Statistical Learning for Data Science: Advanced techniques using R

Stefan Zohren

9 May 2016

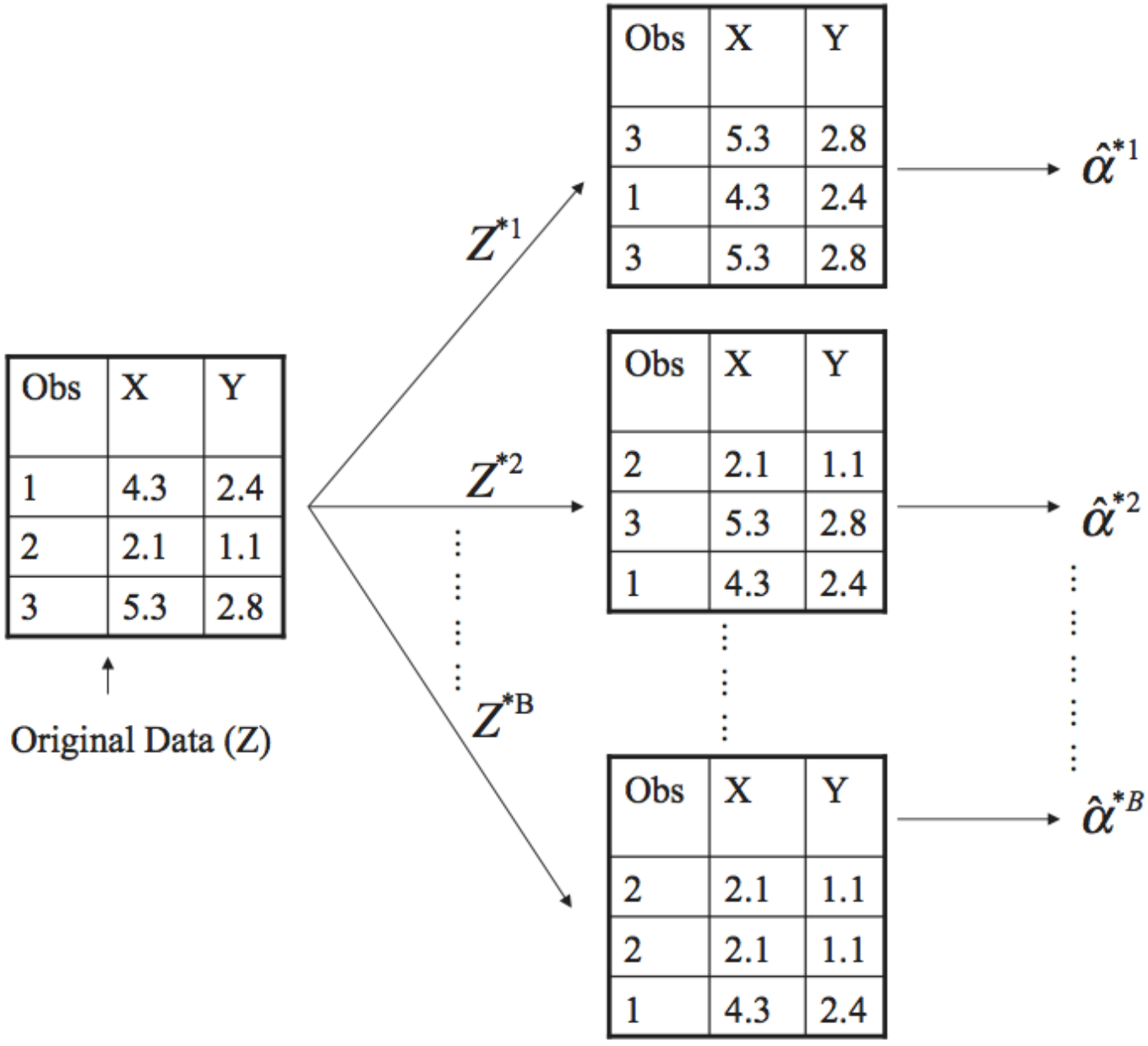
Content of this lecture

- Short introduction to the bootstrap
- Bagging, random forests and boosting
  - Revision of decision trees
  - Bagging
  - Random forests
  - Boosting
- Summary

Short introduction to the bootstrap

- The **bootstrap** uses so-called Monte-Carlo techniques to generate new data
- We think of the data coming from an underlying distribution
- We can obtain an empirical estimator of this data to generate new data
- This amounts to **sampling with replacement from the original data**

Short introduction to the bootstrap (graphical explanation)



Short introduction to the bootstrap

- We create a total of B 'synthetic' or bootstrap data sets
- Each sampled data set contains some observations several times and it does not contain others
- On average only approximately 2/3 of the observations in the original data set are contained in the bootstrap data set

Short introduction to the bootstrap

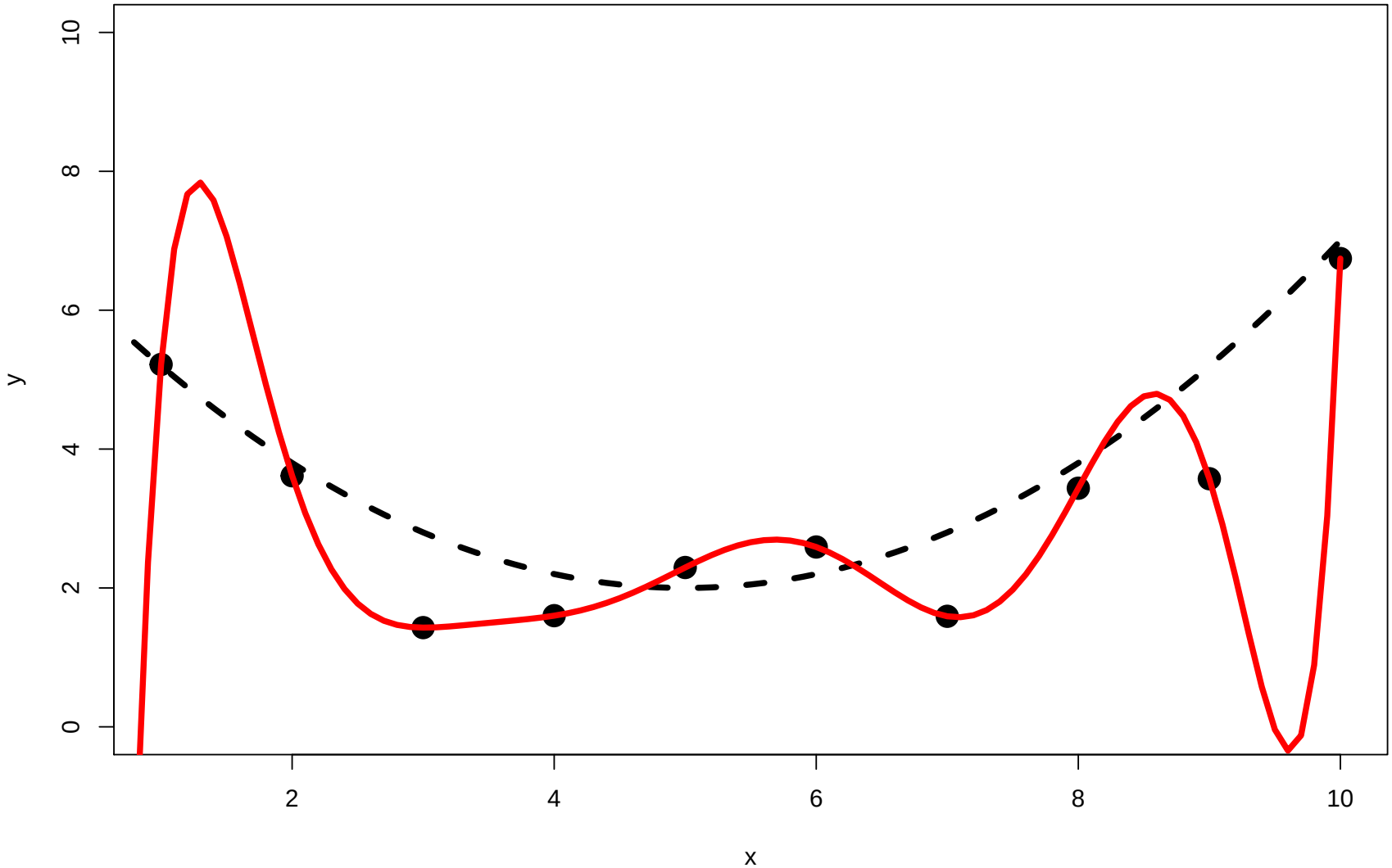
The bootstrap has several important usages. For example:

- Using bootstrap techniques one can obtain **confidence intervals of estimators** or even the entire distribution over estimators.
- Another application of the bootstrap is in prediction and in particular to **reduce the variance of the predictor by averaging various predictors**.

The second use case will be important for the models we will be discussing in this lecture. Let us illustrate it briefly.

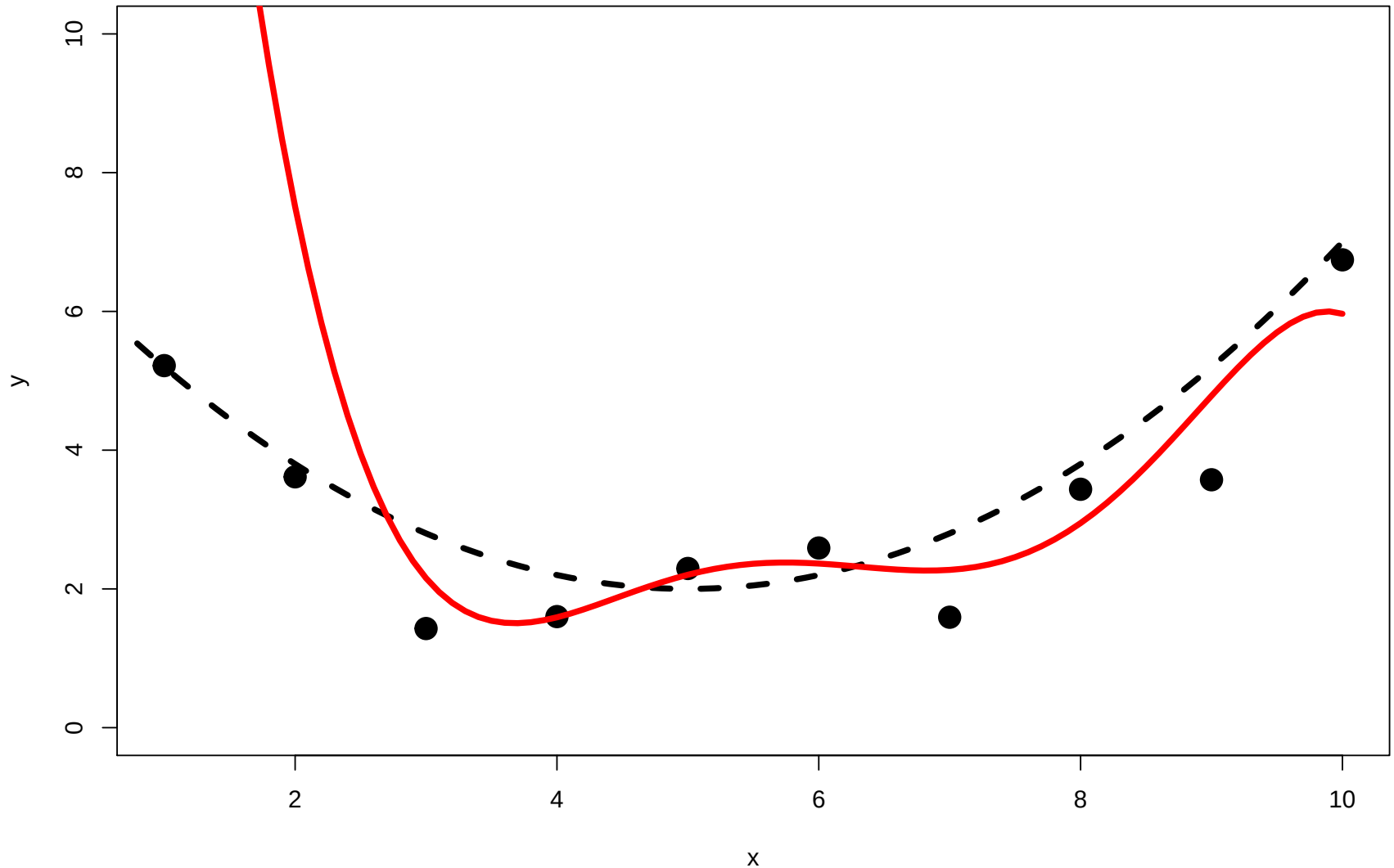
Reduction of variance using the bootstrap

For the illustration we return to the toy example from last class:



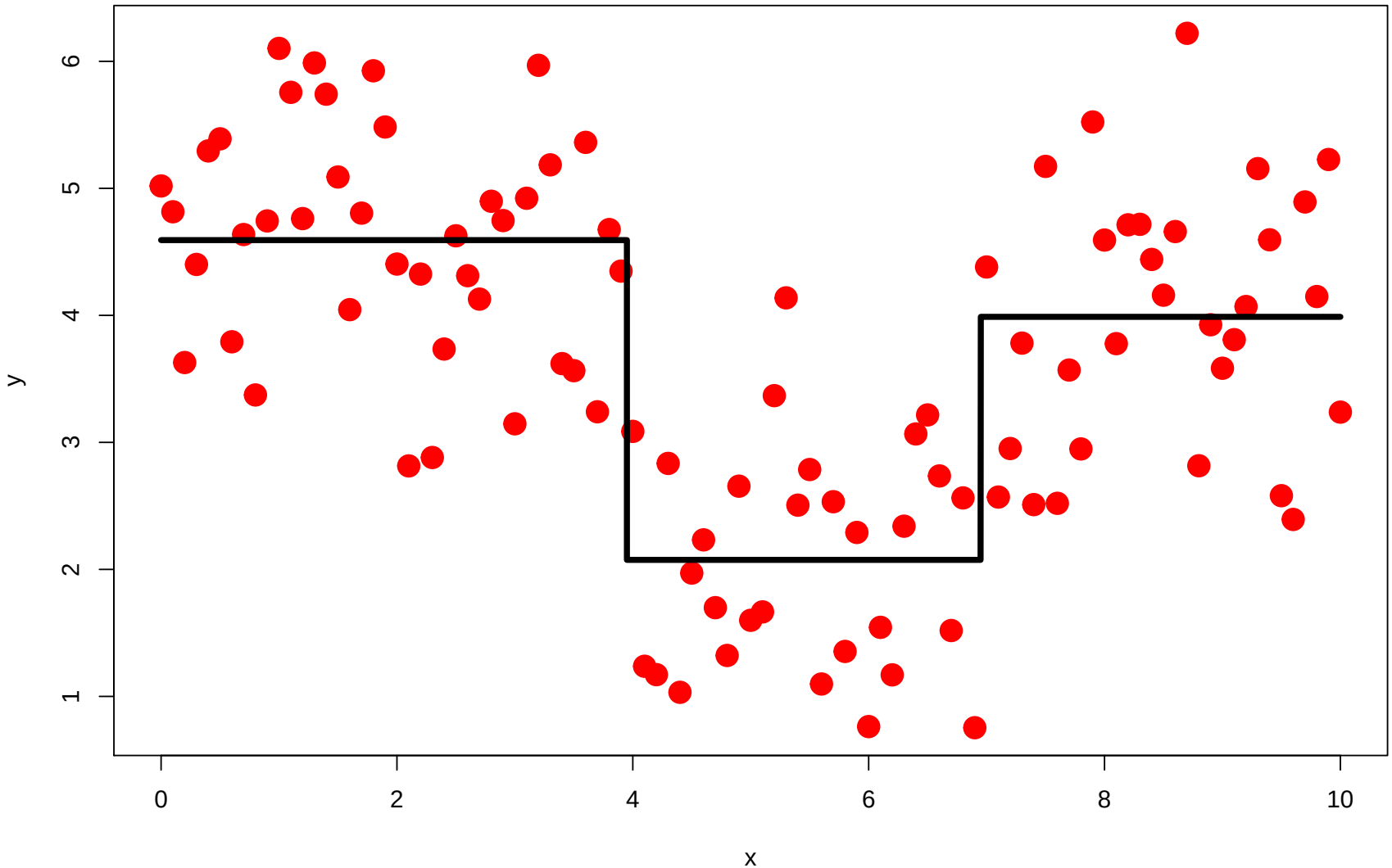
Reduction of variance using the bootstrap

Using bootstrap techniques we can now resample from the data and repeat the above training B=100 times and average:



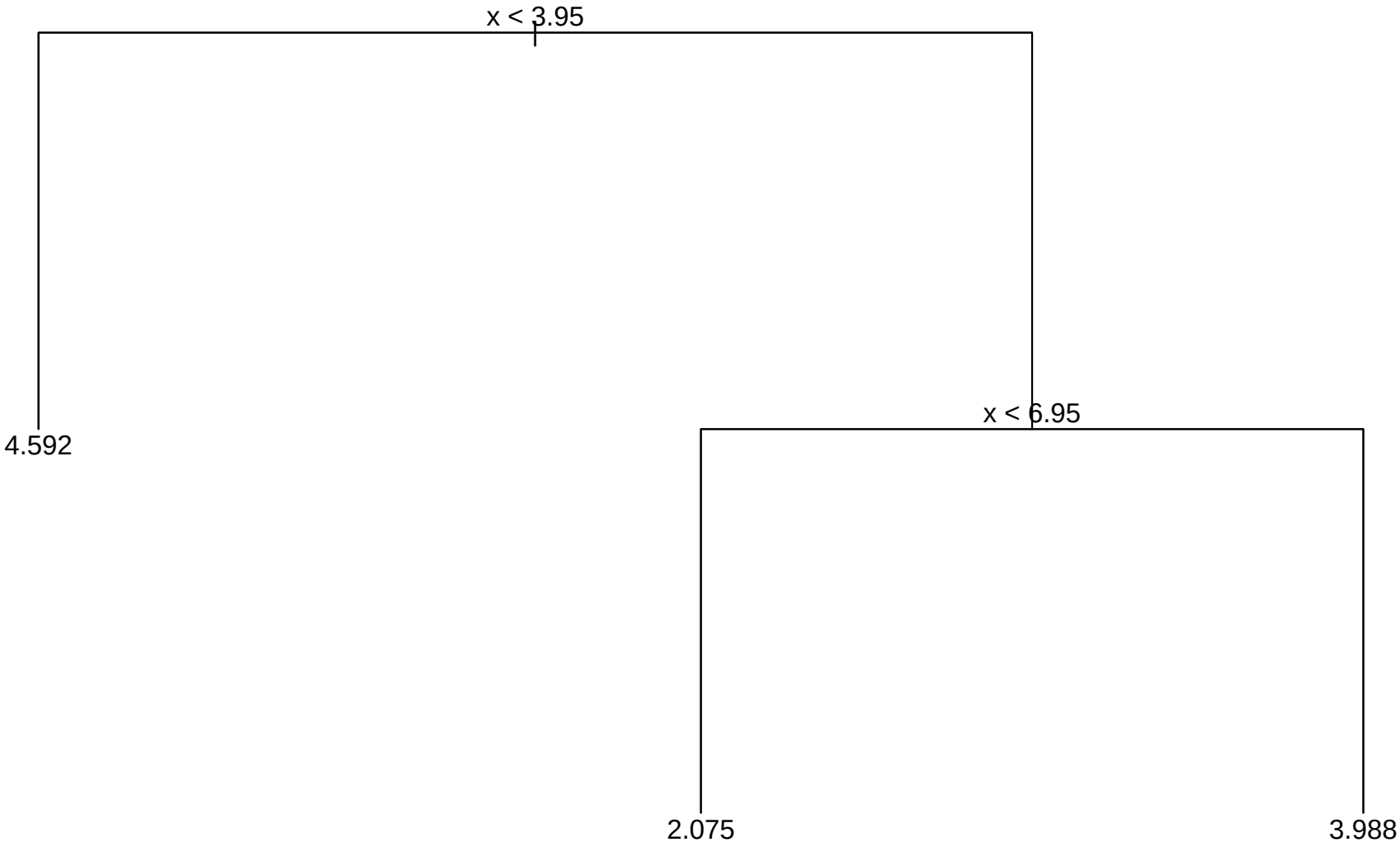
Revision of decision trees

Recall that we fit piecewise constant functions partitioning the predictor space in a greedy fashion:



Revision of decision trees

The resulting function can be represented by a binary decision tree as we discussed last class:



Revision of decision trees

Decision trees have many advantages:

- They are very **simple** and the way prediction is done is very **clear**.
- Another very strong point of decision trees we have seen is that they **can be used with heterogeneous data**.

Disadvantages of decision trees are mainly:

- Decision trees are **limited in their predictive power**.

Bagging

- We now see how we can use ensemble methods to improve the predictive power of decision trees, while maintaining some of their advantages.
- **Bagging** is nothing else than applying the idea of using the bootstrap to reduce the variance predictors to decision trees.
- Repeat the training of a decision tree on B bootstrap data sets then **average the predictors for regression trees**
- To do **bagging for classification trees** use **majority voting**

Bagging: Example in R

We now fit a bagging model using B=100 bootstrap data sets, each of which will have a regression tree trained on it:

```
library(randomForest)
# To do bagging we miust set mtry = #pred. = 8
reg.bag = randomForest(MedianHouseValue ~. , mtry=8,
    data=HousingData.train, ntree=100)
```

We can now calculate the test error:

```
pred.bag = predict(reg.bag, newdata = HousingData.test)
mean((HousingData.test$MedianHouseValue-pred.bag)^2)
```

```
## [1] 0.2104842
```

Random forests

- Random forests are very similar to bagging. We can think of them as bagging with a small tweak
- A potential problem is that the **individual predictors in bagging can be highly correlated**
- In **random forests**, instead of using all predictors each time we train a decision tree on a bootstrap data set, we allow each individual training procedure to only use a smaller random set of the predictors
- This reduces the correlation. In practice, one often random samples m predictors where m is the square root of the total number of predictors

Random forests: Example in R

```
reg.forest = randomForest(MedianHouseValue ~. ,
    data=HousingData.train, mtry=3,
    importance =TRUE,ntree=100)
pred.forest = predict(reg.forest,
                      newdata = HousingData.test)
mean((HousingData.test$MedianHouseValue-pred.forest)^2)
```

```
## [1] 0.2047311
```

Bagging and random forests: Remarks

- Note that both in bagging as well as in random forests we should not prune the trees. Instead we train large flexible trees with individual low bias and high variance and use the averaging procedure to reduce the variance when going to the predictor.
- Bagging and random forests do not require sophisticated fine-tuning of hyper-parameters: We would always want to use as many predictors as possible, as many trees as is computational reasonable and the optimal number of predictors used in random forests is the square root of the total number of predictors.

Boosting

- In boosting we do not use the bootstrap to create new data sets, instead, **in boosting new data sets are created in a sequential manner**
- The idea is that one learns many times a little at a time, always updating the data.
- Even though each individual tree is small (sometimes only a single stem) the resulting **ensemble** of trees is a powerful predictor.

Boosting: Idea of algorithm

Start with full data and a predictors which is zero. Then:

- Fit a decision tree to the data with the given number of splits
- Update the current predictor by adding the new predictor weighted by the shrinkage
- Update the data by 'subtracting' the variance explained by the new tree weighted by the shrinkage
- Repeat

Boosting: Example in R

```
library(gbm)
reg.gbm <- gbm(MedianHouseValue ~. , n.trees=2000,
    data=HousingData.train, distribution= "gaussian",
    interaction.depth = 3,shrinkage = 0.2,verbose =F)
pred.gbm = predict(reg.gbm,
    newdata = HousingData.test, n.trees=2000)
mean((HousingData.test$MedianHouseValue-pred.gbm)^2)
```

```
## [1] 0.1842985
```

Summary

- We gave a short introduction to the bootstrap
- Ensembles methods based on trees were presented: bagging, random forests, boosting
- Bagging and random forests train predictors on many bootstrap data sets
- Boosting uses a sequential approach in training small trees and then updating the data by removing the variance learned
- The last methods, in particular boosting, are powerful ensemble methods in modern data mining