Random Forests

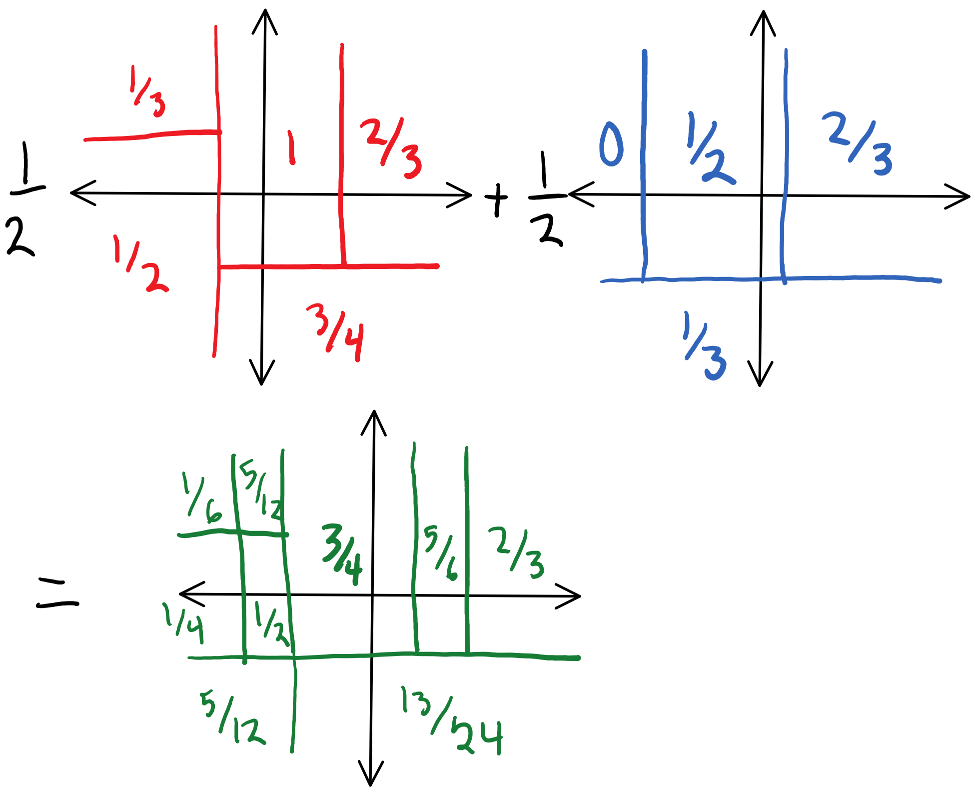
Sunday, May 21, 2017 9:34 PM

Decision Trees

Pros: Automatic variable selection, scalable algorithm, handle mixed features, interpretable (small trees)

Con: poor prediction performance

Averaging predictors: $\hat{f}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b(x)$ (ensemble)

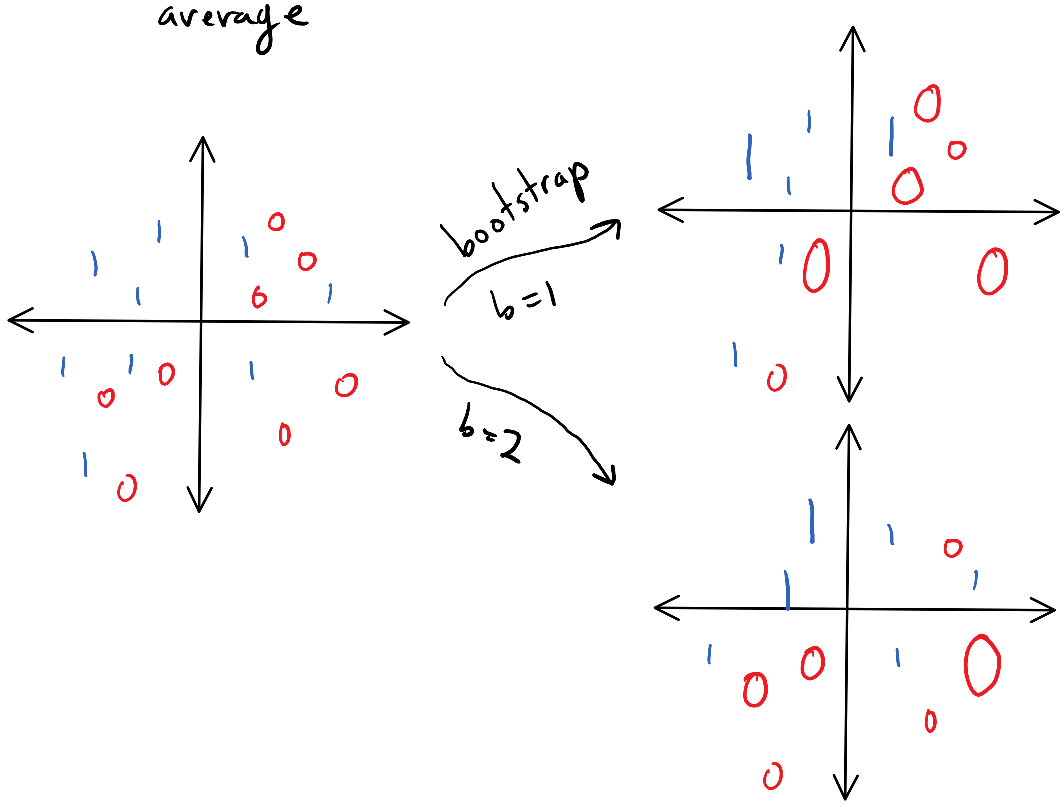


Averaging can make complex predictors! But does it help?

Bias:

Variance:

Bagging: fit $\hat{f}_b$ on bootstrap sample $b = 1, \ldots, B$ and average



How can we reduce bias?

How can we reduce variance?

Random Forests (input # of trees B, # of features m, bootstrap sizes N)

For $b = 1, \ldots, B$

Bootstrap N rows w/ replacement

Sample m features and grow max depth decision tree

Save the trees and sample frequencies

Predict with $\frac{1}{B}\sum_{b=1}^{B} \hat{f}_b(x)$

Out-of-bag (OOB) error: for each (x_i, y_i)

Let B_i be bootstrap samples w/out i

Predict with $\hat{y}_i^o = \frac{1}{B_i} \sum_{b \in B_i} \hat{f}_b(x_i)$. and

Compute OOB risk $\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \hat{y}_i)$

Tradeoffs

B (# of trees) ↗ reduces algorithmic variance
increased computation

m (# of features) ↘ allows more features to have a chance (greedy)
each tree has worse risk

N (Bootstrap size) ↗ increases tree covariance
better training risk for trees
more computation

Variable importance

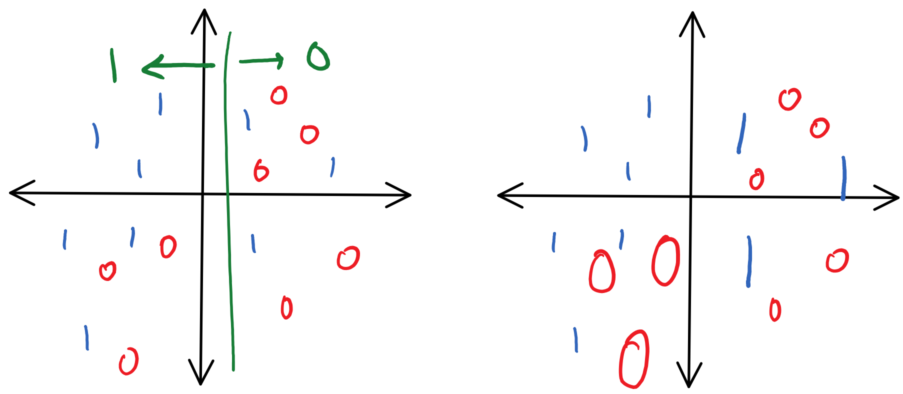
For each bag, when a variable is split on, add reduction in criteria (Emp. Risk) to "importance"

Boosting

Sunday, May 21, 2017 9:35 PM

Bagging: Random data augmentation

Boosting: Adaptive data augmentation



Increase weight on misclassified points

Adaboost $(y_i \in \{-1, 1\})$

Initialize $w_i = \frac{1}{n}$ $i = 1, \ldots, n$

For $b = 1, \ldots, B$

Fit classifier to the training data $\hat{y}_b$ using weights w_i

Compute weighted misclassification error

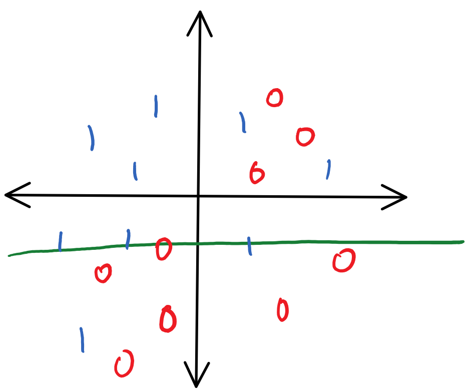
$$\varepsilon_b = \frac{\sum_i w_i I\{y_i \neq \hat{y}_b(x_i)\}}{\sum_i w_i}$$

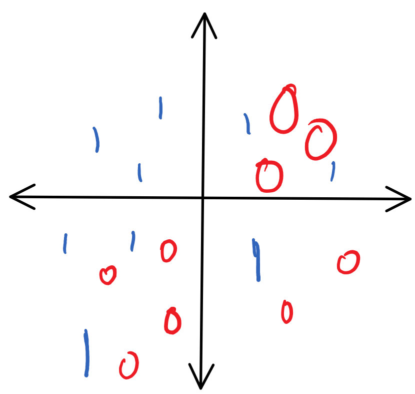
Compute $\alpha_b = \log \frac{1-\varepsilon_b}{\varepsilon_b}$

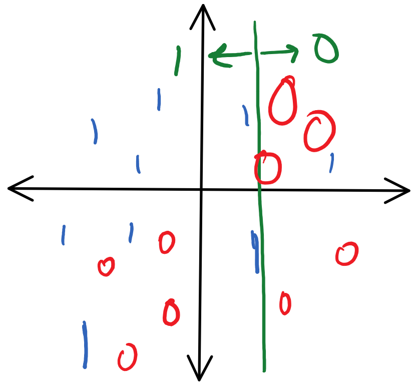
Update weights $w_i \leftarrow w_i \cdot \exp\left\{ \alpha_b I\{y_i \neq \hat{y}_b(x_i)\} \right\}$

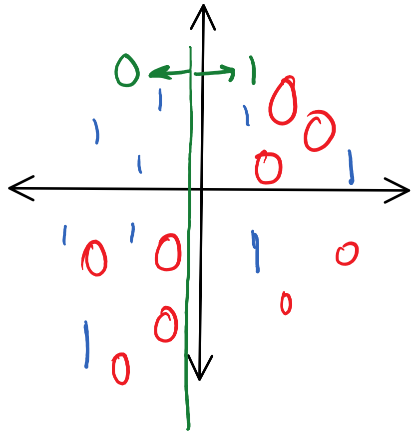
$\hat{f}(x) = \sum_{b=1}^{B} \alpha_b \hat{y}_b(x)$ and $\hat{y}(x) = \text{sign}(\hat{f}(x))$

Train classifiers using stumps (low depth trees)









Test error improved by adaptive data augmentation

Only need each stump to have better than random guessing performance - "weak learner"

Boosting fits a generalized additive model,

$$\hat{f}_B(x) = \sum_{b=1}^{B} \alpha_b f_b(x)$$

Consider $\quad l(y, \hat{f}(x)) = e^{-y\hat{f}(x)}$

$$\min_{\beta, f_B} \sum_{i=1} e^{-y_i(\hat{f}_{B-1}(x_i) + \beta f_B(x_i))}$$