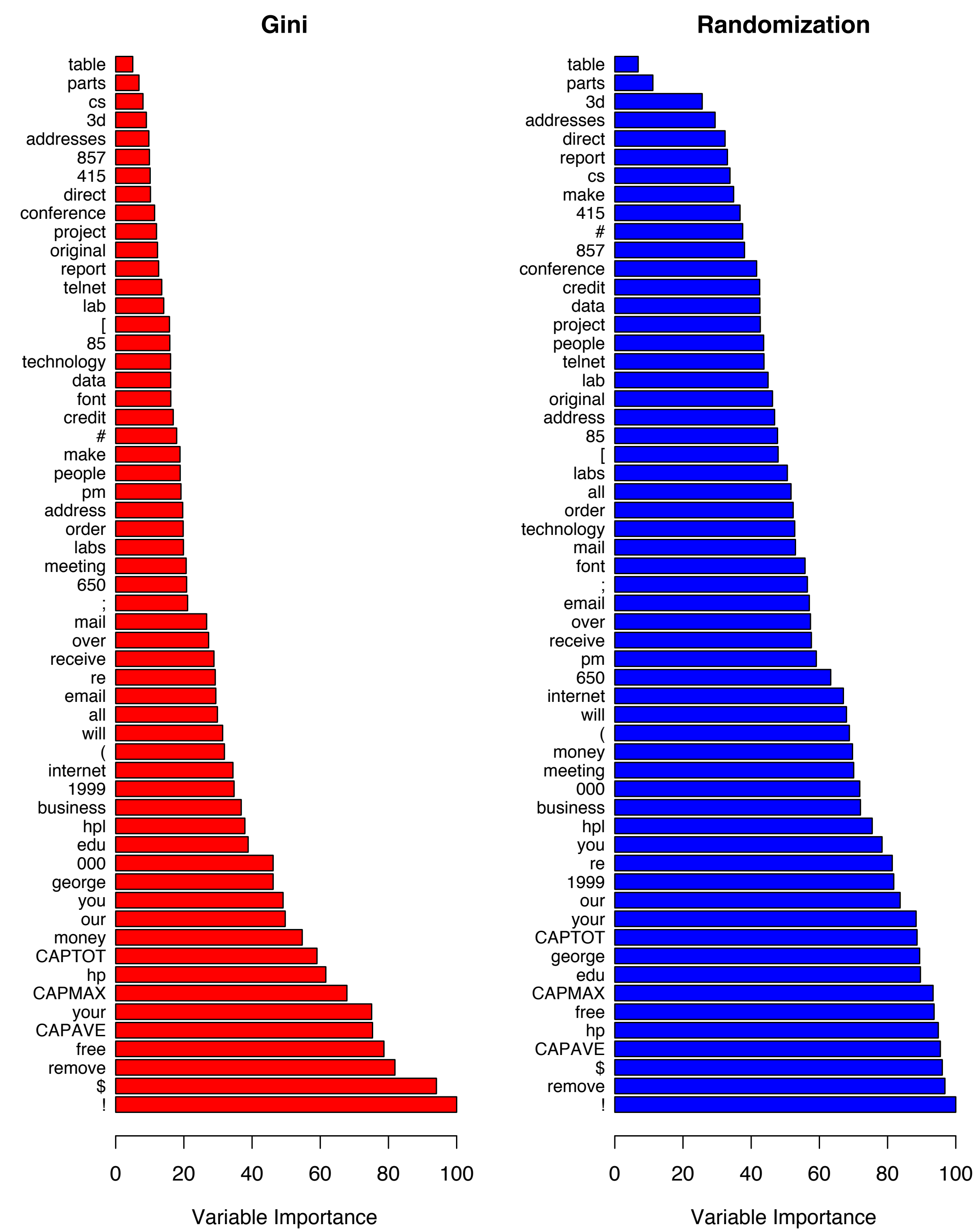


Figure 15.5 (Hastie et al.)

Random Forests: Properties

- State of the art method, generally one of the most accurate general-purpose learners available
- Handles a large number of input variables without overfitting
- Easy to train and tune
- Reduces correlation amongst bagged trees by considering only a subset of variables at each split

Random Forest: Advantages

- Easily parallelized by training
- Robust to errors and outliers
- Can model non-linear boundaries
- Gives variable importance and out of bag error rates

Random Forest: Disadvantages

- Loss of interpretability
- Difficult to analyze as an algorithm and mathematical properties still largely unknown
- Large number of trees is memory-intensive
- Bias towards categorical variables with larger number of levels

Extra Trees

Recap: Tree Growing

1. Start with dataset
2. Pick a splitting feature
3. Pick a splitting cut-point
4. Split dataset into two sets based on feature and cut-point
5. Repeat from step 2 with the partitioned dataset

Recap: C4.5, CART

1. Start with dataset
2. Pick a splitting feature
3. Pick a splitting cut-point

Information gain —> C4.5
Gini impurity —> CART
Variance reduction —> CART

4. Split dataset into two sets based on feature and cut-point
5. Repeat from step 2 with the partitioned dataset

Recap: Random Forest

1. Start with dataset

Bootstrap samples

2. Pick a splitting feature
3. Pick a splitting cut-point

Random subset of features
Find best feature / cutpoint

4. Split dataset into two sets based on feature and cut-point
5. Repeat from step 2 with the partitioned dataset

Extra Trees

1. Start with dataset

2. Pick a splitting feature

3. Pick a splitting cut-point

Select random subset of (feature, cutpoint) pairs
Find best (feature, cutpoint)

4. Split dataset into two sets based on feature and cut-point

5. Repeat from step 2 with the partitioned dataset

Favorite Tradeoff: Bias & Variance

- Recursive partition —> fewer samples as tree grows
- Split features / cutpoints are susceptible to training examples
- Randomization decreases variance

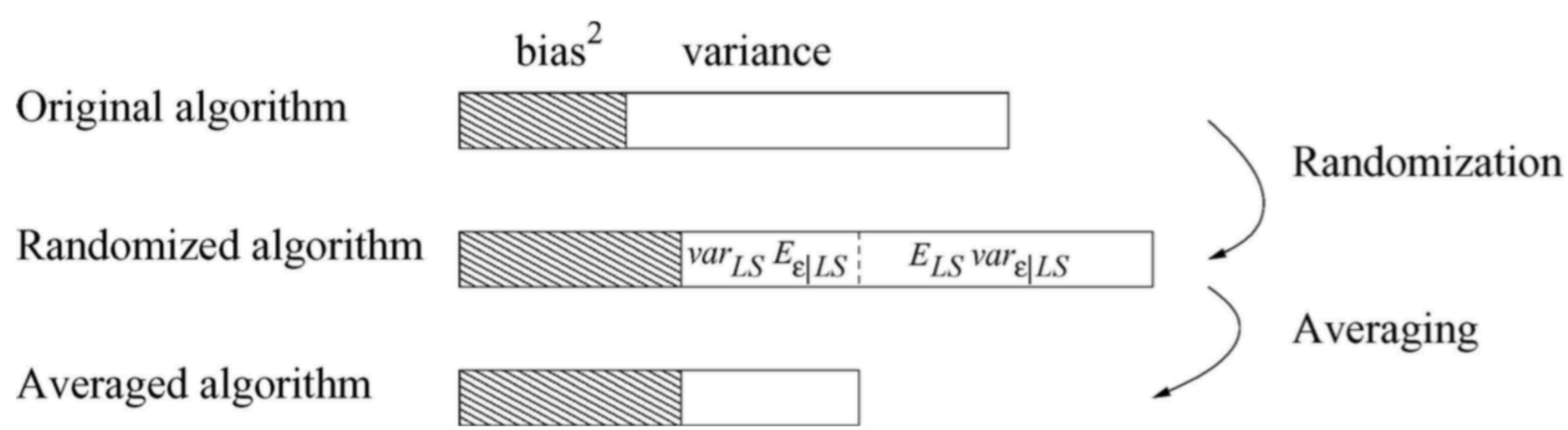


Fig. 11 Expected evolution of bias and variance by randomization and averaging.

Geurts et al. 2006

Example: Predicting MedAdh Scores

- Centers for Medicare and Medicaid Services (CMS) measures the performance of Medicare Advantage (MA) Plans via Star Rating System
- Medication Adherence (MedAdh) is one of the most important quality measures
- MA plans want to know how much their MedAdh scores will change in next two years

Example: MedAdh Data

- Data can be found at CMS Part C and Part D performance webpage
- Datasets
  - Train: MedAdh data from 2012, 2013 to predict 2015
  - Test: MedAdh data from 2013, 2014 to predict 2016

Example: MedAdh Missing Values

- Not all MA plans are measured in a given year

```
X1,X2,X3,X4,X5,X6,X7,X8,X9,Y
...
71.2,72.7,69.9,75.2,75.9,71.0,1.8
-999,-999,-999,75.8,72.5,68.8,-4.8
61.8,59.4,57.7,57.3,59.3,58.3,16.7
...
-999,-999,-999,82.8,80.0,69.8,-11.8
73.8,73.2,71.8,74.5,76.1,72.9,4.5
```

- How to deal with missing data? Mean imputation

Example: MedAdh Model Bakeoff

- Try four different models
  - Linear regression
  - Decision tree
  - Extra tree
  - Gradient boosting

```
from sklearn import linear_model
from sklearn import tree
from sklearn.utils import resample
from sklearn.metrics import mean_squared_error
from sklearn.ensemble import ExtraTreesRegressor
from sklearn.ensemble import GradientBoostingRegressor
```

Example: MedAdh Model Bakeoff

- Code snippet:

```
lm = linear_model.LinearRegression()
dt = tree.DecisionTreeRegressor()
etr = ExtraTreesRegressor(n_estimators=100, max_depth=10)
gbr = GradientBoostingRegressor(n_estimators=500,
                                learning_rate=0.25,
                                max_depth=8)
```

Example: Results

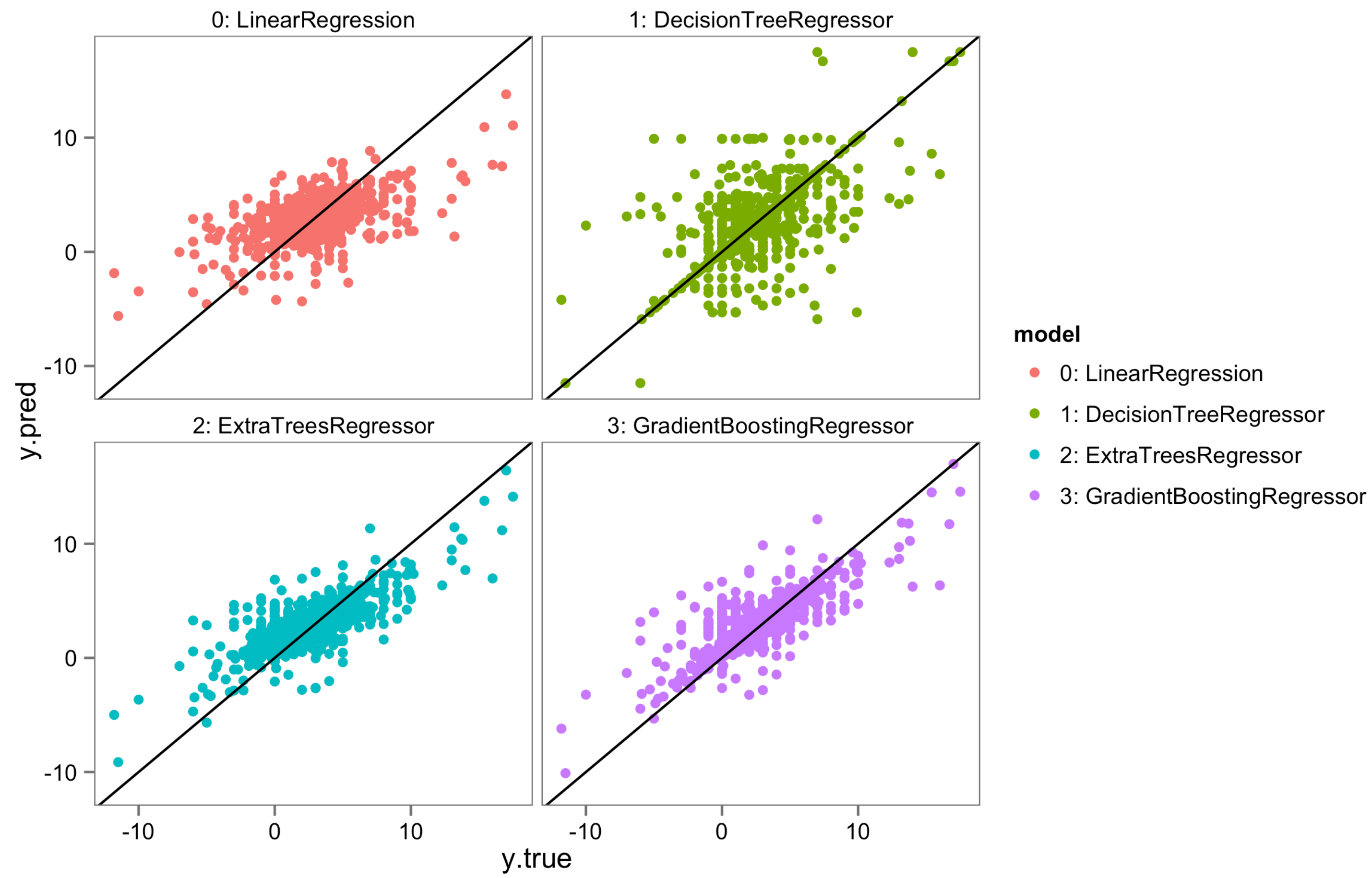
RMSE Results

lm: 2.7125536923

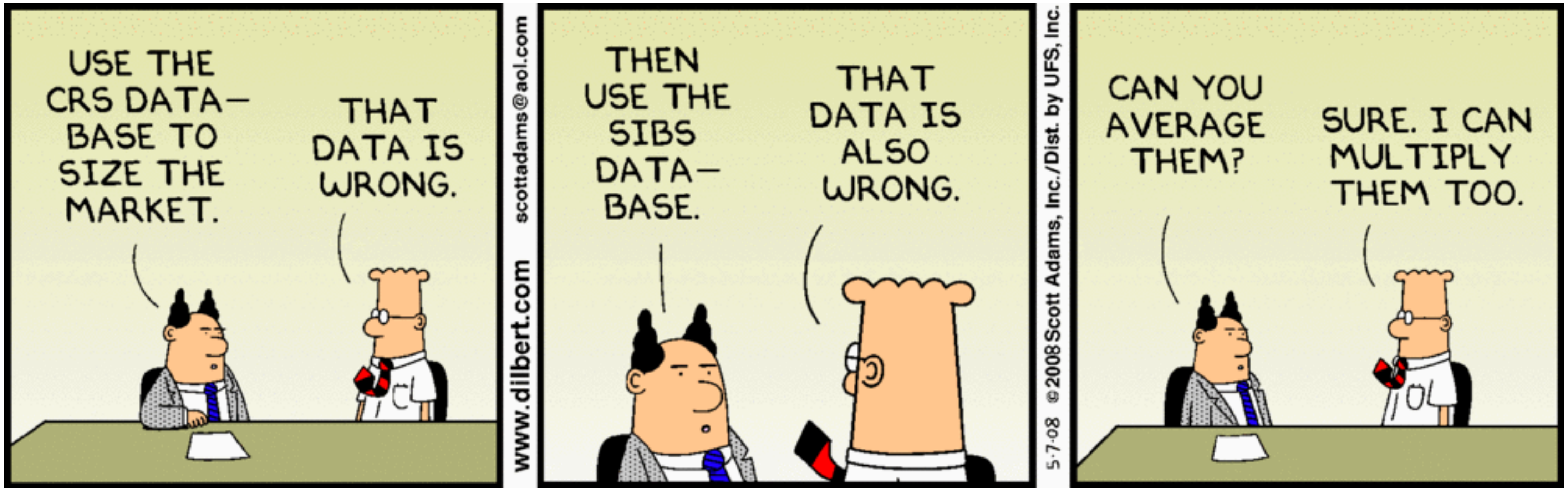
dt: 3.10460672029

etr: 2.18597303421

gbr: 2.02698129388



Extra trees and gradient boosting exhibit better improvement over linear regression & decision tree

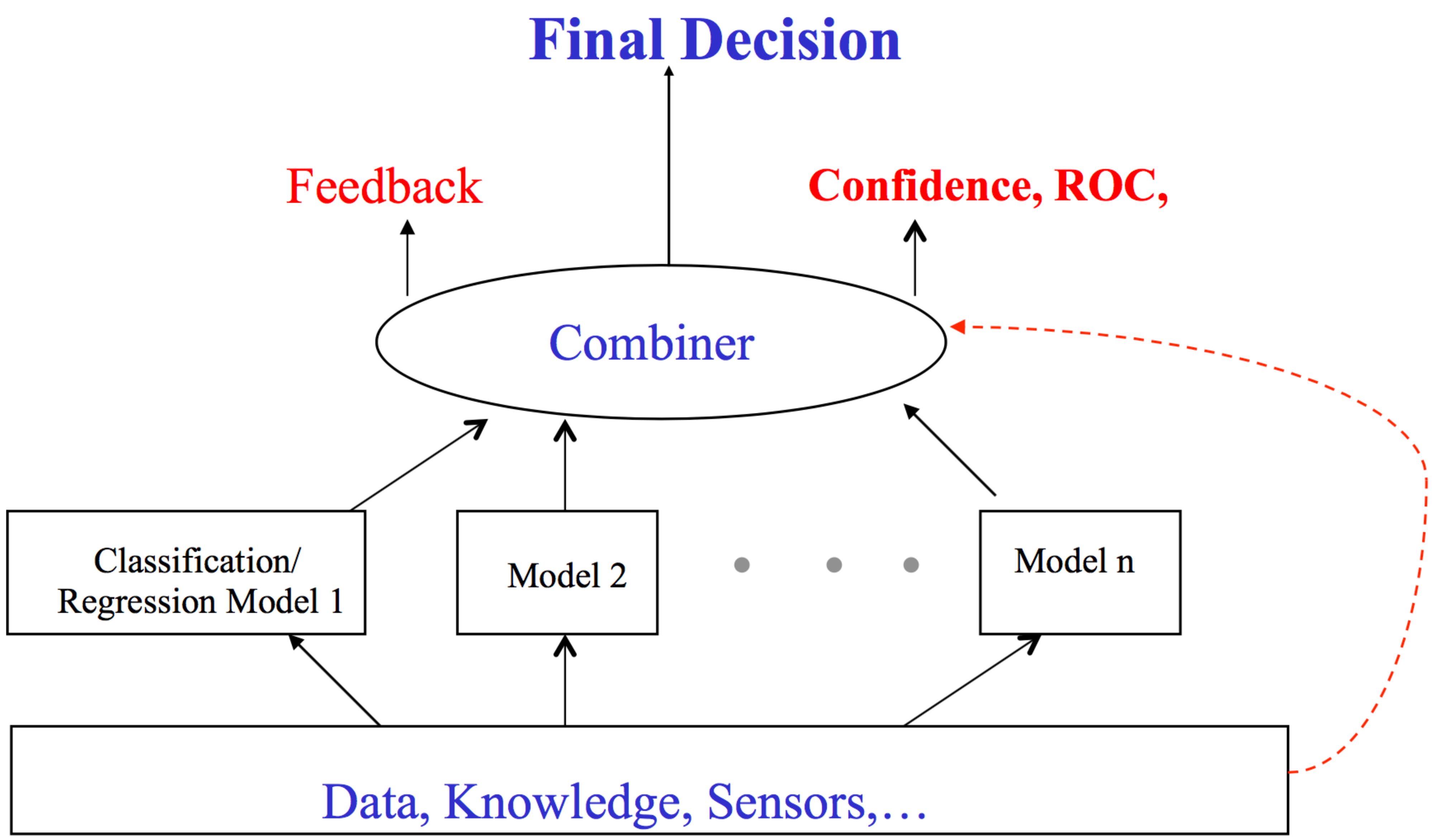


Ensemble & Multi-Learner System

Ensembles & Multi-Learner Systems

- Goal: use multiple “learners” to solve (parts of) the same problem
  - Function approximation
  - Classification
- Ensembles — competing learners with multiple looks at the same problem
- Mixture of experts — cooperative learners with the divide and conquer approach

Generic Multi-Learner System



“This is how you win ML competitions: you take other peoples’ work and ensemble them together.”

–Vitaly Kuznetsov, NIPS 2014

Voting: Ensemble w/o Retraining

- Given existing model predictions, find different ways to team them up
- Voting ensembles mimic error-correcting codes
  - More voters —> potential better signal to noise
  - Lower correlation between models
- Weighted majority (better model gets more weight) vs average

Stacked Generalization

- Introduced by Wolpert, 1992
- Use a pool of base classifiers, then use another classifier to combine their predictions
- Stacker model gains information by using first-stage predictions as features
- If used incorrectly, can lead to information leakage

Example: 2-fold Stacking

- Split training data into 2 parts, A and B
- Fit a first-stage model on A and create predictions on B
- Fit a first-stage model on B and create predictions on A
- Train a second-stage stacker model on probabilities from first-stage model(s)

Blending

- Close to stacked generalization, but a bit simpler
- Instead of out-of-fold predictions, create small holdout set that the stacker is then trained on this set
- Generalizers and stackers use different information
- No need to share seed for stratified folds with teammates — throw models in the 'blender' and blender decides to keep it or not

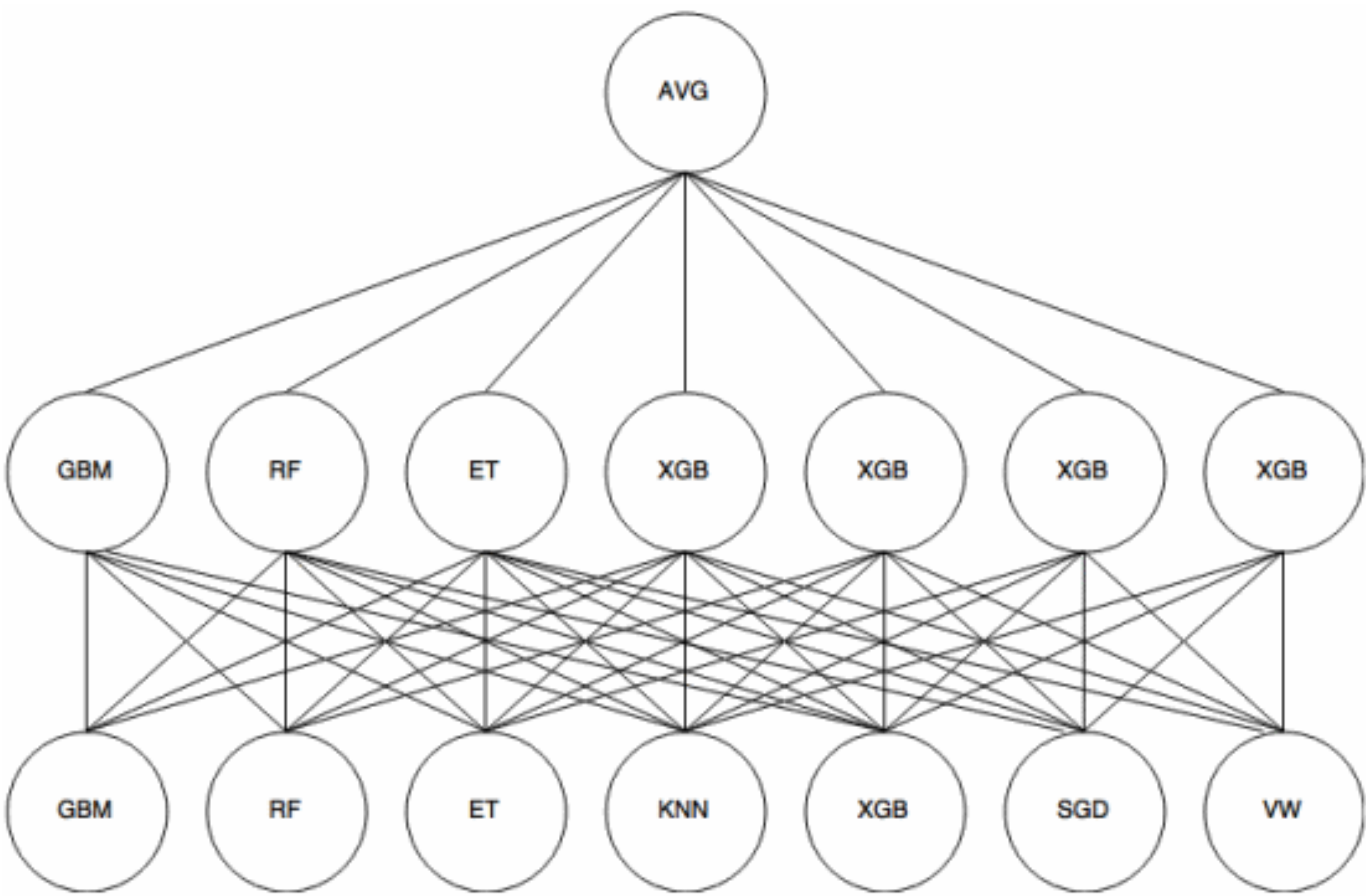
Blending: Disadvantages

- Less data used overall
- Final model may overfit to holdout set
- Single small holdout set won’t necessarily yield good generalization errors

Stacking / Blending

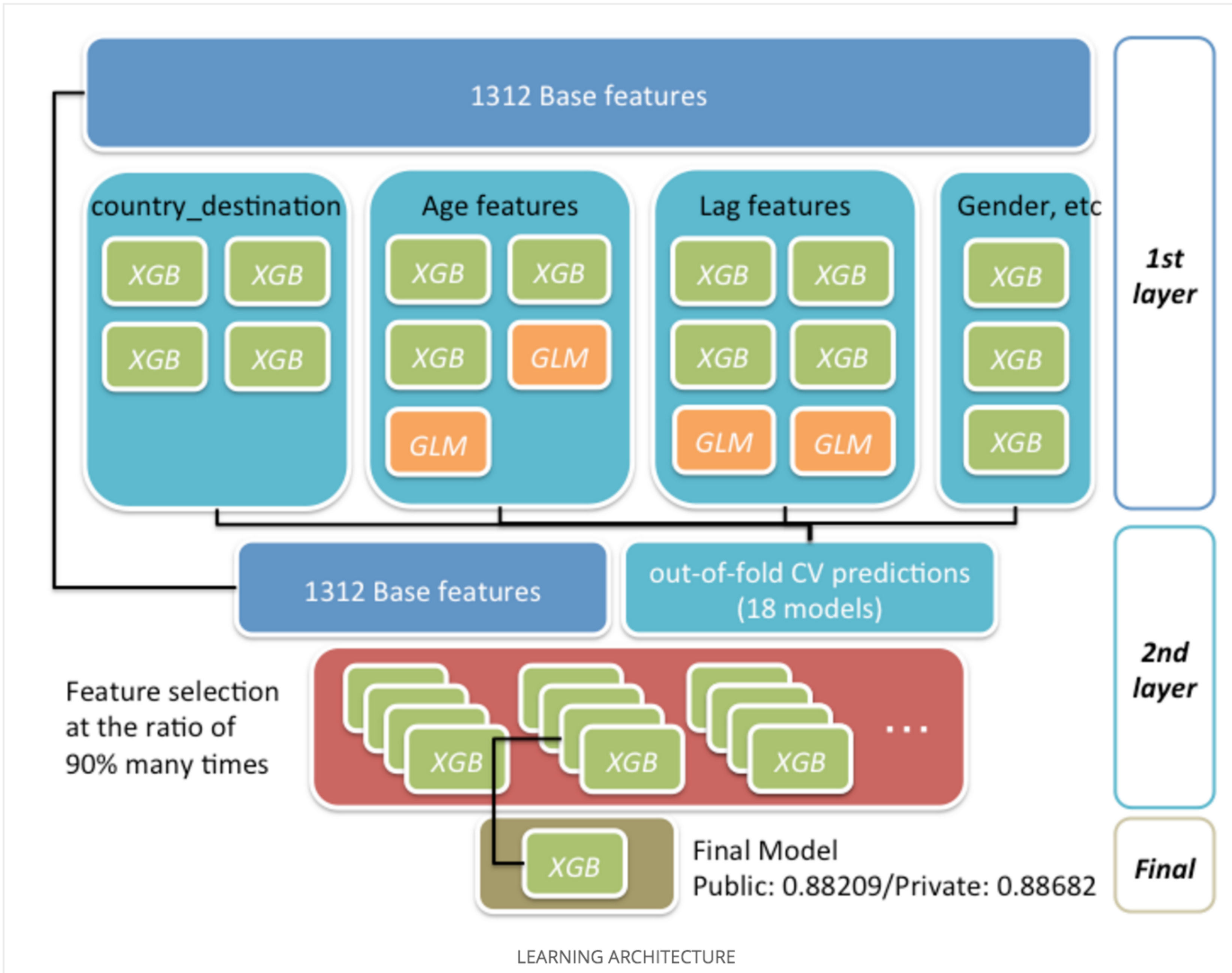
- Everything is a hyperparameter
  - Different preprocessing of the data
  - Imputation
  - Feature selection
- Why stop at two stages? Why not combine multiple ensembles models?

Example: Otto Product Classification



http://mlwave.com/kaggle-ensembling-guide/

Example: Airbnb 2nd Place



http://blog.kaggle.com/2016/03/17/airbnb-new-user-bookings-winners-interview-2nd-place-keiichi-kuroyanagi-keiku/