Lecture 6 – Tree-based methods, Bagging and Random Forests

UPPSALA
UNIVERSITET

Fredrik Lindsten
Division of Systems and Control
Department of Information Technology
Uppsala University.

Email: fredrik.lindsten@it.uu.se

Summary of Lecture 5 (I/III)

In **selecting** models/methods we are interested in their **generalization** performance obtained by investigating their predictive capabilities on **unseen test** data.

The **expected test MSE**

$$\mathbb{E}\left[\left(y_\star - \widehat{f}(x_\star; \mathcal{T})\right)^2\right]$$

refers to the average test MSE obtained if we computed successive estimates of f using a large number of training dataset and test each at $x_\star$.

fredrik.lindsten@it.uu.se

Summary of Lecture 5 (II/III)

$$\text{MSE} = (\text{Bias})^2 + \text{Variance} + \text{irreducible error}$$

$$\underbrace{\mathbb{E}\left[(y_\star - \hat{f}(x_\star;\mathcal{T}))^2\right]}_{\text{Expected test MSE}} = \underbrace{\left(\mathbb{E}\left[\hat{f}(x_\star;\mathcal{T})\right] - f(x_\star)\right)^2}_{(\text{Bias})^2} + \underbrace{\mathbb{E}\left[\hat{f}(x_\star;\mathcal{T}) - \mathbb{E}\left[\hat{f}(x_\star;\mathcal{T})\right]\right]^2}_{\text{Variance}} + \underbrace{\text{Var}(\epsilon)}_{\text{irreducible error}}$$

The **bias–variance trade-off** is key to good generalization performance!

fredrik.lindsten@it.uu.se

Summary of Lecture 5 (III/III)

Cross-validation is a method for estimating the test MSE using the training data.

Two approaches:

1. **Validation set:** Randomly split the data into a **training set** and a **validation set**. Learn the model using the training set. Estimate the test MSE using the validation set.
2. k**-fold cross-validation:** Randomly split the data into k parts (or **folds**) of roughly equal size.
   a) The first fold is treated as a validation set and the model is learned using the remaining $k-1$ folds. MSE_1 is estimated using the validation set.
   b) The procedure is repeated k times, each time a different fold is treated as the validation set.
   c) Resulting estimate of the test MSE:

$$\text{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\text{MSE}_i.$$

fredrik.lindsten@it.uu.se

Contents – Lecture 6

1. Classification and regression trees (CART)
2. Bagging – *a general variance reduction technique*
3. Random forests

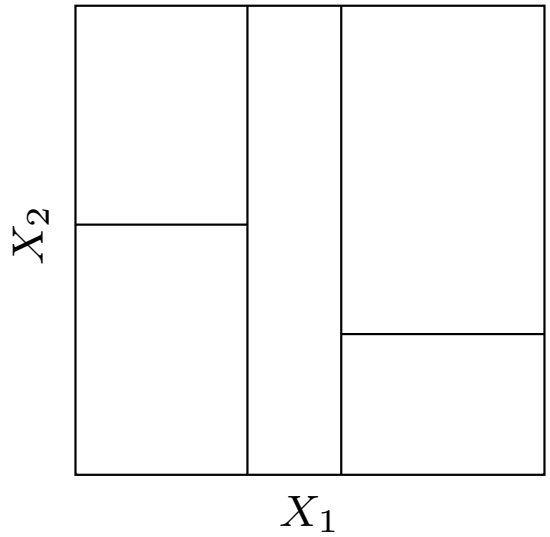
Note: mid-course evaluation available in the student portal. Open until Friday. Please fill it out, your feedback is much appreciated!

fredrik.lindsten@it.uu.se

The idea behind tree-based methods

In both regression and classification settings we seek a function, $\widehat{f}(x)$ or $\widehat{G}(x)$ respectively, which maps the input x into a prediction.

One **flexible** way of designing this function is to partition the input space into disjoint regions and fit a simple model in each region.



- **Classification:** Majority vote within the region.
- **Regression:** Mean of training data within the region.

fredrik.lindsten@it.uu.se

Finding the partition

The key challenge in using this strategy is to find a good partition.

Even if we restrict our attention to seemingly simple regions (e.g. "boxes"), finding an ***optimal*** partition w.r.t. minimizing the training error is ***computationally infeasible!***

Instead, we use a "greedy" approach: **recursive binary splitting**.

1. Select one of the inputs X_j ($j \in \{1, \ldots, p\}$) and a cut-point s. Partition the input space into two half-spaces,

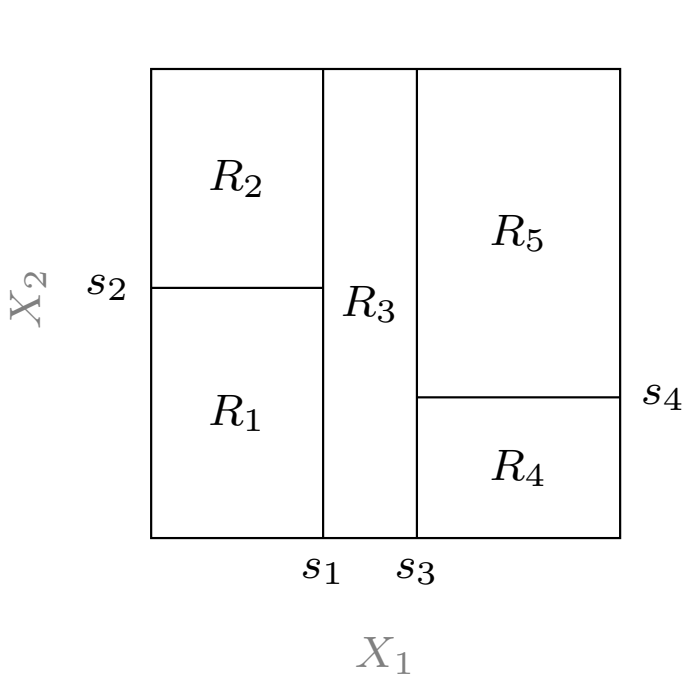
$$\{X : X_j < s\} \qquad\qquad \{X : X_j \geq s\}.$$

2. Repeat this splitting for each region until some stopping criterion is met (e.g., no region contains more than 5 training data points).

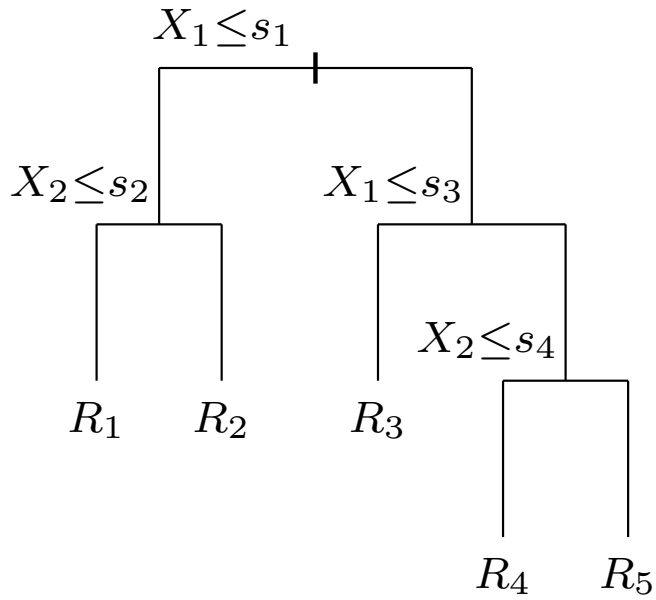
fredrik.lindsten@it.uu.se

Recursive binary splitting

Partitioning of input space



Tree representation



fredrik.lindsten@it.uu.se

Classification trees

Classification trees are constructed similarly to regression trees, but with *two differences*.

Firstly, the class prediction for each region is based on a majority vote within that region. Let

$$\widehat{p}_{mk} = \frac{1}{n_m} \sum_{i:x_i \in R_m} I(y_i = k)$$

be the proportion of training observations in the mth region that belong to the kth class. Then,

$$\widehat{G}(x) = \sum_{m=1}^{M} \widehat{c}_m I(x \in R_m),$$

where $\widehat{c}_m = \arg\max_k \widehat{p}_{km}$.

fredrik.lindsten@it.uu.se

Classification trees

Secondly, the squared loss used to construct the tree needs to be replaced by a measure suitable to qualitative outputs.

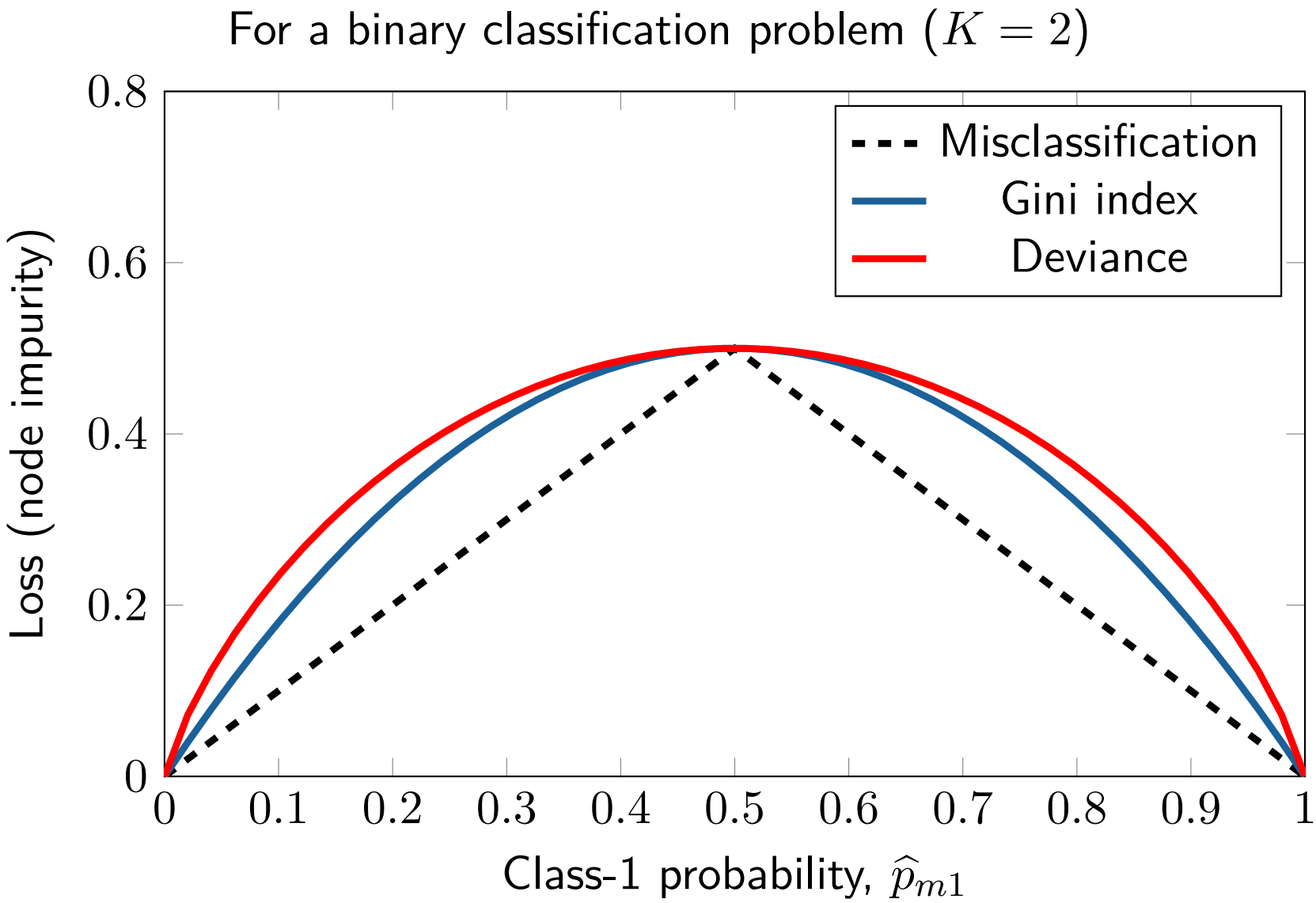
Three common error measures are,

$$\begin{aligned}
\text{Misclassification error:} \quad & 1 - \max_k \widehat{p}_{mk} \qquad [= 1 - \widehat{p}_{m\widehat{c}_m}] \\
\text{Gini index:} \quad & \textstyle\sum_{k=1}^{K} \widehat{p}_{mk}(1 - \widehat{p}_{mk}) \\
\text{Entropy/deviance:} \quad & \textstyle -\sum_{k=1}^{K} \widehat{p}_{mk} \log \widehat{p}_{mk}
\end{aligned}$$

Gini index and deviance are more sensitive to changes in the node probabilities than misclassification error and are often preferred in practice!

fredrik.lindsten@it.uu.se

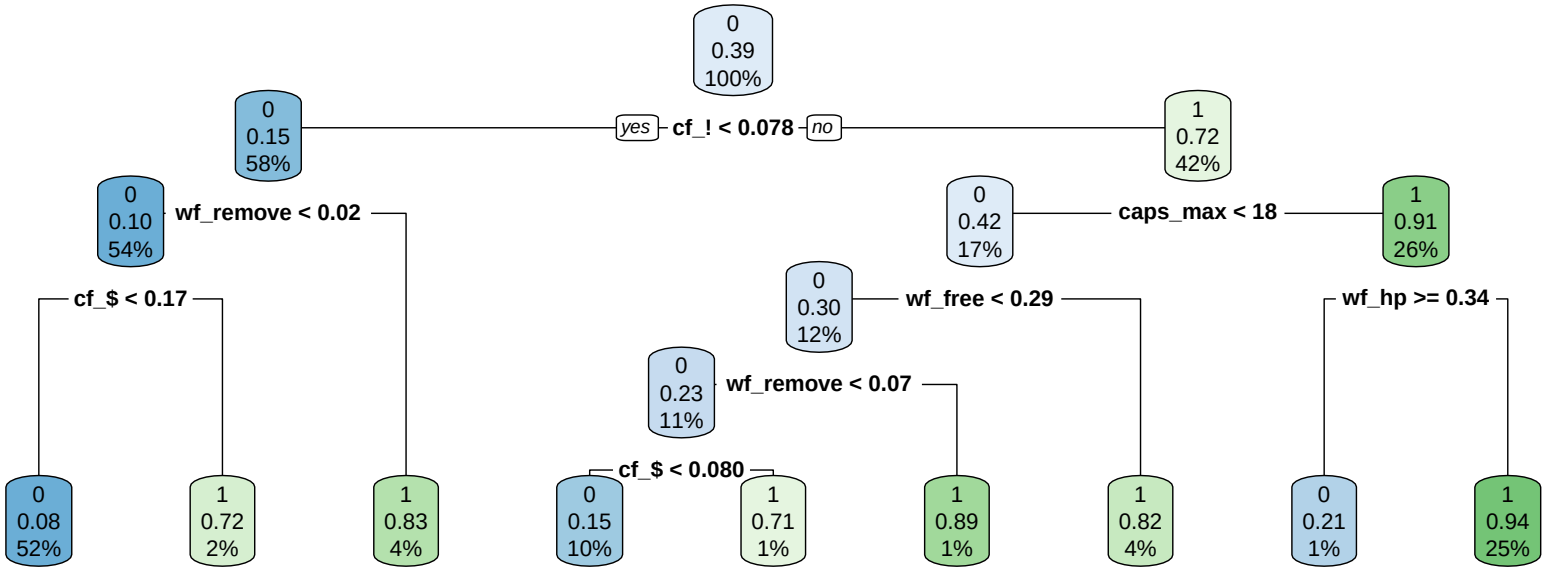
Classification error measures



fredrik.lindsten@it.uu.se

ex) Spam data

Classification tree (`rpart` in R) for spam data:



	Tree	LDA	Random Forest
Test error:	11.3 %	10.9 %	4.5 %

fredrik.lindsten@it.uu.se

Improving CART

The flexibility/complexity of classification and regression trees (CART) is decided by the tree depth.

- ! To obtain a **small bias** the tree need to be grown deep,
- ! but this results in a **high variance!**

The performance of (simple) CARTs is often unsatisfactory!

To improve the practical performance:

- **Pruning** – grow a deep tree (small bias) which is then pruned into a smaller one (reduce variance).
- **Ensemble methods** – average or combine multiple trees.
  - Bagging and Random Forests (this lecture)
  - Boosted trees (next lecture)

fredrik.lindsten@it.uu.se

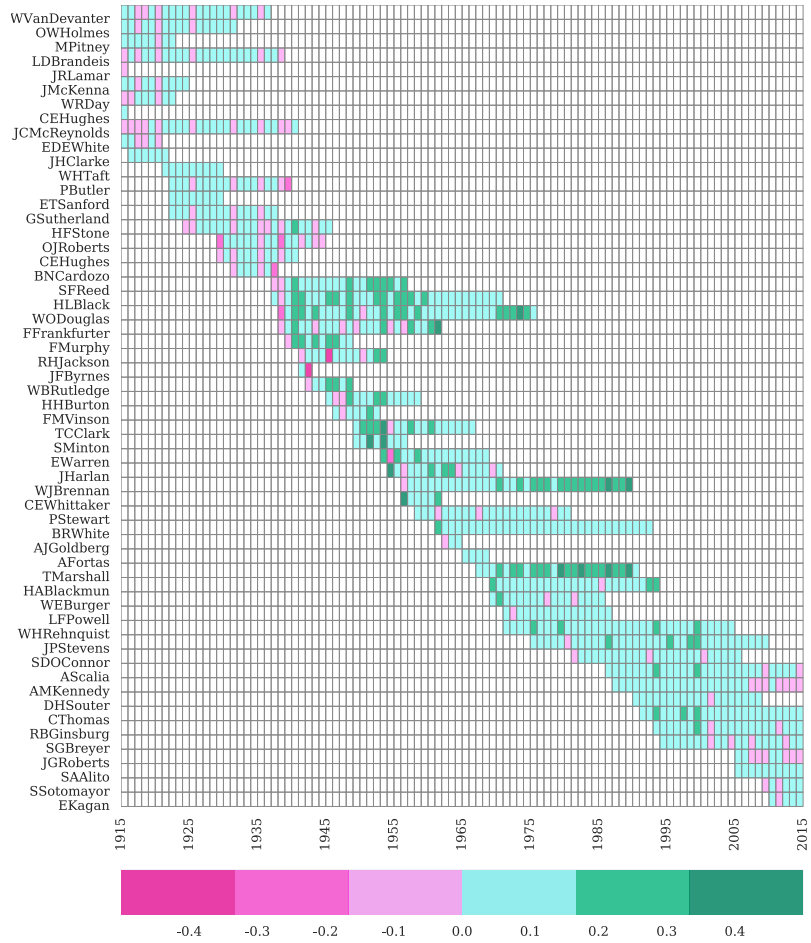
ex) Predicting US Supreme Court behavior

Random forest classifier built on SCDB data[1] to predict the votes of Supreme Court justices:

$Y \in \{\texttt{affirm}, \texttt{reverse}, \texttt{other}\}$

$X \sim 240$-dimensional feature vector

Result: 70% correct classifications



D. M. Katz, M. J. Bommarito II and J. Blackman. **A General Approach for Predicting the Behavior of the Supreme Court of the United States**. *PLoS ONE*, 12(4):1–18, 2017.

[1] `http://supremecourtdatabase.org`

fredrik.lindsten@it.uu.se

ex) Predicting US Supreme Court behavior

Not only have random forests proven to be "unreasonably effective" in a wide array of supervised learning contexts, but in our testing, random forests outperformed other common approaches including support vector machines [...] and feedforward artificial neural network models such as multi-layer perceptron

— Katz, Bommarito II and Blackman (PLoS ONE, 12(4), 2017)

fredrik.lindsten@it.uu.se

Random forests

Bagging can drastically improve the performance of CART!

However, the B bootstrapped dataset are ***correlated***
$\Rightarrow$ the variance reduction due to averaging is diminished.

Idea: De-correlate the B trees by randomly perturbing each tree.

A **random forest** is constructed by bagging, but for each split in each tree only a ***random subset*** of $q \leq p$ inputs are considered as splitting variables.

Rule of thumb: $q = \sqrt{p}$ for classification trees and $q = p/3$ for regression trees.[2]

[2] Proposed by Leo Breiman, inventor of random forests.

fredrik.lindsten@it.uu.se

Random forest pseudo-code

Algorithm Random forest for regression

1. For $b = 1$ to B *(can run in parallel)*
   (a) Draw a bootstrap data set $\mathcal{T}^{\star b}$ of size n from $\mathcal{T}$.
   (b) Grow a regression tree $\widehat{f}^{\star b}(x)$ by repeating the following steps until a minimum node size is reached:
      i. Select q out of the p input variables uniformly at random.
      ii. Find the variable X_j among the q selected, and the corresponding split point s, that minimizes the squared error.
      iii. Split the node into two children with $\{X_j \leq s\}$ and $\{X_j > s\}$.
2. Final model is the average the B ensemble members,

$$\widehat{f}_{\mathsf{rf}}(x) = \frac{1}{B} \sum_{b=1}^{B} \widehat{f}^{\star b}(x).$$

fredrik.lindsten@it.uu.se

Random forests

Recall: For i.d. random variables $\{Z_b\}_{b=1}^{B}$

$$\text{Var}\left(\frac{1}{B}\sum_{i=1}^{B} Z_b\right) = \frac{1-\rho}{B}\sigma^2 + \rho\sigma^2$$

The random input selection used in random forests:

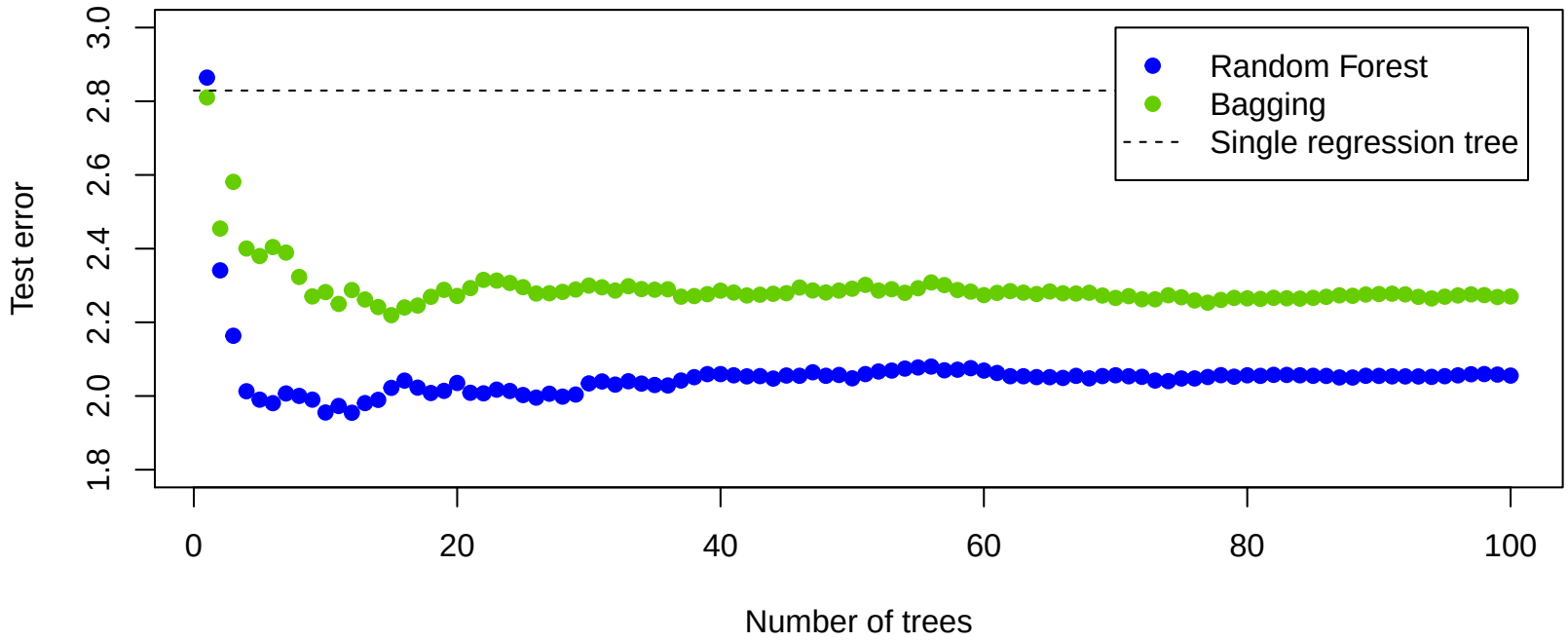
- ▼ increases the bias, but often very slowly
- ▼ adds to the variance (σ^2) of each tree
- ▲ reduces the correlation (ρ) of the trees

The reduction in correlation is typically the dominant effect $\Rightarrow$ there is an overall reduction in MSE!

fredrik.lindsten@it.uu.se

ex) Toy regression model

For the toy model previously considered...



fredrik.lindsten@it.uu.se

Overfitting?

The complexity of a bagging/random forest model increases with an increasing number of trees B.

Will this lead to overfitting as B increases?

No – more ensemble members ***does not*** increase the ***flexibility*** of the model!

Regression case:

$$\widehat{f}_{\text{rf}}(x) = \frac{1}{B}\sum_{b=1}^{B} \widehat{f}^{\star b}(x) \to \mathbb{E}[\widehat{f}^{\star}(x) \,|\, \mathcal{T}], \qquad \text{as } B \to \infty,$$

where the expectation is w.r.t. the randomness in the data bootstrapping and input restriction.

fredrik.lindsten@it.uu.se

Advantages of random forests

Random forests have several **computational advantages**:

- Embarrassingly parallelizable!
- Using $q < p$ potential split variables reduces the computational cost of each split.
- We ***could*** bootstrap fewer than n, say $\sqrt{n}$, data points when creating $\mathcal{T}^{\star b}$ — very useful for "big data" problems.

. . . and they also come with some other benefits:

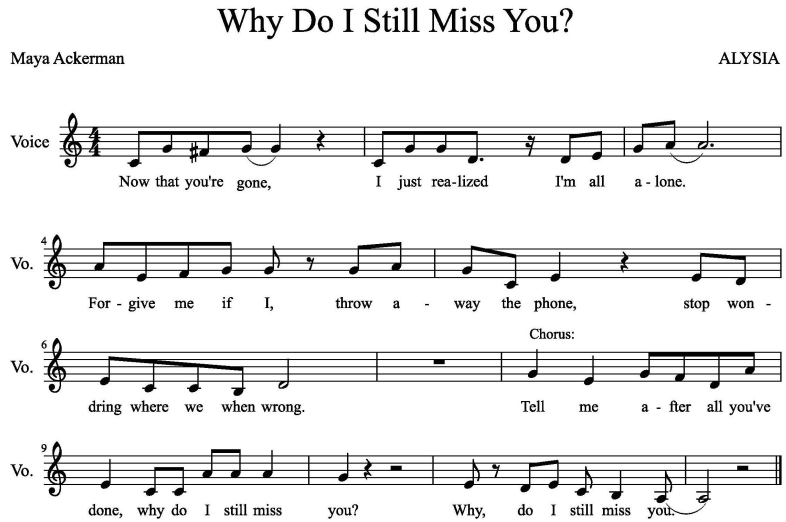
- Often works well off-the-shelf – few tuning parameters
- Requires little or no input preparation
- Implicitly estimates the importance of each input variable X_j

fredrik.lindsten@it.uu.se

ex) Automatic music generation

ALYSIA: automated music generati on using random forests.

- User specifies the lyrics
- ALYSIA generates accompanying music via
  - *rythm model*
  - *melody model*
- Trained on a corpus of pop songs.



`http://www.cs.sjsu.edu/~ackerman/ALYSIA_songs.html`

M. Ackerman and D. Loker. **Algorithmic Songwriting with ALYSIA**. *In: Correia J., Ciesielski V., Liapis A. (eds) Computational Intelligence in Music, Sound, Art and Design. EvoMUSART*, 2017.

fredrik.lindsten@it.uu.se

A few concepts to summarize lecture 6

CART: Classification and regression trees. A class of nonparametric methods based on partitioning the input space into regions and fitting a simple model for each region.

Recursive binary splitting: A greedy method for partitioning the input space into "boxes" aligned with the coordinate axes.

Gini index and deviance: Commonly used error measures for constructing classification trees.

Ensemble methods: Umbrella term for methods that average or combine multiple models.

Bagging: Bootstrap aggregating. An ensemble method based on the statistical bootstrap.

Random forests: Bagging of trees, combined with random feature selection for further variance reduction (and computational gains).

fredrik.lindsten@it.uu.se