Machine Learning in clinical drug development

Markus Lange, Lorenz Uhlmann
BBS Seminar
February 21st, 2022

NOVARTIS | Reimagining Medicine

Agenda

- Introduction
  - Motivational example - the google flu story
  - Why the hype?
  - What is machine learning?
- Introducing key concepts
  - Performance evaluation
  - Cross-validation
  - Bias-Variance-Tradeoff
  - The bootstrap

Agenda (continued)

- Machine Learning techniques
  - Penalized regression
  - Trees, Bagging, Random forests, and Boosting
  - Finding subgroups
  - Unsupervised learning

Acknowledgements

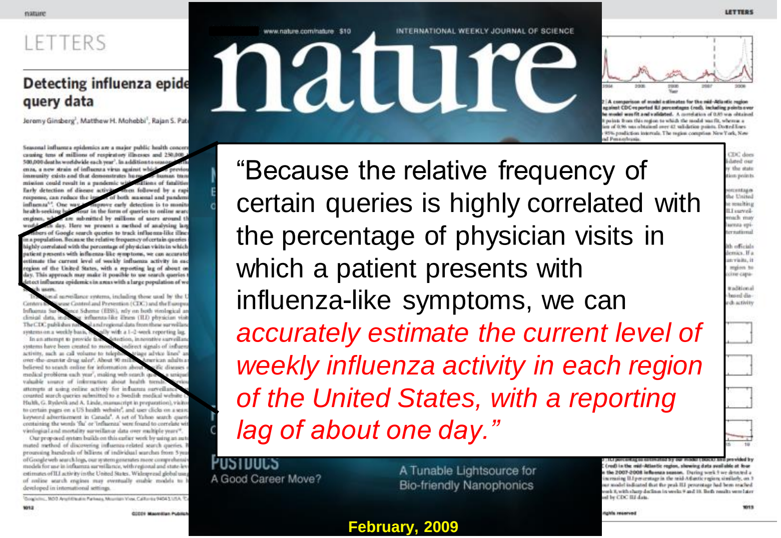
(In no particular order)

- Oliver Sander
- Matthias Kormaksson
- David Ohlssen
- Marc Vandemeulebroecke
- Peter Krusche
- Shu Yang
- Conor Moloney
- Mark Baillie

... who all contributed to this training in one way or another!

The google flu story

Social Media in Action – the google flu story


nature
LETTERS
Detecting influenza epide
query data
www.nature.com/nature $10
INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
nature
POSTDOCS
A Good Career Move?
A Tunable Lightsource for
Bio-friendly Nanophotonics
LETTERS

"Because the relative frequency of certain queries is highly correlated with the percentage of physician visits in which a patient presents with influenza-like symptoms, we can *accurately estimate the current level of weekly influenza activity in each region of the United States, with a reporting lag of about one day."*

February, 2009

Social Media in Action

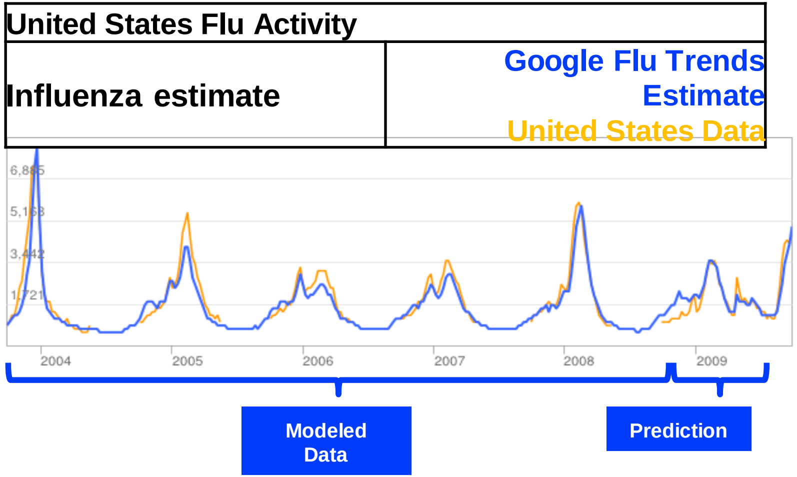
- “Google web search queries can be used to estimate ILI percentages accurately in each of the nine public health regions of the United States. Because search queries can be processed quickly, the resulting ILI estimates were consistently 1–2 weeks ahead of CDC ILI surveillance reports. The early detection provided by this approach may become an important line of defense against future influenza epidemics in the United States, and perhaps eventually in international settings.”

ILI = Influenza-like illness

Triumph of Big Data

United States Flu Activity

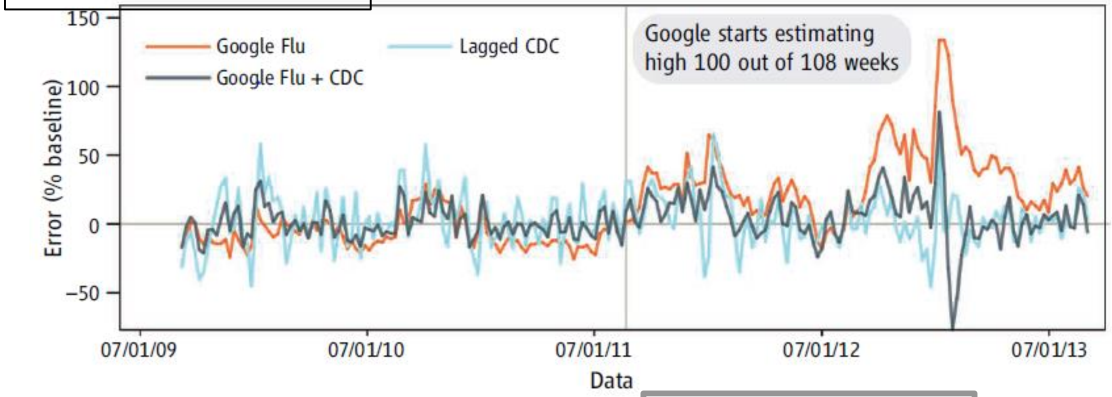
Influenza estimate | **Google Flu Trends Estimate** | **United States Data**



- "… simple models and big data trump more-elaborate analytics approaches."
  - A. McAfee, E. Brynjolfsson
  - Harvard Business Review, 90
  - Oct, 2012, p. 64

Social Media in Action

⇐ Models built on data from 2003-2008.



Predictions become worse over time. ⇒

Social Media in Action

14 Mar 2014

BIG DATA

The Parable of Google Flu: Traps in Big Data Analysis

David Lazer,[1,2*] Ryan Kennedy,[1,3,4] Gary King,[3] Alessandro Vespignani[5,6,3]

Large errors in flu predictions were **largely avoidable**, which offers lessons for the use of big data.

Why the hype?

Hype Cycle for Emerging Technologies

Machine learning community has made great progress on many problems!

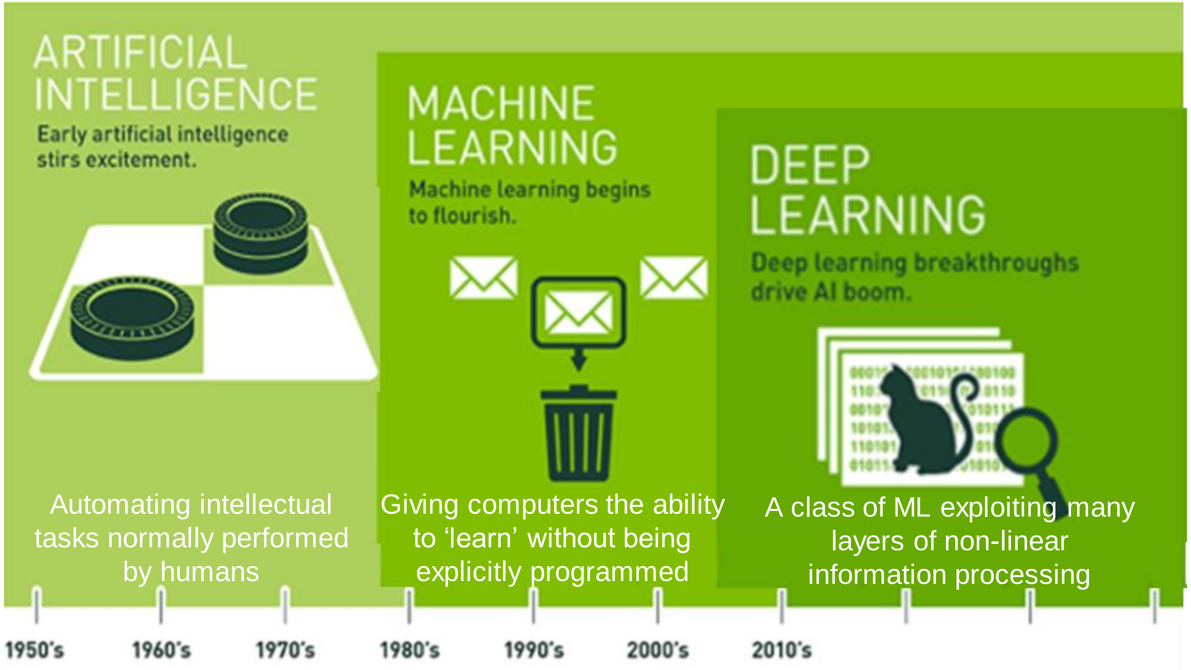
Those problems are very different to “Pharma problems”!

- Machine learning successfully applied in **high signal to noise** settings
  - E.g. Image recognition
  - Easy to classify
  - Lots of available data (e.g. online data bases, Reinforcement learning)
- “Problems” in pharma are oftentimes nothing like this
  - Low signal to noise
  - Hard to classify (When exactly is patient A doing better than patient B?)
  - “Inherent” randomness
  - Data generation is time consuming and expensive

What does machine learning even mean?

Definitions

Alan Turing (1950): a machine is “intelligent” if it can make a human believe that it is human

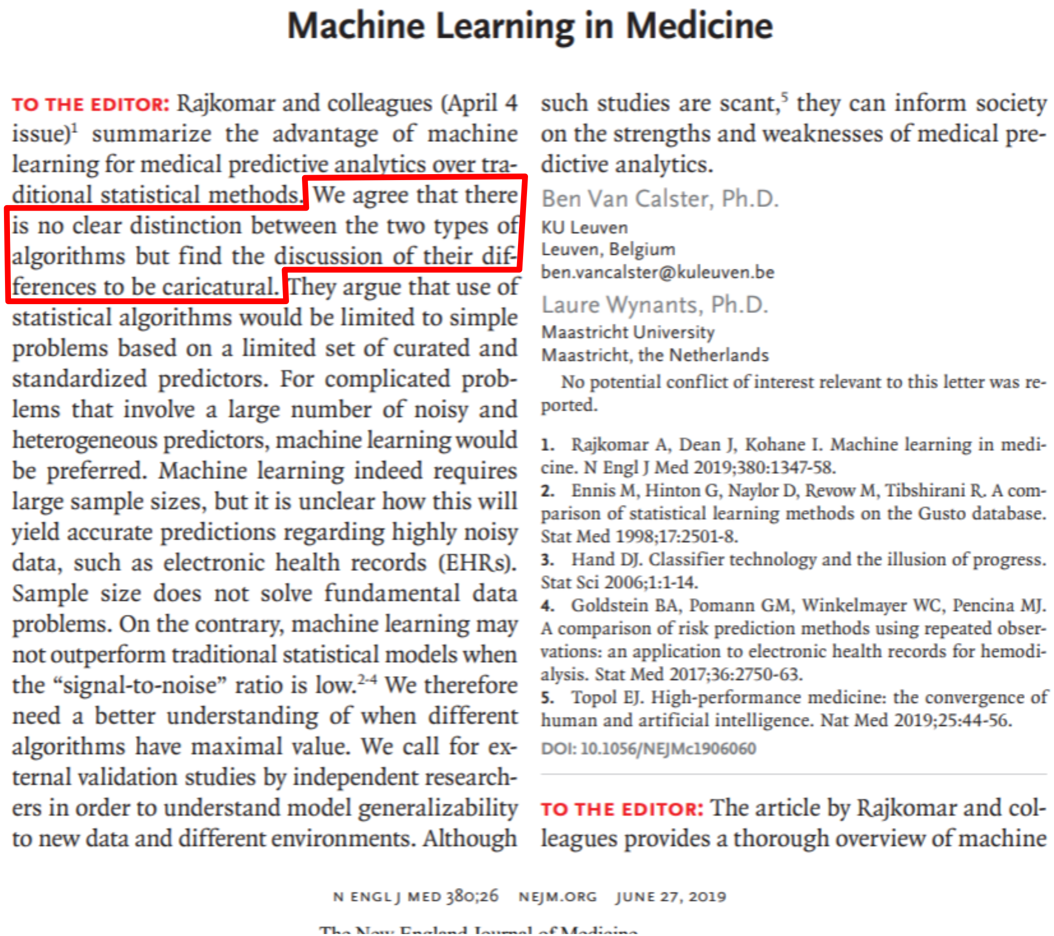

ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
1950's
1960's
1970's
1980's
1990's
2000's
2010's

Automating intellectual tasks normally performed by humans

Giving computers the ability to ‘learn’ without being explicitly programmed

A class of ML exploiting many layers of non-linear information processing

ML vs(?) stats - pretty much the same thing?



Machine Learning in Medicine

TO THE EDITOR: Rajkomar and colleagues (April 4 issue)[1] summarize the advantage of machine learning for medical predictive analytics over traditional statistical methods. We agree that there is no clear distinction between the two types of algorithms but find the discussion of their differences to be caricatural. They argue that use of statistical algorithms would be limited to simple problems based on a limited set of curated and standardized predictors. For complicated problems that involve a large number of noisy and heterogeneous predictors, machine learning would be preferred. Machine learning indeed requires large sample sizes, but it is unclear how this will yield accurate predictions regarding highly noisy data, such as electronic health records (EHRs). Sample size does not solve fundamental data problems. On the contrary, machine learning may not outperform traditional statistical models when the "signal-to-noise" ratio is low.[2-4] We therefore need a better understanding of when different algorithms have maximal value. We call for external validation studies by independent researchers in order to understand model generalizability to new data and different environments. Although such studies are scant,[5] they can inform society on the strengths and weaknesses of medical predictive analytics.

Ben Van Calster, Ph.D.
KU Leuven
Leuven, Belgium
ben.vancalster@kuleuven.be

Laure Wynants, Ph.D.
Maastricht University
Maastricht, the Netherlands

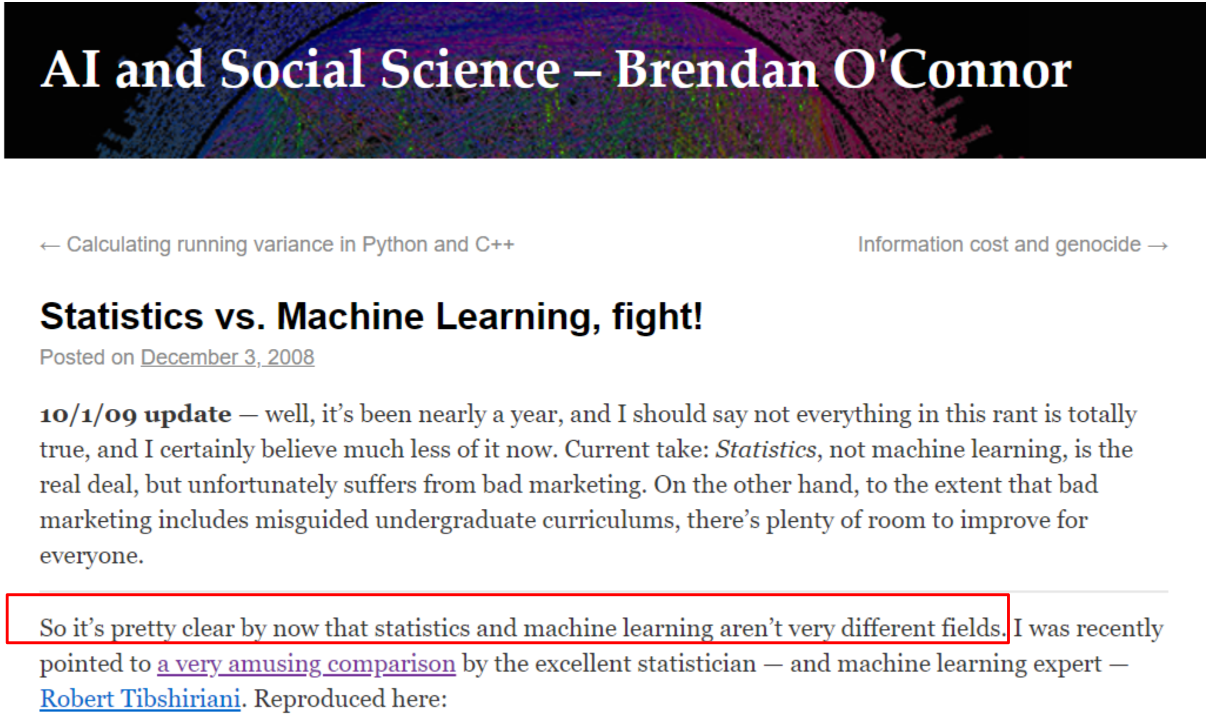
No potential conflict of interest relevant to this letter was reported.

1. Rajkomar A, Dean J, Kohane I. Machine learning in medicine. N Engl J Med 2019;380:1347-58.
2. Ennis M, Hinton G, Naylor D, Revow M, Tibshirani R. A comparison of statistical learning methods on the Gusto database. Stat Med 1998;17:2501-8.
3. Hand DJ. Classifier technology and the illusion of progress. Stat Sci 2006;1:1-14.
4. Goldstein BA, Pomann GM, Winkelmayer WC, Pencina MJ. A comparison of risk prediction methods using repeated observations: an application to electronic health records for hemodialysis. Stat Med 2017;36:2750-63.
5. Topol EJ. High-performance medicine: the convergence of human and artificial intelligence. Nat Med 2019;25:44-56.

DOI: 10.1056/NEJMc1906060

TO THE EDITOR: The article by Rajkomar and colleagues provides a thorough overview of machine

https://www.nejm.org/doi/pdf/10.1056/NEJMc1906060?articleTools=true



AI and Social Science – Brendan O'Connor

← Calculating running variance in Python and C++

Information cost and genocide →

Statistics vs. Machine Learning, fight!

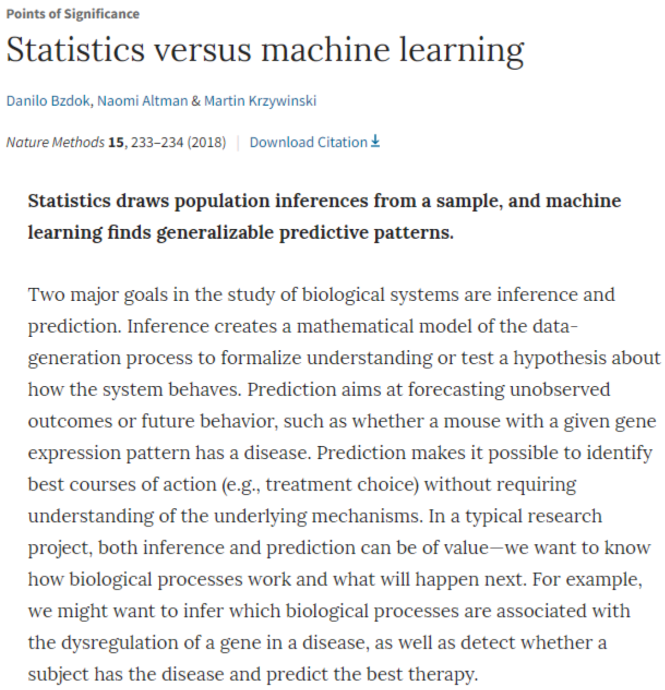
Posted on December 3, 2008

10/1/09 update — well, it's been nearly a year, and I should say not everything in this rant is totally true, and I certainly believe much less of it now. Current take: *Statistics*, not machine learning, is the real deal, but unfortunately suffers from bad marketing. On the other hand, to the extent that bad marketing includes misguided undergraduate curriculums, there's plenty of room to improve for everyone.

So it's pretty clear by now that statistics and machine learning aren't very different fields. I was recently pointed to a very amusing comparison by the excellent statistician — and machine learning expert — Robert Tibshiriani. Reproduced here:

https://brenocon.com/blog/2008/12/statistics-vs-machine-learning-fight/

Or are there distinct differences?



Points of Significance

Statistics versus machine learning

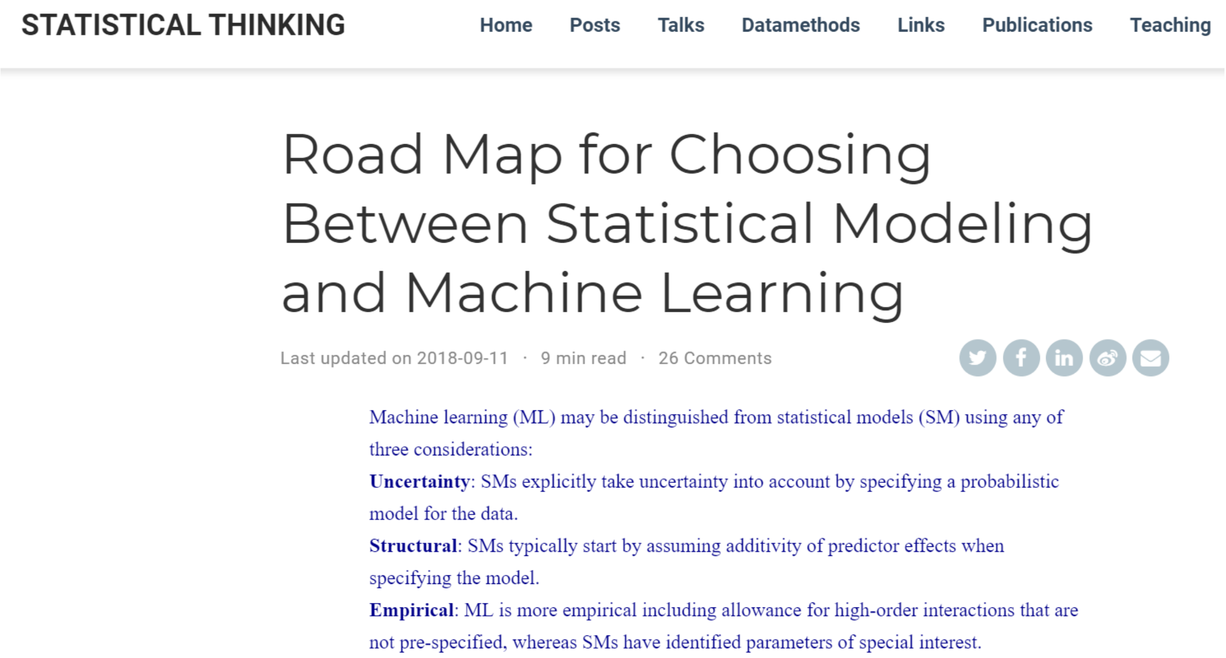
Danilo Bzdok, Naomi Altman & Martin Krzywinski

Nature Methods **15**, 233–234 (2018) | Download Citation

Statistics draws population inferences from a sample, and machine learning finds generalizable predictive patterns.

Two major goals in the study of biological systems are inference and prediction. Inference creates a mathematical model of the data-generation process to formalize understanding or test a hypothesis about how the system behaves. Prediction aims at forecasting unobserved outcomes or future behavior, such as whether a mouse with a given gene expression pattern has a disease. Prediction makes it possible to identify best courses of action (e.g., treatment choice) without requiring understanding of the underlying mechanisms. In a typical research project, both inference and prediction can be of value—we want to know how biological processes work and what will happen next. For example, we might want to infer which biological processes are associated with the dysregulation of a gene in a disease, as well as detect whether a subject has the disease and predict the best therapy.

https://www.nature.com/articles/nmeth.4642



Road Map for Choosing Between Statistical Modeling and Machine Learning

Last updated on 2018-09-11 · 9 min read · 26 Comments

Machine learning (ML) may be distinguished from statistical models (SM) using any of three considerations:

Uncertainty: SMs explicitly take uncertainty into account by specifying a probabilistic model for the data.

Structural: SMs typically start by assuming additivity of predictor effects when specifying the model.

Empirical: ML is more empirical including allowance for high-order interactions that are not pre-specified, whereas SMs have identified parameters of special interest.

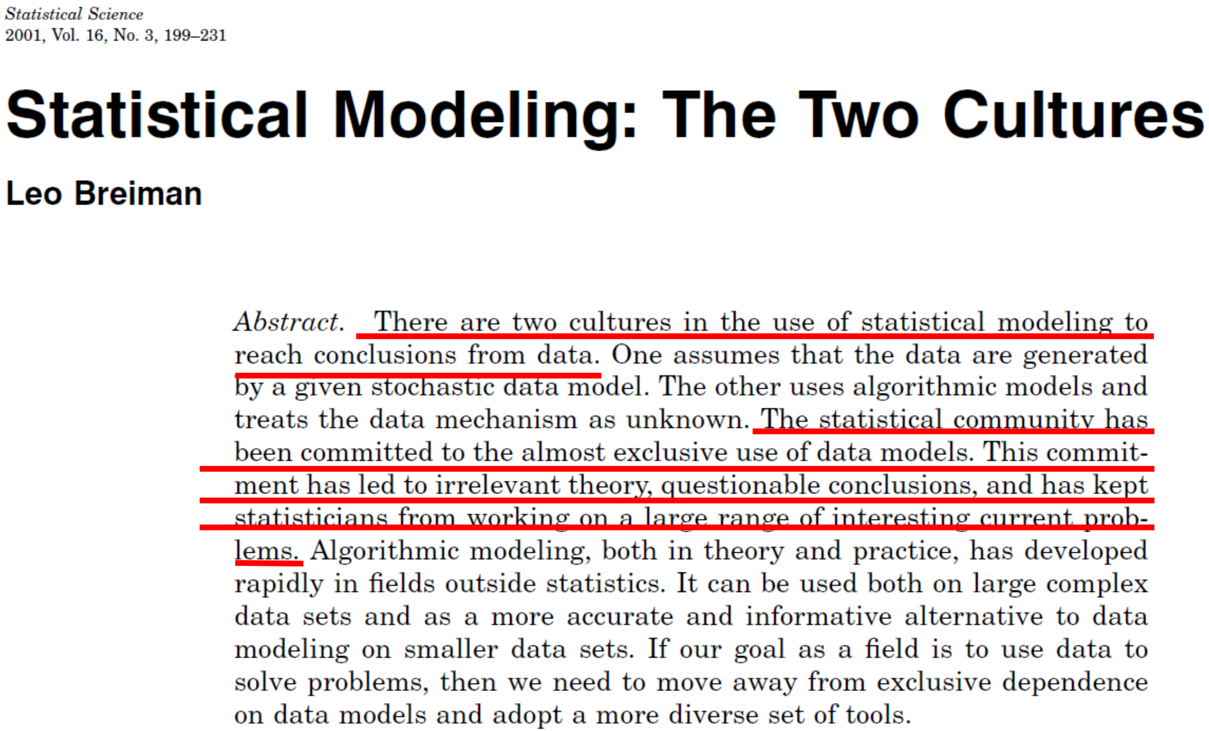
Frank Harrel: https://www.fharrell.com/post/stat-ml/

People often mean different things when comparing the two

- Some focus on the difference in application ...
  - Using a linear model for prediction
    - You are “doing Machine learning”
  - Using a linear model for inference
    - You are “doing statistics”

- Others focus on differences of the underlying methodology/philosophy

How did it all start? Maybe here...



Statistical Science
2001, Vol. 16, No. 3, 199–231

Statistical Modeling: The Two Cultures

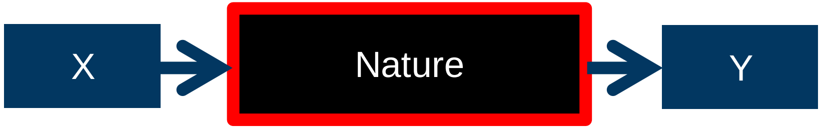
Leo Breiman

Abstract. There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown. The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems. Algorithmic modeling, both in theory and practice, has developed rapidly in fields outside statistics. It can be used both on large complex data sets and as a more accurate and informative alternative to data modeling on smaller data sets. If our goal as a field is to use data to solve problems, then we need to move away from exclusive dependence on data models and adopt a more diverse set of tools.

Culture = Maths/Stats versus Computing Science department

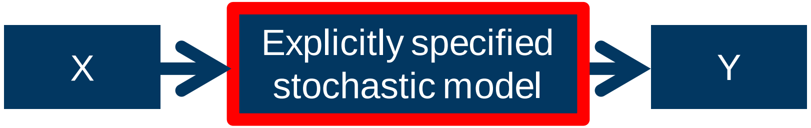
The Two Cultures

Nature



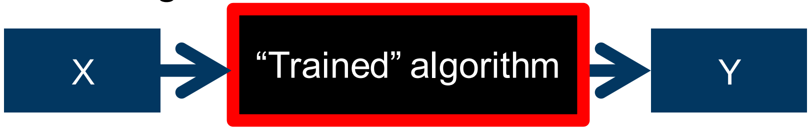
- Nature is a black box

«Data modeling culture»



- Simple models with interpretable parameters
- Emphasis on interpretability and inference

«Algorithmic modeling culture»



- Complex models that are trained rather than explicitly specified
- Emphasis on prediction rather than interpretability

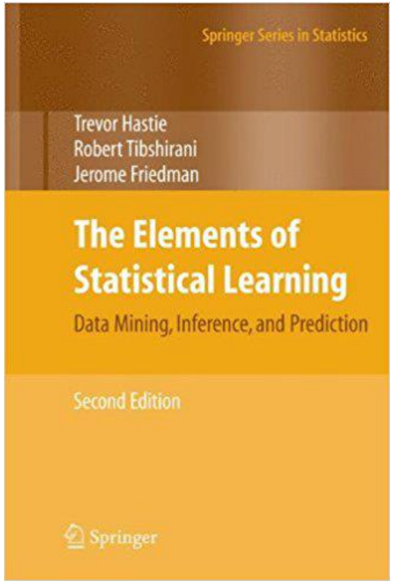
Leo Breiman’s opinion

- Model validation based on goodness of fit and residual examination – should be based on predictive accuracy
- Led to irrelevant theory and questionable scientific conclusions
- Kept statisticians from using more suitable algorithmic models and from working on exciting problems
- Estimated 98% of statisticians follow this approach
- The goal should be accurate information, not interpretability

Comments on the machine learning culture

- In 2001 Breiman claimed about 2% of statisticians would follow the machine learning or algorithmic approach
- However, since then a large literature has developed in statistical machine learning
- More recent approaches combine realistically complex statistical models with the scalability of machine learning algorithms

Statistical learning forms a bridge between the two cultures



Springer Series in Statistics

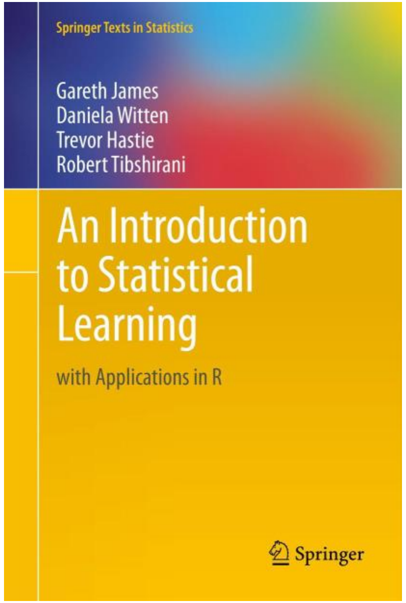
Trevor Hastie
Robert Tibshirani
Jerome Friedman

The Elements of Statistical Learning

Data Mining, Inference, and Prediction

Second Edition

Springer



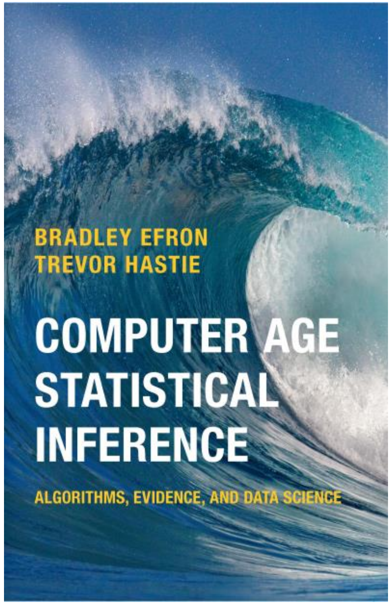
Springer Texts in Statistics

Gareth James
Daniela Witten
Trevor Hastie
Robert Tibshirani

An Introduction to Statistical Learning

with Applications in R

Springer



BRADLEY EFRON
TREVOR HASTIE

COMPUTER AGE STATISTICAL INFERENCE

ALGORITHMS, EVIDENCE, AND DATA SCIENCE

Model performance evaluation

Setting

Binary prediction problem

Given predictors $\boldsymbol{X}$, predict binary outcome (or class) $\mathbf{Y} \in \{\mathbf{0}, \mathbf{1}\}$

Scoring classifier, e.g. predicted class probability $\widehat{\boldsymbol{p}} = \widehat{\boldsymbol{P}}\{(\boldsymbol{Y} = \mathbf{1})$ and a threshold criteria (e.g. $\widehat{\boldsymbol{Y}} = \boldsymbol{I}(\widehat{\boldsymbol{p}} > \mathbf{0.5})$ $\in \{0,1\}$).

Think Logistic Regression

Simple Example: *Identify responders/non-responders at week 16 by baseline characteristics, e.g. demographics, disease severity, mechanistic or genetic markers.*

Predictive performance of models

- **What is “good performance”?**
  → different performance metrics

- **How to find out if your model is doing well?**
  → Model validation strategies:

  hold-out data set, Cross Validation (CV), ...

- **How to make your model do well?**
  → bias-variance tradeoff, regularization, preventing overfitting

What is good performance?

Performance metrics

NOVARTIS | Reimagining Medicine

Many ways to look at a 2x2 contingency table...

			$Y \in \{0,1\}$	
			True Class	
			Positive	Negative
$\hat{Y} \in \{0,1\}$	**Predicted Class**	Positive	**T**rue **P**ositives	**F**alse **P**ositives
		Negative	**F**alse **N**egatives	**T**rue **N**egatives
		Column Total	**P**	**N**

- Many performance metrics exist. Choose wisely!

Accuracy
weights each sample in the same way

$Y \in \{0,1\}$

True Class

			Positive	Negative
$\hat{Y} \in \{0,1\}$	Predicted Class	Positive	True Positives	False Positives
		Negative	False Negatives	True Negatives
		Column Total	P	N

Σ

# of correct predictions

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP + TN}}{\Sigma}$$

# of predictions

$$\boldsymbol{Misclassification\ Rate = 1 - Accuracy}$$

- Can be misleading in case of class imbalance (if 95% of samples are negative, we can achieve 95% accuracy, by always predicting “negative”)

True/False Positive Rate - TPR/FPR condition on the true label

		$Y \in \{0,1\}$ **True Class**	
		Positive	Negative
$\hat{Y} \in \{0,1\}$ **Predicted Class**	Positive	***T**rue **P**ositives*	***F**alse **P**ositives*
	Negative	**F**alse **N**egatives	**T**rue **N**egatives
	Column Total	**P**	**N**

$$\boldsymbol{TPR = \frac{TP}{P}}$$

# positives

$$\boldsymbol{FPR = \frac{FP}{N}}$$

# negatives

- **Important for ROC curves**
- Note alternative terminology: TPR = sensitivity = recall,
  FPR = 1-specificity

TPR = true positive rate, FPR = false positive rate

Pos/Neg Predictive Value – PPV/NPV condition on the predicted label

$Y \in \{0,1\}$

True Class

		Positive	Negative	
$\hat{Y} \in \{0,1\}$ **Predicted Class**	Positive	**T**rue **P**ositives	**F**alse **P**ositives	Positive predicted
	Negative	**F**alse **N**egatives	**T**rue **N**egatives	Negative predicted
	Column Total	**P**	**N**	

$$\boldsymbol{FDR = 1 - PPV}$$

$$\boldsymbol{PPV} = \frac{\boldsymbol{TP}}{\underbrace{TP + FP}_{\text{\# of predicted positives}}}$$

$$\boldsymbol{NPV} = \frac{\boldsymbol{TN}}{\underbrace{FN + TN}_{\text{\# of predicted negatives}}}$$

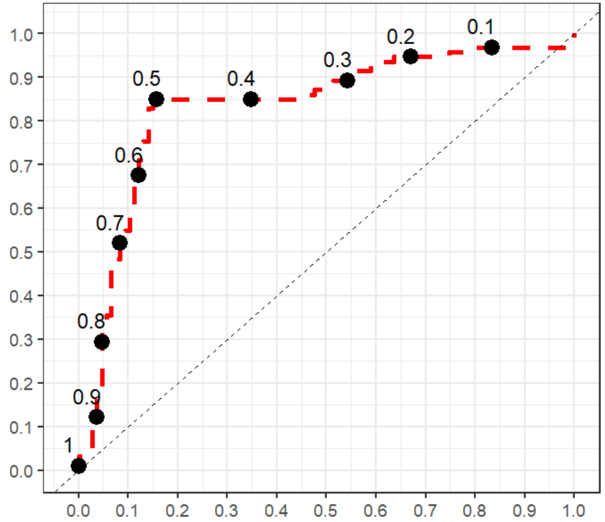
- Note alternative terminology: PPV = precision
- Conditioning on predicted label can be useful in situations with high imbalance (e.g. diagnostic screening or information retrieval)

PPV = positive predictive value, NPV = negative predictive value

ROC curve and Area Under Curve (AUC)

sweep across range of possible thresholds of scoring classifier

$$TPR = \frac{TP}{\mathrm{P}}$$



$$FPR = \frac{FP}{\mathrm{N}}$$

$Y \in \{0,1\}$

True Class

Predicted Class $\hat{Y} \in \{0,1\}$

	Positive	Negative
Positive	***TP***	***FP***
Negative	False Negatives	True Negatives
Column Total	**P**	**N**

Each threshold produces a new table!

$$\hat{Y} = I(\hat{p} > threshold)$$

$$threshold = 0, 0.1, 0.2, \ldots, 0.9, 1$$

- AUC integrates over all possible thresholds / predictions you could make
- AUC = P(Randomly-chosen positive is ranked more highly than a randomly-chosen negative)
- AUC close to 1 is optimal, AUC close to 0.5 is no better than chance

Trade-off between measures and mis-classification costs

- Resolving trade-offs is hard
- Beware of implicit resolutions – e.g. all weights equal (accuracy, ...)
- **Make decisions based on the use case** of the prediction algorithm. Unclear trade-offs often a warning sign of unclear use case.
- **Think of consequences of prediction**!
- Examples of different trade-off situations

Mass screening for disease	First diagnosis of disease	Treatment decision
- N (healthy) >> P (disease) - FP will lead to costs of further diagnosis - FN will leave people undiagnosed/untreated	- Patient presents with problems - FP will lead to further tests - FN will leave patient undiagnosed/untreated	- Should patient be treated? - FP will lead to treating someone that will not respond - FN will not use treatment although would have responded

How to find out if your model is doing well?

→ **Model validation**

Source: Shutterstock.com

How do we obtain performance measures?

- Distinguish between ...
  - Model evaluation for model selection or model improvement (“tuning”)
    FROM
  - Final model evaluation

- Mixing model selection/improvement with final evaluation tends to overestimate the performance

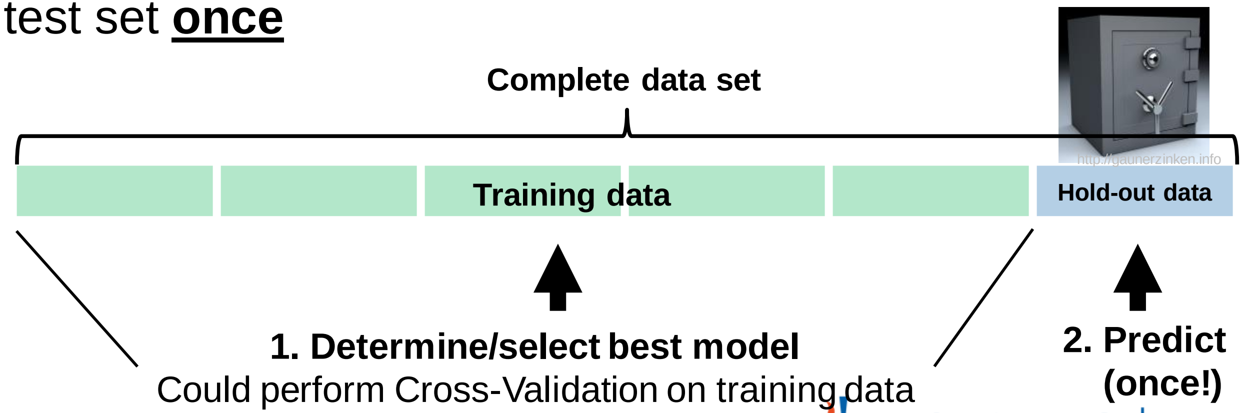
Hold-out test sets are the gold standard

for model evaluation

- Training and testing on the same data set will overestimate performance → **Don't do this!**

Training\Test

- The **gold standard** is to evaluate the trained, optimized and selected model on a hold-out test set **once**



1. Determine/select best model
Could perform Cross-Validation on training data

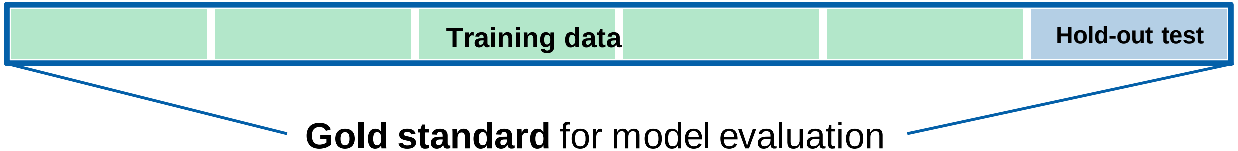
2. Predict (once!)

Use n-fold cross-validation for tuning and hold-out testing for evaluation

1) Cross-Validation for **model selection** (and/or tuning)

				Validation
			Validation	
		Validation		
	Validation			
Validation				

2) Train best model and **evaluate performance** on hold-out test set

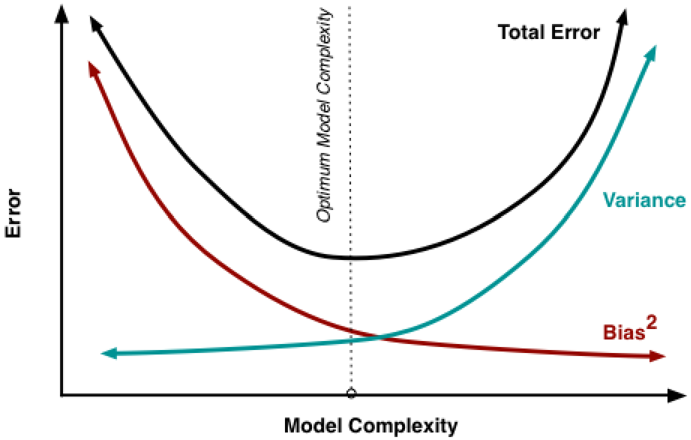


Gold standard for model evaluation

Some words of caution

From: *Validation in prediction research: the waste by data splitting* by author *Ewout W. Steyerberg*

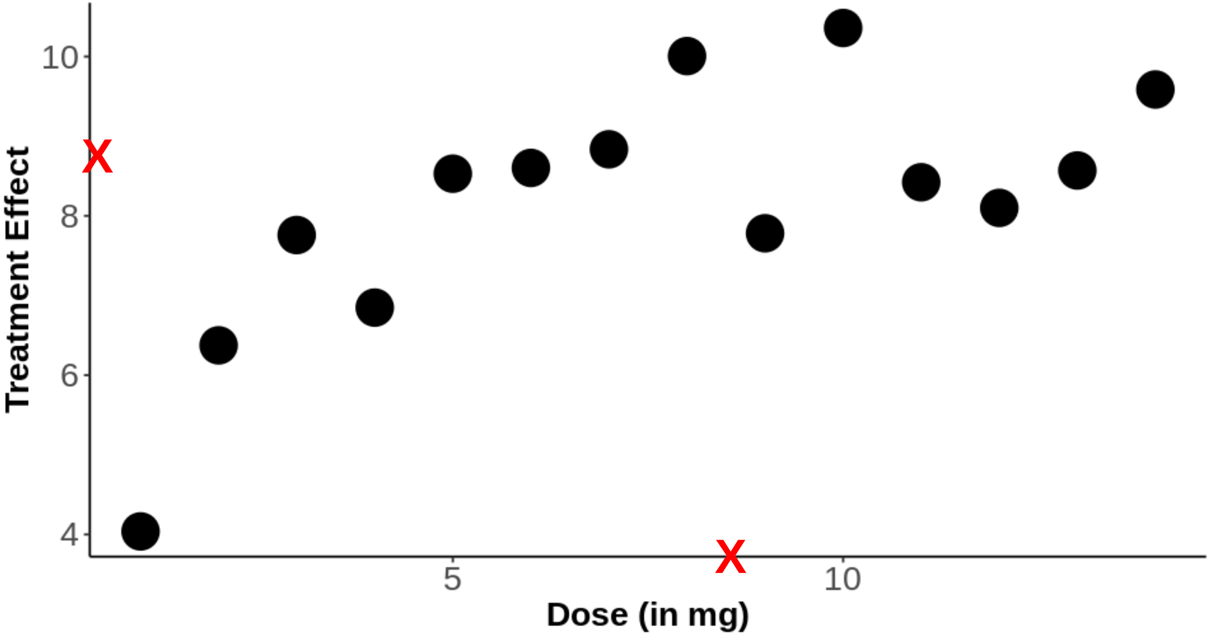
- In the absence of sufficient sample size, independent validation is misleading and should be dropped as a model evaluation step.
  - Independent validation in small samples, such as with 3 events among 10 patients, is merely window dressing.
  - Validation studies should have at least 100 events to be meaningful. In Big Data, heterogeneity in model performance should be quantified rather than average performance.
- In small samples, we should accept that small size studies on prediction merely are exploratory in nature. We should use cross-validation and bootstrapping as more efficient approaches to assess average model performance.



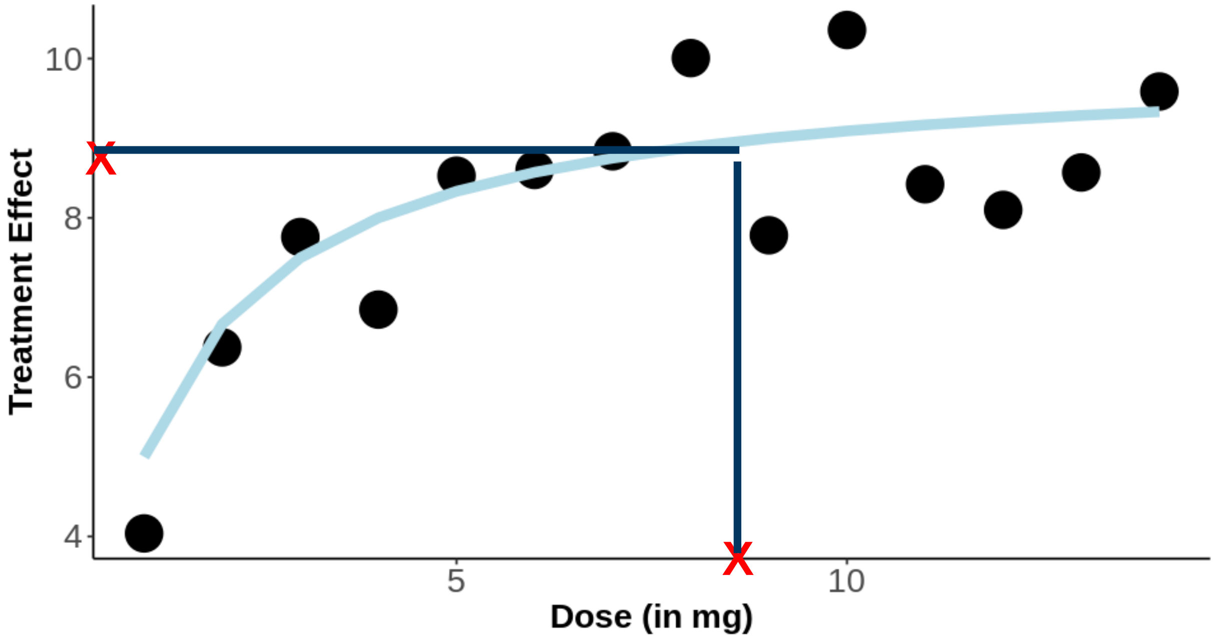
Bias-Variance Tradeoff

and how Machine Learning finds the balance

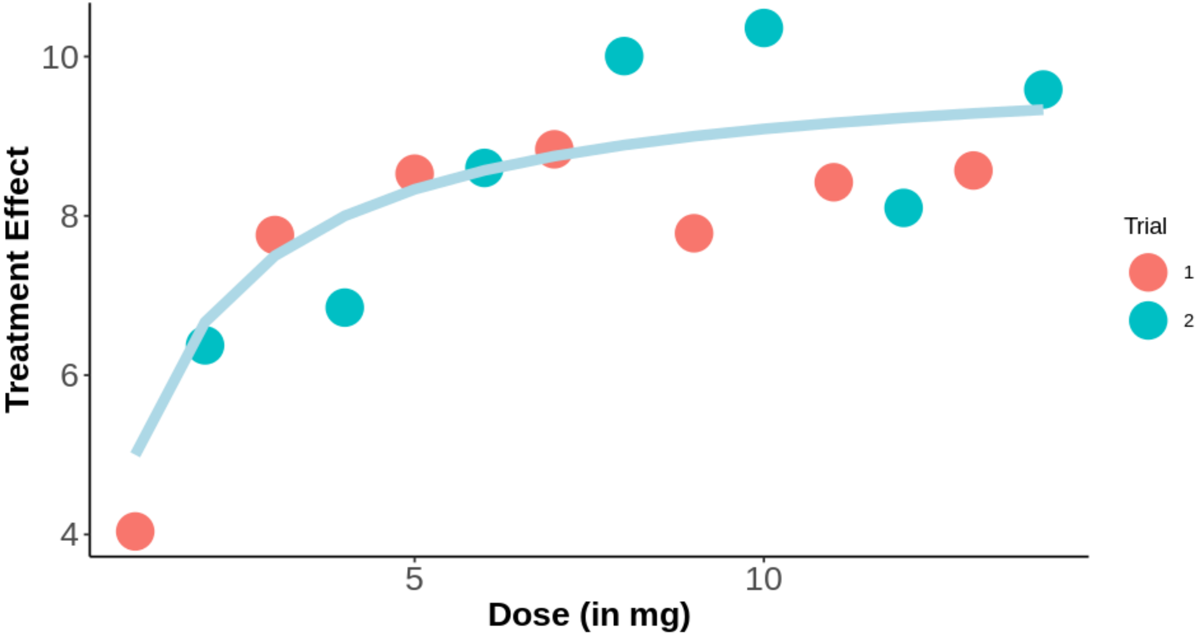
Bias and variance tradeoff



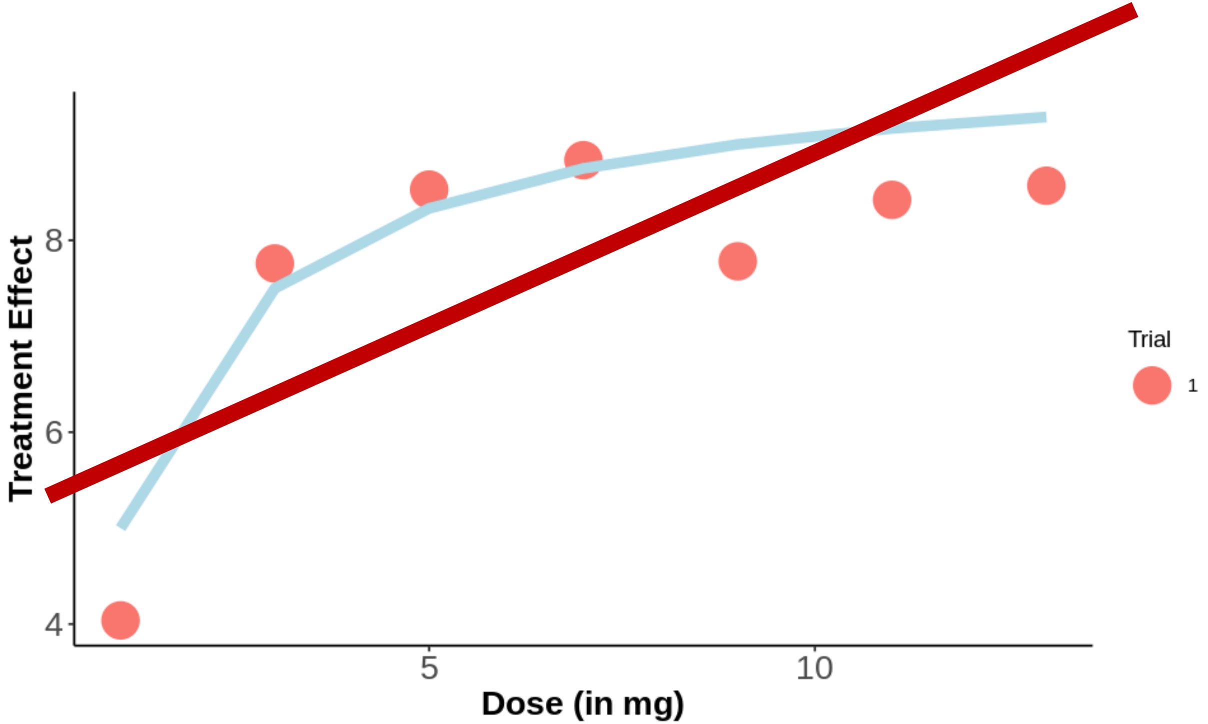
Bias and variance tradeoff



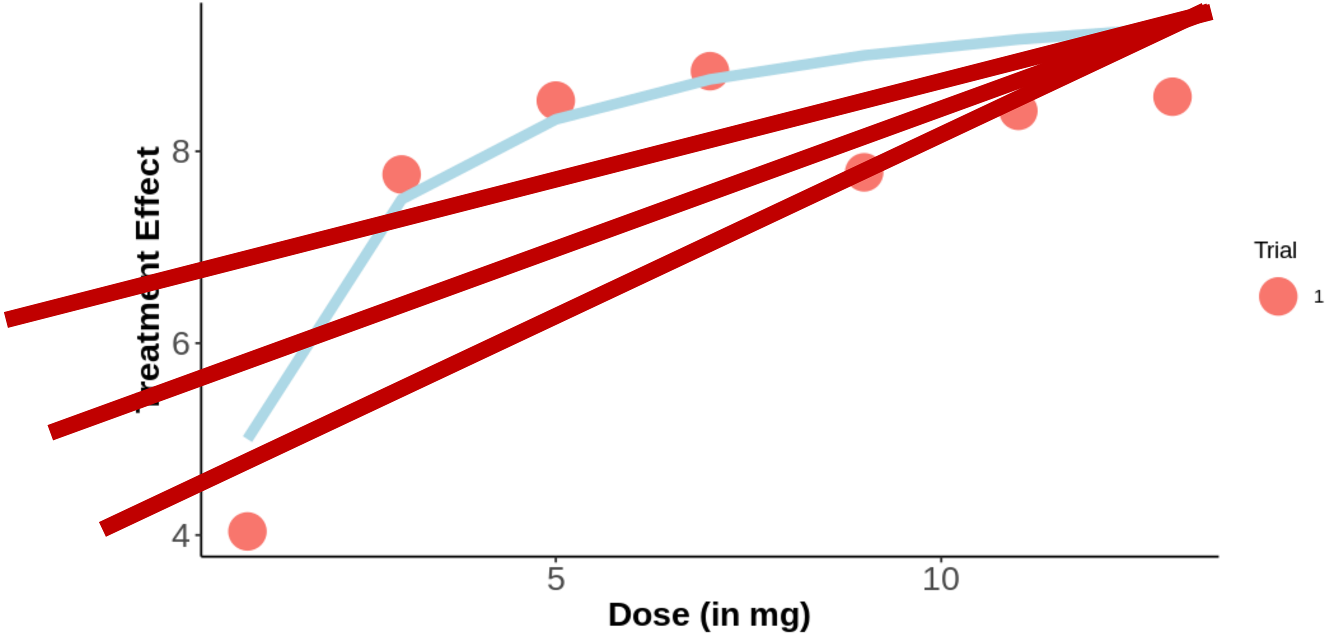
Bias and variance tradeoff



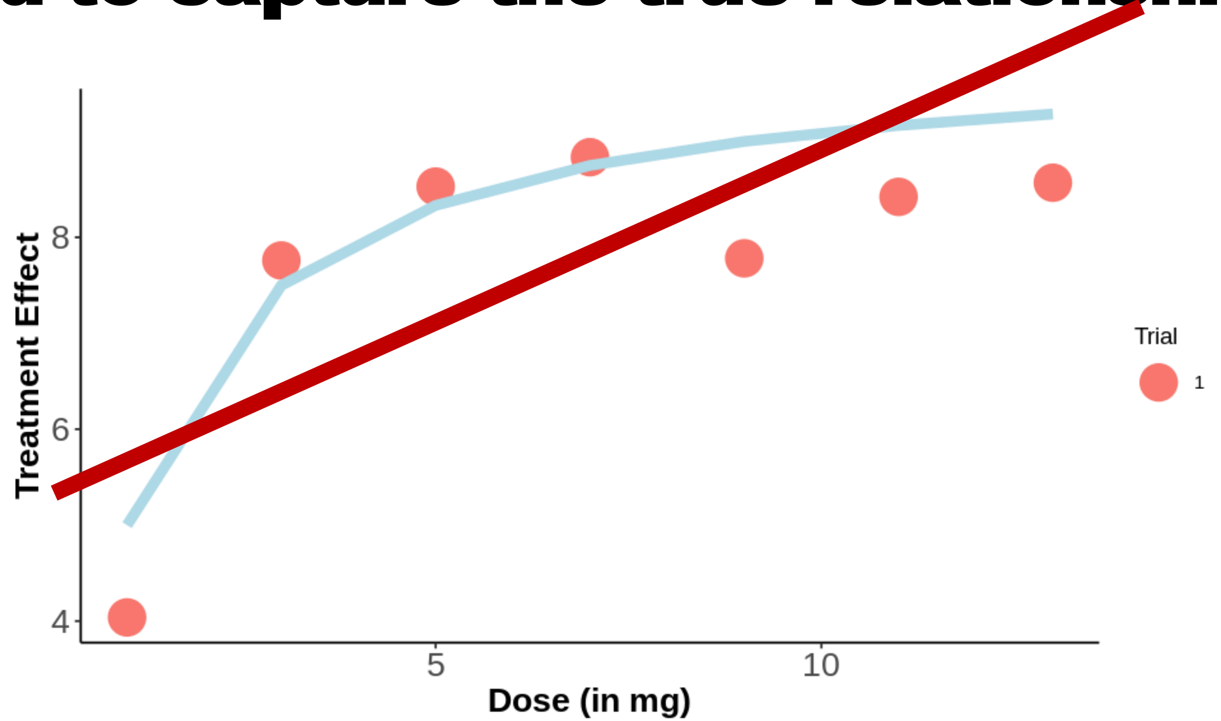
Model 1: Linear model



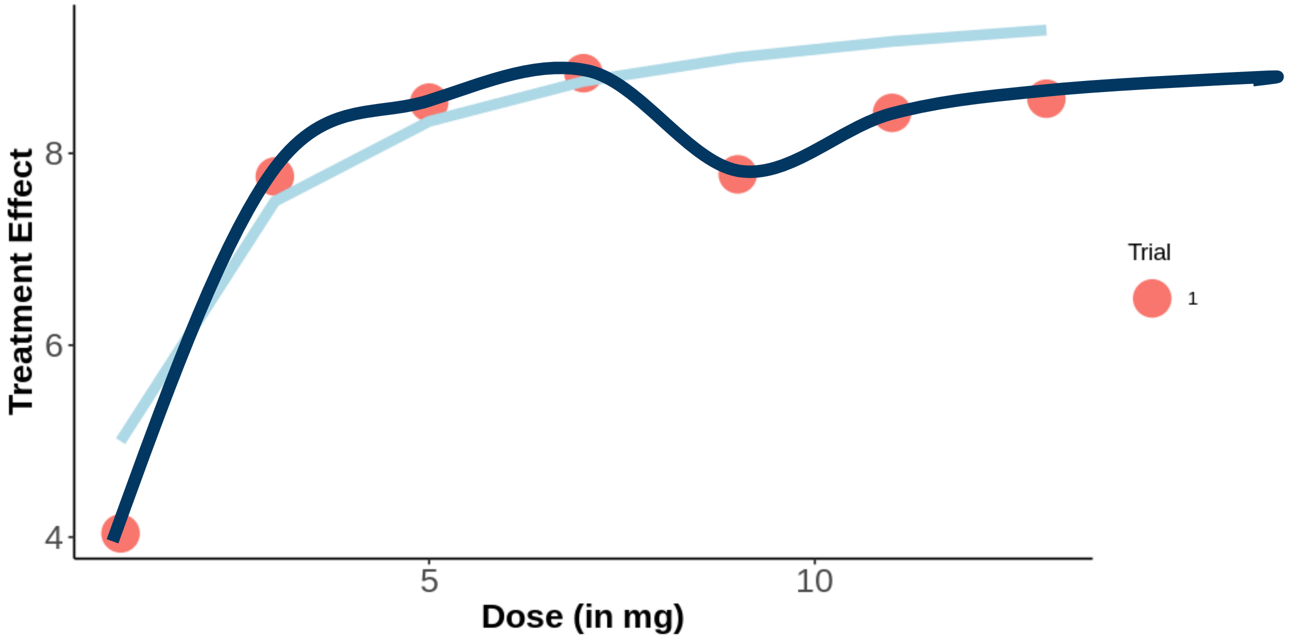
Bias and variance tradeoff



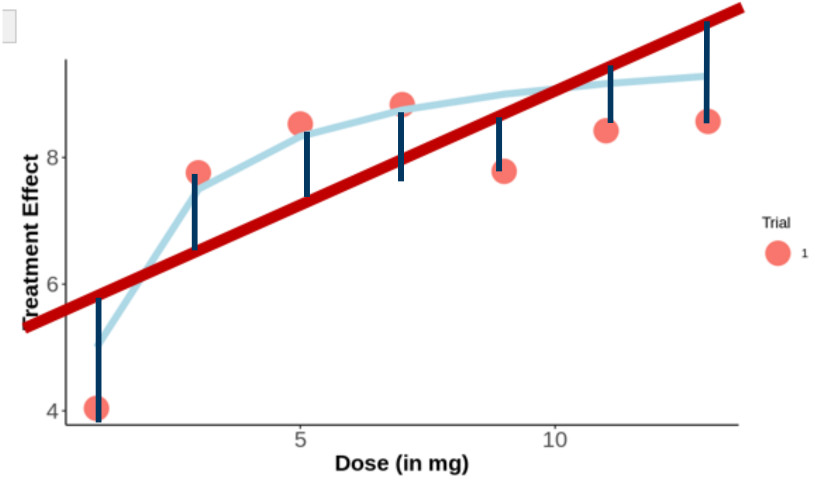
Bias: The inability of a machine learning method to capture the true relationship

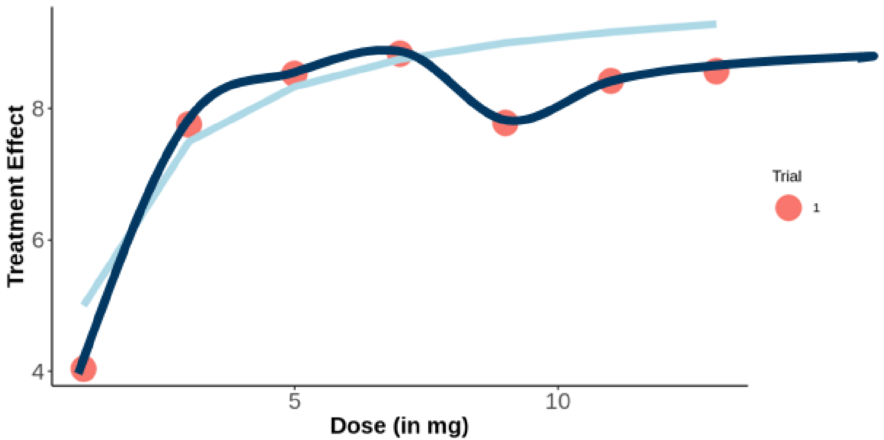


Model 2: Flexible line

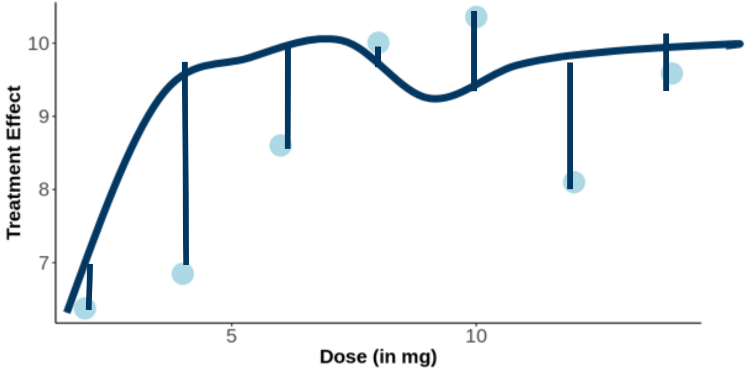


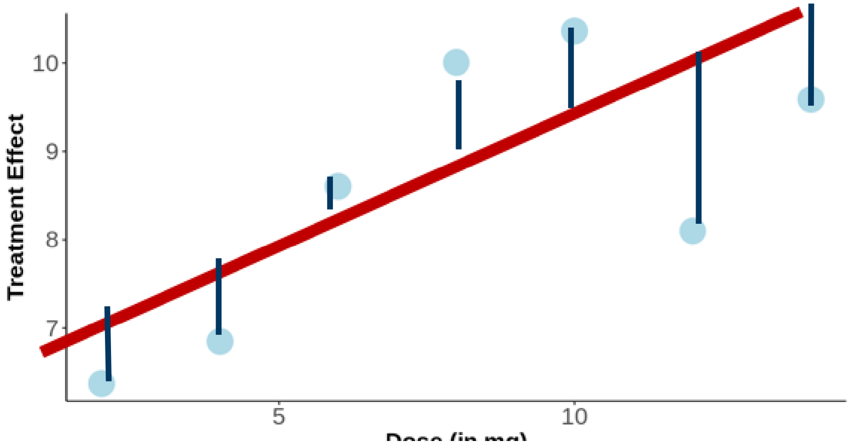
How do the two models compare?





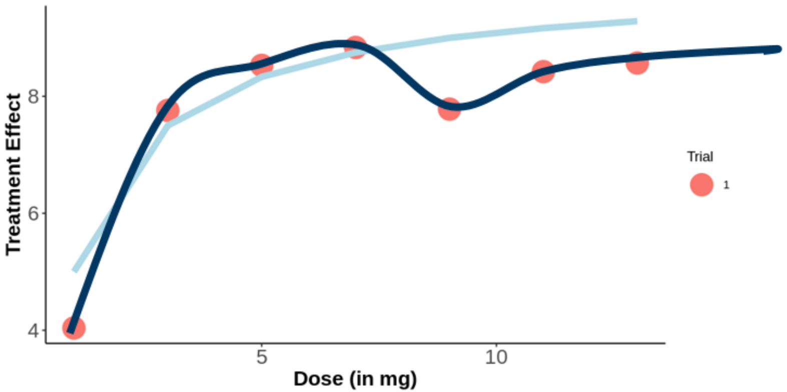
On the data from the other trial, the linear model wins!

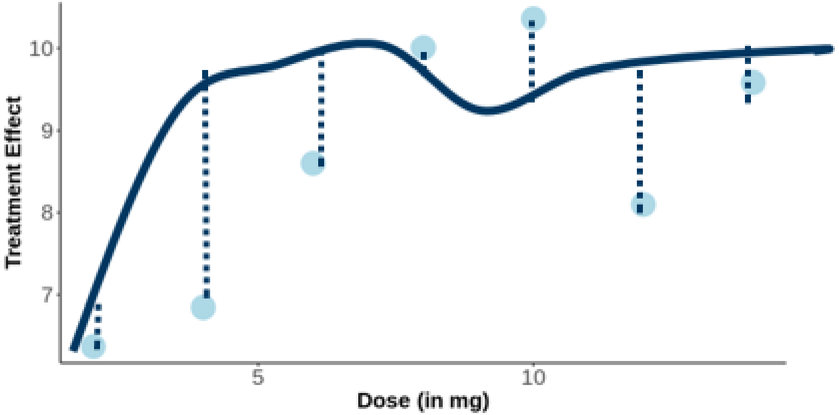




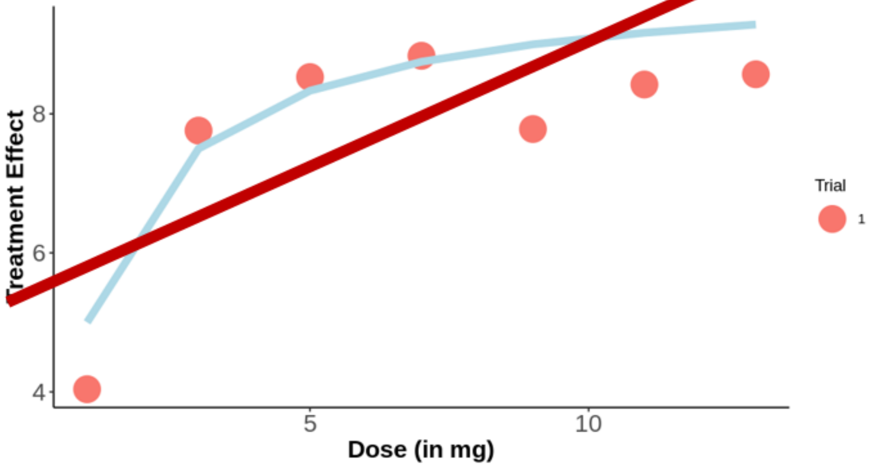
Variance: the difference in fits between data sets

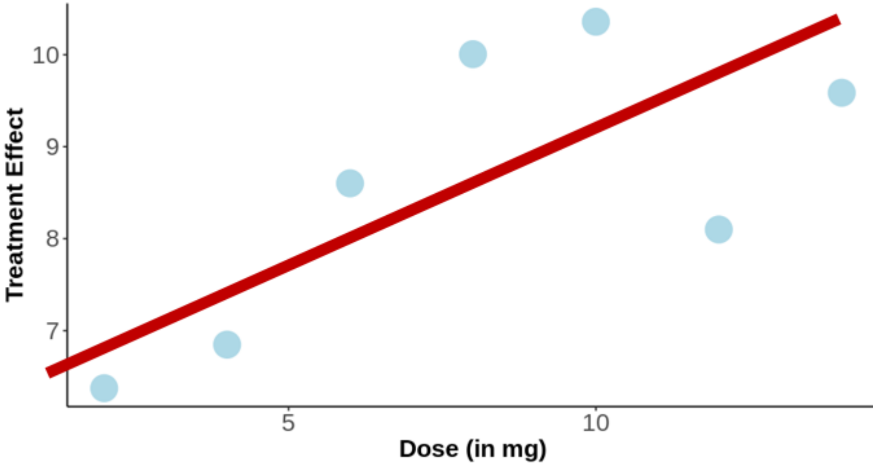
The “flexible model” has low bias, but high variability





The linear model has high bias, but low variability



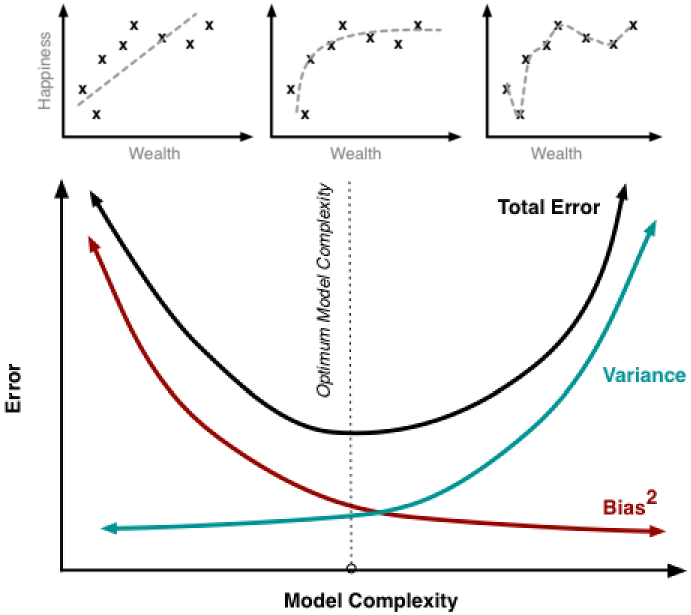


Summary of bias/variance tradeoff

- Bias: The inability of a machine learning method to capture the true relationship
- Variance: the difference in fits between data sets

The ideal algorithm has low bias, i.e. is able to accurately describe the true relationship. It should also have low variability, such that is produces consistent predictions across different datasets.

Over-fitting can be understood as bias / variance trade-off



$$Err(x) = E\left[(Y - \hat{f}(x))^2\right]$$

$$Err(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

Most machine learning methods use regularization to “tune” along the bias-variance axis

- lasso, ridge regression → penalty parameter (lambda)
- nearest neighbor → n (number of neighbors to take into account)
- SVM → C (cost)
- decision trees → pruning criteria
- random forests → tree depth
- ...

- Often these hyper-parameters are tuned empirically
  (be aware of risk when tuning towards test set performance)

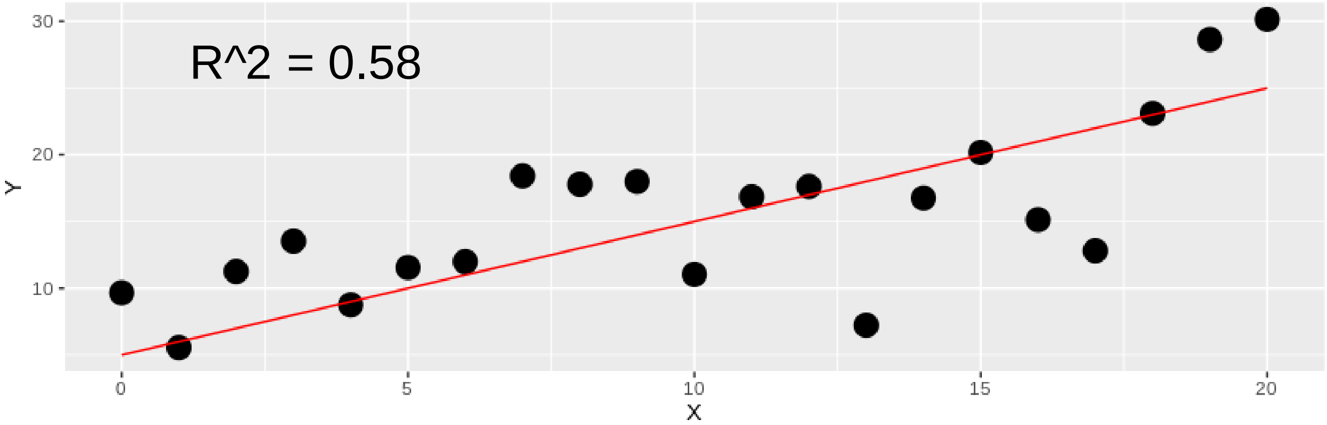
References

- Hastie, Tibshirani – Elements of Statistical Learning https://web.stanford.edu/~hastie/ElemStatLearn/
- Fawcett, T., 2006. An introduction to ROC analysis. *Pattern recognition letters*, *27*(8), pp.861-874. http://people.inf.elte.hu/kiss/12dwhdm/roc.pdf

The bootstrap

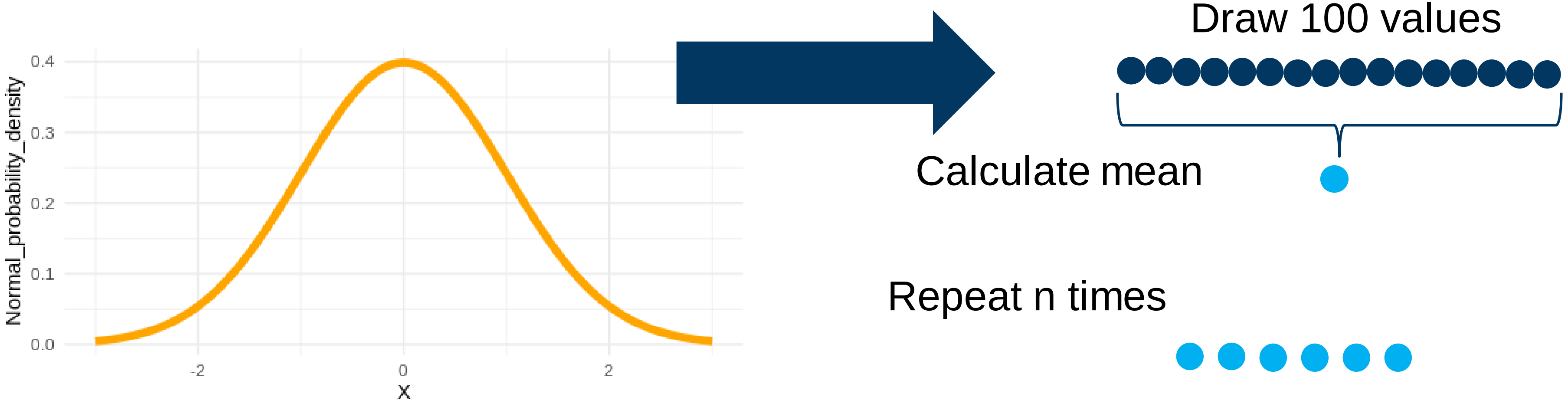
The bootstrap

- Commonly used flexible and powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or Machine learning method
- For example: deriving confidence intervals on a single parameters



Deriving confidence intervals

If we took many samples from the population, 95% of the confidence intervals build using those samples would include the true mean



Draw 100 values

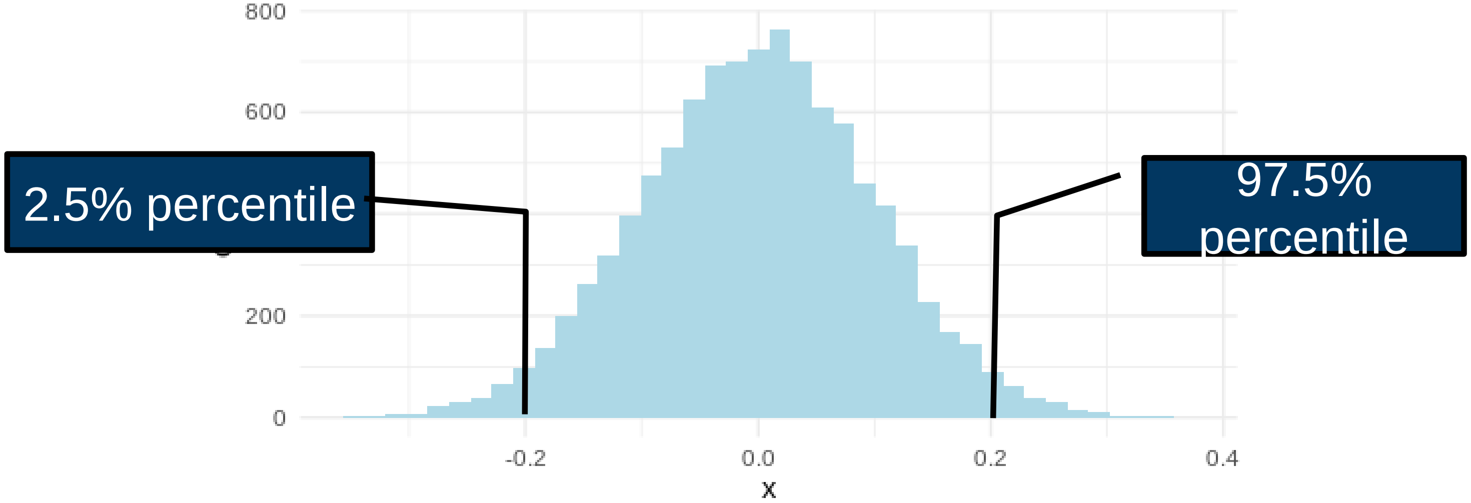
Calculate mean

Repeat n times

This generates a **sampling distribution of means**

Slide idea from Matthew E. Clapham

Derive quantities of interest from resampling distribution



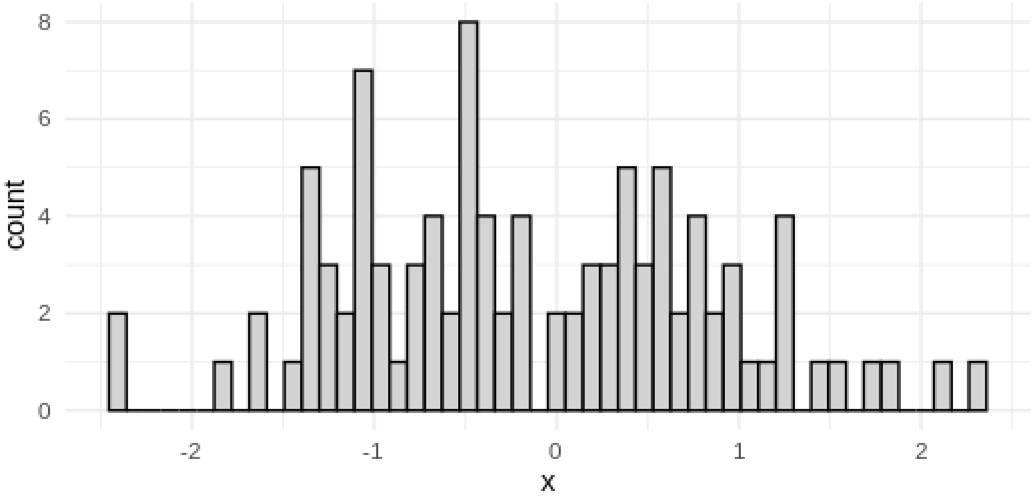
2.5% percentile

97.5% percentile

Back to reality …

We never know the true population parameters, so we cannot apply above's method!

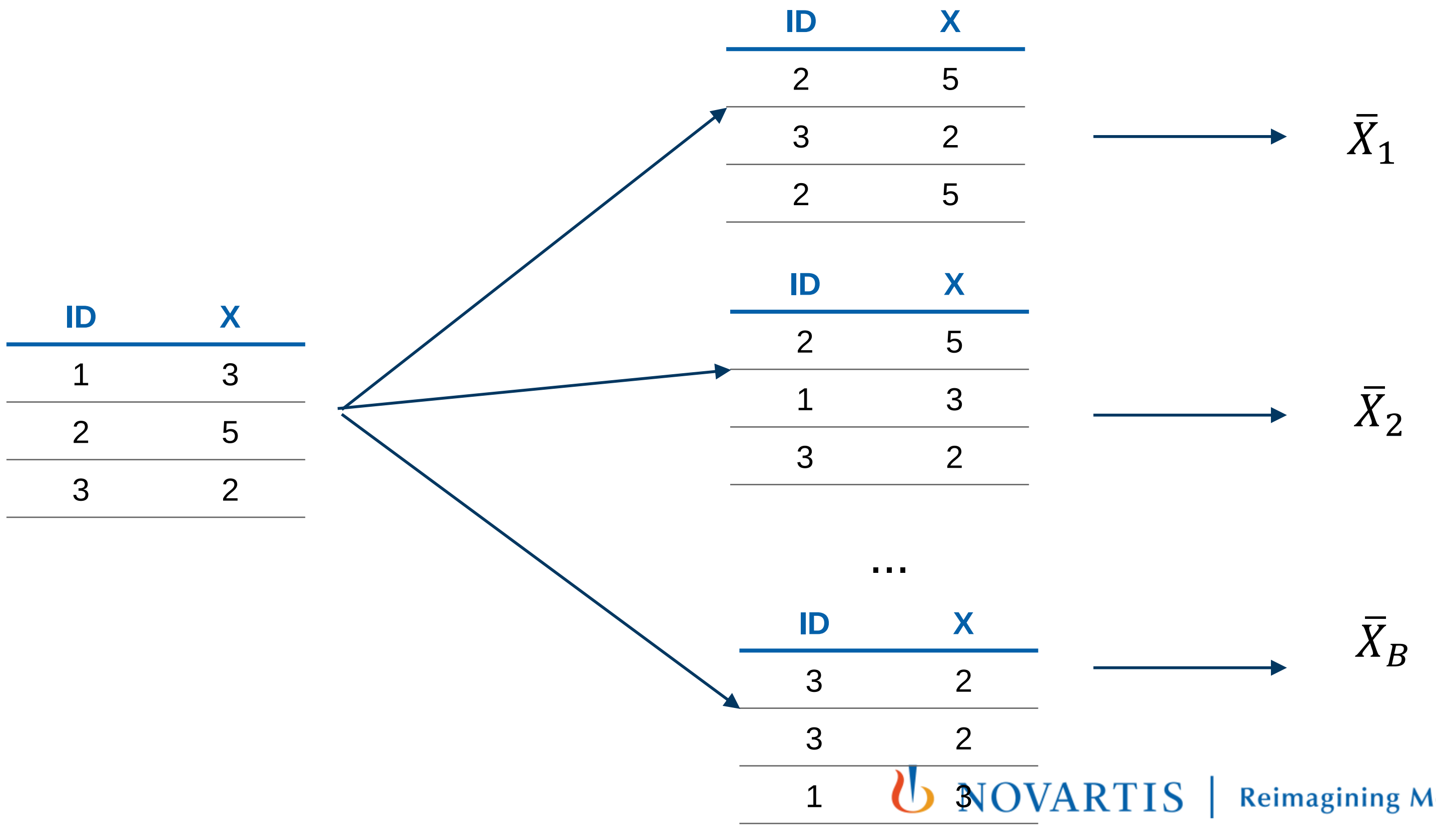
We only ever have a single sample!



Back to reality (2)

- Bootstrap approach allows us to use a computer to mimic the process of obtaining independent samples from the population
- We cannot repeatedly obtain independent data sets from the population, so instead we obtain distinct data sets by repeatedly sampling observations from original data set **with replacement**
- Each bootstrap data set is the same size as our original dataset
  - Some observations may appear more than once in a given bootstrap dataset
  - Some observation will not appear at all

Simple bootstrap example



ID	X
1	3
2	5
3	2

ID	X
2	5
3	2
2	5

$\bar{X}_1$

ID	X
2	5
1	3
3	2

$\bar{X}_2$

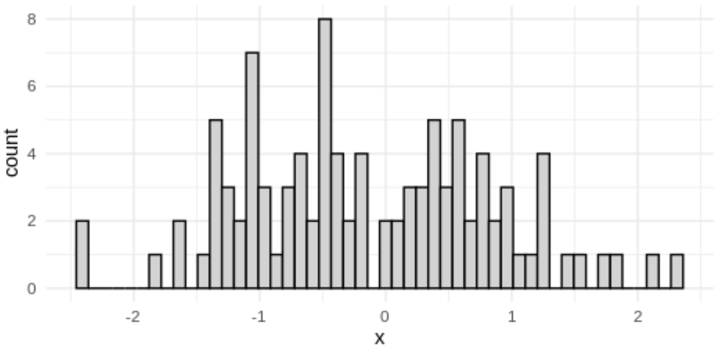
…

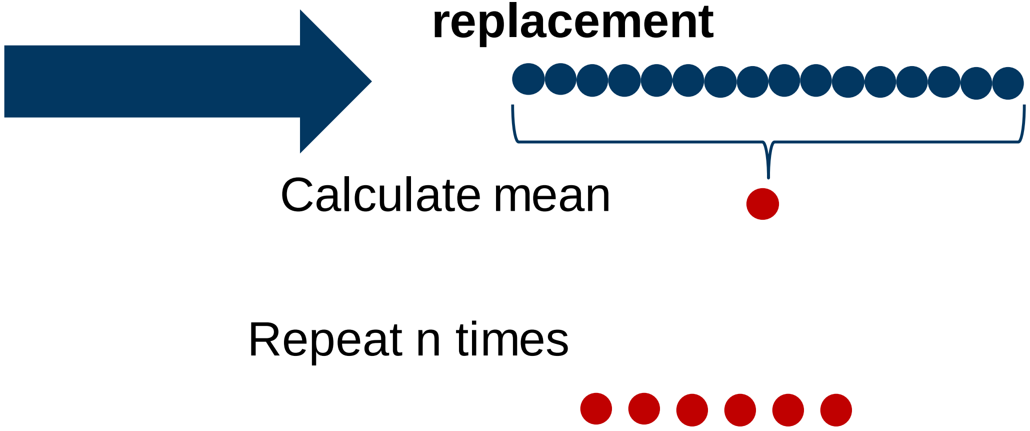
ID	X
3	2
3	2
1	3

$\bar{X}_B$

Bootstrapping the single sample we have

The single sample is the best (and only) information we have about the population





Sample values **with replacement**

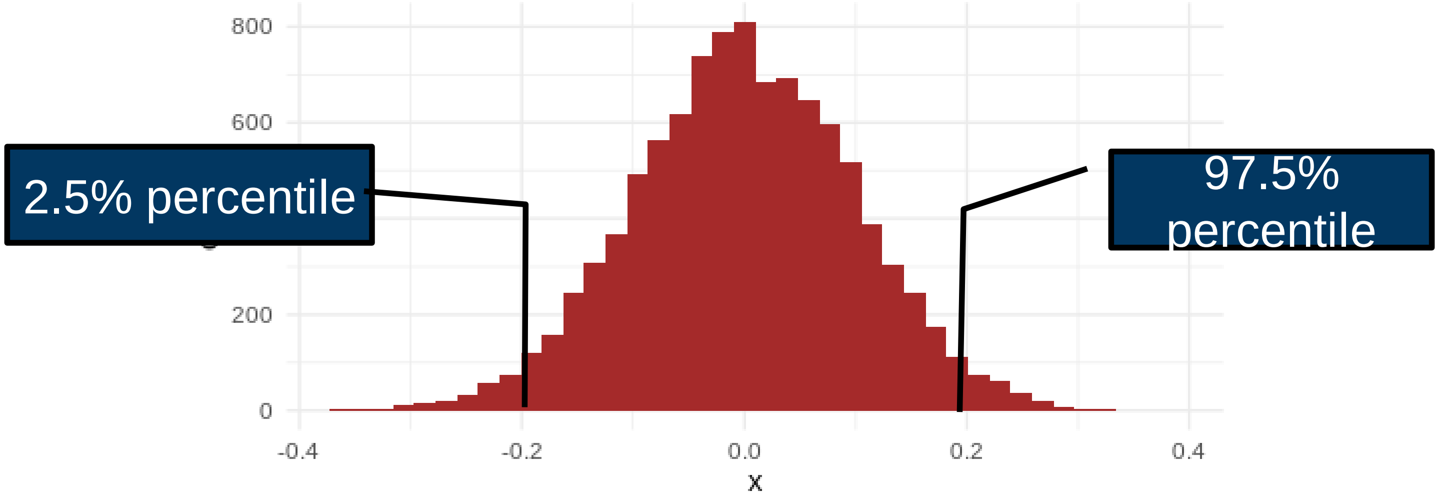
Calculate mean

Repeat n times

This generates a **sampling distribution of means**

Slide idea from Matthew E. Clapham

Inference on bootstrapped resampling distribution



2.5% percentile

97.5% percentile

Uses of the bootstrapping

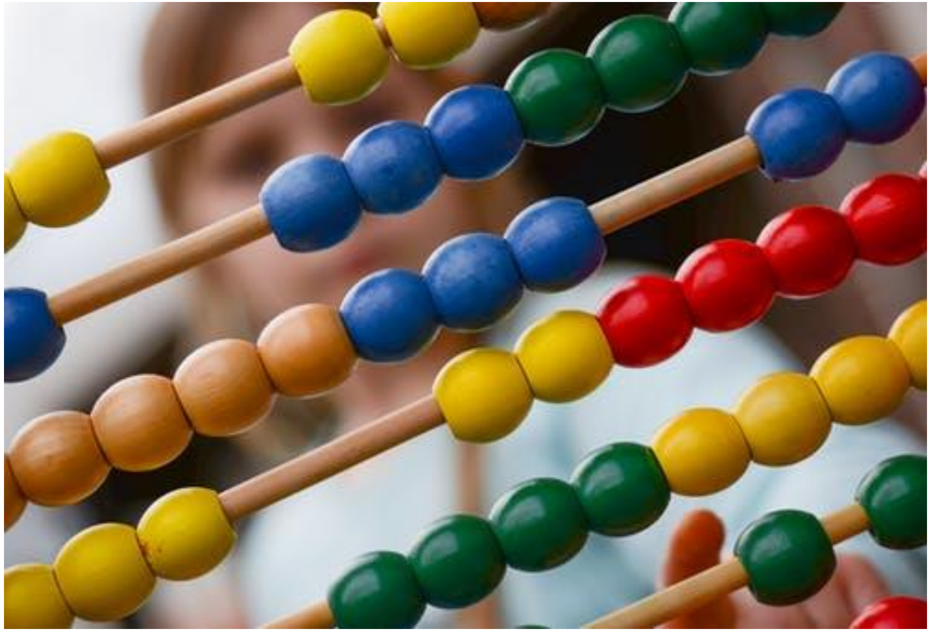
- Estimating statistical parameters where data are non-normal
- Estimating parameters that lack a standard calculation (e.g. 95% CI on R-squared)
- Can also be used to estimate the prediction error
- Essential to the idea of bagging and random forests
- Great to assess the consistency of your methods

Conclusion and looking back

- We covered a lot
  - Overview of machine learning
  - Performance measures (2x2, accuracy, TPR, FPR, PPV, NPV, AUC, ...)
  - Performance evaluation strategies (hold-out, cross-validation, ...)
  - Overfitting / bias-variance tradeoff
  - The bootstrap
- Many topics not covered here
  - Evaluating multi-class predictions
  - Evaluating continuous predictions
  - Evaluating multi-dimensional / longitudinal / correlated or grouped predictions
  - Learning curves (performance vs. # of training samples)
  - Calibration of probabilistic predictions (calibration curves)
  - Taking predictors apart to understand (opening the black box)
  - ...
- Many predictive problems pose hard engineering problems (i.e. in practice) around seemingly simple concepts (i.e. in theory)

References

- Hastie, Tibshirani – Elements of Statistical Learning
  https://web.stanford.edu/~hastie/ElemStatLearn/
- James, Witten, Hastie, Tibshirani - An Introduction to Statistical Learning
  https://www.statlearning.com/



Machine Learning Techniques

NOVARTIS | Reimagining Medicine

Agenda (continued)

- Machine Learning techniques
  - Penalized regression
  - Trees, Bagging, Random forests, and Boosting
  - Finding subgroups
  - Unsupervised learning

Penalized Regression

Multiple Linear Regression

- Aim: Modelling of (linear) relationship between outcome and predictors:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i$$

  with

  - y_i $(i = 1, \dots, n)$: outcome
  - x_{ik} $(\mathrm{k} = 1, \dots, p)$: covariate values for observation i
  - $\beta_0, \beta_1, \dots, \beta_p$: regression parameters
  - ϵ_i: error term / residual

- Least squares solution by minimizing:

$$|| y \ - X\beta||^2 = \sum_{i=1}^{n} \left(y_i - \left(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} \right) \right)^2$$

General idea of penalized regression

- Take a multiple linear regression model and add a “penalty term”.
- Penalization of the regression parameters:
  => Not a “full-grown” model anymore
- Advantages:
  - Improvement in terms of prediction (making use of the bias variance trade-off).
  - Allows estimation of regression parameters in the p>n case.
  - It is still a parametric model (no “black box”).

Ridge regression

Definition:

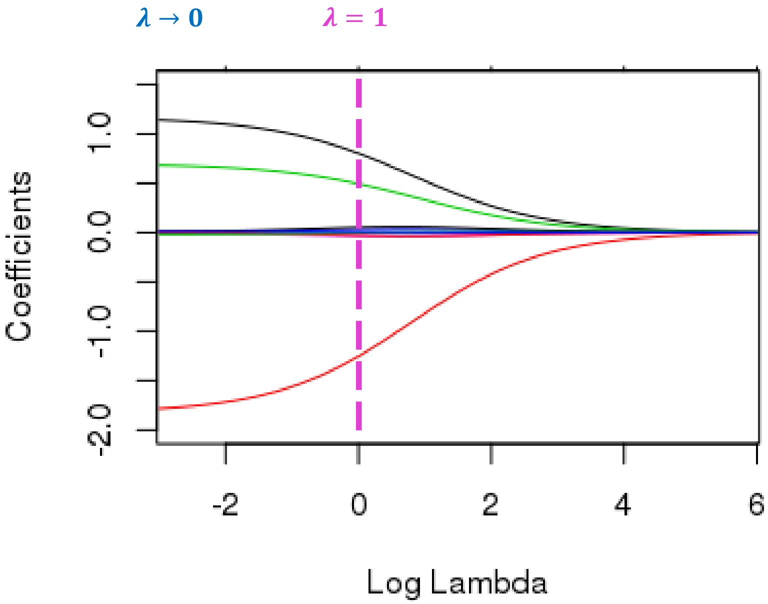
$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}$$

$$= \underset{\beta}{\operatorname{argmin}} \left\{ \underbrace{|| y - X\beta||^2}_{\text{Least Squares part}} + \underbrace{\lambda \sum_{i=1}^{n} \beta_i^2}_{\text{Shrinkage } L_2\text{-penalty}} \right\}$$

λ controls the weight of the penalty:

- $\lambda \to \infty \Rightarrow \hat{\beta}^{\text{ridge}} = 0$
- $\lambda \to 0 \Rightarrow \hat{\beta}^{\text{ridge}} = \hat{\beta}^{OLS}$ (=least squares estimate)

Parameter paths



Ridge solution paths of a linear regression model

Least Squares estimate (excluding intercept) is at $\lambda \to 0$.

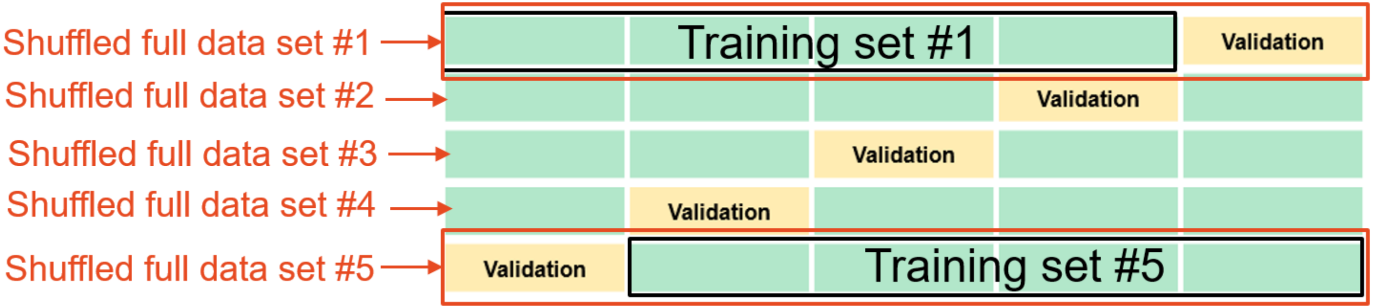
Some notes on ridge regression

- In the penalty term, β_0 is not included to make it robust against adding a constant term to y.
  => Center y (or estimate β_0 by $\bar{y} = \frac{1}{n}\sum_i y_i$) and then estimate the ridge coefficients.
  => $\boldsymbol{X}$ includes only p columns.
- Scaling of the predictors affects the ridge solutions.
  => Standardize the predictors

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_{ij} - \bar{x}_j\right)^2}}$$

How do we find “the best” λ?

Cross-Validation (e.g. 5-fold)



Shuffled full data set #1 → Training set #1 | Validation

Shuffled full data set #2 → Validation

Shuffled full data set #3 → Validation

Shuffled full data set #4 → Validation

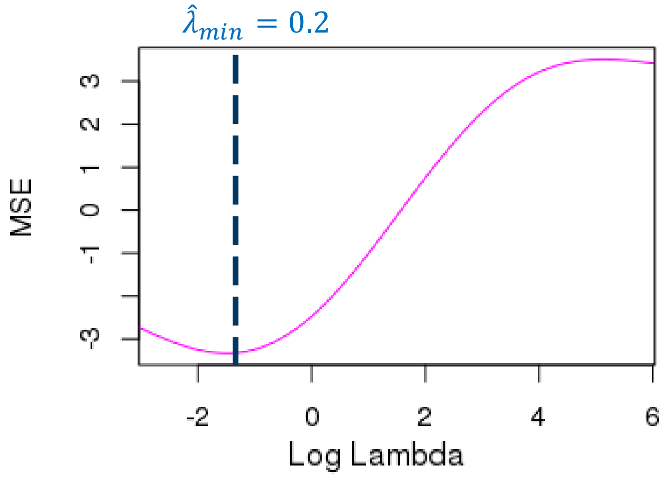
Shuffled full data set #5 → Validation | Training set #5

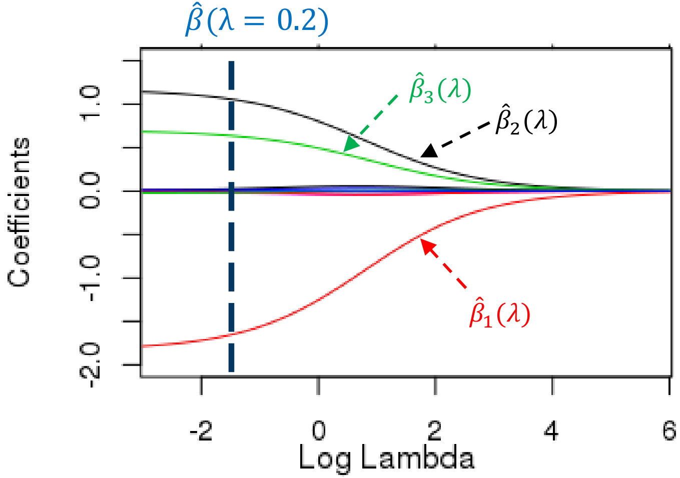
- For each $\lambda \in \{0.001, 0.01, 0.1, 0.5, \ldots, 10\}$ over some grid of values do the following:
  - For each shuffle find solution of $\min\{\, || y - X\beta ||^2 + \lambda \sum_{i=1}^{n} \beta_i^2 \,\}$ on training set and predict on validation set.
  - Calculate pooled error $\mathrm{MSE}(\lambda) = || y - X\beta ||^2$ over all validation sets.
- Find λ that minimizes the pooled $MSE(\lambda)$.

Ridge regression coefficient estimate

Regression model with 10 covariates:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_{10} x_{10} + \varepsilon_i$$





Ridge estimate (excluding intercept): $\hat{\boldsymbol{\beta}} = ($-1.66, 1.06, 0.64, 0.01, 0.002, -0.02, 0.02, -0.01, -0.02, 0.01$)$

Lasso Regression
(Least absolute shrinkage and selection operator)

The Lasso Page

L1-constrained fitting
for statistics and data mining

(source: http://statweb.stanford.edu/~tibs/lasso.html)

Lasso Regression (Tibshirani, 1996)

Definition:

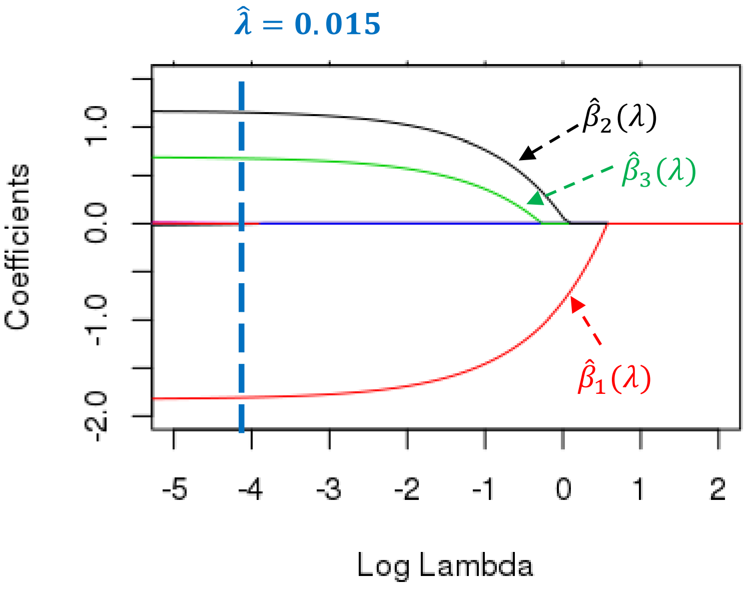
$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

$$= \underset{\beta}{\operatorname{argmin}} \underbrace{|| \, y \; - X\beta ||^2}_{\text{Least Squares part}} + \underbrace{\lambda \sum_{i=1}^{n} |\beta_i|}_{\text{Shrinkage } L_1\text{-penalty}}$$

λ controls the weight of the penalty:

- $\lambda \to \infty \;\Rightarrow\; \hat{\beta}^{\text{lasso}} = 0$
- $\lambda \to 0 \;\Rightarrow\; \hat{\beta}^{\text{lasso}} = \hat{\beta}^{OLS}$ (=least squares estimate)

Lasso paths



LASSO solution paths of a linear regression model

Best λ found through Cross-validation: $\hat{\lambda} = 0.015$.

$\hat{\boldsymbol{\beta}} = (-1.78, 1.12, 0.65, 0, 0, 0, 0, 0, 0, 0)$

Some parameters will be set to 0. ➡ Variable selection!

Summary LASSO & Ridge Regression

- Standardize the predictors and center the response!
- Lasso and Ridge regression make use of the Bias-Variance tradeoff.
- Main advantage of LASSO: variable selection.
- Neither ridge regression nor the lasso will universally dominate the other.
- If there are only few “true” predictors, LASSO may be the better choice.
- Cross-validation may be used to determine the final model.

Graphical representation - preparation

Ridge and lasso regression can be written as follows:

- Ridge regression:

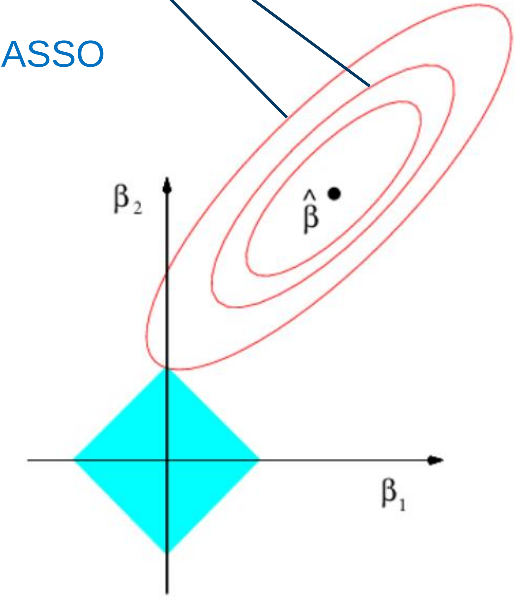
$$\underset{\beta}{\operatorname{argmin}} || y - X\beta||^2 \ , \qquad s.t. \ |\beta|_2 \leq C$$

- Lasso regression:

$$\underset{\beta}{\operatorname{argmin}} || y - X\beta||^2 \ , \qquad s.t. \ |\beta|_1 \leq C$$

There is a direct connection between C and λ.

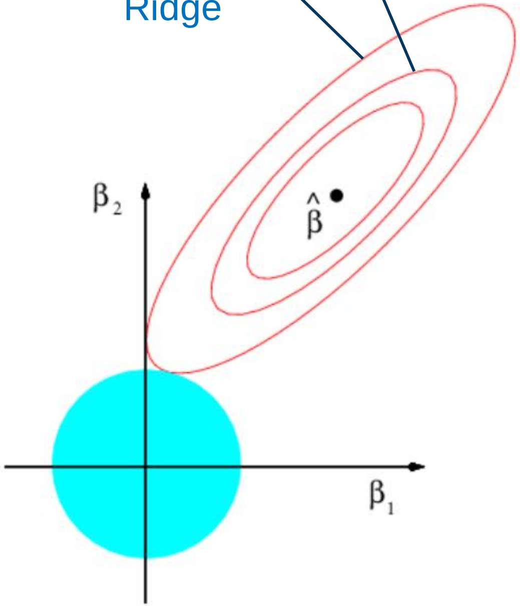
Graphical representation



LASSO

Contours of $|| y \; - X\beta||^2$

$$\underset{\beta}{\text{argmin}} \, || y \; - X\beta||^2 \quad s.t. \; |\beta|_1 \leq C$$



Ridge

Contours of $|| y \; - X\beta||^2$

$$\underset{\beta}{\text{argmin}} \, || y \; - X\beta||^2 \quad s.t. \; |\beta|_2 \leq C$$

Source https://bookdown.org/egarpor/PM-UC3M/lm-iii-shrink.html

Elastic Net (Zou and Hastie, 2005)

$$\hat{\beta}^{\mathrm{EN}} = \underset{\beta}{\operatorname{argmin}} \left\{ \underbrace{||\,y \; - X\beta||^2}_{\text{Least Squares part}} + \lambda \left[\underbrace{(1-\alpha)\sum_{j=1}^{p}\beta_j^2 + \alpha\sum_{j=1}^{p}|\beta_j|}_{\text{Elastic net penalty}} \right] \right\}$$

- If $\alpha = 0$ ➔ Ridge Regression.
- If $\alpha = 1$ ➔ LASSO.

➔ Do Cross-Validation to find the optimal $\alpha \in [0, 1]$.

Main advantage of Elastic Net is that it encourages grouped variable selection (while e.g. LASSO tends to pick only one variable among correlated variables)

Some final notes

- Software: R package “glmnet” allows to implement Ridge, Lasso, Elastic Net.
- Extension to generalized linear models in a straightforward way (by adjusting the likelihood and the link function).
- In case of p>n (more covariates than observations), the OLS estimate cannot be calculated. Adding a penalty term solves the issue.
- Several other extensions available (group lasso, fused lasso, ...)
- Bayesian interpretation of Lasso by implementing Laplace prior distributions for the regression coefficients.

Data example

- Simulated data based on real study data
- Population: Patients with psoriatic arthritis
- Response: American College of Rheumatology 20 (ACR20) response (binary)
- Two groups: active treatment vs. placebo
- Additional covariates/predictors:
  - Patient demographics and other baseline skin characteristics
  - Background characteristics
  - Laboratory values

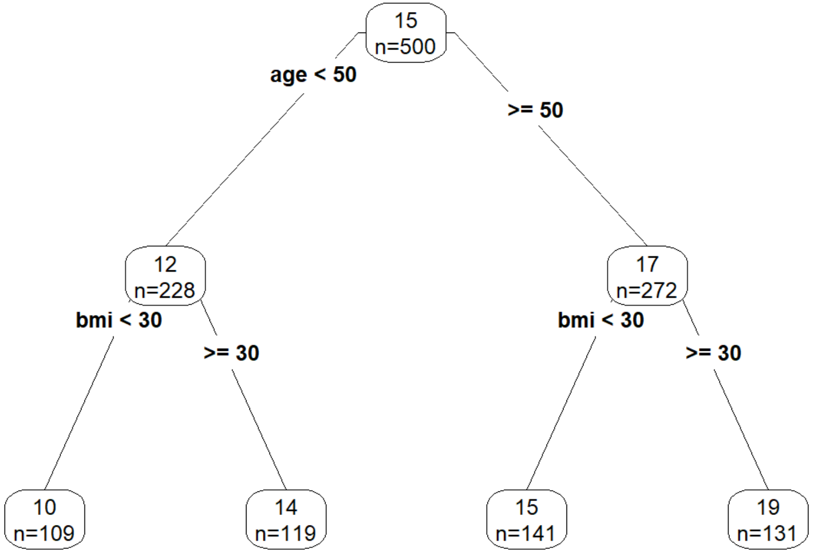
Trees, Bagging, Random forests, and Boosting

NOVARTIS | Reimagining Medicine

Regression trees

- Set-up: continuous response $\boldsymbol{y}$ and predictors $\boldsymbol{x_1}, \dots, \boldsymbol{x_p}$.
- Goal: predict the response based on predictors.
- A tree is defined by (several) splits which result in branches.
- Each split is based on only one variable.
- Result: Predictor space is devided into distinct regions.
- Prediction: “Run” the new observation through the tree. Predict the mean response value of the leaf where the observation ends up.

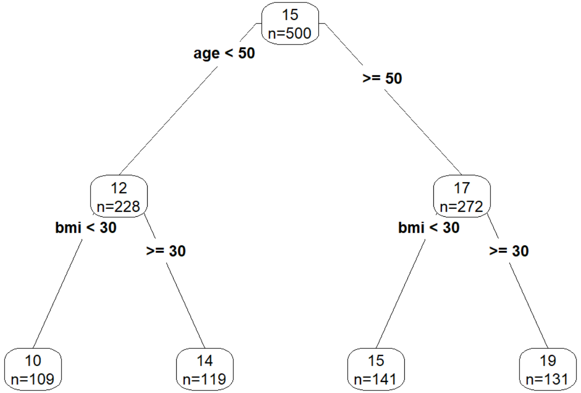
Regression trees



1. First split is done at age < 50 vs. >=50
2. Secon splits are based on BMI.
3. The predictions are the numbers on the top in each box.

Regression trees – algorithm

- Start from the root and go top-down.
- Split the data into two branches:
  - For each predictor $\boldsymbol{x}_j$ (j $= 1, \dots, p$), select the cut-point that leads to greatest reduction of the residual sum of squares (RSS).
  - Select the predictor with the biggest reduction in RSS for the split.
- Repeat the splitting until some stopping criteria is met (e.g., each node has fewer observations than a limit).



Classification trees

- Work basically the same way as regression trees.
- Set-up: Categorical response $\boldsymbol{y}$ and predictors $\boldsymbol{x_1}, \ldots, \boldsymbol{x_p}$.
- Goal: Predict the response category based on the predictors.
- Create a tree as done before.
- Prediction:
  - Run through the tree
  - take the most frequent class (mode) in the final leaf

Classification trees – split criteria

Often used:

Gini index (a measure of total variance across the k classes)

$$G = \sum_{k=1}^{K} \hat{p}_{mk} \left(1 - \hat{p}_{mk}\right)$$

with $\hat{p}_{mk}$ as proportion of observations in the mth leaf which belong to the kth category.

Small if the $\hat{p}_{mk}$ are close to 0 or 1
(most observations in a leaf belong to the same category)

Trees – discussion

- Easy to explain and display
- Can handle non-linearity
- Useful for exploratory and explanatory purposes
- Usually not being used as a stand alone predictive model due to limited prediction accuracy.

Using an Ensemble of models

As stated above, a single tree does not necessary lead to good predictions.

➡ Combine several trees (or more generally, predictions based on some function $f(x)$) and use the average over the trees for prediction.

➡ Reduction of the variance

Examples for ensemble methods:

- Bagging
- Random forests
- Boosting

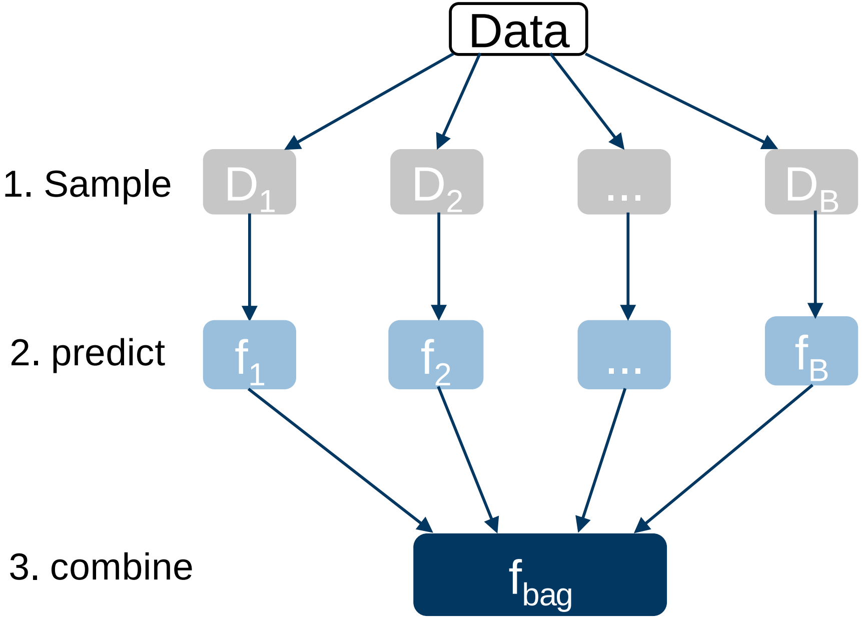
Bagging (Bootstrap Aggregating)

1. Repeatedly sample from training set.

2. Get single predictor of $\hat{f}^{*b}(x)$ from the bth dataset.

3a. For continuous response, average all the predictors as the final $\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x)$.

3b. For categorical response, use majority vote for classification.



Bagging Discussion

- Bagging model
  - Improves accuracy over prediction of a single tree.
  - Hard to interpret the results.
  - Important predictors can be identified by checking the impact on RSS or Gini index or by counting the number of splits which are based on a specific predictor.
- Out-of-bag (OOB) error
  - Can be calculated from the predictions based on OOB observations.
  - Provides a valid estimate of the test error for the model.

Random Forest

(Breiman, 2001)

- Follow the same steps as in bagging.
- However add the following additional step:

  At each split, randomly choose m predictors out of the full set of p predictors.

  (Usually m is set to $\sqrt{p}$ or log2p.)

- The random choice of predictors avoid strong predictors to dominate the lower nodes.

Random Forest – discussion

- Bagging may not reduce the variance enough:
  Strong predictors may dominate the lower level of tree and hence induce correlation among the trees.

- Random forest
  - Random predictor selection as well as bootstrap samples from data.
  - This helps make the trees less correlated.
  - If m=p, then random forest is bagging.
  - Variance importance measures available (to do variables selection)

Boosting

Idea:

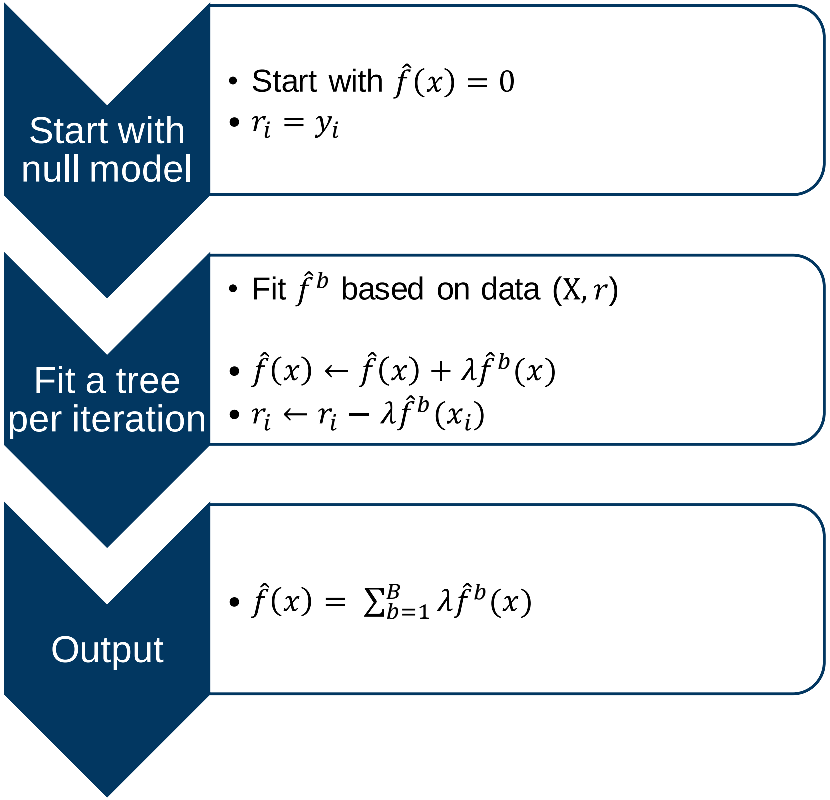
- Sequentially build up a model based on “weak learners”.
- The “ensemble” will create a powerful model.
- Use, for example, trees as learners.

Note the “sequential nature” as compared to bagging and random forests.

Boosting with trees

- Incrementally build the ensemble by training each new model based on the residuals from the previous model.

- Main tuning parameters:
  - Number of trees B.
  - Shrinkage λ controls the learning rate, typical values: 0.01 or 0.001.
  - Number of splits d to control the complexity of the trees. When d = 1, each tree is a stump.



Start with null model

- Start with $\hat{f}(x) = 0$
- $r_i = y_i$

Fit a tree per iteration

- Fit $\hat{f}^b$ based on data (X, r)
- $\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x)$
- $r_i \leftarrow r_i - \lambda \hat{f}^b(x_i)$

Output

- $\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$

Boosting – some remarks

- Boosting comes with great performance in many situations (mostly greater performance to random forests and bagging).
- Many parameters that can be optimized (compared to random forests).
- Several variations available:
  - AdaBoost (Adaptive Boosting, by Freund and Schapire 1997)
  - Stochastic gradient boosting (Friedman, 1999)
  - Gradient boosting (Friedman, 2001)
  - ...

Summary

- Decision trees are simple and interpretable models for regression and classification.
- However, they are often not competitive with other methods in terms of prediction accuracy.
- Bagging, random forests and boosting are good methods for improving the prediction accuracy of trees. They work by growing many trees on the training data and then combining the predictions of the resulting ensemble of trees.
- Random forests and boosting are among the state-of-the-art methods for supervised learning. However their results can be difficult to interpret.

Finding subgroups

NOVARTIS | Reimagining Medicine

Overview

- Setting:
  - One endpoint variable (for example, binary)
  - Two treatment arms (placebo vs. active treatment)
  - Several covariates (demographics, lab parameters, etc.)
- Goal: Finding subgroups of an increased **treatment effect** based on the covariates

General procedure

- Identify most influential covariates:
  - Test interaction between treatment group and covariates or
  - Apply the **virtual twins** method or
  - Implement **causal forests.**
- Do some graphical assessment of potential subgroups: **Funnel plot**
- Define a subgroups based on a decision tree.

Data example - reminder

- Patient population: Patients suffering from psoriatic arthritis.
- Treatment groups: placebo vs. active treatment.
- Endpoint: musculoskeletal endpoint of American College of Rheumatology (ACR) 20 response (binary).
- Covariates (continuous or binary):
  - Patient demographics (age, BMI, etc.)
  - Laboratory variables
  - ...

Test for interaction with treatment

Procedure:

- For each covariate: Fit a regression model with the following predictors:
  - The treatment group.
  - The covariate of interest.
  - An interaction term of the two above.
- Test the interaction term for significance.
- Select all variables below a certain threshold (for example, $p<0.05$).
- Note: This is rather a “univariate” approach!

Virtual twins

(Foster et al., 2011)

General idea: Create a virtual twin for each patient and analyze the difference:

- Select all placebo patients and fit a random forest.
- Select all treatment patients and fit a random forest.
- Predict outcomes using both random forests for all patients ($\hat{Y}_1$ and $\hat{Y}_0$)
- Take the difference $Z = \hat{Y}_1 - \hat{Y}_0$ and fit another random forest on Z.

Causal forests

(Athey et al., 2019)

Fit a random forest to the data (including treatment group and all covariates).

However, use causal trees:

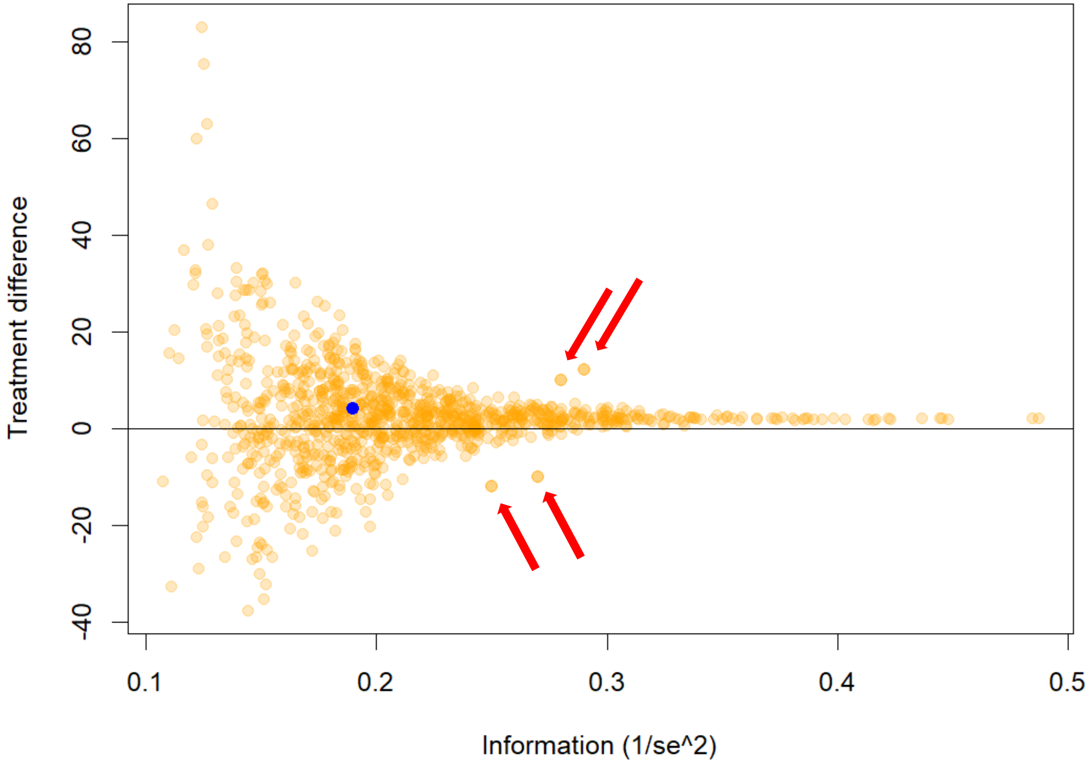
- They work the same way as “normal” trees.
- However, maximize the difference between treatment groups in each split.

After applying virtual twins or causal forests use some variable importance measure to select the most influential covariates.

Funnel plot

- Visualization / tool to assess if there are any potential subgroups at all.
- Idea:
  - Select a set of covariates.
  - Build subgroups; in case of continuous covariates use cut-offs.
  - Calculate the treatment effect for each subgroup.
  - Display all treatment effects in one plot
- Great distances between dots indicate differences in subgroups.

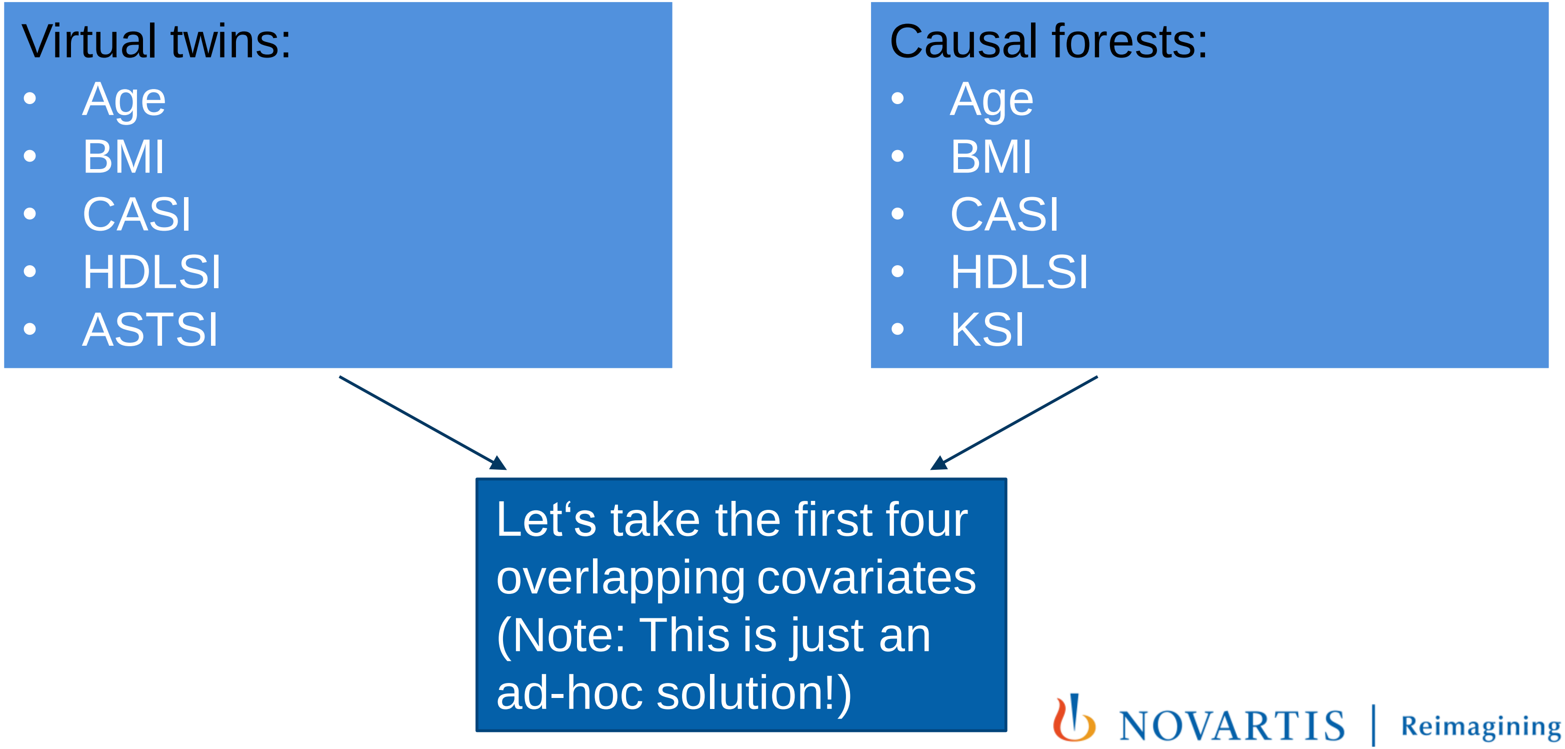
Data example: Funnel plot



The blue dot represents the treatment difference in the overall population.

Potentially interesting subgroups.

Data example: selected variables



Virtual twins:

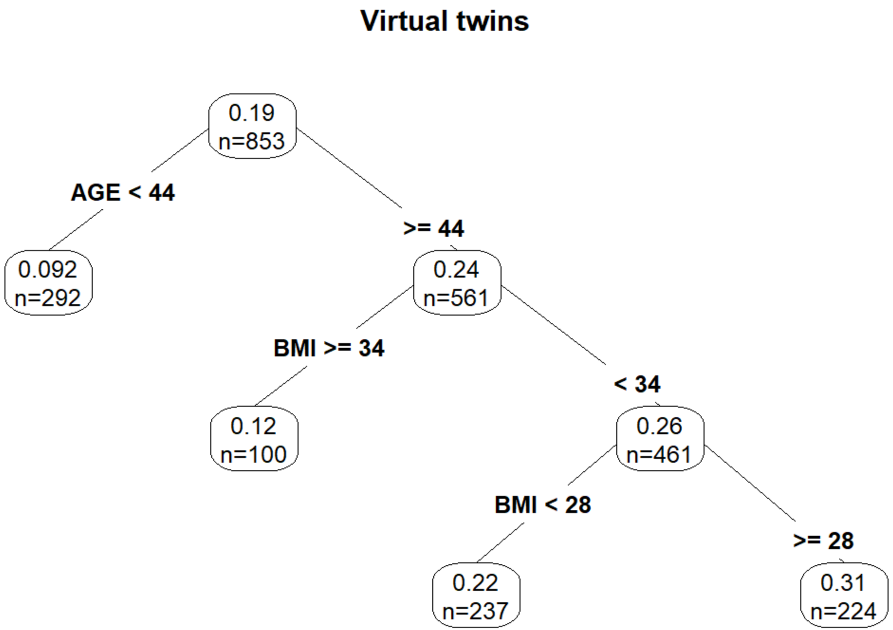
- Age
- BMI
- CASI
- HDLSI
- ASTSI

Causal forests:

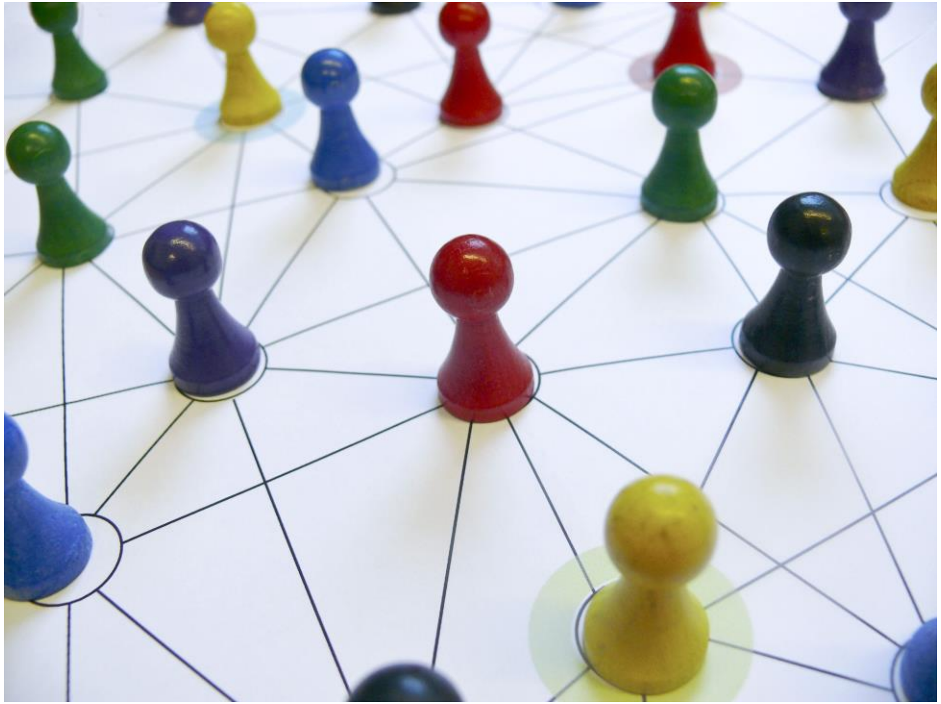
- Age
- BMI
- CASI
- HDLSI
- KSI

Let‘s take the first four overlapping covariates (Note: This is just an ad-hoc solution!)

Data example: Create the tree



Look at the treatment effects in the resulting subgroups



Unsupervised learning

Supervised vs. unsupervised learning

<u>Supervised learning:</u>

- Outcome variable / response: $\boldsymbol{y} = (y_1, \dots, y_n)$
- Predictors: $\boldsymbol{X}^T = (\boldsymbol{X}_1, \dots, \boldsymbol{X}_p)$
- Goal: predict $\boldsymbol{y}$ using $\boldsymbol{X}$, resulting in $\widehat{\boldsymbol{y}}$.
- Idea: Minimize some loss function $L(\boldsymbol{y}, \widehat{\boldsymbol{y}})$, for example, $L(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = |\boldsymbol{y} - \widehat{\boldsymbol{y}}|^{\mathbf{2}}$

It is called supervised, because (in some training set) we know $\boldsymbol{y}$.

Predictions for $\boldsymbol{y}$ can be made based on new data ($\widehat{\boldsymbol{y}}$).

More generally:

- Assume a joint probability density Pr$(\boldsymbol{Y}, \boldsymbol{X})$.
- We are interested in the properties of Pr$(\boldsymbol{Y}|\boldsymbol{X})$.

Supervised vs. unsupervised learning

Unsupervised learning:

We do not have any $\boldsymbol{y}$ variables (no response).

➡ We would like to characterize Pr($\boldsymbol{X}$).

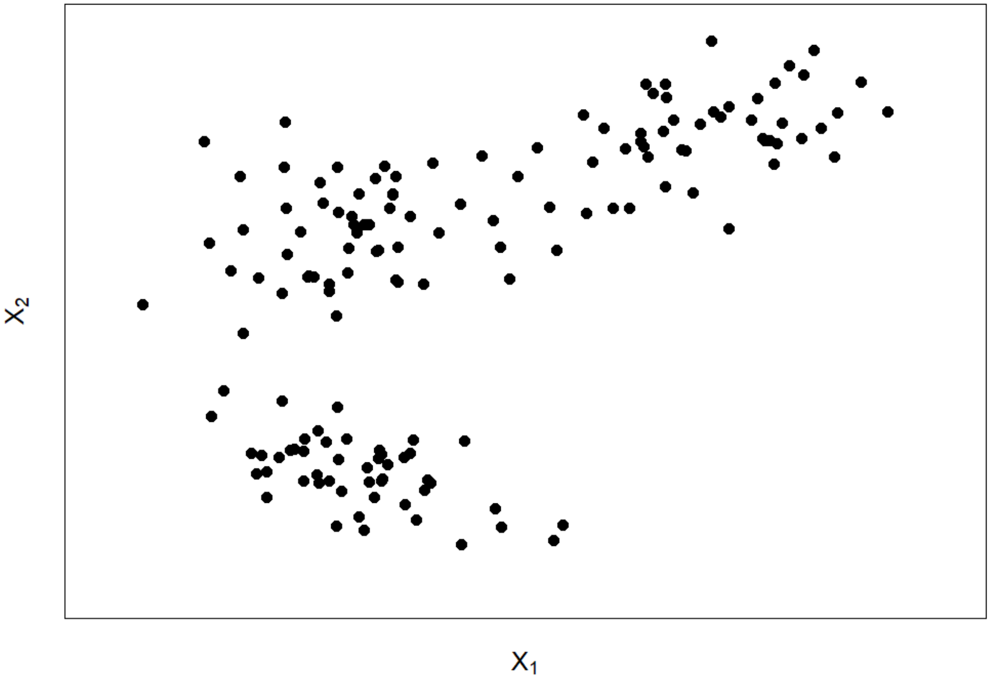
What does that mean?

- Find patterns.
- Find groups of subjects with similar characteristics.
- Find associations between variables.
- Combine variables to a smaller set of “latent” variables.
- ...

Cluster analysis

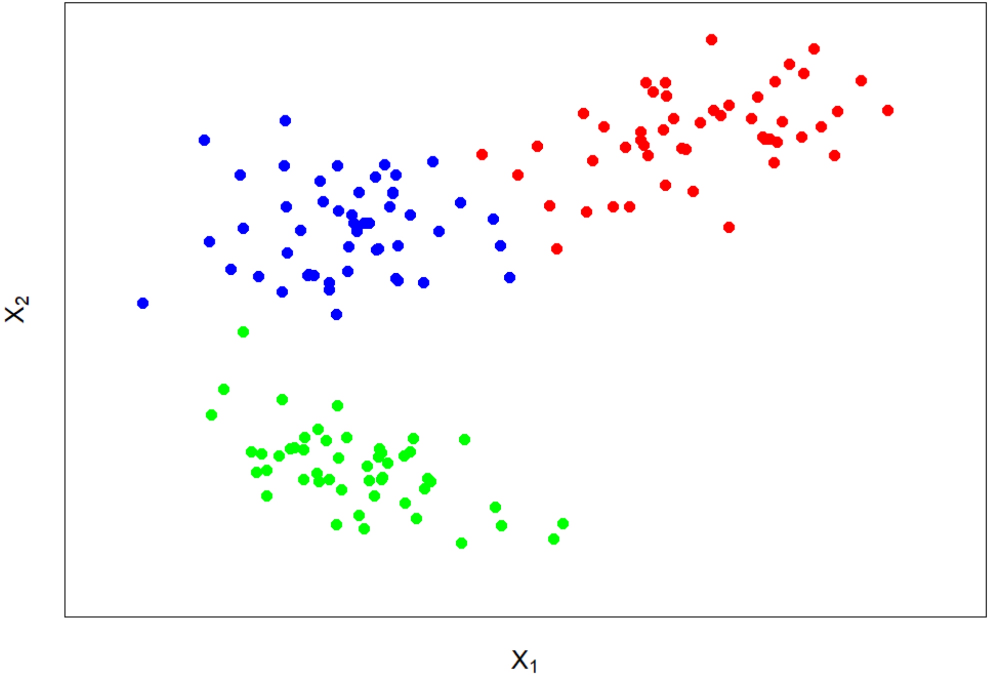
- Goal: Identify groups or “clusters” of subjects.
  - Subjects within the same clusters are supposed to be “similar”.
  - Subjects from different clusters are supposed to be “different”.
- How do we identify clusters (what does “similar” mean)?
  - Similarity is based on $\boldsymbol{X}$ (usually all variables in the data set).
  - We need some distance measure.
  - Different distance measures lead to different results
- What is this useful for?
  - Descriptive analysis of your (patient) population.
  - Identification of subgroups with different characteristics.

Two-dimensional example



<u>Goal:</u> Find groups of observations which are “similar”.

Two-dimensional example



Goal: Find groups of observations which are “similar”.

Degree of similarity

To identify similar patients, we need to define similarity.

Pairwise definition for subjects i and i':

$$D(x_i, x_i') = \sum_{j=1}^{p} d_j(x_{ij}, x_{i'j})$$

Most common choice is the

Euclidean distance:

$$d_j\left(x_{ij}, x_{i'j}\right) = \left(x_{ij} - x_{i'j}\right)^2$$

in case of quantitative (continuous) variables.

Degree of (dis)similarity

Alternative definitions (examples)

Continuous variables:

$$d_j(x_{ij}, x_{i'j}) = |x_{ij} - x_{i'j}|$$

Ordinal variables: replace the M categories with

$$\frac{i - 1/2}{M}, \qquad i = 1, \dots, M$$

and treat them as continuous variables.

Nominal variables:

$$d_j(x_{ij}, x_{i'j}) = \begin{cases} 1, & x_{ij} \neq x_{i'j} \\ 0, & x_{ij} = x_{i'j} \end{cases}$$

Degree of (dis)similarity

Some additional remarks

- Weights can be added to the dissimilarity measure:

$$D(x_i, x_i') = \sum_{j=1}^{p} w_j \cdot d_j(x_{ij}, x_{i'j}); \quad \sum_{j=1}^{p} w_j = 1$$

- Choice of the dissimilarity measure seems to be more important than the clustering algorithm.
  - Distance measures should be chosen wisely.
  - Clinical input may be very helfpul.
  - Using equal weights to all variables ($w_j = 1/\bar{d}_j$) may not always be the best choice.

Clustering algorithm: *k*-means

Assumptions:

- Only continuous variables
- Euclidean distance as dissimilarty measure

Thus, we would like to minimize “within cluster” point scatter:

$$W(C) = \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2$$

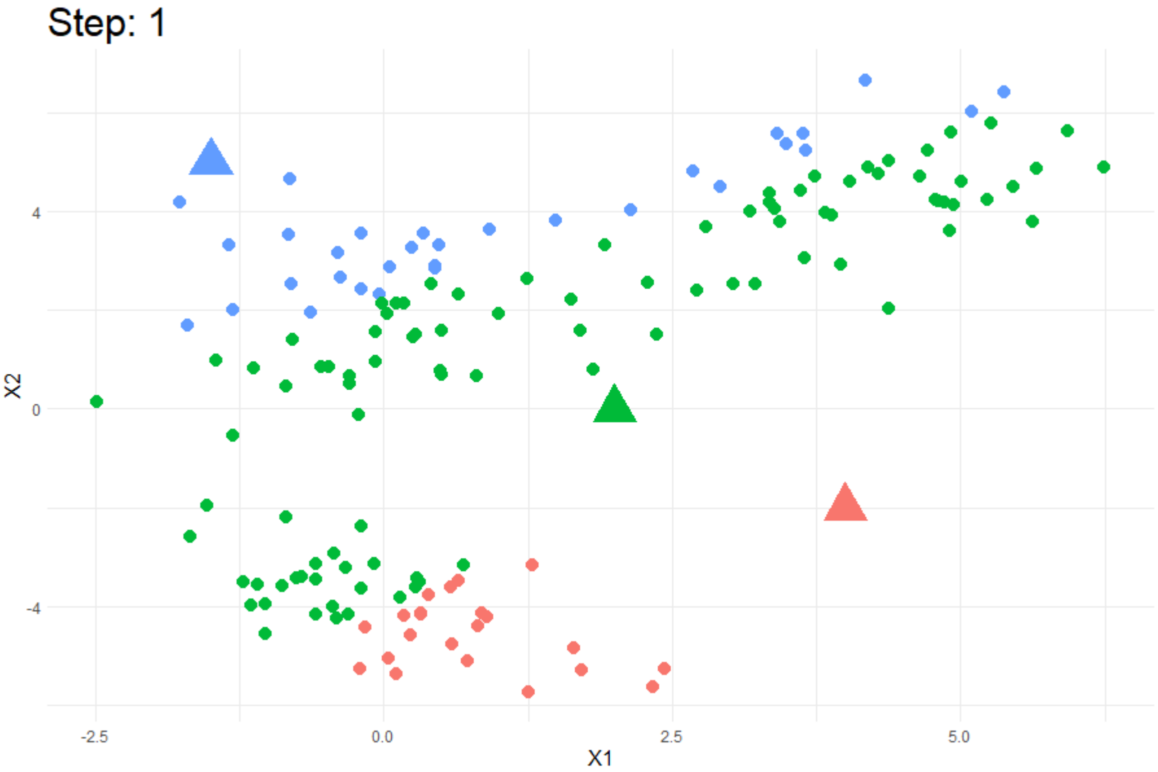
$C(i) = k$ assigns observation i to cluster k.

Clustering algorithm: *k*-means

The algorithm looks as follows:

1. Given a set of clusters, find the mean of each cluster to minimize the variance within the cluster around that mean.
2. Given these means, assign each observation to the cluster with the closest mean.
3. Repeat steps 1 and 2 until there is no change.

Animated visualization



Some remarks on *k*-means

- Computationally simple, but very expensive. ➡ Greedy descent algorithms are being implemented.
- We need to decide on k and start with initial values (e.g., random values).
- How to choose k?
  - Sometimes defined by research question.
  - Use $W(C)$ as a criterion. However, it will always decrease with increasing k.
  - Stop, for example, if the decrease gets sufficiently small (use for example the *Gap statistic* (Tibshirani, 2001)).

Extensions and other algorithms

- Include categorical data by:
  - using some recoding (dummy coding) or preferably by
  - adjusting the dissimilarity measure (for example, Gower distance)
- Hierarchical clustering (do the clustering in hierarchical steps)
- *k*-medoids using actual data points as center of clusters (commonly used algorithm: Partitioning Around Medoids (PAM)).
- Self organizing maps. Idea:
  - Projection to a low-dimensional space (similar to principal component analysis).
  - Finding clusters on this low-dimensional space.

References

The slides on machine learning techniques are mainly based on:
Hastie, T., Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: Data Mining, Inference, and Prediction*. New York: Springer, 2nd edition.

Additional references can be found on the next slide.

References

- Athey, S., Tibshirani, J., Wager, S. (2019): Generalized random forests. The Annals of Statistics, 47:1148–1178.
- Breiman, L. (2001): Random Forests. Machine Learning, 45:5–32.
- Foster, J. C., Taylor, J. M., Ruberg, S. J. (2011): Subgroup identification from randomized clinical trial data. Statistics in medicine, 30:2867–2880.
- Freund, Y., Schapire, R. E. (1997): A Decision-Theoretic Generalization of On-Line Learningand an Application to Boosting. Journal of computer and system sciences, 55:119–139.
- Friedman, J. (1999): Stochastic gradient boosting. Technical report, Stanford University.
- Friedman, J. (2001): Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29:1189–1232.

References

- Tibshirani, R. (1996): Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society. Series B. 58:267–288.
- Tibshirani, R., Walther, G., Hastie, T. (2001): Estimating the Number of Clusters in a Data Set Via the Gap Statistic. Journal of the Royal Statistical Society Series B. 63:411–423.
- Zou, H., Hastie, T. (2005): Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society. Series B. 67:301–320.