Supervised Learning Methodology

Malte Schierholz & Christoph Kern

University of Mannheim, MZES
Institute for Employment Research (IAB)

Malte.Schierholz@iab.de
c.kern@uni-mannheim.de

March 21 and 22, 2018

Supervised Learning from 1000 miles above

Goal: Find function $f(x)$ that makes optimal predictions in a **new data set** ($\sim$ only X available)

Supervised Learning from 1000 miles above

Goal: Find function $f(x)$ that makes optimal predictions in a **new data set** ($\sim$ only X available)

Prerequisites:

- **Representation**: What is the *hypothesis space*, the family of functions to search over?
  - Describes possible relationships between X and Y
  - Examples: $f(x) = x'\beta$ is linear, or f is a tree.

Supervised Learning from 1000 miles above

Goal: Find function $f(x)$ that makes optimal predictions in a **new data set** ($\sim$ only X available)

Prerequisites:

- **Representation**: What is the *hypothesis space*, the family of functions to search over?
  - Describes possible relationships between X and Y
  - Examples: $f(x) = x'\beta$ is linear, or f is a tree.
- **Evaluation**: What is the criterion to choose between different functions?
  - Measures predictive performance
  - Examples: Mean Squared Error, Logistic Loss

Supervised Learning from 1000 miles above

Goal: Find function $f(x)$ that makes optimal predictions in a **new data set** ($\sim$ only X available)

Prerequisites:

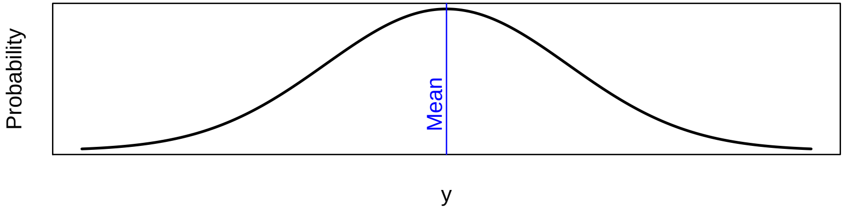
- **Representation**: What is the *hypothesis space*, the family of functions to search over?
  - Describes possible relationships between X and Y
  - Examples: $f(x) = x'\beta$ is linear, or f is a tree.
- **Evaluation**: What is the criterion to choose between different functions?
  - Measures predictive performance
  - Examples: Mean Squared Error, Logistic Loss
- **Computation**: How is f actually calculated?
  - Speed and memory space may be limiting factors
  - Examples: Stochastic gradient descent, handling of sparse matrices

Loss Functions for Estimation

Loss Functions for Estimation and Prediction

Estimation: Standard View

Given a distribution $\mathbb{P}_\theta$, find (ML-)estimator $\hat{\theta}$ that maximizes $P(\text{Data}|\hat{\theta})$



Example:

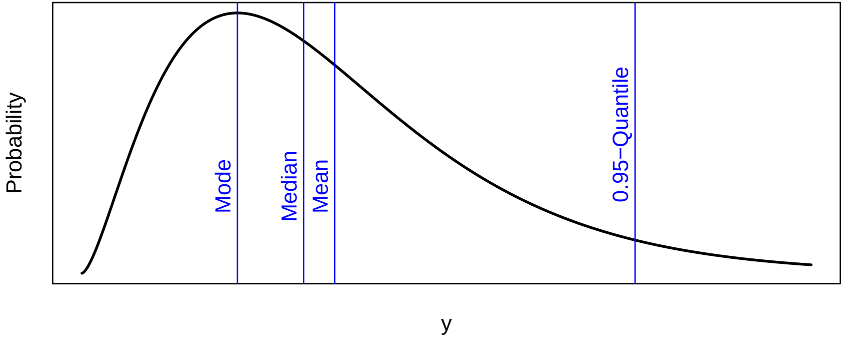
- To estimate the mean from a Gaussian, maximize

$$\max_\theta P(y_1, ..., y_n|\theta) \propto \exp(-\frac{1}{2\sigma^2}\sum(y_i - \theta)^2) \qquad (1)$$

- or, equivalently, find least squares estimate

$$\min_\theta \sum(y_i - \theta)^2 \qquad \Rightarrow \qquad \hat{\theta} = \frac{1}{n}\sum y_i \qquad (2)$$

Estimation: Complementary View



- What if we don't know the distribution?
- What if we are interested in arbitrary functionals?

(M-)Estimation: Complementary View

Estimation is a decision problem: Select estimator $\hat{\theta}$ which has minimal costs. How to define cost?

- Squared distance from “true” θ:

$$\min_{\theta} \sum (y_i - \theta)^2 \qquad \Rightarrow \qquad \hat{\theta} = \frac{1}{n} \sum y_i \tag{3}$$

(M-)Estimation: Complementary View

Estimation is a decision problem: Select estimator $\hat{\theta}$ which has minimal costs. How to define cost?

- Squared distance from “true” θ:

$$\min_{\theta} \sum (y_i - \theta)^2 \qquad \Rightarrow \qquad \hat{\theta} = \frac{1}{n} \sum y_i \tag{3}$$

- Absolute distance from “true” θ:

$$\min_{\theta} \sum |y_i - \theta| \qquad \Rightarrow \qquad \hat{\theta} = \text{Median}(y_1, ..., y_n) \tag{4}$$

(M-)Estimation: Complementary View

Estimation is a decision problem: Select estimator $\hat{\theta}$ which has minimal costs. How to define cost?

- Squared distance from "true" θ:

$$\min_{\theta} \sum (y_i - \theta)^2 \qquad \Rightarrow \qquad \hat{\theta} = \frac{1}{n} \sum y_i \tag{3}$$

- Absolute distance from "true" θ:

$$\min_{\theta} \sum |y_i - \theta| \qquad \Rightarrow \qquad \hat{\theta} = \text{Median}(y_1, ..., y_n) \tag{4}$$

- As a generalization of ML-estimation (density f known):

$$\min_{\theta} \sum - \log f(y_i|\theta) \qquad \Rightarrow \qquad \hat{\theta} = \hat{\theta}_{ML} \tag{5}$$

(M-)Estimation: Complementary View

Estimation is a decision problem: Select estimator $\hat{\theta}$ which has minimal costs. How to define cost?

- Squared distance from "true" θ:

$$\min_{\theta} \sum (y_i - \theta)^2 \qquad \Rightarrow \qquad \hat{\theta} = \frac{1}{n} \sum y_i \tag{3}$$

- Absolute distance from "true" θ:

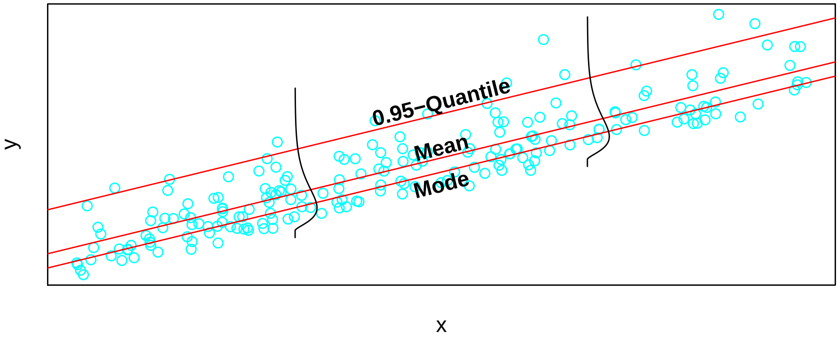
$$\min_{\theta} \sum |y_i - \theta| \qquad \Rightarrow \qquad \hat{\theta} = \text{Median}(y_1, ..., y_n) \tag{4}$$

- As a generalization of ML-estimation (density f known):

$$\min_{\theta} \sum - \log f(y_i|\theta) \qquad \Rightarrow \qquad \hat{\theta} = \hat{\theta}_{ML} \tag{5}$$

→ The specification of "cost" determines what we aim to estimate

Estimation: Conditional Distributions



Depending on the distribution/cost function used, different target functionals $\hat{\theta}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ are estimated.

Loss functions for estimation

Estimation requires (at least implicitly)

- a target distribution $\mathbb{P}_{Y|X}$, or
- a loss function

Possible choices:

(watch out in R for parameters named "`family`" or "`distribution`", but implementations differ.)

Setting	Loss	Target $f(x)$	R implemention
Regression	$(y - f(x))^2$	mean$(y\|x)$	Gaussian
Regression	$\|y - f(x)\|$	median$(y\|x)$	Laplace
Regression	$\rho_\tau(y - f(x))$	$F^{-1}_{y\|x}(\tau)$	τ-Quantile
Classification	Deviance	$\pi_{y\|x}$	Binomial

- Many more options exist, e.g., for censored (survival) data and (multi-)categorical outcomes

Loss functions for prediction

Same for prediction: Prediction requires

- a target distribution $\mathbb{P}_{Y|X}$, or
- a loss function

A small difference:

- Machine Learning mindset is more focused on evaluation criteria, i.e., loss functions

Loss functions for prediction

Same for prediction: Prediction requires

- a target distribution $\mathbb{P}_{Y|X}$, or
- a loss function

A small difference:

- Machine Learning mindset is more focused on evaluation criteria, i.e., loss functions

General lessons:

- Estimation can be framed as loss minimization
- Loss functions describe the target functional we wish to estimate or predict

Estimation vs. Prediction

Estimation vs. Prediction

Estimation vs. Prediction

Goal of Estimation

$$\mathbb{E}(\hat{f}) = f$$

Goal of Prediction

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \mathbb{E}(L(f, y))$$

- $L(f, y) = (f - y)^2$ in the following
- Choose function $\hat{f}$ that predicts new observation y well
    - Low test error ($\neq$ training error)

Estimation vs. Prediction

Standard assumption in both cases:

- Stationary distribution: $(y_i, x_i) \overset{iid}{\sim} \mathbb{P}_{Y,X}$
- ML extrapolates this to future observations

Estimation:

- Interested in parameters of data generating process $\mathbb{P}_{Y,X}$

Prediction:

- Interested in predicting $y = f(x)$

High-dimensional Prediction Problems

ML-methods are most useful in high-dimensional prediction problems

- Many predictor variables
- Unforeseeable interactions
- Local problems with many hikes

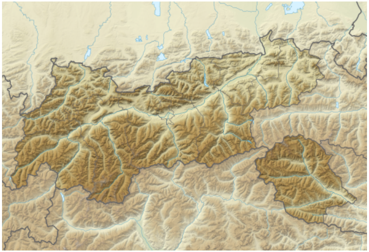


Figure: Map of Tyrol

For a conceptual understanding, lets look at low dimensions first.

Image-Source: Wikipedia By NordNordWest CC BY-SA 2.0, [1]

Bias-Variance-Tradeoff

The Bias-Variance-Tradeoff

A Simple Example

True data generating process:

$$y = \mu + \epsilon \qquad \epsilon \sim N(0, \sigma_\epsilon^2) \tag{6}$$

- Very low dimensional, no covariates

An Estimation Problem

$$y = \mu + \epsilon \qquad \epsilon \sim N(0, \sigma_\epsilon^2) \tag{7}$$

Goal: Estimate μ. Which $\hat{f}_\alpha$ is a good estimator for μ?

$$\hat{f}_\alpha = \alpha \bar{y} \tag{8}$$

$$\mathbb{E}_{\bar{Y}}(\hat{f}_\alpha) = \alpha \mu \tag{9}$$

Bias minimizer: Set $\alpha = 1$ (sample mean is unbiased for μ)

- You get the same result with OLS regression on a constant

A Prediction Problem

Setting:

- Distribution $\mathbb{P}_Y$ and expectation $\mu = \mathbb{E}_Y(y)$ known
- Make a good guess $\hat{f}$ about a future observation y
- **Goal: minimize quadratic loss** of prediction $\hat{f}$

$$\begin{aligned} \mathbb{E}_Y(L(y, \hat{f})) &= \mathbb{E}_Y((y - \hat{f})^2) && (10) \\ &= \mathbb{E}_Y((y - \mu + \mu - \hat{f})^2) && (11) \\ &= (\mu - \hat{f})^2 + \sigma_\epsilon^2 && (12) \end{aligned}$$

A Prediction Problem

Setting:

- Distribution $\mathbb{P}_Y$ and expectation $\mu = \mathbb{E}_Y(y)$ known
- Make a good guess $\hat{f}$ about a future observation y
- **Goal: minimize quadratic loss** of prediction $\hat{f}$

$$\begin{aligned}\mathbb{E}_Y(L(y,\hat{f})) &= \mathbb{E}_Y((y-\hat{f})^2) \quad (10)\\ &= \mathbb{E}_Y((y-\mu+\mu-\hat{f})^2) \quad (11)\\ &= (\mu-\hat{f})^2 + \sigma_\epsilon^2 \quad (12)\end{aligned}$$

Result:

- $\hat{f} = \mu$ is optimal for squared loss.
- σ_ϵ^2 is irreducible noise.

If μ were unknown, could we just plug-in $\bar{y}$ for it?

A Supervised Learning Problem

Setting:

- Same as before, but distribution $\mathbb{P}_Y$ unknown
- We are given n training observations to learn $\hat{f}_\alpha$

$$\hat{f}_\alpha = \alpha \bar{y} \qquad \text{(Hypothesis space)} \tag{13}$$

$$\mathbb{E}_{\bar{Y}}(\hat{f}_\alpha) = \alpha \mu \qquad \rightarrow \text{Unbiased with } \alpha = 1 \tag{14}$$

Training data are random.

A Supervised Learning Problem

Setting:

- Same as before, but distribution $\mathbb{P}_Y$ unknown
- We are given n training observations to learn $\hat{f}_\alpha$

$$\hat{f}_\alpha = \alpha \bar{y} \qquad \text{(Hypothesis space)} \tag{13}$$

$$\mathbb{E}_{\bar{Y}}(\hat{f}_\alpha) = \alpha\mu \qquad \rightarrow \text{Unbiased with } \alpha = 1 \tag{14}$$

Training data are random.
This changes the expected loss

$$\mathbb{E}_{Y,\bar{Y}}(L(y, \hat{f}_\alpha)) = (1-\alpha)^2\mu^2 + \alpha^2\frac{1}{n}\sigma_\epsilon^2 + \sigma_\epsilon^2 \tag{15}$$

Loss is minimal at $\alpha = \frac{\mu^2}{\mu^2+\sigma_\epsilon^2/n} < 1$!!!!

Key Lesson

Estimation != Prediction

- Parameters are biased if a function is fitted for prediction purposes.
- Very different compared with traditional statistics
    - Unbiased estimators are often considered essential (remember that OLS is BLUE, best among all linear unbiased estimators)

The Bias-Variance Trade-Off

Expected test error decomposition
Minimize

$$\mathbb{E}_{Y,\bar{Y}}(L(y,\hat{f}_\alpha)) = \overbrace{(\mathbb{E}_{\bar{Y}}(\hat{f}_\alpha) - \mu)^2}^{(\text{Bias})^2} + \overbrace{\mathbb{E}_{\bar{Y}}(\hat{f}_\alpha - \mathbb{E}_{\bar{Y}}(\hat{f}_\alpha))^2}^{\text{Variance}} + \overbrace{\sigma_\epsilon^2}^{\text{Noise}} \quad (16)$$

$$= \underbrace{(1-\alpha)^2\mu^2}_{\alpha\rightarrow 1} + \underbrace{\alpha^2\frac{1}{n}\sigma_\epsilon^2}_{\alpha\rightarrow 0} + \sigma_\epsilon^2 \quad (17)$$

Usually not possible to minimize both

- the bias (Average deviation from optimal (but unknown) μ)
- the variance (Variability of using different training data)

→ Trade-Off

Intuition about Bias-Variance Trade-Off

You know this already:

- Consider a single test score to estimate a person's ability
    - Would you predict the same score for a future test?

→ Unbiased, yet you probably wouldn't trust a single observation

Intuition about Bias-Variance Trade-Off

You know this already:

- Consider a single test score to estimate a person's ability
    - Would you predict the same score for a future test?

→ Unbiased, yet you probably wouldn't trust a single observation

- You run a one variable regression (correctly specified) and get
    - $\hat{\beta}_0^{OLS} = 0 \pm 0.3$
    - $\hat{\beta}_1^{OLS} = 3 \pm 20$
    - Would you use this model to make predictions?

→ Unbiased, yet with different data the estimates could look very different

Does it matter?

Does the Bias-Variance Trade-Off matter?

Why?

- Theoretical insight about learning goal
- Practical guidance is limited (because bias and variance are unknown)

When?

- Variance term is negligible in low-dimensional problems (few parameters, N large)
- Variance term becomes relevant for high-dimensional problems

High-dimensional Problems

High-dimensional Problems

High-dimensional Problems

ML-methods are most useful in high-dimensional prediction problems

- Many predictor variables
- Unforeseeable interactions
- Local problems with many hikes

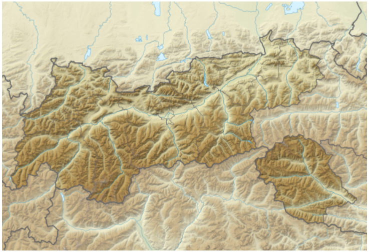


Figure: Map of Tyrol

Our hypothesis space needs to be very flexible to cover these situations

Image-Source: Wikipedia By NordNordWest CC BY-SA 2.0, [1]

Flexible functions

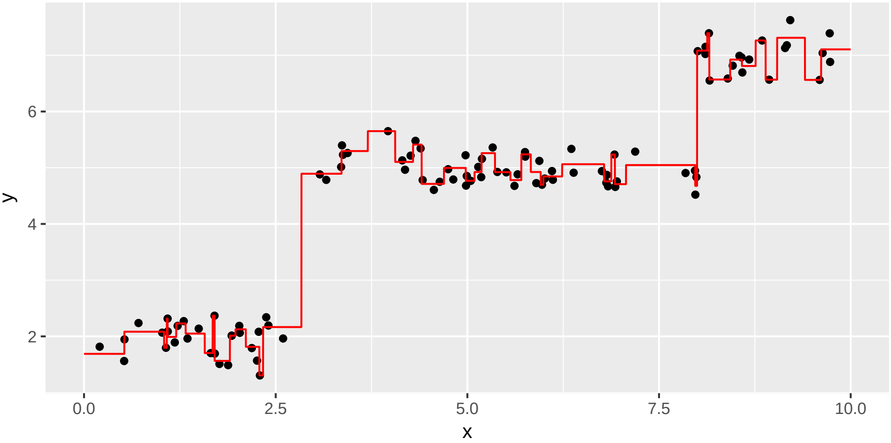
Remember that we need to specify a family of functions (a hypothesis space) to describe possible relations between X and Y

Different hypothesis spaces:

- Linear regression (with many covariates)
- Classification and Regression Trees
- Random Forests
- Boosting
- Neural Networks
- ...

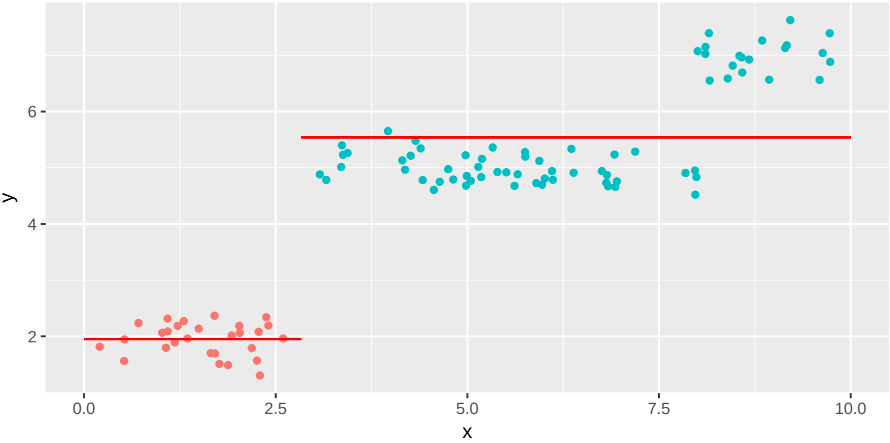
Methods allow specification of flexible functions using many covariates
→ High variance becomes an issue

High Variance in Trees



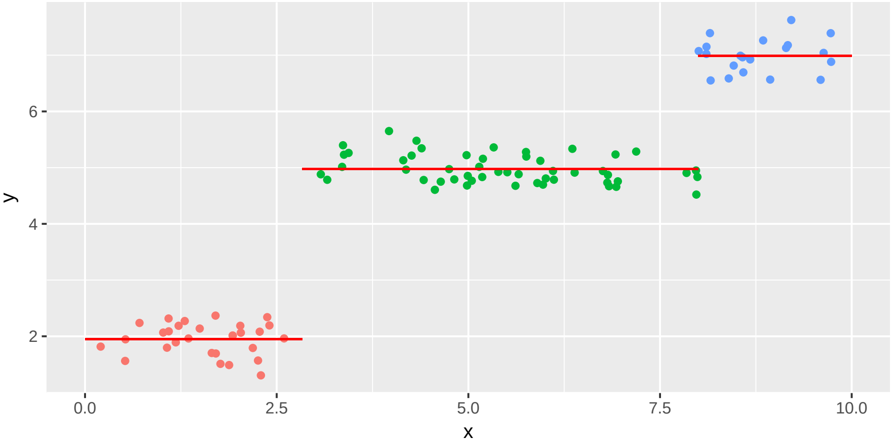
- High Variance = Different data would lead to a different function
- Overfitting = Poor generalization to new data

High Bias in Trees



- High Bias = Blue points are poorly predicted
- Underfitting = Function should adapt better to the data

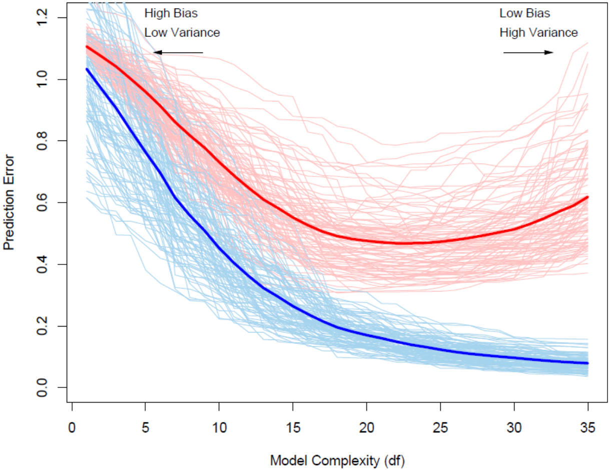
Optimal Solution



- Goal: Find optimal compromise between bias and variance

Bias-Variance Tradeoff

Training Error and Test Error by model capacity



(Source: Hastie et al. 2009)

Key Lesson

Goal of prediction:

$$\arg\min_{f \in \mathcal{F}} \mathbb{E}(L(f(x), y)) \tag{18}$$

but we cannot simply minimize its empirical analogue in training data

$$\arg\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} L(f(x_i), y_i) \tag{19}$$

because this would overfit if the capacity of f is high enough.

Regularization

Solution: Solve (as before)

$$\arg\min_{f \in \mathcal{F}_K} \frac{1}{N} \sum_{i=1}^{N} L(f(x_i), y_i) \tag{20}$$

but f must come from a restricted hypothesis space (limited capacity)

- Tree with at most K leaves
- Regression with $\sum |\beta_j| < K$
- General form: Penalty$(f) < K$

This is **regularization** and usually written as

$$\arg\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} L(f(x_i), y_i) + \lambda \cdot \text{Penalty}(f) \tag{21}$$

$\rightarrow$ How to choose model capacity, i.e., the regularization parameter λ?

Quiz Question

If we have a high bias problem (underfitting), what can be done?

- Add more predictors (= collect more variables or transform existing ones)?
- Allow higher function capacity (= reduce regularization parameter)?
- Use more flexible algorithms (e.g., a tree instead of linear regression)?

If we have a high variance problem (overfitting), what can be done?

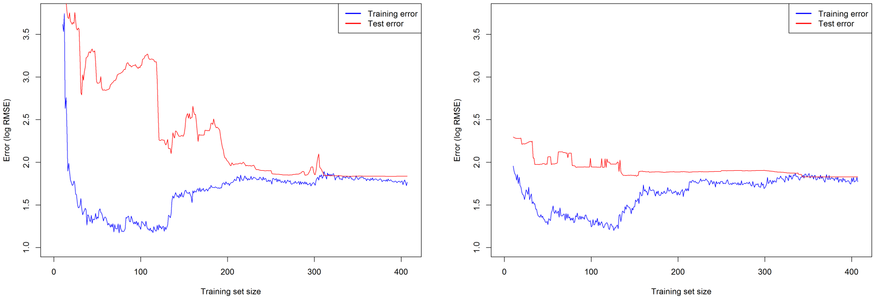
- Add more predictors (= collect more variables or transform existing ones)?
- Allow higher function capacity (= reduce regularization parameter)?
- Use more flexible algorithms (e.g., a tree instead of linear regression)?
- Collect more training data?

Learning curves

How much data are needed?

- Idea: Plot training and validation error against training set size
- Allows to study the gain of adding more data
  - Convergence of validation error curve towards training curve
- Can also be used as a diagnosis tool to asses
  - High bias (Underfitting): Curves converge at a high value
  - High variance (Overfitting): Large gap between curves

Figure: Learning curves



(a) Linear regression

(b) Regression trees

References

Hastie, Trevor, Tibshirani, Robert & Friedman, Jerome (2009)
The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer. Chapter 7.

Athey, Susan & Imbens, Guido (2018)
Machine Learning and Econometrics, January 7-9, 2018:
`https://www.aeaweb.org/conference/cont-ed/2018-webcasts`