unicef | for every child

Implementing Causal Machine Learning in

Online Workshop 3: Machine Learning

August / September / October 2024

Michael Lechner
Professor of Econometrics | University of St. Gallen | Switzerland

unicef | for every child

The workshop series | 1

4 online workshops

- August 27, 13-15: Correlation & causation
- September 4, 13-15: Available identification strategies & classical estimation
- September 10, 13-15: Machine Learning
- September 11, 13-14: The data for the Astana workshop & useful descriptive statistics

unicef | for every child

The workshop series | 2

The Astana Workshop September 30 to October 4, 2024 (10-13, 14:30-17:30)

- Monday
  - Morning: Identification with experiments & selection on observables
  - Afternoon: Discussion of potential programmes to be evaluated
- Tuesday: Causal Machine Learning (theory)
- Wednesday
  - Morning: 2 empirical examples
  - Afternoon: The mcf package – how to use it & how to interpret the results
- Thursday: Doing an empirical study in groups with the data introduced in online workshop 4
- Friday: Discussion of programmes to be evaluated continued (core team only)

unicef | for every child

Quick introduction

Participants

- Professional background?
- Knowledge in the estimation of causal effects, machine learning, Python?

Myself

- Professor of Econometrics at the University of St. Gallen
- Co-head of The Swiss Institute for Empirical Economic Research at the University of St. Gallen
- Empirical Economic Research | SEW-HSG | University of St.Gallen (unisg.ch)

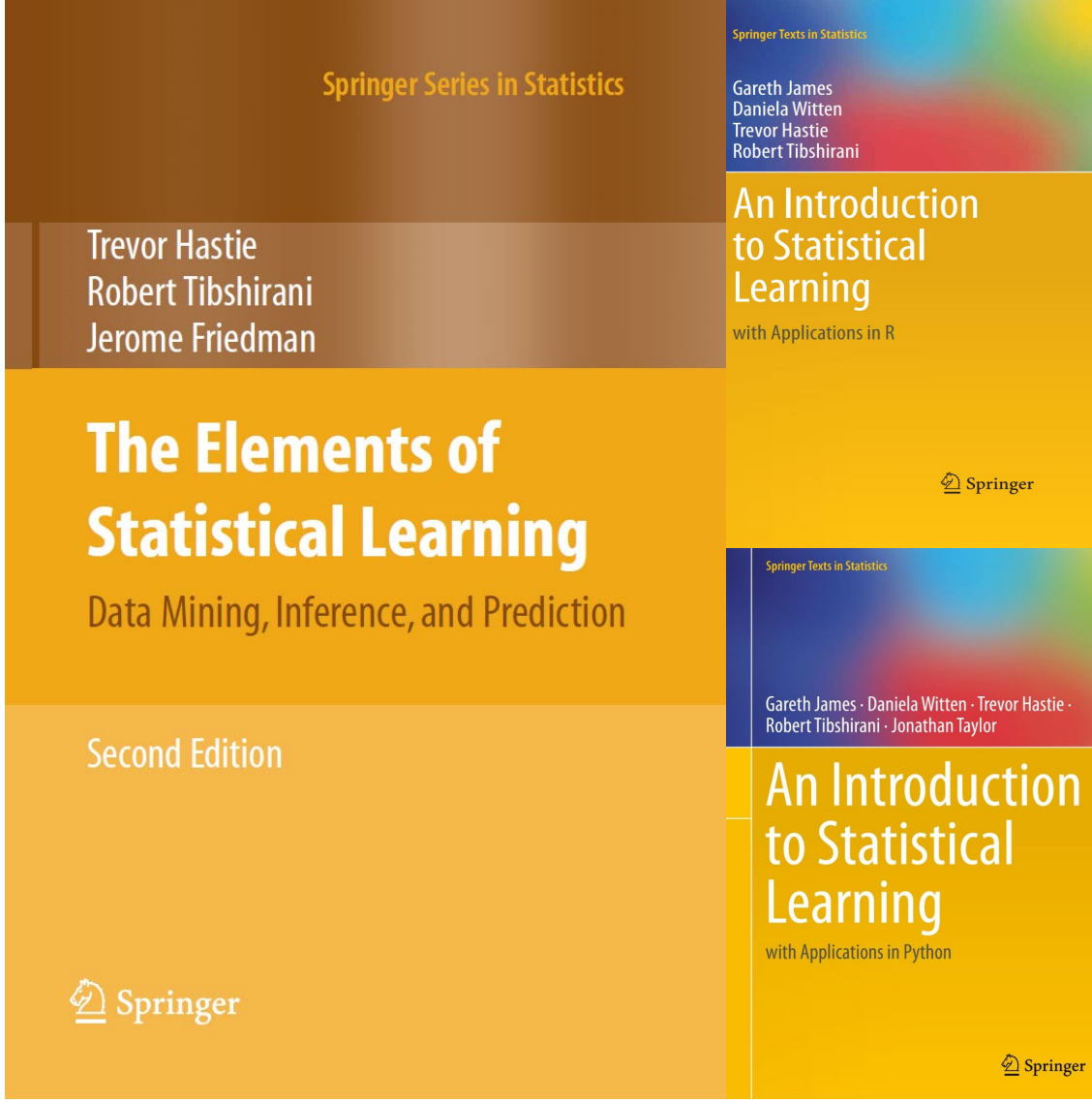
  www.michael-lechner.eu
- Research interest in Causal Machine Learning, AI, programme evaluation, …

unicef | for every child

Plan for today's workshop

Introduction to machine (statistical) learning

- Introduction: Some terminology & concepts
- Supervised learning I: Shrinkage methods
- Supervised learning II: Neural networks
- Supervised learning III: Trees & Forests
- Unsupervised learning: K-means clustering
- Econometrics & statistical learning



unicef | for every child

Machine Learning | 1

What is machine learning?

- Everything & nothing ➔ Here: *Statistical Learning*
- Flexible prediction methods

Types of ML

- **Supervised** (*y, x)* & unsupervised learning (*x*)
- Supervised: Classification (discrete *y*) & **regression** (cond. expectations)

Examples of supervised, regression ML (details later)

- All classical econometric estimators
- Regularized, shrinkage estimators (Lasso, Ridge, ELN, …)
- Neural Networks
- Trees & Random Forests

Statistical learning |2

Standard (non-linear) estimation problem: $y = f(x) + e$

- Estimate unknown function *f(.)* from data
  - Parametric example (econometrics): Coefficients of probit or linear regression model
  - Non-parametric example (econometrics): Kernel estimator of regression function
  - Statistical learning methods provide alternative approaches that …
    - Combine the efficiency features of the parametric models with the flexibility of the non-parametric approaches (to find a 'good' variance bias trade-off)
    - May be able to deal with higher dimension in *X*
    - May be computationally more attractive

Statistical learning | Language & concepts

The prediction error when using *f(.)* to predict y can be reduced in the *training data* by making the model more flexible

But: Not a good idea to use the same data to judge quality of estimation in the population or another (future) sample

Divide the (large) data!

- *Training data*: Sample used for estimating *f(x)*
- *Test and/or validation data*: Different data to evaluate performance of estimator 'out-of-sample'

unicef | for every child

Supervised & unsupervised learning

Supervised Learning: Find structure in *X* that allows to predict *Y*

- If particular *X* are important for *Y*: Similar values of those *X* will be related to similar values of *Y*
- Comparing predictions with realisations of *Y* provides metric for the similarity of high-dimensional *X (Y is 'teacher'; Y 'tells' algorithm how 'good' its prediction based on its current combination of X is)*
- Regression (classification)-type methods

Unsupervised learning: Find similarity of *X* without having access to *Y*

- Clustering etc.

unicef | for every child

1 | Introduction

2 | Supervised learning 1: Shrinkage methods

3 | Supervised learning 2: Neural networks

4 | Supervised learning 3: Trees & Forests

5 | Unsupervised learning: K-means clustering

6 | Econometrics & statistical learning

7 | Conclusions & outlook

Methods |2

Shrinkage methods are based on classical regression estimators (e.g., OLS) but with penalty terms penalizing large coefficient values

- $p >> N$ (ultra-high dimensional models) possible

Ridge regression & LASSO |1

$$\hat{\beta}^{Ridge} = \arg\min_{\beta} \sum_{i=1}^{N} \left(y_i - x_i\beta\right)^2 + \lambda \underbrace{\sum_{j=1}^{p} \beta_j^2}_{=\|\beta\|_2^2 \quad (\ell_2-norm)^2} \quad ; \quad \lambda \geq 0$$

$$\hat{\beta}^{Lasso} = \arg\min_{\beta} \sum_{i=1}^{N} \left(y_i - x_i\beta\right)^2 + \lambda \underbrace{\sum_{j=1}^{p} \left|\beta_j\right|}_{=\|\beta\|_1 \quad \ell_1-norm} \quad ; \quad \lambda \geq 0$$

λ : penalty multiplier

- If $\lambda = 0$: OLS
- If $\lambda \rightarrow \infty$: Only constant term remains in model (others almost zero)
- Constant term is not penalized

Ridge regression & LASSO |2

Estimates are shrunken (usually) towards zero

- This leads to (usually downward) biased coefficient estimates (no causal interpretation) but better predictions

Penalty term

- Bias-variance trade-off
  - Increasing penalty term (λ) increases bias & reduces variance
- Effect of λ on MSE a priori unclear
- Choose penalty term that leads to best predictive performance
  - Optimal value of λ determined by grid search (cross-validation)

Ridge regression (RR)

RR has explicit solution (in linear model)

May be an attractive choice in *dense* models

Least **A**bsolute **S**hrinkage & **S**election **O**perator |2

LASSO has **no** explicit solution (in linear model)

- Iterative methods available

Many estimated coefficients will be exactly zero: *Explicit variable selection*

- Lasso yields *sparse(r)* models
- Eases interpretability of estimates (at the danger of misinterpretation)

May be an attractive choice in *sparse* models

Elastic Net |1

LASSO is expected to do better than Ridge regression in *sparse* models, while Ridge regression does better in *dense* models

Elastic Net combines these properties to allow for *somewhat sparse* settings by combining the penalty terms

$$\hat{\beta}^{EN} = \arg\min_{\beta} \sum_{i=1}^{N} \left(y_i - x_i\beta\right)^2 + \lambda_1 \sum_{j=1}^{p} \left|\beta_j\right| + \lambda_2 \sum_{j=1}^{p} \beta_j^2; \quad \lambda_1, \lambda_2 \geq 0$$

Find optimal values of λ_1, λ_2 by CV using grid search on a 2-dimensional grid

Shrinkage methods

Advantage

- Should do well when the relation of *Y* & *X* is approximately as specified (e.g., linear)

Disadvantage

- Should **not** do well when the relation of *Y* & *X* is **not** approximately as specified

unicef | for every child

Introduction | 1

Today: Only **N**eural **N**etworks for regression type learning problems considered

Marketing type of definition

- Flexible Learning Model achieving good prediction properties by using a model structure that is similar to the human brain

Statistical definition

- Flexible, regularized, global, non-linear estimator

Mathematical approximation theorem suggest that a (suitable) NN can approximate any functional relationship

Introduction | 2

Heavily used in modern ML in practice for very many different tasks

Very powerful, but also computationally very demanding

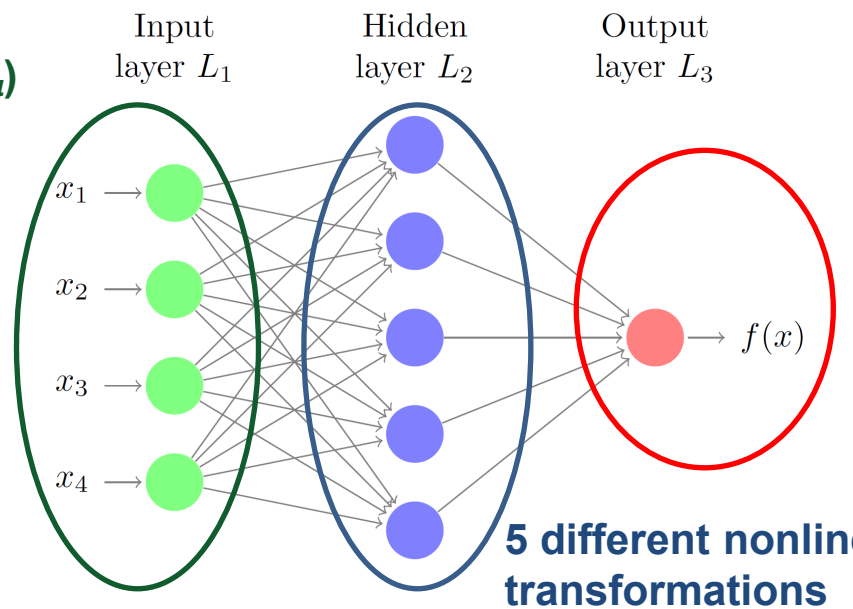
- May need specific hardware architecture to be feasible

Huge number of parameters needed for good performance

- Tuning very important
  - Choice of architecture
  - Regularization
- ‘More art than science’

Graphical representation of a very simple NN

Covariates (X_1, …, X_4)



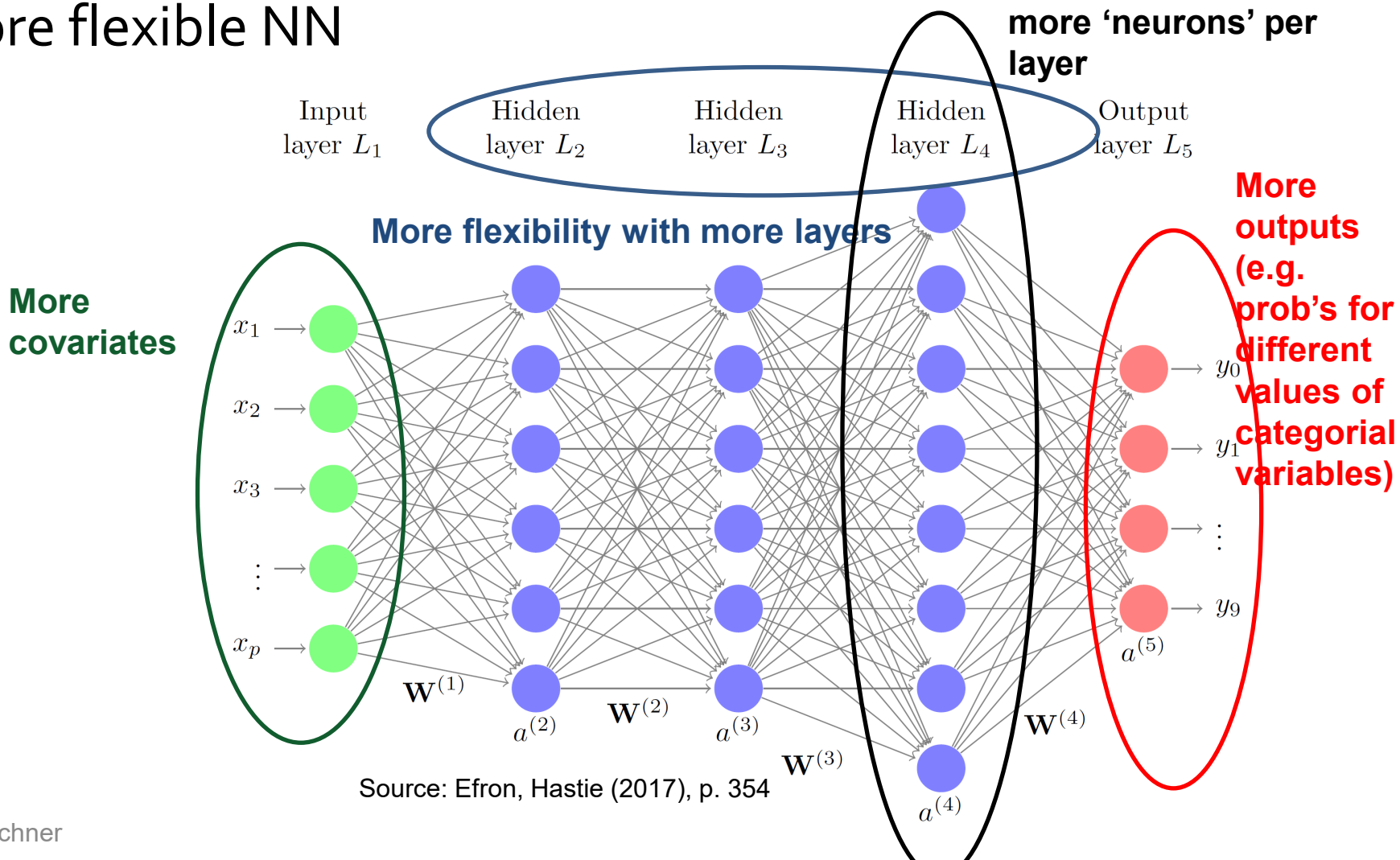
Outcome specific linear or nonlinear combination of 5 non-linear functions leads to prediction for *Y*

5 different nonlinear transformations

Figure 18.1 Neural network diagram with a single hidden layer. The hidden layer derives transformations of the inputs—nonlinear transformations of linear combinations—which are then used to model the output. Source: Efron, Hastie (2017), p. 351

A more flexible NN

More flexibility with more 'neurons' per layer



Source: Efron, Hastie (2017), p. 354

Notation | 1

When writing the statistical model, it is useful to account for the particular architecture in the notation

- P covariates, L layers, K^l neurons in layer l, $z^{nl'}$ output of neuron n in layer l', M possible outcomes

$$z^{k,1}(x) = g^1(\alpha^{k,1} + \sum_{p=1}^{P} x^p \beta^{p,k,1}), \quad l = 1, \quad \forall k^1$$

$$\vdots$$

$$z^{k,l}(x) = g^l(\alpha^{n^1} + \sum_{k=1}^{K^{l-1}} z^{k,l-1} \beta^{k,l}), \quad l > 1, \quad \forall k^l$$

$$\vdots$$

$$z^{k,L}(x) = g^L(z^{1,L-1}, \ldots, z^{K^{L-1},L-1}), \quad l = L, \quad \forall k^L$$

$$\hat{y}^m = f^m(z^{1,L}, \ldots, z^{K^L,L}), \quad \forall m = 1, \ldots, M$$

Notation | 2

$$z^{k,1}(x) = g^1(\alpha^{k,1} + \sum_{p=1}^{P} x^p \beta^{p,k,1}), \quad l = 1$$

$$z^{k,l}(x) = g^l(\alpha^{n^1} + \sum_{k=1}^{K^{l-1}=1} z^{k,l-1}\beta^{k,l}), \quad l > 1$$

$$z^{k,L}(x) = g^L(z^{1,L-1}, ..., z^{K^{L-1},L-1}), \quad l = L$$

$$\hat{y}^m = h^m(z^{1,L}, ..., z^{K^L,L})$$

The major nonlinearity is captured by the layers & neurons

The transformation in the last layer is outcome specific

The transformation into a prediction is outcome- & task- (classification, regression) specific

Coefficients ('weights') & tuning parameters

Huge # of coefficients (called 'weights') to estimate

- # of covariates *x* avg. # of neurons *x* # of layers (minimum)

Many tuning parameters

- # of neurons in specific layers
- # of layers
- Different functions in each layer
- Functions used for generating predictions

Depending on architecture, estimation is computationally very expensive

- May require explicitly designed hardware

unicef | for every child

1 | Introduction

2 | Supervised learning 1: Shrinkage methods

3 | Supervised learning 2: Neural networks

4 | Supervised learning 3: Trees & Forests

5 | Unsupervised learning: K-means clustering

6 | Econometrics & statistical learning

7 | Conclusions & outlook

unicef | for every child

Regression Trees

Concept

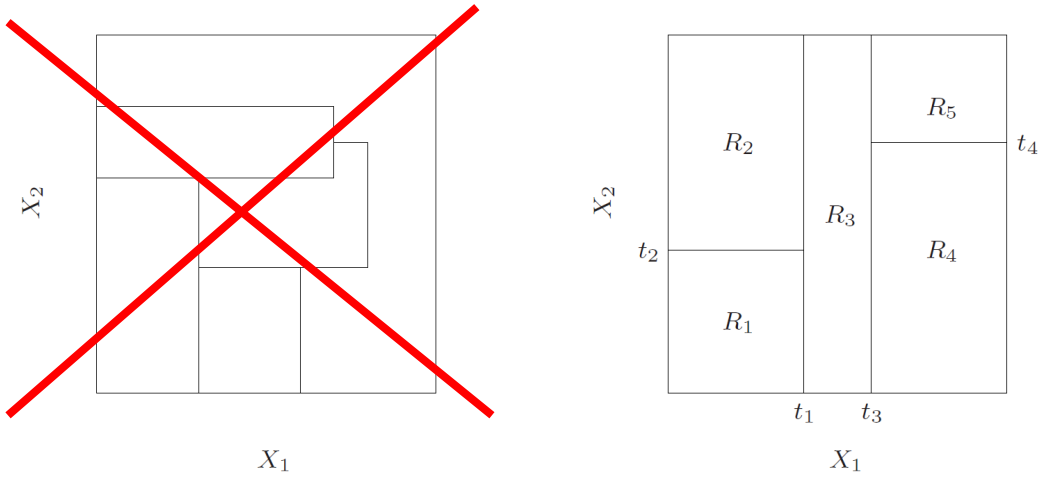
$$y_i = f(x_i) + e_i \qquad \hat{y}_i = \widehat{f(x_i)}$$

Main idea

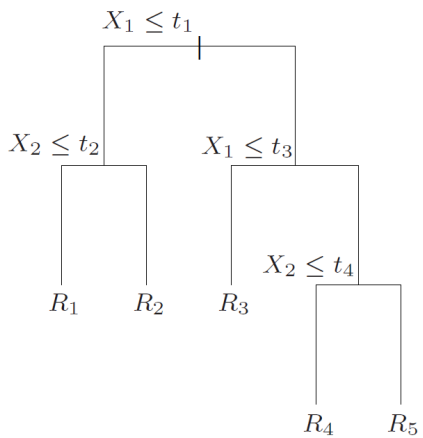
- Stratify data in distinct strata (leaves) with similar values of *x*
- Mean of *y* leaf is predictor for *y* for all observation with values of *x* that belon to this leaf

Build tree by further splitting existing strata with recursive binary splits

- In each stratum (=leaf), for each feature in X, & each value, split old stratum into two new strata such that prediction error (e.g., MSE) of the new prediction is minimized



The splitting rule



Estimate of *f(x)*

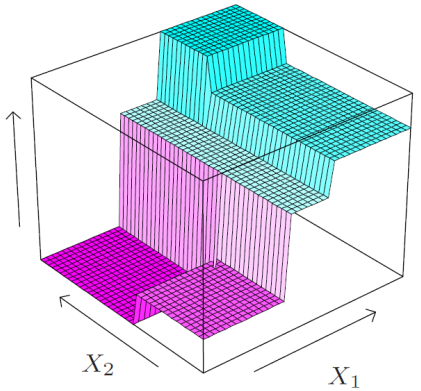


FIGURE 9.2. *Partitions and CART. Top right panel shows a partition of a two-dimensional feature space by recursive binary splitting, as used in CART, applied to some fake data. Top left panel shows a general partition that cannot be obtained from recursive binary splitting. Bottom left panel shows the tree corresponding to the partition in the top right panel, and a perspective plot of the prediction surface appears in the bottom right panel.*

Illustration of a tree with two continuous covariates

Hastie et al. (2013), p. 306

Growing the tree |3

The variance-bias trade-off

- If tree is too large → overfitting → too much variance
- If tree is too small → not precise enough → too much bias

Find stopping rule that prevents overfitting

Regression vs. classification trees

Regression vs. classification

- Quality of fit measure (objective function) & predictor may be different

Regression tree

- Goal is to approximate *f(x)* by a conditional expectation *E(y|x)*
- Taking averages in leaf as prediction & MSE as quality measure seems 'natural'

Classification tree

- If *y* is discrete (& multivalued), an approximation of *f(x)* by *E(y|x)* may not be optimal in a predictive sense (e.g., underrepresentation of extreme categories)
- 'Majority rule' may be more natural than average
- Alternative quality-of-prediction measure required

Pro & con's of trees

Pro's

- Very flexible
  - they capture substantial non-linearities
- Nice interpretation of leaves in terms of sets of values of *x*
- Easy to understand for non-statisticians (easy to plot)

Con's

- They do not scale in *p* → if *p* is large, computationally very expensive
- They may be very unstable (path dependence) → slight change in sample may lead to very different tree ➔ loss of predictive power compared to other methods

Bagging trees|1

Simple way to reduce the variability of tree predictions is **b**ootstrap-**a**vera**ging** (*bagging*)

Idea

- Use a *deep* tree (not pruned): Low bias, large variance
- Do this for many bootstrap samples & average over the predictions over these samples → variance reduction

$$\widehat{f(x)}^{bagged} = \frac{1}{B}\sum_{i=1}^{B}\left\{\sum_{m=1}^{M^b}\overline{y}_m^b \underline{I}(x \in R_m^b)\right\}, \qquad \overline{y}_m^b = \frac{1}{\underline{I}(x_i^b \in R_m^b)}\sum_{i=1}^{N} y_i^b \underline{I}(x_i^b \in R_m^b)$$

Bagged trees|2

Advantages compared to single tree

- Variance reduction → better predictive properties

Disadvantages

- Increased computation time
  - *B* determined such that *Out-of-bag* MSE stabilizes (or *B* predetermined)
- Loss of interpretation
  - Because averaging is over many different trees (which are likely to overlap across bootstrap samples)

Random Forests

Random Forests

Special Bagged Tree (used for RF)

- When deciding on the next split, only a random selection of variables is evaluated
- This decorrelates the bagged trees & reduces computation

Prediction of a Random Forests is the mean of the predictions of many of such trees

- Computationally efficient & very effective in non-linear settings

unicef | for every child

1 | Introduction

2 | Supervised learning 1: Shrinkage methods

3 | Supervised learning 2: Neural networks

4 | Supervised learning 3: Trees & Forests

5 | Unsupervised learning: K-means clustering

6 | Econometrics & statistical learning

7 | Conclusions & outlook

Introduction | 1

As supervised learning, unsupervised learning is used for dimension reduction

- Helps ‘understanding’ / describing data (or findings of estimations, …)
- May be necessary for some more structural estimations to avoid some form of curse of dimensionality

Many different methods exist

- Here, we briefly discuss one cluster method as example

Intro & notation

K-means clustering is very popular because of its

- Simplicity (& computational efficiency)

Similarity measure as

Euclidian distance $$d(x_i, x_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = \|x_i - x_{i'}\|^2$$

Find clusters such that observations are most similar within clusters & most different across clusters

The algorithm

Set number of clusters to *C*

Designate *C* data points as (starting) cluster centres (randomly or more sophisticated)

1) Assign each observation to the closest cluster centre (in terms of Euclidian distance)

2) Compute new cluster centres as mean of x-values of observations in each cluster

Repeat 1) and 2) until no observation changes cluster in step 1)

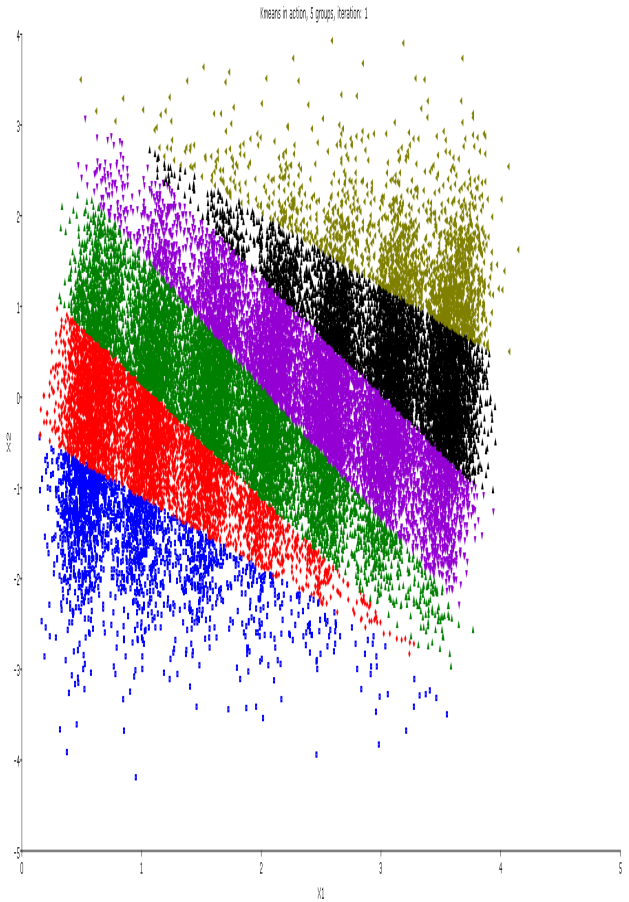
Simulation | DGP

$$X \sim N\left(\begin{pmatrix}0\\ \mu\end{pmatrix}, \begin{pmatrix}1 & 0.5\\ 0.5 & 2\end{pmatrix}\right); \quad \mu_i \in [1,2,3,4,5,6,7] \text{ (with equal probability)}$$

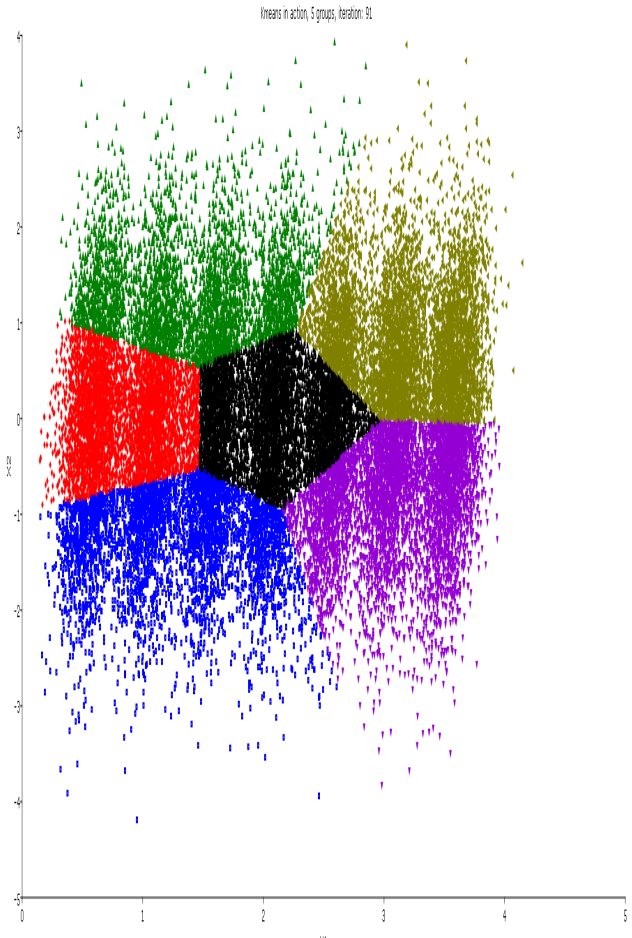
N=30’000

K-means with 6 (below) & 7 (live) groups (no retries)

Simulation | Start



End (92 iterations)



unicef | for every child

1 | Introduction

2 | Supervised learning 1: Shrinkage methods

3 | Supervised learning 2: Neural networks

4 | Supervised learning 3: Trees & Forests

5 | Unsupervised learning: K-means clustering

6 | Econometrics & statistical learning

7 | Conclusions & outlook

unicef | for every child

How to compare estimators? *SL/ML*

Goal: Reliable prediction (& classification)

- Limited interest in inference

How to: Use many reference data sets

- Split data in estimation data (e.g., used for estimation)
- Compare predictive performance relative to true values in test data
  - Test data has not been used at all for estimation
- Many references to such data sets can be found at
  - https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research

Machine learning meets econometrics |2

ML & econometrics have different goals

Goal of machine learning

y = f(x) + unobservables

- Get best guess of *y* once you know *x* → *learn* (=estimate) *f(x)*
- Carefully ensure that estimated *f(x)* does not depend on unobservables (i.e. avoid in-sample overfitting)

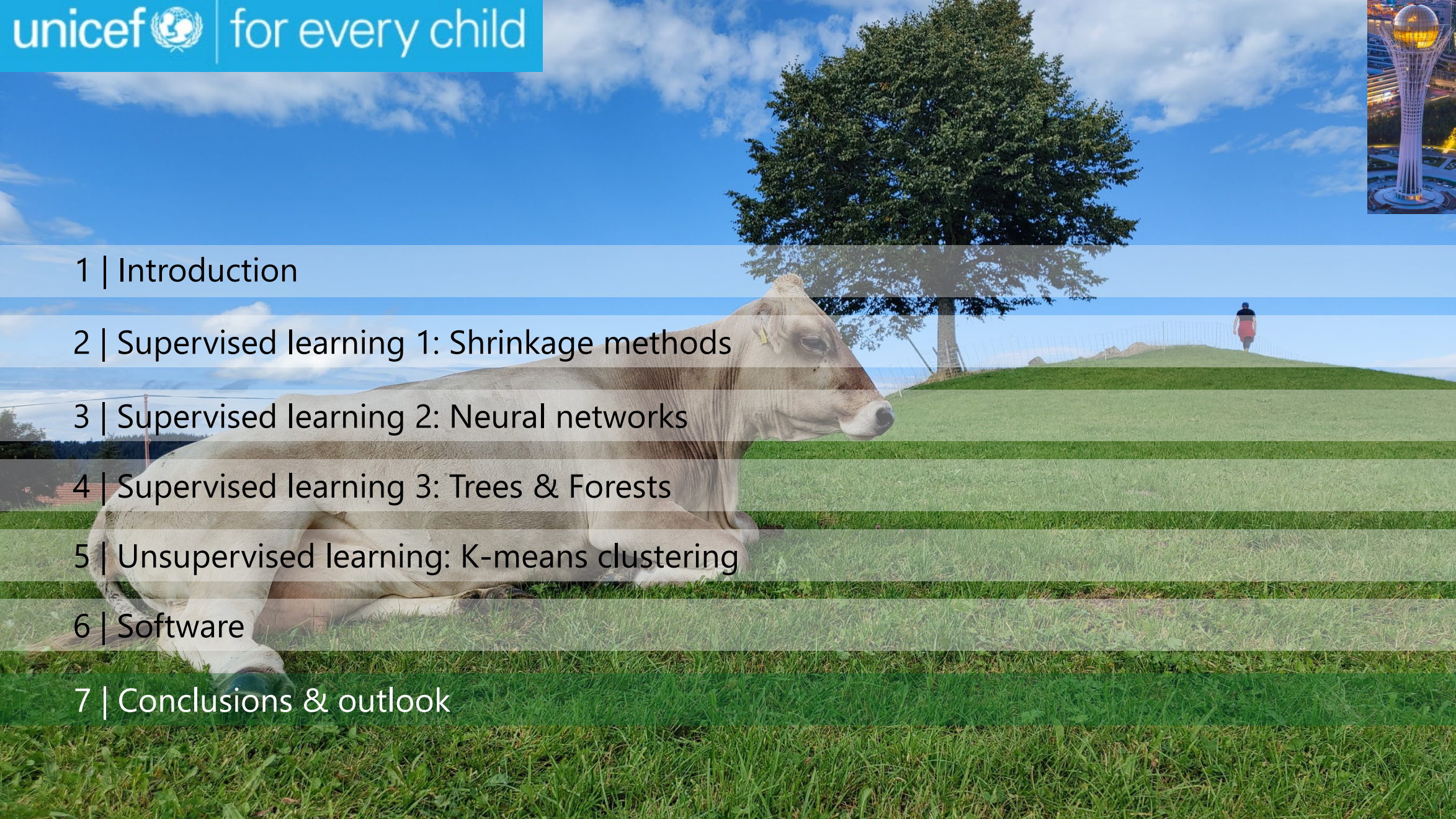
Goal of econometrics

- Learn *parameters* capturing how a change in *x* changes *y*
  - Get respective uncertainty measures for these causal parameters
  - Examples: regression coefficients, average treatment effects, other 'structural' parameters

This table oversimplifies (a bit)

Classical Econometrics & Machine Learning

Issue	Classical econometrics	Supervised statistical learning
Target of interest (θ)	Structural & causal parameters (*low dimensional*)	Prediction (*Ey|x)* or classification of *y*
Sample analogue of θ	--	*y*
Judging quality of estimation	*Indirect* (fit, …), in-sample	*Direct* ($\widehat{y}$ vs *y),* out-of-sample
Inference & theoretical properties	Very important	Less important (irrelevant?)
Sample size (*N*)	Large *N* is nice to have	Large *N* may be required
# of variables (k)	Much smaller than *N*	Smaller or larger than *N*
Preferred model complexity	Simple, likely to be parametric (linearity popular)	Complicated (overparametrized; nonlinear)
Names of methods	Boring	Cool



Conclusions

Machine/statistical learning massively improved our ability to predict well

Many innovations come from computer science ➔ algorithms are (usually) efficient

SL/ML & Causal Machine Learning (→ Astana workshop!)

- Limited direct usefulness of SL for causal analysis: Prediction & parameter estimation are different tasks
- *But*: Solve specific prediction problems inside a causal estimation strategy
- *But*: Modify regression ML algorithm to predict a parameter (Random Forest → Causal Forest)
- *But*: Modify classification algorithm to obtain improved programme assignments

unicef | for every child

Tomorrow

Descriptive tools useful for causal analysis

Introduction of the data used in hands-on ‘I do my own CML evaluation’ sessions in Astana (Thursday)

Michael Lechner

Swiss Institute for Empirical Economic Research (SEW)
University of St. Gallen | Switzerland