UPPSALA UNIVERSITET

- Decision trees
- Ensemble methods
  - Bagging
  - Random forests
  - (Gradient) Boosting

Machine learning – Block 2

Måns Magnusson
Department of Statistics, Uppsala University

Autumn 2023

UPPSALA UNIVERSITET

This week's lecture

- Trees
- Bagging
- Random Forest
- Boosting (Trees)

UPPSALA UNIVERSITET

- Decision trees
- Ensemble methods
  - Bagging
  - Random forests
  - (Gradient) Boosting

Assignment 1

1. Research ammanuens positions (https://swerik-project.github.io/)
2. Master Thesis project proposals
3. Assignment 1: Overall satisfaction

UPPSALA UNIVERSITET

Section 1

Decision trees

UPPSALA UNIVERSITET

Decision trees: basic idea

- A popular method that can be used for both classification and regression is decision trees.

UPPSALA UNIVERSITET

Decision trees: basic idea

- A popular method that can be used for both classification and regression is decision trees.
- Have you ever played the game "20 questions"?

UPPSALA UNIVERSITET

Decision trees: basic idea

- A popular method that can be used for both classification and regression is decision trees.
- Have you ever played the game "20 questions"?
- Decision trees is more or less that game!

UPPSALA UNIVERSITET

Decision trees: basic idea

- A popular method that can be used for both classification and regression is decision trees.
- Have you ever played the game "20 questions"?
- Decision trees is more or less that game!
- In the case of classification, the idea is to classify the new observation by
  1. Asking a questions
  2. Based on the previous answer, ask new question
  3. Questions are asked until a conclusion is reached

UPPSALA UNIVERSITET

Decision trees: basic idea

Name	Body temp	Gives birth	Legs	Class
Human	warm-blooded	yes	yes	mammal
Whale	warm-blooded	yes	no	mammal
Cat	warm-blooded	yes	yes	mammal
Cow	warm-blooded	yes	yes	mammal
Python	cold-blooded	no	no	reptile
Komodo dragon	cold-blooded	no	yes	reptile
Turtle	cold-blooded	no	yes	reptile
Salmon	cold-blooded	no	no	fish
Eel	cold-blooded	no	no	fish
Pigeon	warm-blooded	no	yes	bird
Penguin	warm-blooded	no	yes	bird

UPPSALA UNIVERSITET

Decision trees: basic idea

Name	Body temp	Gives birth	Legs	Class
Human	warm-blooded	yes	yes	mammal
Whale	warm-blooded	yes	no	mammal
Cat	warm-blooded	yes	yes	mammal
Cow	warm-blooded	yes	yes	mammal
Python	cold-blooded	no	no	reptile
Komodo dragon	cold-blooded	no	yes	reptile
Turtle	cold-blooded	no	yes	reptile
Salmon	cold-blooded	no	no	fish
Eel	cold-blooded	no	no	fish
Pigeon	warm-blooded	no	yes	bird
Penguin	warm-blooded	no	yes	bird

Classify Komodo dragon with a decision tree:

1. Does it give live birth?

UPPSALA UNIVERSITET

Decision trees: basic idea

Name	Body temp	Gives birth	Legs	Class
Human	warm-blooded	yes	yes	mammal
Whale	warm-blooded	yes	no	mammal
Cat	warm-blooded	yes	yes	mammal
Cow	warm-blooded	yes	yes	mammal
Python	cold-blooded	no	no	reptile
Komodo dragon	cold-blooded	no	yes	reptile
Turtle	cold-blooded	no	yes	reptile
Salmon	cold-blooded	no	no	fish
Eel	cold-blooded	no	no	fish
Pigeon	warm-blooded	no	yes	bird
Penguin	warm-blooded	no	yes	bird

Classify Komodo dragon with a decision tree:

1. Does it give live birth? (No!)
2. Is it warm-blooded?

UPPSALA UNIVERSITET

Decision trees: basic idea

Name	Body temp	Gives birth	Legs	Class
Human	warm-blooded	yes	yes	mammal
Whale	warm-blooded	yes	no	mammal
Cat	warm-blooded	yes	yes	mammal
Cow	warm-blooded	yes	yes	mammal
Python	cold-blooded	no	no	reptile
Komodo dragon	cold-blooded	no	yes	reptile
Turtle	cold-blooded	no	yes	reptile
Salmon	cold-blooded	no	no	fish
Eel	cold-blooded	no	no	fish
Pigeon	warm-blooded	no	yes	bird
Penguin	warm-blooded	no	yes	bird

Classify Komodo dragon with a decision tree:

1. Does it give live birth? (No!)
2. Is it warm-blooded? (No!)
3. Does it have legs?

UPPSALA UNIVERSITET

Decision trees: basic idea

Name	Body temp	Gives birth	Legs	Class
Human	warm-blooded	yes	yes	mammal
Whale	warm-blooded	yes	no	mammal
Cat	warm-blooded	yes	yes	mammal
Cow	warm-blooded	yes	yes	mammal
Python	cold-blooded	no	no	reptile
Komodo dragon	cold-blooded	no	yes	reptile
Turtle	cold-blooded	no	yes	reptile
Salmon	cold-blooded	no	no	fish
Eel	cold-blooded	no	no	fish
Pigeon	warm-blooded	no	yes	bird
Penguin	warm-blooded	no	yes	bird

Classify Komodo dragon with a decision tree:

1. Does it give live birth? (No!)
2. Is it warm-blooded? (No!)
3. Does it have legs? (Yes!) $\rightarrow$ Reptile

Regression trees: The regions of a tree

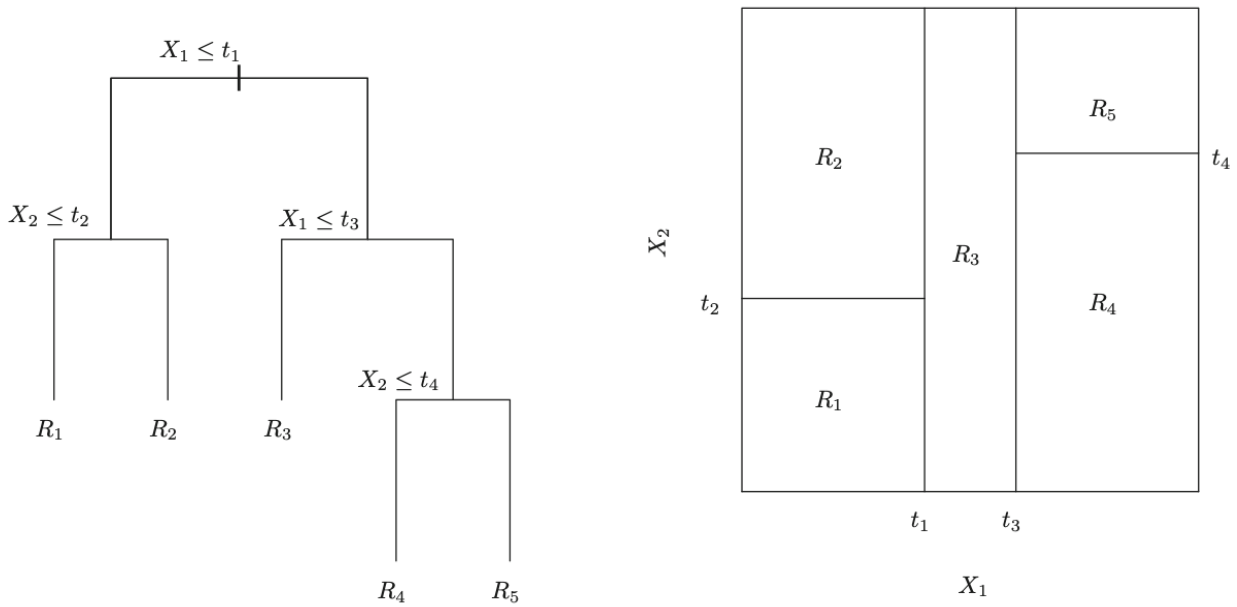

Figure: Regions of a tree (Garreth et al, 2013, Fig. 8.3)

UPPSALA UNIVERSITET

Regression Trees

$$T(x) = \sum_{m=1}^{M} \gamma_m I(x \in R_m),$$

where M is the total number of regions and $I(x \in R_m)$ is an indicator variable if x_i belongs to region R_m and γ_m is the prediction for region m.

UPPSALA UNIVERSITET

- Decision trees
- Ensemble methods
  - Bagging
  - Random forests
  - (Gradient) Boosting

Regression Trees

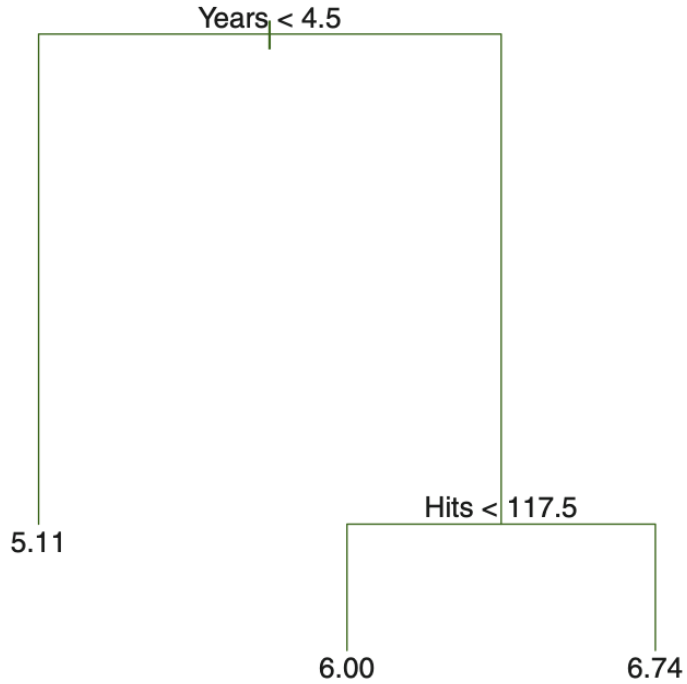

Figure: Regression Tree (Garreth et al, 2013, Fig. 8.1.)

- The `Hitters dataset`: log Salaries of Baseball players.
- The end of the tree contain the observations.

UPPSALA UNIVERSITET

- Decision trees
- Ensemble methods
  - Bagging
  - Random forests
  - (Gradient) Boosting

Regression Trees

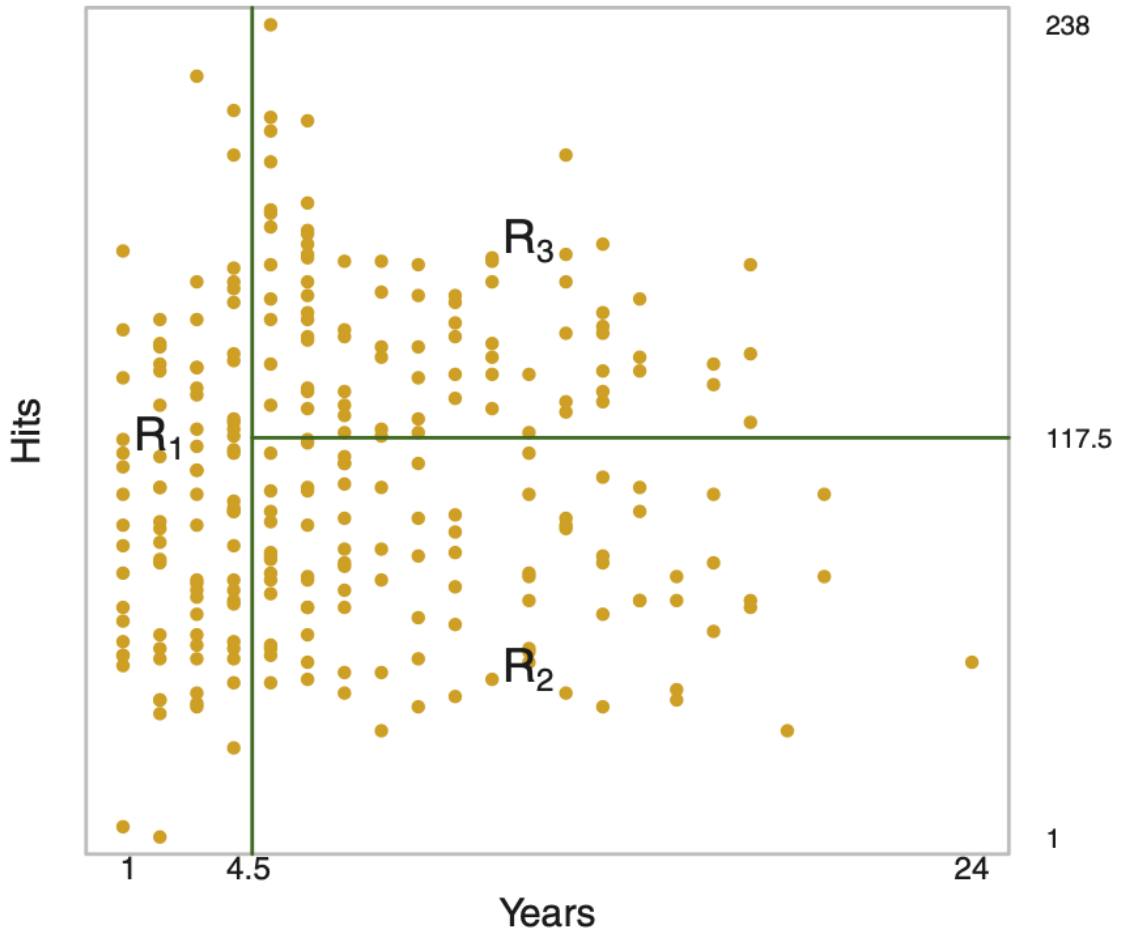

Figure: `Hitters` data and regression tree regions (Garreth et al, 2013, Fig. 8.2.)

UPPSALA UNIVERSITET

Estimating a Decision Tree

1. A tree has two groups of parameters $\Theta = (\gamma, R)$ that we need to learn.

UPPSALA UNIVERSITET

Estimating a Decision Tree

1. A tree has two groups of parameters $\Theta = (\gamma, R)$ that we need to learn.
2. We want a tree that minimize $L(\theta) = \sum_i^N (y_i - T_\Theta(x_i))^2$, here the squared loss

UPPSALA UNIVERSITET

Estimating a Decision Tree

1. A tree has two groups of parameters $\Theta = (\gamma, R)$ that we need to learn.
2. We want a tree that minimize $L(\theta) = \sum_i^N (y_i - T_\Theta(x_i))^2$, here the squared loss
3. Usually we estimate γ_m as the mean of y_i in the region as:

$$\hat{\gamma}_m = \frac{1}{N_m} \sum_{x_i \in R_m}^{N_m} y_i \, ,$$

where N_m is the number of observations in region R_m.

UPPSALA UNIVERSITET

Estimating a Decision Tree

1. A tree has two groups of parameters $\Theta = (\gamma, R)$ that we need to learn.
2. We want a tree that minimize $L(\theta) = \sum_i^N (y_i - T_\Theta(x_i))^2$, here the squared loss
3. Usually we estimate γ_m as the mean of y_i in the region as:

$$\hat{\gamma}_m = \frac{1}{N_m} \sum_{x_i \in R_m}^{N_m} y_i \, ,$$

   where N_m is the number of observations in region R_m.
4. Learning R_m exact is generally computationally infeasable so we use a greedy heuristic.

UPPSALA UNIVERSITET

Growing a Decision Tree: Greedy Algorithm

Let $\mathcal{S}$ be the set of all observations $\{1, ..., N\}$ and `S[[m]]` be the set of observation indecies in R_m and `l` is the minimal number of leafs per node.

Input: $\mathcal{S}$, `X`, `y`, `l`

1. `S[[1]]` $= \mathcal{S}$,`M = 1, m = 1`
2. `while m <= M` then do:
   - 2.1 `if(size(S[[m]]) >= 2*l)`
     - 2.1.1 `S[[M+1]], S[[M+2]], j[m], s[m] = split_tree(X[S[[m]],], y[S[[m]],],l)`
     - 2.1.2 `M = M + 2`
   - 2.2 `else`
     - 2.2.1 compute $\hat{\gamma}$ for `S[[m]]`
   - 2.3 `m = m + 1`

Output: `j`, `s`, γ

Example of a tree: `j = {Years, Hits}`, `s = {4.5, 117.5}`, $\hat{\gamma} = \{122, 317, 245\}$

UPPSALA UNIVERSITET

How to do a split?

Here we try to compute Eq. (9.12-9.14) in ESL:

$$R_1(j,s) = \{X|X_j \leq s\} \text{ and } R_2(j,s) = \{X|X_j > s\}$$

$$\min_{j,s}\left(\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2 + \min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right)$$

Inner minimization is solved by:

$$\hat{c}_m = \frac{1}{N_m}\sum_{x_i\in R_m}^{N_m} y_i\ ,$$

UPPSALA UNIVERSITET

How to do a split? Pseudo-code

Input: $\mathbf{X}, \mathbf{y}, l$

1. $SS = \text{Inf}$ # Store sum of squares in matrix of dim $P \times N_s$
2. $S = \text{Inf}$ # Store split point in matrix of dim $P \times N_s$
3. for $j \in \{1, ..., P\}$ # all features
   - 3.1 for $k \in \{1, ..., N_s\}$ # all observations in set s
     - 3.1.1 $s = x_{k,j}$ # Split point (use the value of x)
     - 3.1.2 if ($|R_1(s,j)| < l$ or $|R_2(s,j)| < l$) next # Dont create too few leaves
     - 3.1.3 $\hat{c}_1 = \frac{1}{|R_1(s,j)|} \sum_{x_i \in R_1(s,j)} y_i$
     - 3.1.4 $\hat{c}_2 = \frac{1}{|R_2(s,j)|} \sum_{x_i \in R_2(s,j)} y_i$
     - 3.1.5 $SS_{k,j} = \sum_{x_i \in R_1(s,j)} (y_i - c_1)^2 + \sum_{x_i \in R_2(s,j)} (y_i - c_2)^2$ # Compute Sum of Squares
     - 3.1.6 $S_{k,j} = \text{s}$
4. $k_{final}, j_{final} = \min_{k,j} SS$
5. $s_{final} = S_{k_{final}, j_{final}}$
6. return $R_1(s_{final}, j_{final}), R_2(s_{final}, j_{final}), s_{final}, j_{final}$

UPPSALA UNIVERSITET

Decision trees: Classification Trees

- How do we do if we have a classification tree?

UPPSALA UNIVERSITET

Decision trees: Classification Trees

- How do we do if we have a classification tree?
- We just change the loss function $L(\theta)$.

UPPSALA UNIVERSITET

Decision trees: Classification Trees

- How do we do if we have a classification tree?
- We just change the loss function $L(\theta)$.
- Let $p(j|t)$ be the fraction of observations in class j at the node t and let J be the number of classes.

UPPSALA UNIVERSITET

Decision trees: Classification Trees

- How do we do if we have a classification tree?
- We just change the loss function $L(\theta)$.
- Let $p(j|t)$ be the fraction of observations in class j at the node t and let J be the number of classes.
- The Gini for node t is defined as

$$Gini(t) = 1 - \sum_{j=1}^{J} p(j|t)^2$$

UPPSALA UNIVERSITET

Decision trees: Classification Trees

- How do we do if we have a classification tree?
- We just change the loss function $L(\theta)$.
- Let $p(j|t)$ be the fraction of observations in class j at the node t and let J be the number of classes.
- The Gini for node t is defined as

$$Gini(t) = 1 - \sum_{j=1}^{J} p(j|t)^2$$

- Gini is a measure of "impurity". If all observations belong to the same class, then

$$Gini(t) = 1 - 1^2 - 0 - \ldots - 0 = 0.$$

UPPSALA UNIVERSITET

Decision trees: Classification Trees

- How do we do if we have a classification tree?
- We just change the loss function $L(\theta)$.
- Let $p(j|t)$ be the fraction of observations in class j at the node t and let J be the number of classes.
- The Gini for node t is defined as

$$Gini(t) = 1 - \sum_{j=1}^{J} p(j|t)^2$$

- Gini is a measure of "impurity". If all observations belong to the same class, then

$$Gini(t) = 1 - 1^2 - 0 - \ldots - 0 = 0.$$

- The Gini is maximized when all classes have the same number of observations at t.

UPPSALA UNIVERSITET

Decision trees: Classification Trees

- How do we do if we have a classification tree?
- We just change the loss function $L(\theta)$.
- Let $p(j|t)$ be the fraction of observations in class j at the node t and let J be the number of classes.
- The Gini for node t is defined as

$$Gini(t) = 1 - \sum_{j=1}^{J} p(j|t)^2$$

- Gini is a measure of "impurity". If all observations belong to the same class, then

$$Gini(t) = 1 - 1^2 - 0 - \ldots - 0 = 0.$$

- The Gini is maximized when all classes have the same number of observations at t.
- One criterion for splitting could be to minimize the Gini in the next level of the tree. That way we will get "purer" nodes.

UPPSALA UNIVERSITET

Important concepts in trees

- Tree depth: the length of the longest path from the root to a leaf (i.e. greatest number of questions that the tree can ask).

UPPSALA UNIVERSITET

Important concepts in trees

- Tree depth: the length of the longest path from the root to a leaf (i.e. greatest number of questions that the tree can ask).
- Leaf size: the number of observations in a leaf.

UPPSALA UNIVERSITET

Important concepts in trees

- Tree depth: the length of the longest path from the root to a leaf (i.e. greatest number of questions that the tree can ask).
- Leaf size: the number of observations in a leaf.

Decision trees can become quite large, which may lead to:

- Overfitting (high variance)
- Difficulties interpreting the tree

UPPSALA UNIVERSITET

Important concepts in trees

- Tree depth: the length of the longest path from the root to a leaf (i.e. greatest number of questions that the tree can ask).
- Leaf size: the number of observations in a leaf.

Decision trees can become quite large, which may lead to:

- Overfitting (high variance)
- Difficulties interpreting the tree

The solution to this is

- Pruning: forcing the tree to be smaller by adding a stopping condition, e.g. a maximum depth or minimal leaf size.

UPPSALA UNIVERSITET

Important concepts in trees

- Tree depth: the length of the longest path from the root to a leaf (i.e. greatest number of questions that the tree can ask).
- Leaf size: the number of observations in a leaf.

Decision trees can become quite large, which may lead to:

- Overfitting (high variance)
- Difficulties interpreting the tree

The solution to this is

- Pruning: forcing the tree to be smaller by adding a stopping condition, e.g. a maximum depth or minimal leaf size.
- But decision trees are quite bad predition models...

UPPSALA UNIVERSITET

Section 2

Ensemble methods

UPPSALA UNIVERSITET

Wisdom of the crowd

Simulated example (Prize academy, see ESL):

1. 50 members vote in 10 categories, each with 4 nominations.

UPPSALA UNIVERSITET

Wisdom of the crowd

Simulated example (Prize academy, see ESL):

1. 50 members vote in 10 categories, each with 4 nominations.
2. For any category, only 15 voters have some knowledge ($p > 0.25$)

UPPSALA UNIVERSITET

Wisdom of the crowd

Simulated example (Prize academy, see ESL):

1. 50 members vote in 10 categories, each with 4 nominations.
2. For any category, only 15 voters have some knowledge ($p > 0.25$)
3. For each category, the 15 experts are chosen at random from the 50.

UPPSALA UNIVERSITET

- Decision trees
- Ensemble methods
  - Bagging
  - Random forests
  - (Gradient) Boosting

Wisdom of the crowd

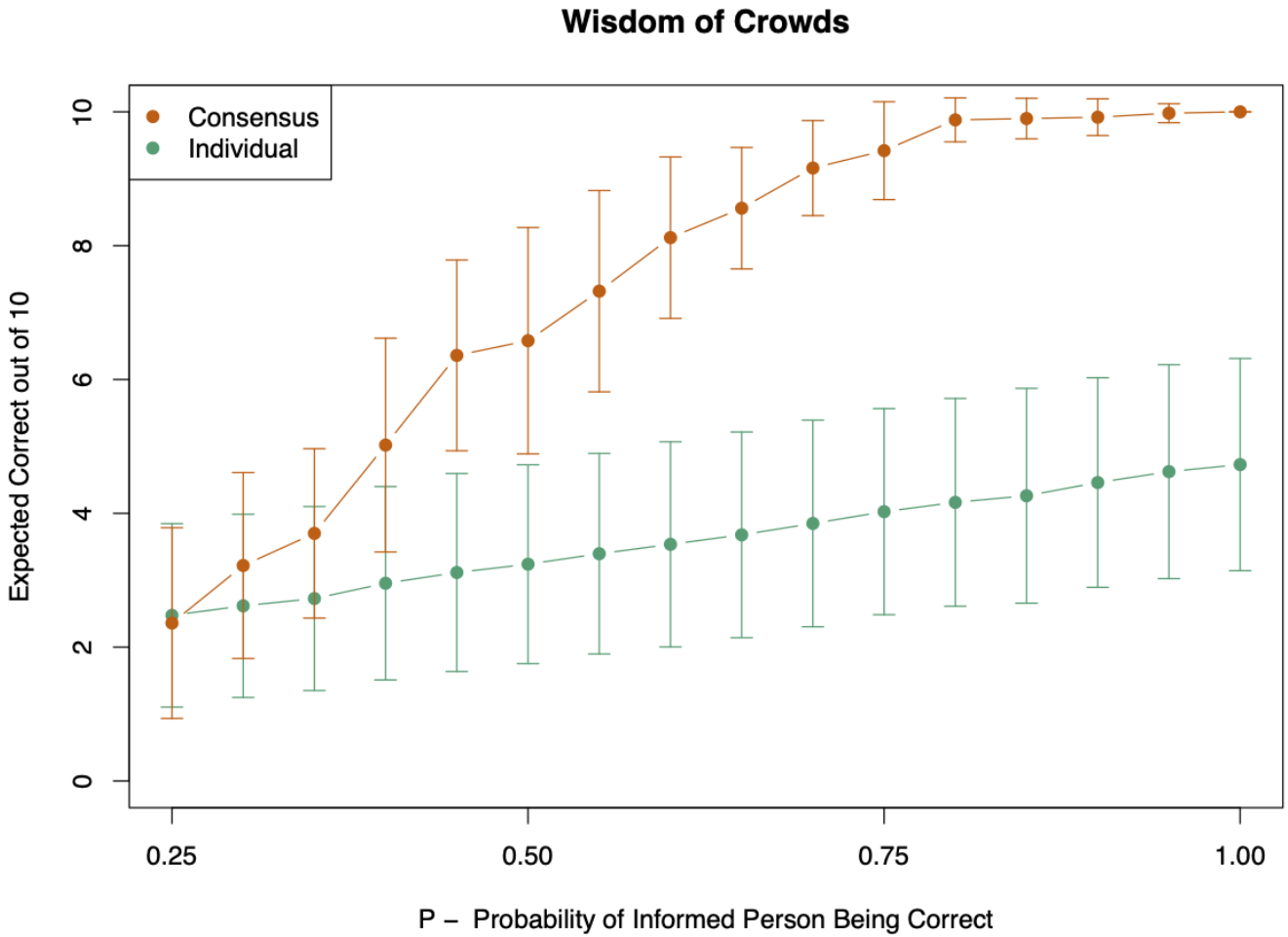

Figure: Simulated Award Voting, Fig. 8.11 (ESL)

UPPSALA UNIVERSITET

General idea of ensembles

The idea of an ensemble is simple: If it difficult to find one really good model perhaps we can find several weaker models and combine their predictions.

UPPSALA UNIVERSITET

General idea of ensembles

The idea of an ensemble is simple: If it difficult to find one really good model perhaps we can find several weaker models and combine their predictions.

A simple example: Say you have one outcome Y and 4 covariates X_1, X_2, X_3, X_4. The goal is to predict Y. A possible ensemble would be to fit

$$y = \alpha_1 + \beta_1 X_1 + \epsilon_1$$
$$y = \alpha_2 + \beta_2 X_2 + \epsilon_2$$
$$y = \alpha_3 + \beta_3 X_3 + \epsilon_3$$
$$y = \alpha_4 + \beta_4 X_4 + \epsilon_4$$

and then use the mean of their predictions

$$\hat{y}_{ensemble} = \frac{1}{4}\sum_{i=1}^{4}\hat{y} = \frac{1}{4}\sum_{i=1}^{4}(\hat{\alpha}_i + \hat{\beta}_i X_i) \tag{1}$$

UPPSALA UNIVERSITET

Two key parts of an ensemble

1. The prediction models (sometimes called 'learners')
   - A single model in an ensemble can be a simple or a complex model
   - Often the ensemble contains many simple models.

UPPSALA UNIVERSITET

Two key parts of an ensemble

1. The prediction models (sometimes called 'learners')
   - A single model in an ensemble can be a simple or a complex model
   - Often the ensemble contains many simple models.
2. The weighting of each prediction in the final ensemble prediction
   - Many different algorithms for weighting together the predictions from many models
   - Models/learners with better predictive power can be given larger weights in the final prediction

UPPSALA UNIVERSITET

Ensambles of decision trees

- A common type of ensembles is ensembles of decision trees.
- We will focus on this case, but note that any type of model can be included in an ensemble in principle.

UPPSALA UNIVERSITET

Bagging and Boosting

Remember, the error of a prediction/classification can be decomposed as

$$error = bias^2 + variance + bayeserror. \quad (2)$$

UPPSALA UNIVERSITET

Bagging and Boosting

Remember, the error of a prediction/classification can be decomposed as

$$error = bias^2 + variance + bayeserror. \quad (2)$$

- Complex models/strong learners (with many parameters) tend to have small bias and large variance (tend to be overfitted)

UPPSALA UNIVERSITET

Bagging and Boosting

Remember, the error of a prediction/classification can be decomposed as

$$error = bias^2 + variance + bayeserror. \quad (2)$$

- Complex models/strong learners (with many parameters) tend to have small bias and large variance (tend to be overfitted)
- Shallow models/weak learners (with few parameters) tend to have small variance and large bias

UPPSALA UNIVERSITET

Bagging and Boosting

Remember, the error of a prediction/classification can be decomposed as

$$error = bias^2 + variance + bayeserror. \tag{2}$$

- Complex models/strong learners (with many parameters) tend to have small bias and large variance (tend to be overfitted)
- Shallow models/weak learners (with few parameters) tend to have small variance and large bias

Bagging: Ensemble methods that aim to decrease the variance of complex/strong learners with low bias and large variance

Boosting: Ensemble methods that aim to decrease the bias of shallow/weak learners with low variance and large bias

UPPSALA UNIVERSITET

Subsection 1

Bagging

UPPSALA UNIVERSITET

Bagging (Bootstrap AGGregating)

- Train several deep trees and combine their results
- Use bootstrap to train different trees

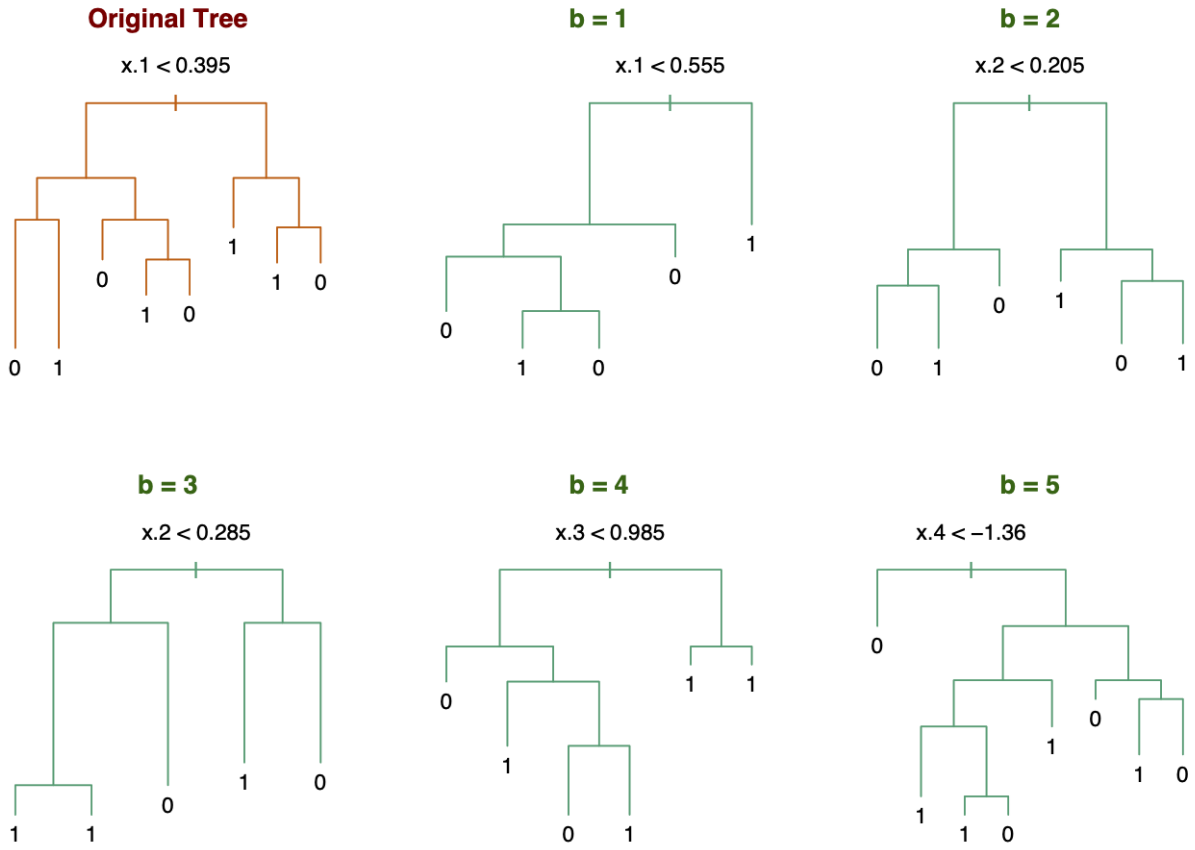

Figure: Bagging trees, Fig. 8.9 (ESL)

UPPSALA UNIVERSITET

Bagging (Bootstrap AGGregating)

1. Draw, *with replacement*, a random sample of N units from the original sample
2. Fit a prediction model (e.g., a deep decision tree)
3. Repeat steps 1-2 B times
4. Weight together the predictions from the B models into a final ensemble prediction as

$$\hat{f}_{\text{bag}}(x_i) = \frac{1}{B}\sum_b \hat{f}^b(x_i) = \frac{1}{B}\sum_b \hat{T}(x_i|\Theta_b)$$

UPPSALA UNIVERSITET

Subsection 2

Random forests

UPPSALA UNIVERSITET

Random Forest

- A bagging ensemble method, but...

UPPSALA UNIVERSITET

Random Forest

- A bagging ensemble method, but...
- Sample of N observations and K covariates/features

UPPSALA UNIVERSITET

Random Forest

- A bagging ensemble method, but...
- Sample of N observations and K covariates/features
- It is common to use $k = \sqrt{K}$ (rounded down) for classification and $k = K/3$ for regression. But these are only rules of thumb: k is a *tuning parameter*.

UPPSALA UNIVERSITET

Random Forest: Algorithm

1. Draw, *with replacement*, a random sample of N units from the original sample
2. **Draw, without replacement, a random subset of k covariates/features**
3. Fit a prediction model (e.g., a decision tree)
4. Repeat step 1-3 B times
5. Weight together the predictions from the B models into a final ensemble prediction

$$\hat{f}_{\text{rf}}(x_i) = \frac{1}{B} \sum_b \hat{T}(x_i | \Theta_b)$$

UPPSALA UNIVERSITET

Random Forest variance reduction

- Consider each tree to be an i.i.d. random variable with variance σ^2.
- The average of these trees then have variance

$$\mathbb{V}(\hat{f}(x) = \frac{1}{B}\sigma^2.$$

UPPSALA UNIVERSITET

Random Forest variance reduction

- Consider each tree to be an i.i.d. random variable with variance σ^2.
- The average of these trees then have variance

$$\mathbb{V}(\hat{f}(x) = \frac{1}{B}\sigma^2.$$

- Trees constructed from the same set of covariates will be correlated and therefore not independent.
- The variance of the average of these correlated trees then becomes

$$\mathbb{V}(\hat{f}(x)) = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2.$$

UPPSALA UNIVERSITET

Random Forest variance reduction

- The variance of the average of correlated trees:

$$\mathbb{V}(\hat{f}(x)) = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2.$$

- The second term will vanish with increasing B leaving just the first term left: a function of the correlation between the trees

UPPSALA UNIVERSITET

Random Forest variance reduction

- The variance of the average of correlated trees:

$$\mathbb{V}(\hat{f}(x)) = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2.$$

- The second term will vanish with increasing B leaving just the first term left: a function of the correlation between the trees
- The remaining part of the variance is minimized by only consider a subset of the covariates when constructing trees - reducing the correlation between trees.

UPPSALA UNIVERSITET

Bagging vs. Random forest

- In bagging, the trees are often highly correlated
  - If some covariates are strong predictors of the outcome (in the training data), many trees in the 'bag' will us the same covariates in their decisions

UPPSALA UNIVERSITET

Bagging vs. Random forest

- In bagging, the trees are often highly correlated
  - If some covariates are strong predictors of the outcome (in the training data), many trees in the 'bag' will us the same covariates in their decisions
- In a random forest, the trees are less similar/correlated since all covariates are not available when each tree is constructed.

UPPSALA UNIVERSITET

Bagging vs. Random forest

- In bagging, the trees are often highly correlated
  - If some covariates are strong predictors of the outcome (in the training data), many trees in the 'bag' will us the same covariates in their decisions
- In a random forest, the trees are less similar/correlated since all covariates are not available when each tree is constructed.
- Intuition: A random forest (with many trees) uses the predictive ability of all covariates rather than just a few $\rightarrow$ improved out of sample performance.

UPPSALA UNIVERSITET

Subsection 3

(Gradient) Boosting

UPPSALA UNIVERSITET

Boosting

1. In boosting, models/trees are trained sequentially

UPPSALA UNIVERSITET

Boosting

1. In boosting, models/trees are trained sequentially
2. Each new model tries to target weak spots of the previous models

UPPSALA UNIVERSITET

Boosting (AdaBoost)

1. Initialize weight for all observations as $w_i = n^{-1}$
2. Repeat B times
   - 2.1 Fit a simple classifier $h_b(x)$ using w_is
   - 2.2 Compute the weighted error (between 0-1) as

$$e_b = \frac{\sum w_i I(y_i \neq h_b(x_i))}{\sum w_i}$$

   - 2.3 Compute the importance of the classifier as

$$\alpha_b = \log((1 - e_b)/e_b) = \text{logit}(1 - e_b)$$

   - 2.4 Add the classifier to the ensemble

$$\hat{f}_b(x) = \hat{f}_{b-1}(x) + \alpha_b h_b(x)$$

   - 2.5 Update weights so that badly classified observations is weighted more

$$w_i \leftarrow w_i \exp(\alpha_b I(y_i \neq h_b(x_i)))$$

3. Output the final ensemble $\hat{f}_B(x)$

UPPSALA UNIVERSITET

- Decision trees
- Ensemble methods
  - Bagging
  - Random forests
  - (Gradient) Boosting

Boosting example

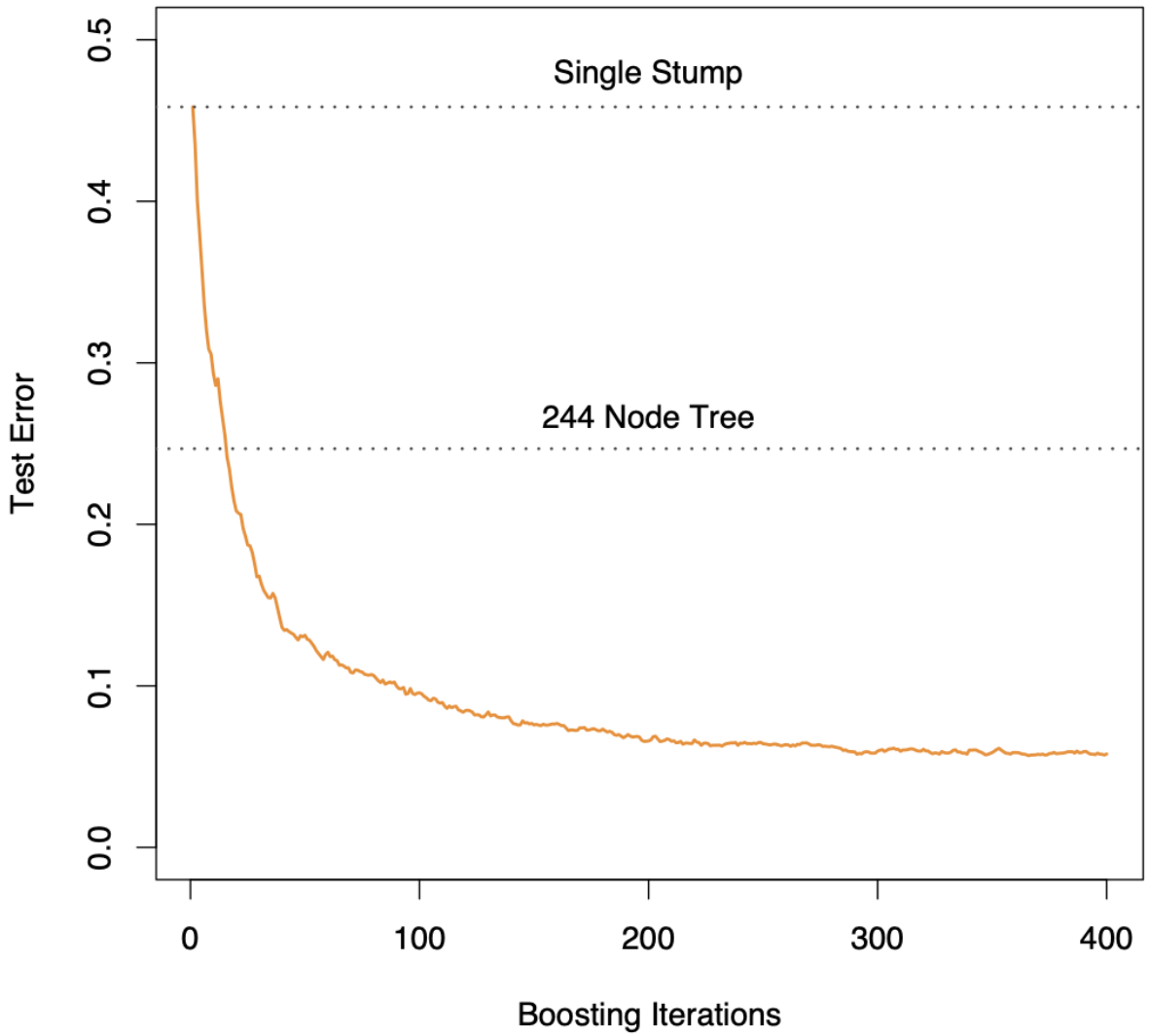

Figure: Boosting example (using stumps as h_b), Fig. 10.2 (ESL)

UPPSALA UNIVERSITET

- Decision trees
- Ensemble methods
  - Bagging
  - Random forests
  - (Gradient) Boosting

Boosting example

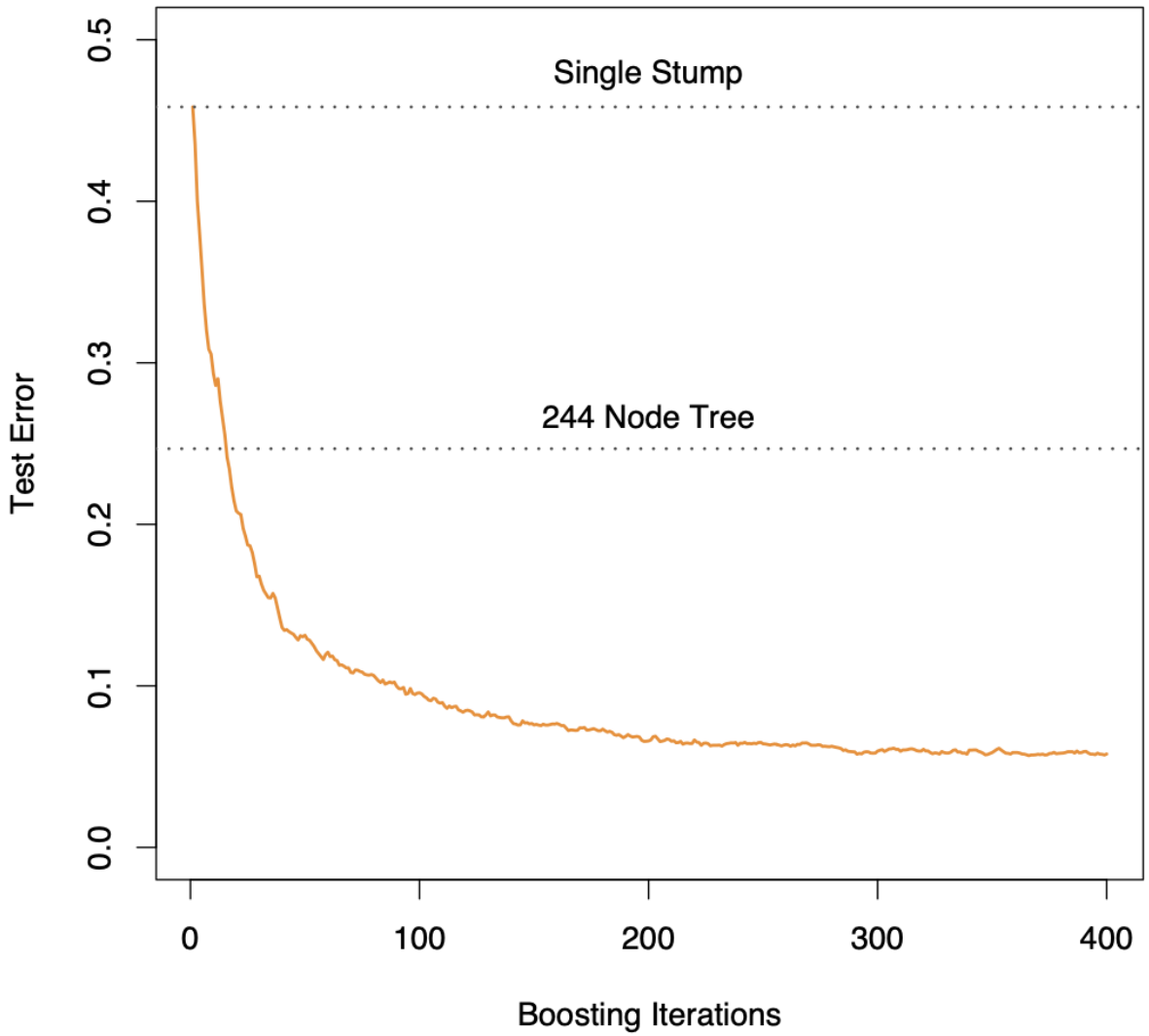

Figure: Boosting example (using stumps as h_b), Fig. 10.2 (ESL)

UPPSALA UNIVERSITET

Boosting trees

- A more general approach is (gradient) boosting trees

UPPSALA UNIVERSITET

Boosting trees

- A more general approach is (gradient) boosting trees
- Let

$$f_{\text{gb}}(x) = \sum_{b}^{B} \hat{T}(x|\Theta_b)$$

UPPSALA UNIVERSITET

Boosting trees

- A more general approach is (gradient) boosting trees
- Let

$$f_{\text{gb}}(x) = \sum_{b}^{B} \hat{T}(x|\Theta_b)$$

- We want to minimize the loss

$$L(y, f_{\text{gb}}(x))$$

UPPSALA UNIVERSITET

Boosting trees

- A more general approach is (gradient) boosting trees
- Let

$$f_{\text{gb}}(x) = \sum_{b}^{B} \hat{T}(x|\Theta_b)$$

- We want to minimize the loss

$$L(y, f_{\text{gb}}(x))$$

- This means finding

$$\Theta_b = \arg\min \sum_{i}^{N} L(y_i, f_{b-1}(x_i) + \hat{T}(x|\Theta_b))$$

UPPSALA UNIVERSITET

Gradient Boosting Trees

1. Initialize $f_0(x)$
2. Repeat B times
   2.1 For $i = 1, 2, ..., N$ compute

$$r_{ib} = -\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}$$

   2.2 Compute a regression tree $\hat{T}(x|\Theta_b)$ on r_b
   2.3 Add the classifier to the ensemble

$$\hat{f}_b(x) = \hat{f}_{b-1}(x) + \hat{T}(x|\Theta_b)$$

3. Output the final ensemble $\hat{f}_B(x)$

UPPSALA UNIVERSITET

- Decision trees
- Ensemble methods
  - Bagging
  - Random forests
  - (Gradient) Boosting

Comparisons

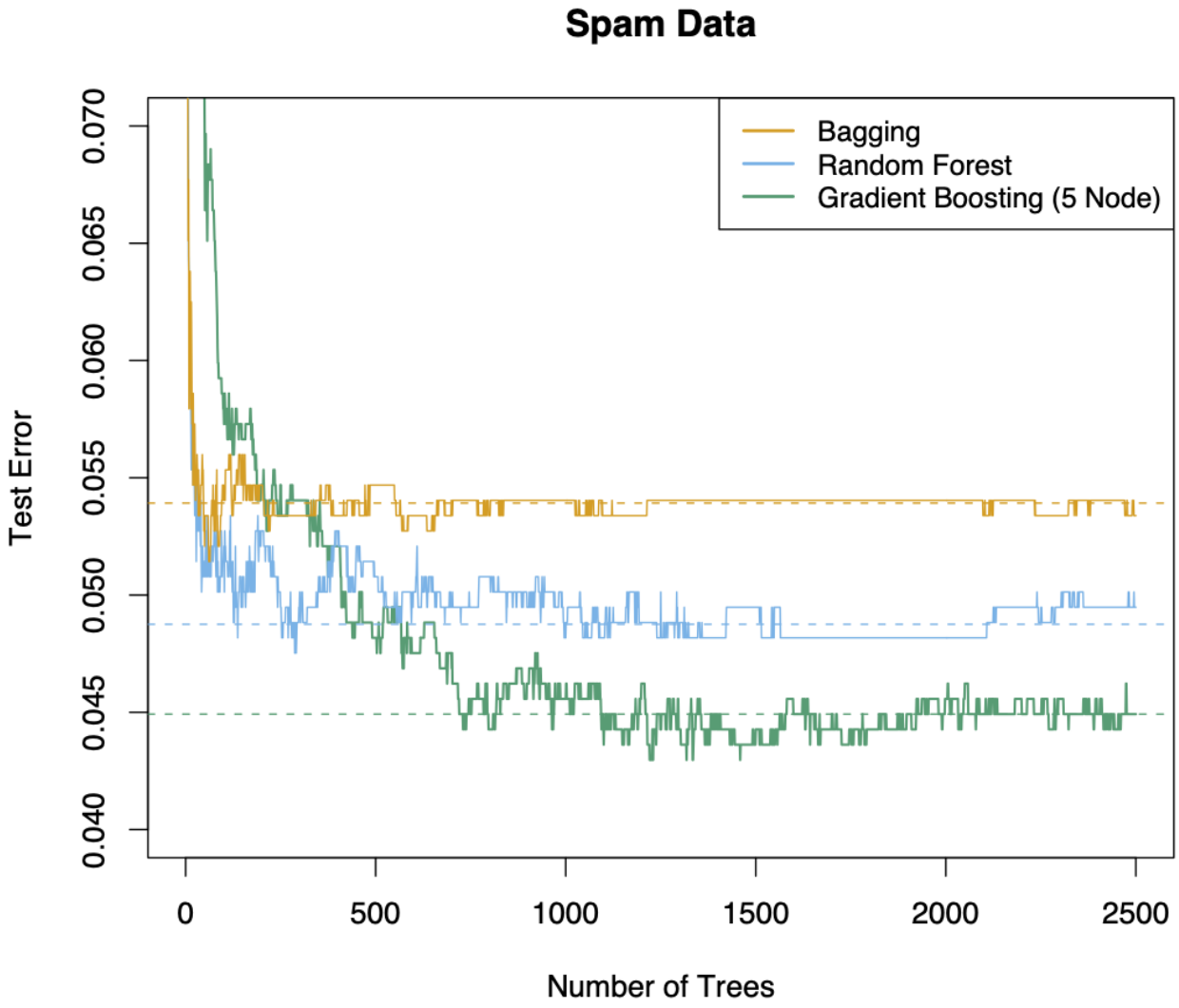

Figure: Comparing bagging, random forests and boosting, Fig. 15.1 (ESL)

UPPSALA UNIVERSITET

XGBoost

- State of the Art method

UPPSALA UNIVERSITET

XGBoost

- State of the Art method
- Use gradient boosting trees with regularization

$$L(y, f_{\text{boost}}(x)) + \sum_b \Omega(\hat{T}(x_i|\Theta_b))$$

UPPSALA UNIVERSITET

XGBoost

- State of the Art method
- Use gradient boosting trees with regularization

$$L(y, f_{\text{boost}}(x)) + \sum_b \Omega(\hat{T}(x_i|\Theta_b))$$

where

$$\Omega(\hat{T}(x_i|\Theta_b)) = \lambda_1 Q_b + \lambda_2 ||\gamma_b||_2^2$$

where Q_b is the number of leafs in tree b and $||\cdot||_2$ is the Euclidian/L^2 norm.

UPPSALA UNIVERSITET

XGBoost

- State of the Art method
- Use gradient boosting trees with regularization

$$L(y, f_{\text{boost}}(x)) + \sum_b \Omega(\hat{T}(x_i|\Theta_b))$$

where

$$\Omega(\hat{T}(x_i|\Theta_b)) = \lambda_1 Q_b + \lambda_2 ||\gamma_b||_2^2$$

where Q_b is the number of leafs in tree b and $||\cdot||_2$ is the Euclidian/L^2 norm.

- Is scalable to very large data