Ensemble Methods:
committee-based learning

Jay Hyer

linkedin.com/in/jayhyer

@aDataHead

Overview

- Why Ensemble Learning?
  - What is learning ?
  - How is ensemble learning different?
- Boosting
  - Weak and Strong Learners
  - AdaBoost
- Bagging
  - Bootstrapping and Aggregating
  - Out-of-Bag Error
  - Random Forest
- Stacking
  - Beyond Averaging
- Special Applications
  - Proximity and outliers
  - Class imbalance
- Conclusion
- Questions?

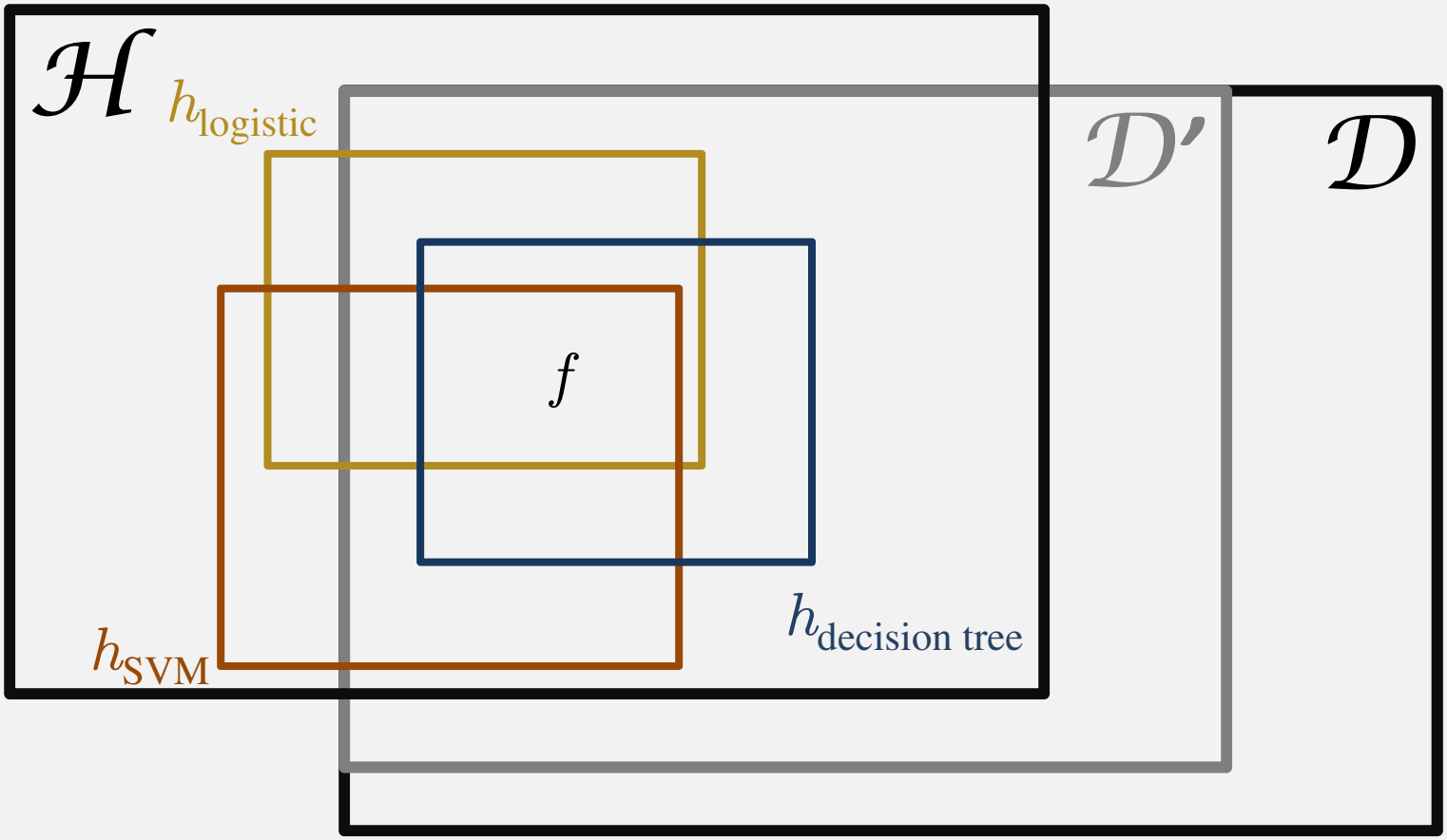
Why Ensemble Learning?
What is learning?

- $\mathcal{D}$ is the phenomenon under investigation: credit card transactions, clinical trails, handwritten digits, ect…
- $\mathcal{D}'$ is the dataset of the observed phenomenon: $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$.
- f is the *target* or *ground-truth* function: $f(x_i) = y_i$.
- $\mathcal{H}$ is the hypothesis space, which is a collection of all possible estimates of the target function f. We will refer to a specific estimate of f from $\mathcal{H}$ as h_i.

[Abu-Mustafa et. al., 2012] (Chapter 1), [Zhou, 2012] (Chapter 1)

Why Ensemble Learning?

What is learning?



Why Ensemble Learning?
What is learning?

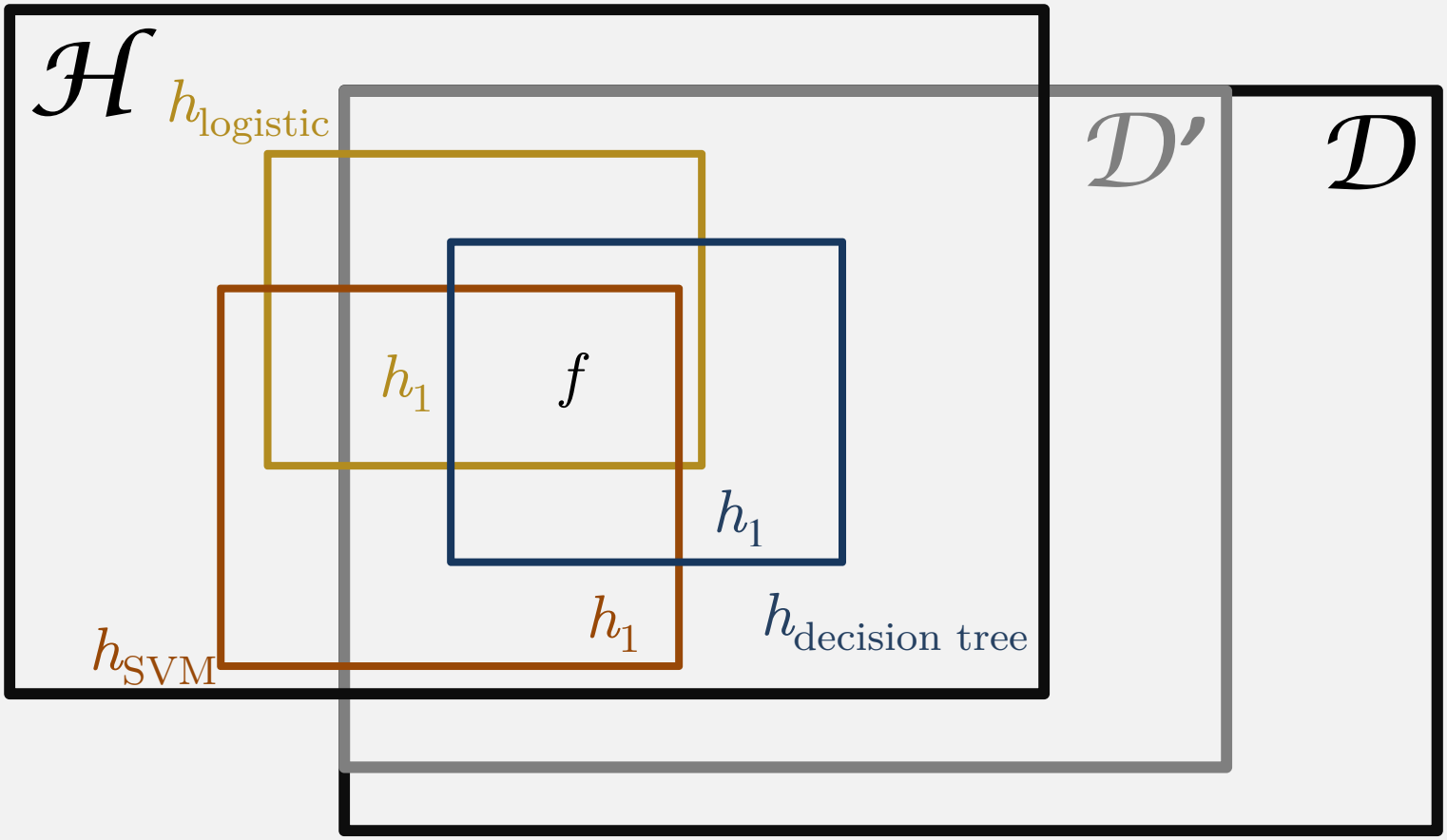
- $h_{\text{logistic}}: 1/1+e^{-y}$ s.t. $y = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$.
- h_{SVM}: $\max_\alpha \Sigma_i \alpha_i - 0.5 \Sigma_j \Sigma_i \alpha_i \alpha_j y_i y_j K(x_i, x_j)$.
- $h_{\text{decision tree}}: Ent(S) - \Sigma_k |S_k|/|S| \cdot Ent(S_k)$, information gain.

[Hastie, et. al.,2009] (Chapters 4, 9 & 12), [Mitchell,1997] (Chapter 3)

Why Ensemble Learning?

What is learning?

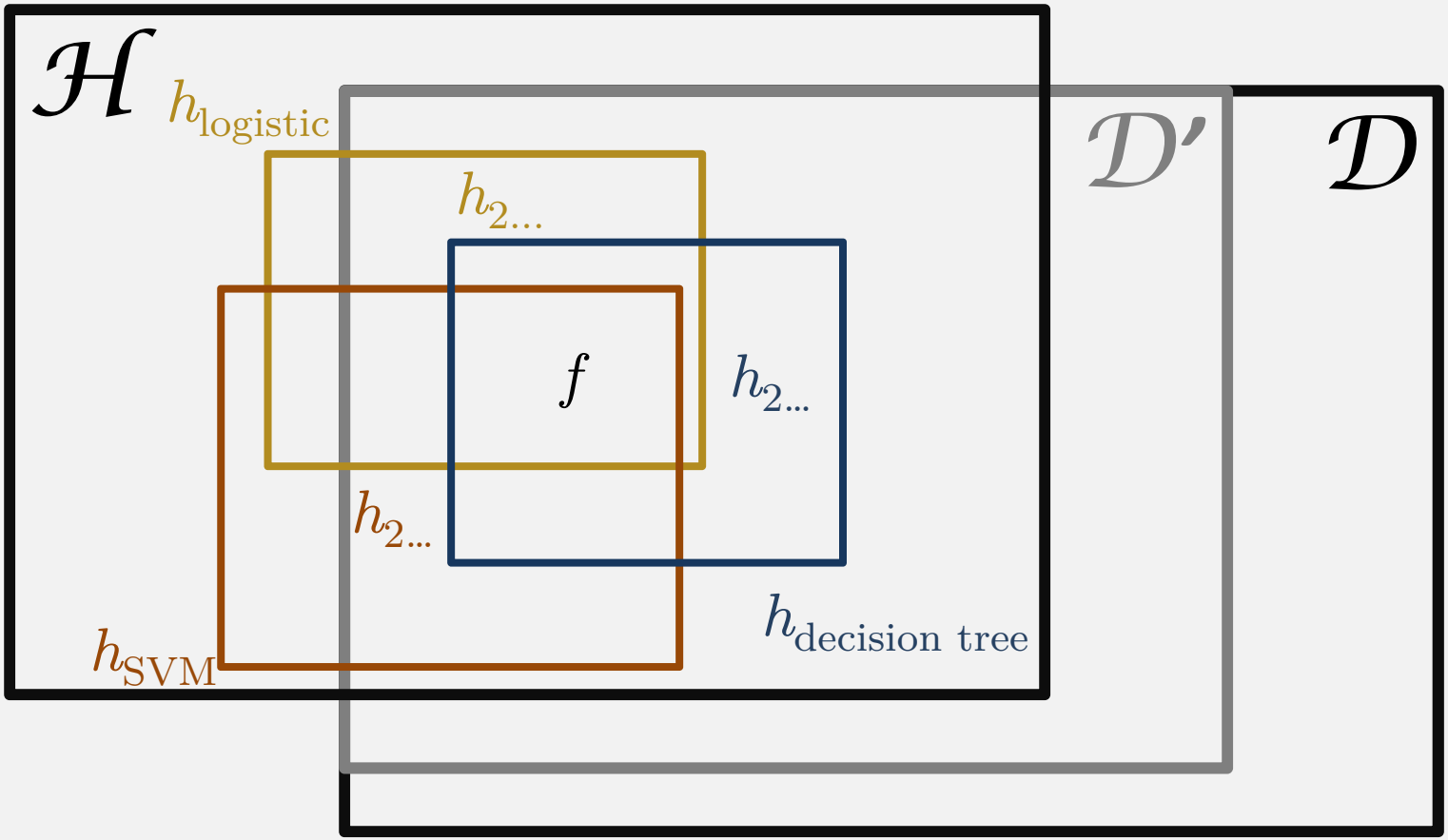
model training …



Why Ensemble Learning?

What is learning?

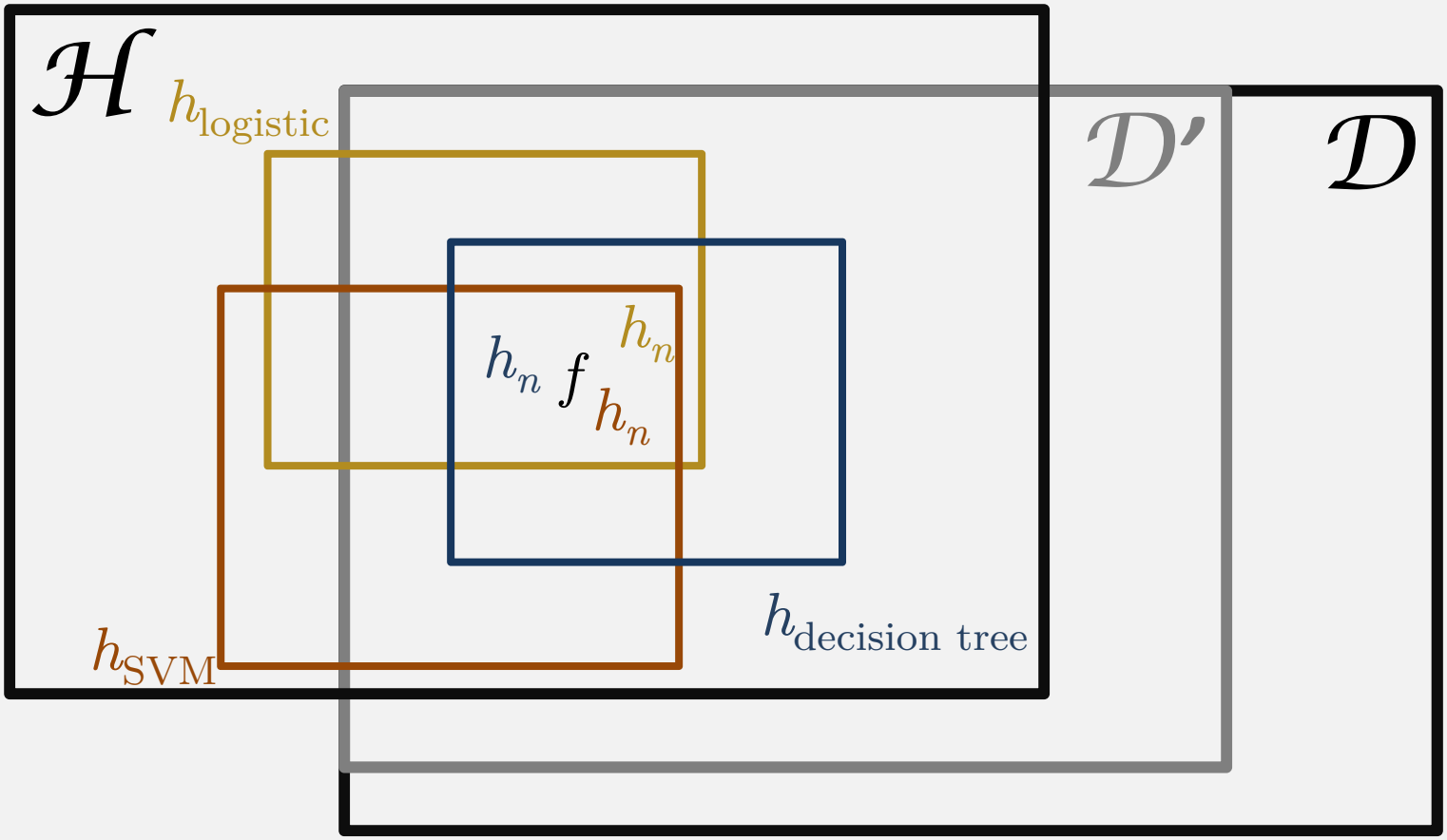
model training … …



Why Ensemble Learning?

What is learning?

… … model trained



Why Ensemble Learning?
How is ensemble learning different?

- Typically, we choose a single model from $\{h_1, h_2, \ldots, h_m, h_1, h_2, \ldots, h_m, h_1, h_2, \ldots, h_m\}$.

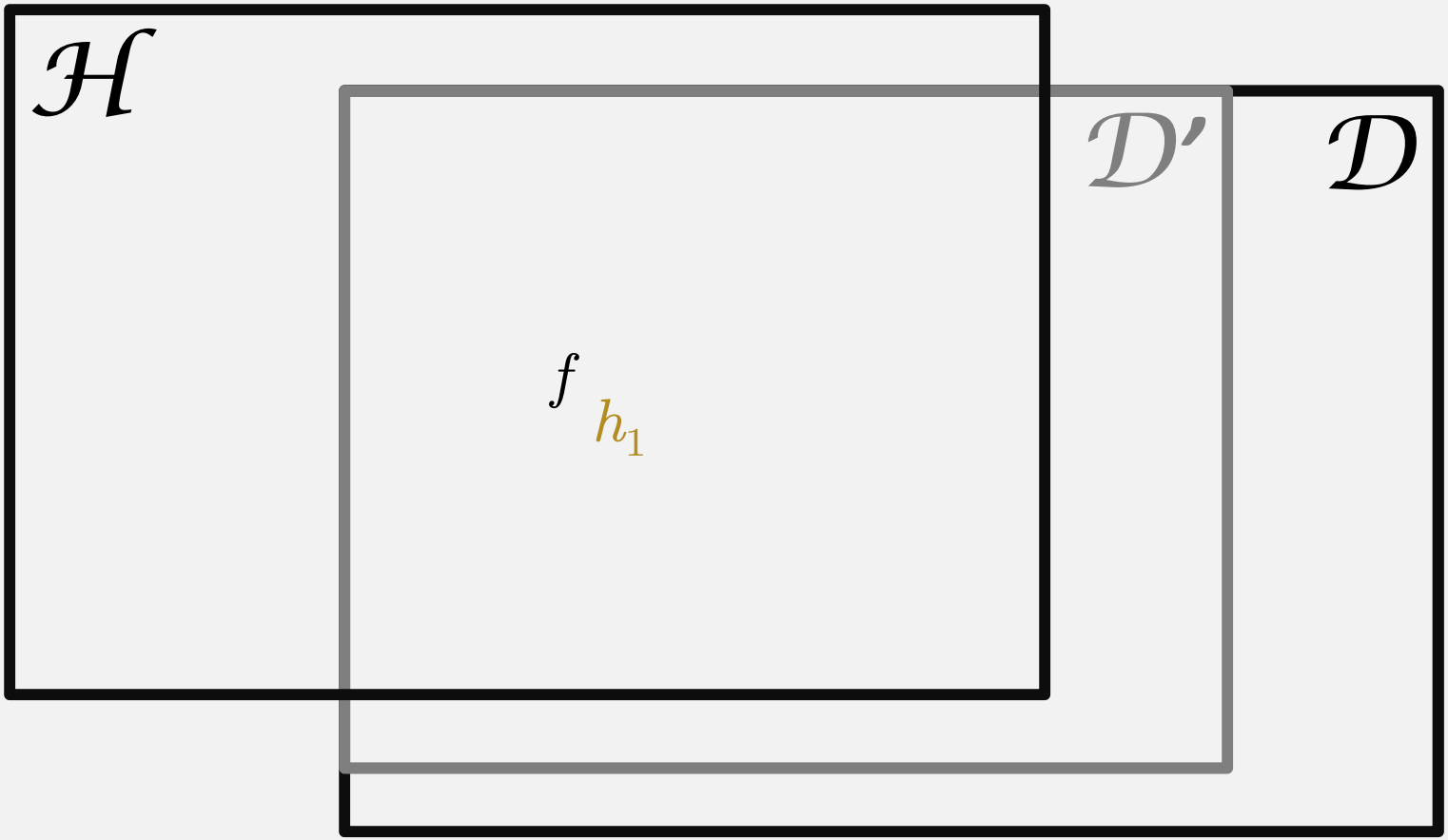
- With ensemble learning, we combine these models with an ensemble learning algorithm.

- Boosting: $\Sigma_{j=1:M}\, \alpha_j h_j(x)$ (weighted average).
- Bagging: $\arg\max_y \Sigma_{j=1:M}\, I(h_j(x)=y)$ (majority vote).
- Stacking: $w_1 h_1 + \cdots + w_{i\text{-}1} h_{i\text{-}1} + w_i h_i + \cdots + w_{j\text{-}1} h_{j\text{-}1} + w_j h_j + \cdots + w_m h_m$ (meta learning).

Why Ensemble Learning?

How is ensemble learning different?

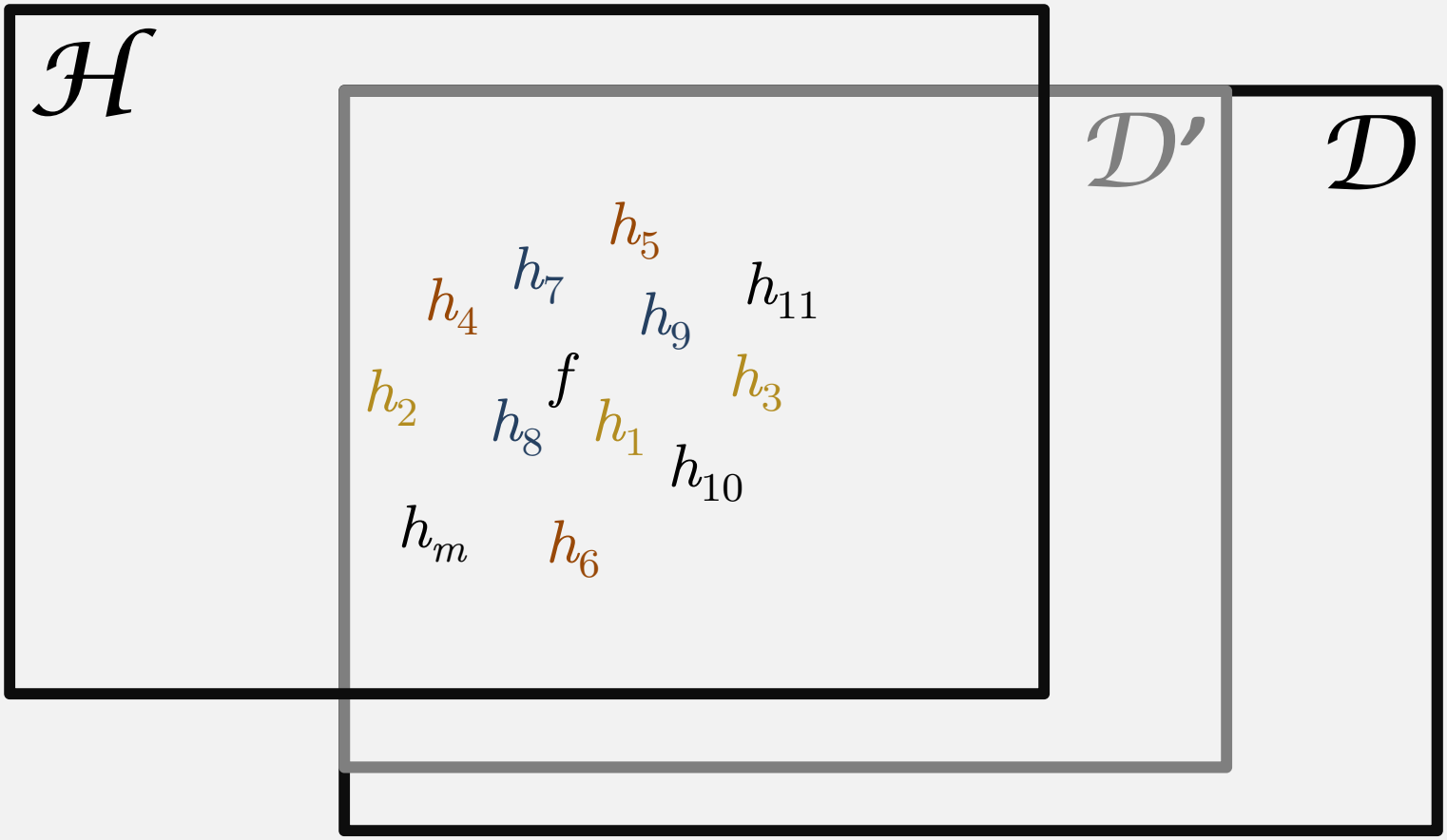
Traditional learning.



Why Ensemble Learning?

How is ensemble learning different?

Ensemble learning.



Why Ensemble Learning?
How is ensemble learning different?

- Can reduce *bias* by increasing the coverage of the hypothesis space with many diverse models.
- Can reduce *variance* by averaging the output of many diverse models.
- So what's the secret to ensemble learning?: Diversity.
- Diversity can be achieved by augmenting the input data with weights (boosting) and resampling (bagging), selecting different input variables (Random Forest), or simply trying many different models and combining them (stacking).

Boosting
Weak and Strong Learners

- A *weak learner* is slightly better than random.
- A *strong learner* closely predicts the target function.
- Can a group of weak learners perform as well as a single strong learner?

  Yes! "…these two notions of learnability are equivalent [Schapire, 1990]."

Boosting

AdaBoost

Input: Dataset$\{(x_1,y_1),(x_2,y_2),...,(x_n,y_n)\}$;

Learning algorithm (decision tree, ect.);

Number of base learners M;

Process:

1. Initialize all weights to $1/n$ (equal values).
2. Estimate learner with set of weights w.
3. Calculate error ϵ_j of learner h_j.
4. Check if learner is better than average.
5. Set weight α_j for each h_j.
6. Update weight w_i for each x_i .
7. Combine the M base learners.

Input: Dataset D, $y_i = \{\text{-}1,1\}$;

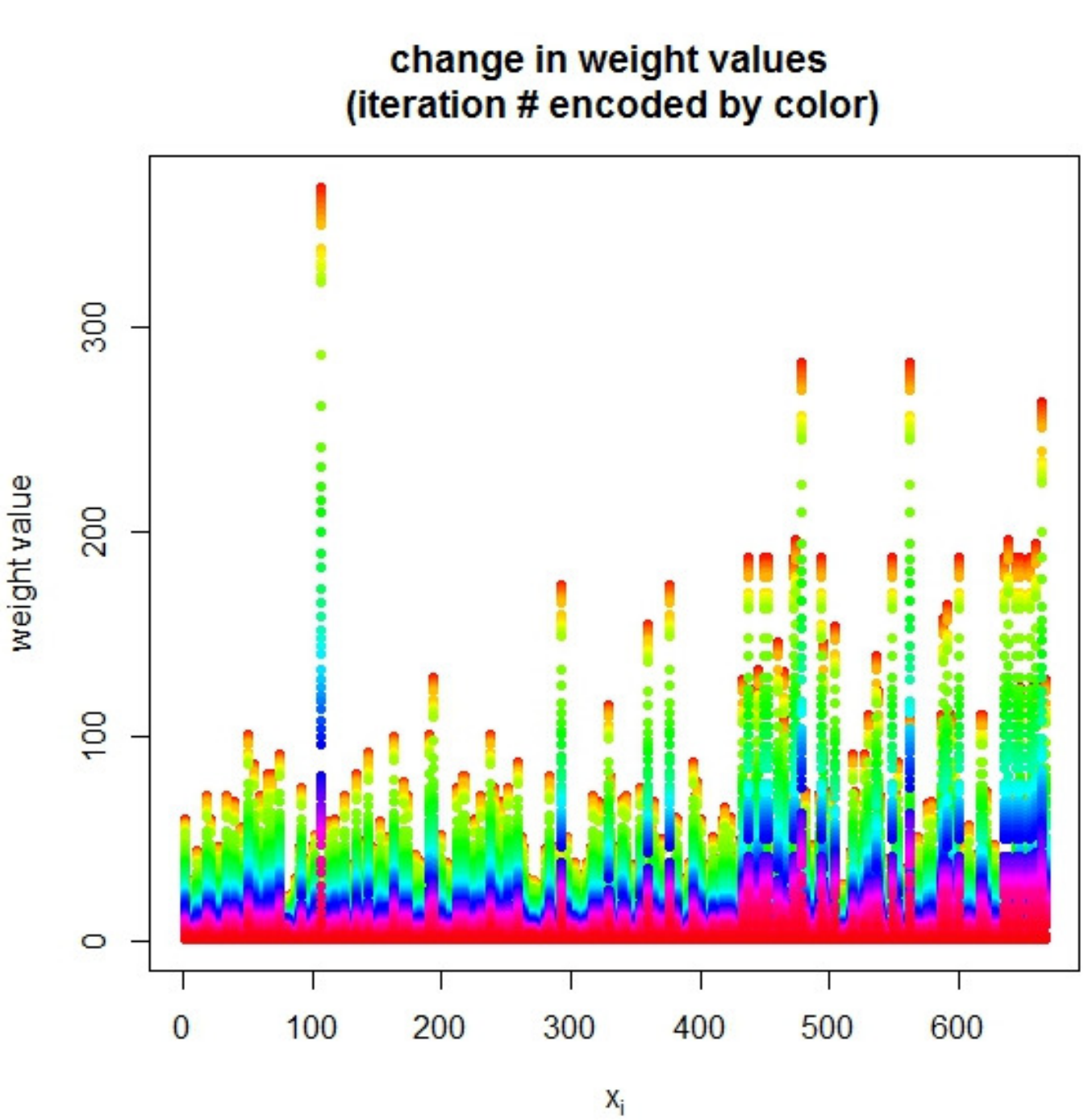
Learning algorithm $\mathcal{L}$;

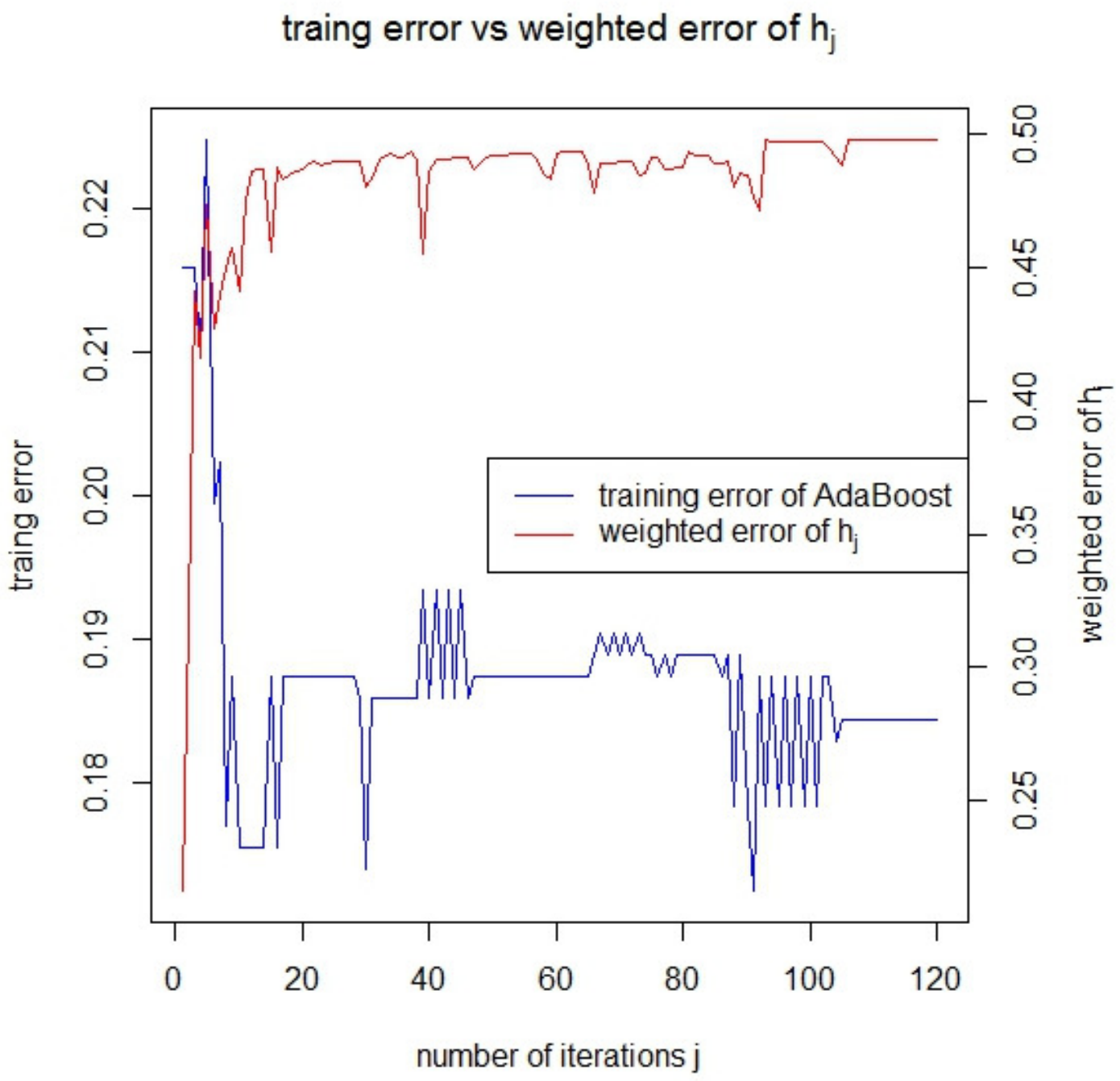
Set number of iterations to M;

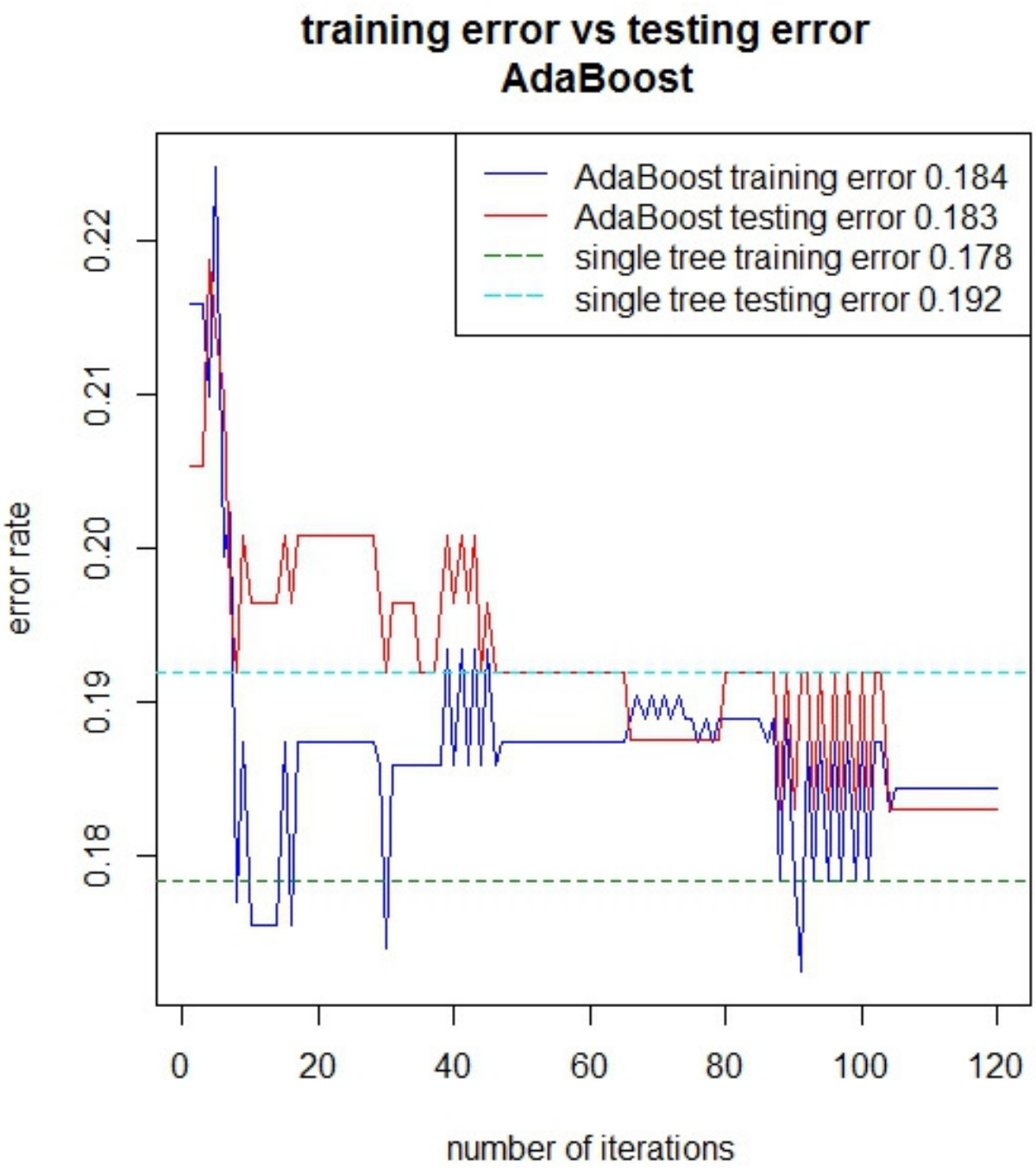
Process:

1. $w_i(x) = 1/n$
2. $h_j = \mathcal{L}(D,\ w)$ **for** j in $1{:}M$
3. $\epsilon_j = \Sigma_{i=1:n} w_i I(h_j(x_i) \neq y_i) / \Sigma_{i=1:n} w_i$
4. **If** $\epsilon_j > 0.5$ **break**
5. $\alpha_j = 0.5\ln(1\text{-}\epsilon_j\,/\epsilon_j)$
6. $w_i \leftarrow w_i \exp(\alpha_j I(h_j(x_i) \neq y_i))$
7. $H(x) = \text{sign}(\Sigma_{j=1:M}\ \alpha_j h_j(x))$

[Hastie, et. al.,2009] (Chapter 9), [Zhou, 2012] (Chapter 2)


change in weight values
(iteration # encoded by color)
weight value
0
100
200
300
0
100
200
300
400
500
600
x_i


traing error vs weighted error of h_j
traing error
weighted error of h_j
number of iterations j
training error of AdaBoost
weighted error of h_j
0.18
0.19
0.20
0.21
0.22
0.25
0.30
0.35
0.40
0.45
0.50
0
20
40
60
80
100
120


training error vs testing error
AdaBoost
AdaBoost training error 0.184
AdaBoost testing error 0.183
single tree training error 0.178
single tree testing error 0.192
error rate
0.18
0.19
0.20
0.21
0.22
0
20
40
60
80
100
120
number of iterations

Bagging
Bootstrapping and Aggregating

- Given a dataset of n observations $D = \{x_1, x_2, \ldots, x_n\}$, we can take n samples from D with **replacement** for a total of B bootstrap samples.
- $D = \{1,2,3,4,5\}$
  - $D_1^* = \{5,5,1,3,3\}$
  - $D_2^* = \{2,3,2,4,1\}$
  - $D_3^* = \{2,3,4,3,2\}$
  - $D_4^* = \{4,1,1,5,2\}$
  - $D_j^* = \ldots$
  - $D_B^* = \{4,4,1,5,3\}$

[L. Breiman, 1996], [Hastie, et. al.,2009] (Chapter 7), [Efron and Tibshirani, 1993] (Chapter 6)

Bagging
Bootstrapping and Aggregating

Input: Dataset $\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$

Learning algorithm (decision tree, ect.)

Number of base learners M

Process:

1. Estimate learner from bootstrap sample.
2. Combine the M base learners and evaluate using majority vote.

Input: Dataset D

Learning algorithm $\mathcal{L}$

Set number of iterations to M

Process:

1. $h_j = \mathcal{L}(D_j^*)$ **for** j in $1{:}M$
2. $H(x) = \arg\max_y \Sigma_{j=1:M} I(h_j(x)=y)$

[Hastie, et. al.,2009] (Chapter 8), [Zhou, 2012] (Chapter 3)

Bagging
Out-of-Bag Error

- For each learner, h_j, in the bagging ensemble, we keep track of the data points not in the bootstrap sample D_j^* and refer to them as the *out-of-bag* samples, $D_j^{*\text{oob}}$.
- The out-of-bag error is the proportion of predicted values for each data point in $D_j^{*\text{oob}}$ that is misclassified averaged over the entire data set D:

$$1/|D|\Sigma_j \; 1/|D_j^{*\text{oob}}|\Sigma_{(x,y)\epsilon D_j^{*oob}} \; I(h_j(x)\neq y)$$

Bagging
Random Forest

Input: Dataset $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$;
Attributes of x;
Random Subset f of features $\mathcal{F}$;

Process:

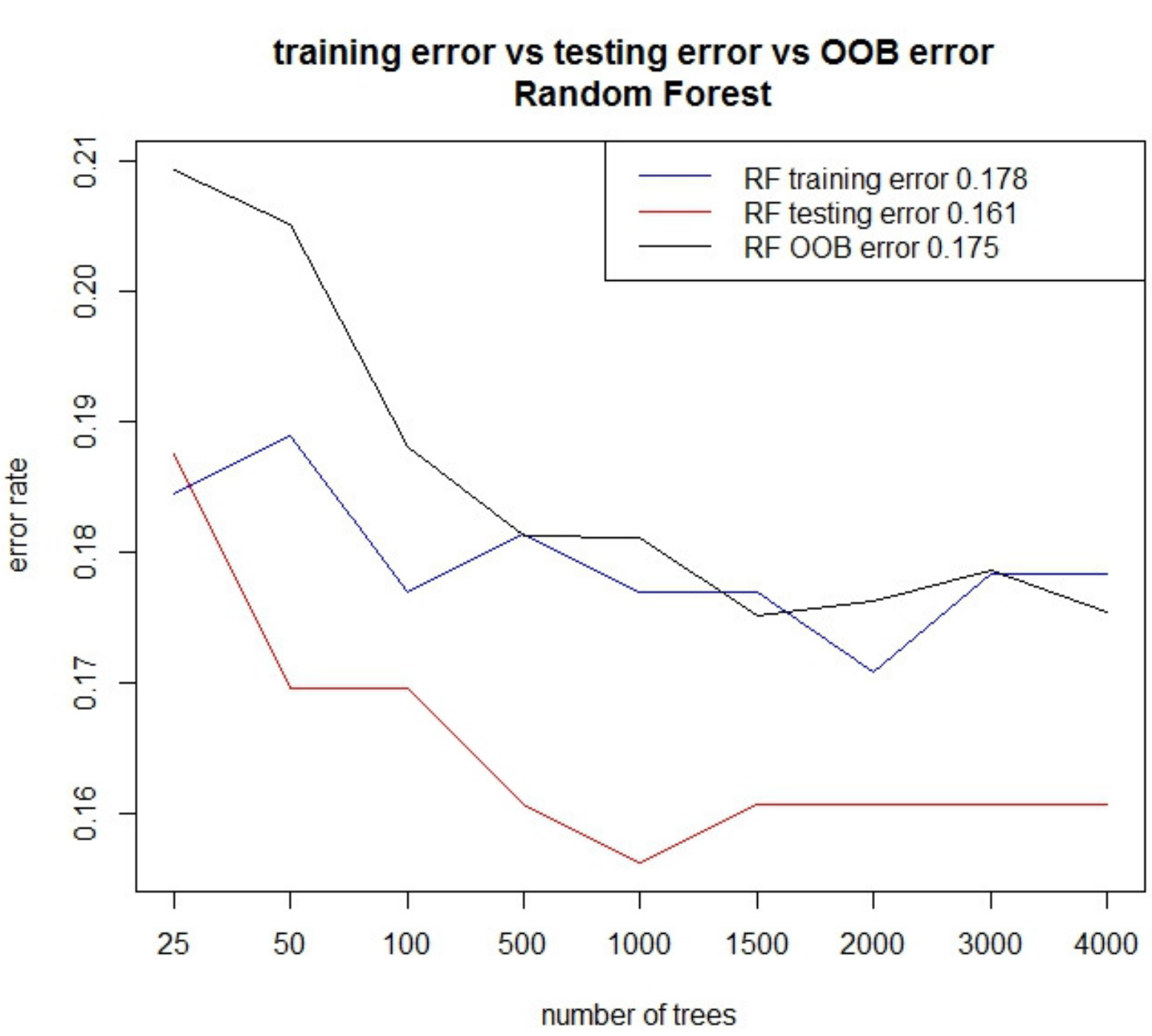
1. Create node i of tree T_j with D_j^* and f_i.
2. If everything is in the same class return h_j.
3. Create list $\mathcal{F}'$ of attributes that can be split.
4. If $\mathcal{F}'$ is empty then return tree h_j.
5. Select subset f_{i+1} from $\mathcal{F}'$ and grow tree.
6. Combine the M trees and evaluate using majority vote.

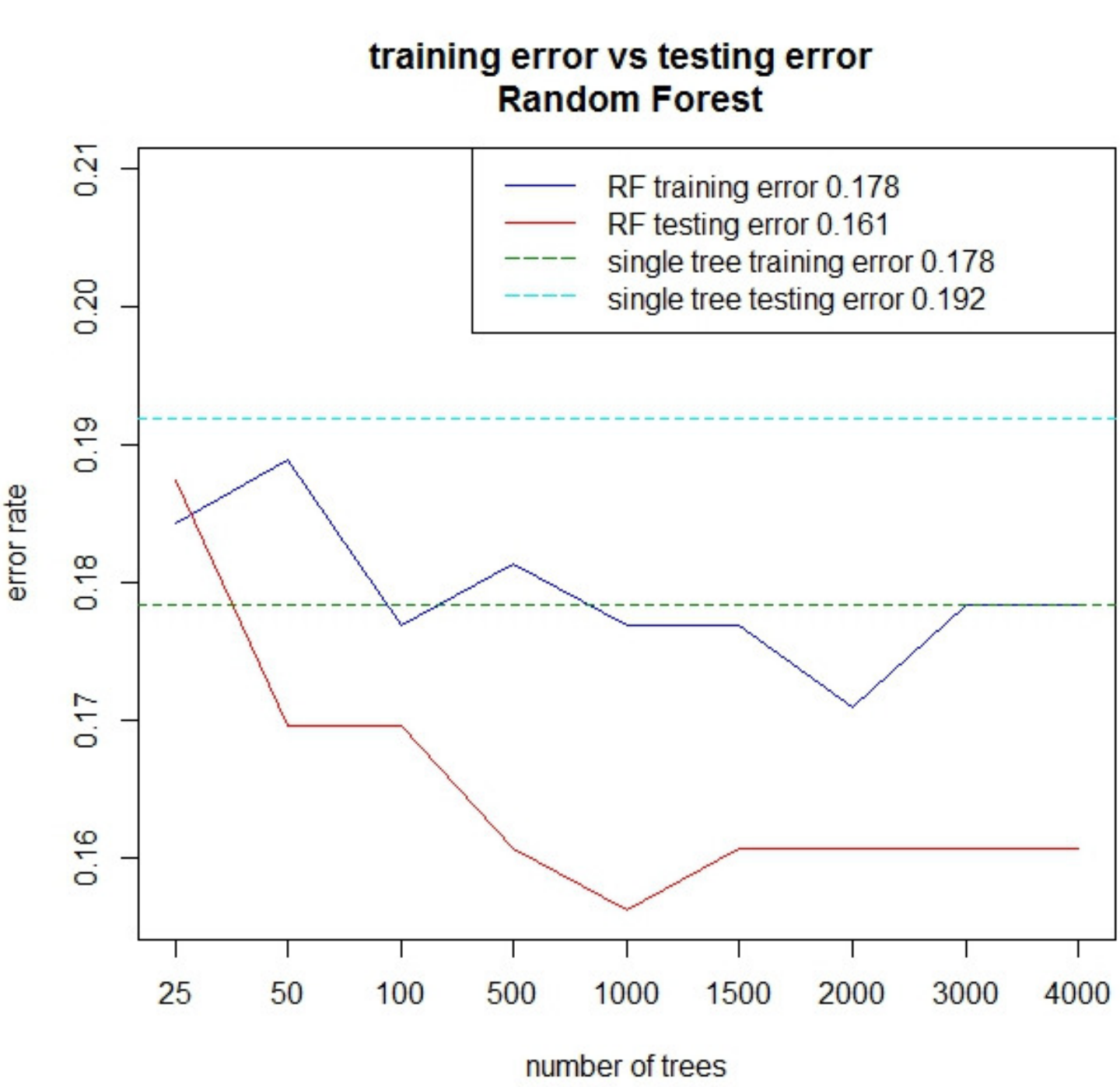
Input: Dataset D;
Set of features $\mathcal{F}$;
Random Subset f of features $\mathcal{F}$;

Process:

1. $h_j \leftarrow T_{ji}(D_j^*, f_i)$ **for** j in $1{:}M$
2. **If** $h_j(x_i^*) = y$ for all i **return** h_j
3. $\mathcal{F}' \leftarrow$ set of features left to split
4. **If** $\mathcal{F}'$ is NULL **return** h_j
5. select f_{i+1} from $\mathcal{F}'$ and repeat step 1.
6. $H(x) = \arg\max_y \Sigma_{j=1:M} I(h_j(x)=y)$

[Hastie, et. al.,2009] (Chapter 9), [Zhou, 2012] (Chapter 2)


training error vs testing error vs OOB error
Random Forest
RF training error 0.178
RF testing error 0.161
RF OOB error 0.175
error rate
0.16
0.17
0.18
0.19
0.20
0.21
25
50
100
500
1000
1500
2000
3000
4000
number of trees


training error vs testing error
Random Forest
RF training error 0.178
RF testing error 0.161
single tree training error 0.178
single tree testing error 0.192
error rate
0.16
0.17
0.18
0.19
0.20
0.21
25
50
100
500
1000
1500
2000
3000
4000
number of trees

Stacking
Beyond Averaging

Input: Dataset$\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$;
Different learning algorithms $\mathcal{L}_j$;
Learning algorithm $\mathcal{L}$ to combine $\mathcal{L}_j$;

Process:

1. Train learners h_j with algorithm $\mathcal{L}_j$.
2. predict on training dataset with trained learners.
3. Combine output from each h_j for each x_i.
4. Train learner h' with dataset D' and algorithm $\mathcal{L}$.
5. Combine output from base learners h_j as input for stacking learner h'.

Input: Base dataset D;
Set of base learning algorithm $\mathcal{L}_j$;
Stacking learning algorithm $\mathcal{L}$;

Process:

1. $h_j \leftarrow \mathcal{L}_j(D)$ **for** j in $1{:}M$
2. $z_j = h_j(x)$ **for** j in $1{:}M$
3. D'= $\{((z_{11},\ldots,z_{1M},)y_1),\ldots,((z_{1n},\ldots,z_{1M},),y_n)\}$
4. $h' \leftarrow \mathcal{L}(D')$
5. $H(x) = h'(h_1(x),\ldots,h_M(x)) = h'(z_1,\ldots,z_M)$.

[Hastie, et. al.,2009] (Chapter 9), [Zhou, 2012] (Chapter 2)

Advances and Special Applications

Proximity and Outlier Detection

- After building a Random Forest run all data (training and OOB data) down the trees and count the number of times each pair of data points fall within the same terminal node. This is the pairwise proximity.
- Comparing an observation's proximity to data points within the same class can detect outliers as data points with below average proximity.

Advances and Special Applications

Class imbalance

- Most real-world datasets do not have evenly distributed classes and some of the most interesting datasets contain a relatively small population for a specific class (among all classes).
- Balancing classes in the training set can be done by downsamlping. But this discards valuable training data.
- An ensemble of models can be created from many downsampled training sets.

[Zhou, 2012] (Chapter 2)

Why Ensemble Methods?

- “My first option with a dataset is almost always tree-based (boosted or bagged).”
  - Jose Guerrero

    #1 ranked on Kaggle

Conclusion

- The key ingredient in ensemble learning is creating diversity among the models: augmenting input, selecting input variables, or using completely different models.
- Understand the inherent biases of your constituent models and lower overall variance by increasing the number of ensemble models.
- Understand the quality and deficiencies of your input data, this can effect which ensemble method you will be able to use.

Thank You!

Questions?

Bibliography

Y.S. Abu-Mustafa , M. Magdon-Ismail and H.T. Lin. *Learning Form Data*: *A Short Course*. AMLBook, 2012.

B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

L. Breiman. Bagging Predictions. *Machine Learning*, 24(2):123-140, 1996.

L. Breiman. Random Forests. Machine Learning, 45(1):5-32, 2001.

Y. Freund and R.E. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting Journal of Computer and System Sciences, 55(1): 119-139, 1997.

T. Hastie, R. Tibshirani, J. Friedman. *The Elements of Statistical Learning*, 2nd ed. Springer, 2009.

T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

R.E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2): 197-227, 1990.

Z.H. Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC, 2012.

Further Reading/R packages

Boosting

G. Ridgeway. *Generalized Boosted Models: A guide to the GBM package*. 2007. Available online: http://cran.open-source-solution.org/web/packages/gbm/vignettes/gbm.pdf

J. Zhu, S. Rosset, H. Zou and T. Hastie. Multi-Class AdaBoost. Technical report, Department of Statistics, University of Michigan, Ann Arbor, MI, 2006.

R packages: 'ada', 'adabag' , 'gbm' and 'mboost'

Random Forest

R packages: 'randomForeast' ,' party' and 'Boruta' (feature selection with Random Forest)

Comparing different ensemble methods

T.G. Dietterich. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting and Randomization. *Machine Learning*. 40(2) : 139-157, 2000.