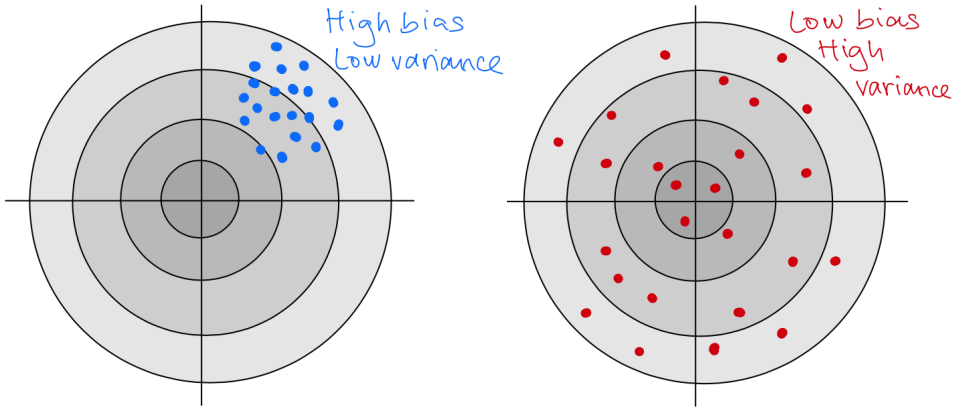
Machine learning essentials

Lukáš Lafférs

Matej Bel University, Dept. of Mathematics

- Bias variance trade-off
- Penalized regression - LASSO and Ridge
- Regression Tree
- Random Forrest
- Machine learning and causality

Bias vs variance trade-off



Which would you choose?

- As always: It depends.

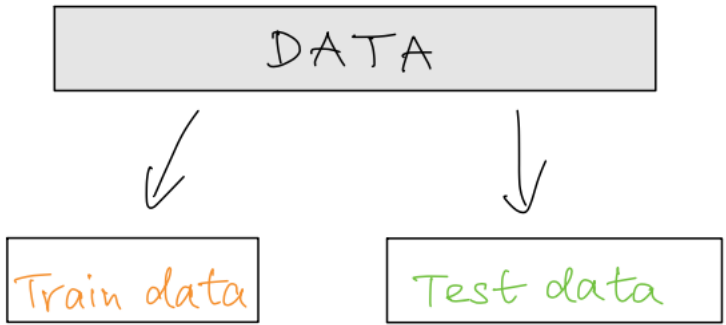
$$Y = f(X) + \varepsilon$$

$$\underbrace{EPE(Y, \hat{f}(X))}_{\text{expected prediction error}} = E\left[\left(Y - \hat{f}(X)\right)^2\right] = \underbrace{E\left[(Y - f(X))^2\right]}_{\text{unsystematic error}} + \underbrace{E\left[(f(X) - \hat{f}(X))^2\right]}_{\text{systematic error}}$$

At a particular point $X = x$:

$$\underbrace{MSE(f(x), \hat{f}(x))}_{\text{mean squared error}} = E\left[(f(x) - \hat{f}(x))^2\right] = \underbrace{\left(f(x) - E[\hat{f}(x)]\right)^2}_{\text{bias}^2} + \underbrace{E\left[(\hat{f}(x) - E[\hat{f}(x)])^2\right]}_{\text{variance}}$$

Train/Test data



Prediction is easy: we can measure on the test data how well the estimated model (on the train data) works.

LASSO

Least Absolute Shrinkage and Selection Operator

Motivation 1 - model selection + estimation

How to choose regressors?

- Hypothesis testing (Backward elimination,...)
- Information criteria (AIC, BIC, FIC,...)
- Model Averaging

2-step procedures: [MODEL SELECTION] + [ESTIMATION]

LASSO does it in one step.

Motivation 2 - many regressors

We have MANY regressors in comparison with observations.

$$p >> N$$

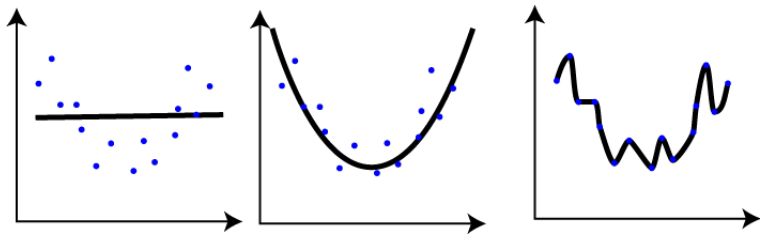
Parameters of

$$y = \boldsymbol{X}\beta + \varepsilon$$

are not uniquely defined.

LASSO can handle such situations.

Motivation 3 - Bias-Variance trade-off



LASSO takes this trade-off into account. It aims to predict well.

$$y = \boldsymbol{X}\beta + \varepsilon$$

$$\hat{\beta}_{OLS} = \arg\min(y - \boldsymbol{X}\beta)^T(y - \boldsymbol{X}\beta)$$

- under Correct specification and Gaussian homoskedastic errors ($\varepsilon \sim N(0, \sigma^2 I)$), this has the lowest variance among linear unbiased estimators
- also Maximum likelihood estimator (asymptotically efficient)

so why do we care with anything else?

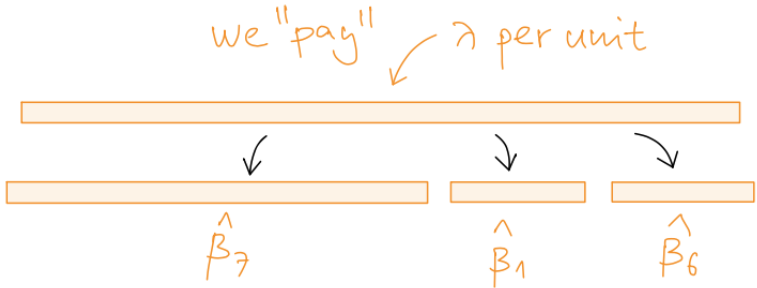
- Should we care about unbiasedness so much?
- Interpretation: OLS estimators are non-zero, all of them
- Sparsity principle: ESL: 'Use a procedure that does well in sparse problems, since no method does well in dense problems.'
- Highly correlated regressors lead to a high variance of estimators

But is the truth sparse?

$$(y - \boldsymbol{X}\beta)^T(y - \boldsymbol{X}\beta) + \textit{PENALTY}$$

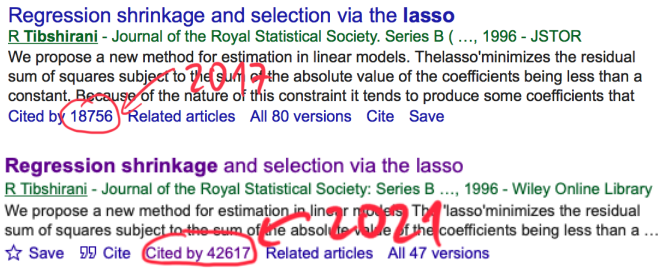
$$(y - \boldsymbol{X}\beta)^T(y - \boldsymbol{X}\beta) + \lambda||\beta||_1$$

- λ - 'price' of β
- $||\beta||_1 = \sum_{i=1}^{p} |\beta_i|$
- We shrink β towards zero!



And the credit goes to...

Robert Tibshirani

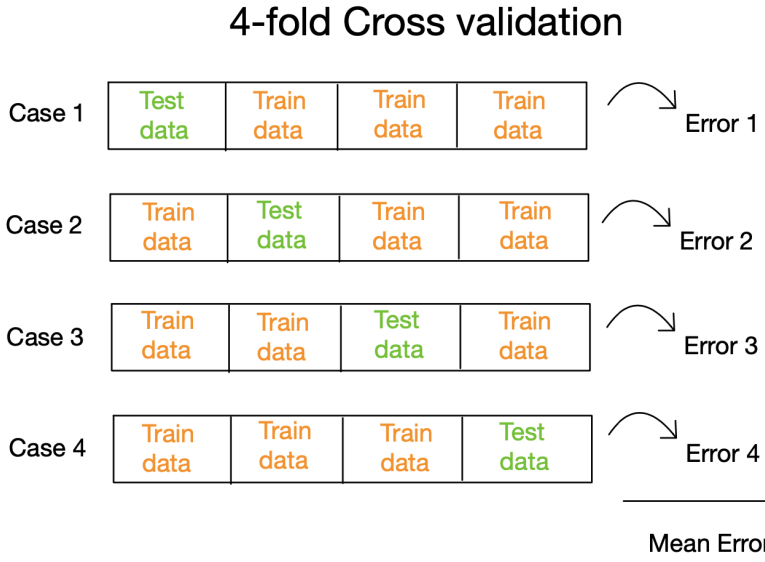


Cross-validation

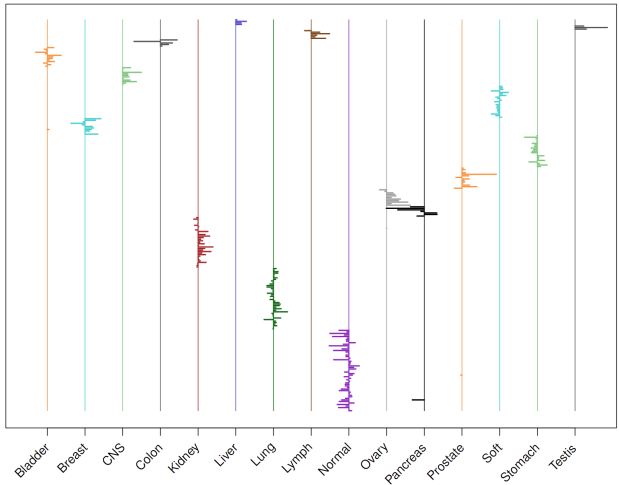
How to choose λ ?

Choose it in a way so that the model predicts well.

Minimize cross-validation error.



Example 1



215 genes out of 4718 with non-zero coefficients of a multinomial lasso classifier (source: SLS)

The Lasso Estimator

For a linear model LASSO estimator is the solution of the following problem

$$\min_{\beta_0,\beta_1} \left\{ \frac{1}{2N} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \right\}$$

$$\text{subject to } \sum_{j=1}^{p} |\beta_j| \leq t$$

$$t \to 0 \implies \hat{\beta} \to 0$$
$$t \to \infty \implies \hat{\beta} \to \beta_{OLS}$$

..in a more compact form

$$\min_{\beta} \left\{ \frac{1}{2N} ||y - \boldsymbol{X}\beta||_2^2 \right\}$$

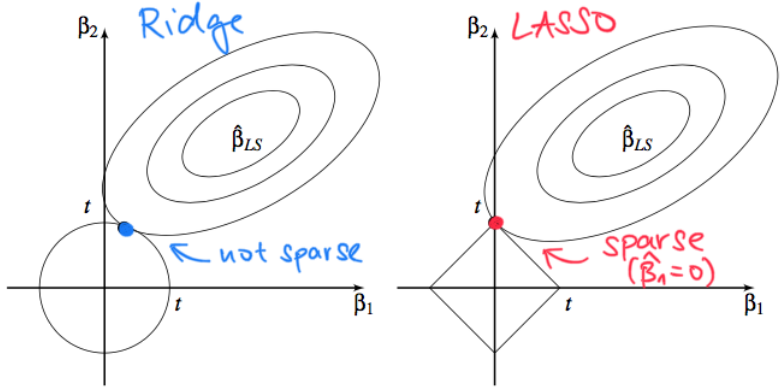
$$\text{subject to } ||\beta||_1 \leq t$$

which is equivalent to

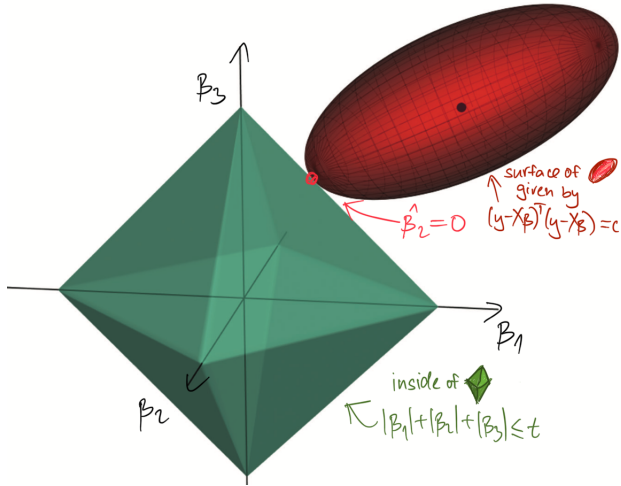
$$\min_{\beta} \left\{ \frac{1}{2N} ||y - \boldsymbol{X}\beta||_2^2 + \lambda ||\beta||_1 \right\}$$

A few remarks...

- If the predictors are not on a same scale, we rescale and recenter then. intercept disappears
- there is a relationship between λ and t and this relationship is data-dependent
- the factor $\frac{1}{2N}$ makes it easier to compare λ across different sample sizes

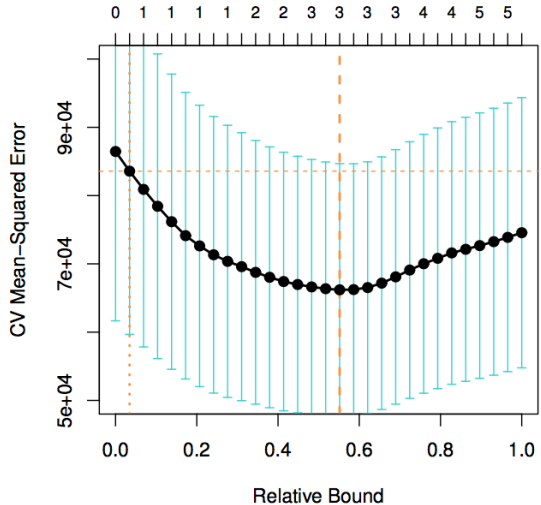


Ridge regression vs LASSO. (source: Faraway (2014))



LASSO in 3D - $(\beta_1, \beta_2, \beta_3)$. (source: SLS)

Choice of λ by Cross-validation

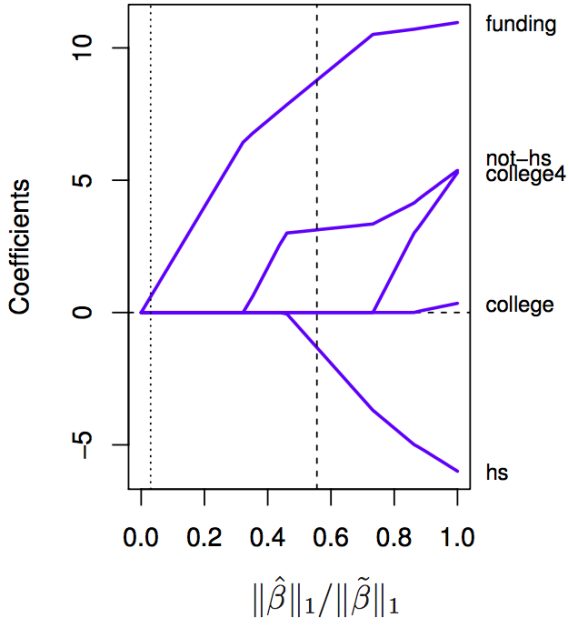


Relative bound = $||\hat{\beta}||_1 / ||\hat{\beta}_{OLS}||_1$ (source: SLS)

Example - Crime rate in U.S. cities

city	funding	hs	not-hs	college	college4	crime rate
1	40	74	11	31	20	478
2	32	72	11	43	18	494
3	57	70	18	16	16	643
4	31	71	11	25	19	341
5	67	72	9	29	24	773
⋮	⋮	⋮	⋮	⋮		
50	66	67	26	18	16	940

(source: SLS)


Coefficients
10
5
0
−5
0.0
0.2
0.4
0.6
0.8
1.0
$\|\hat{\beta}\|_1/\|\tilde{\beta}\|_1$
funding
not−hs
college4
college
hs

(*)Consistency

We may be interested in MSE consistency

$$||\boldsymbol{X}(\hat{\beta} - \beta^*)||_2^2/N \leq C||\beta^*||_1\sqrt{\log(p)/N}.$$

what does it mean?

If $||\beta^*||_1 = o(\sqrt{N/\log(p)})$ (read as: does not grow too fast, alternatively read as: the truth is "sparse"), then lasso is consistent for prediction.

Or we may be interested in *sparsistency*, that is, recoverability of the non-zero support set of the true regression parameters. This is difficult to prove and for this we would need higher level assumptions.

Uniqueness

Suppose we have x_1 and x_2 with corresponding β_1 and β_2.

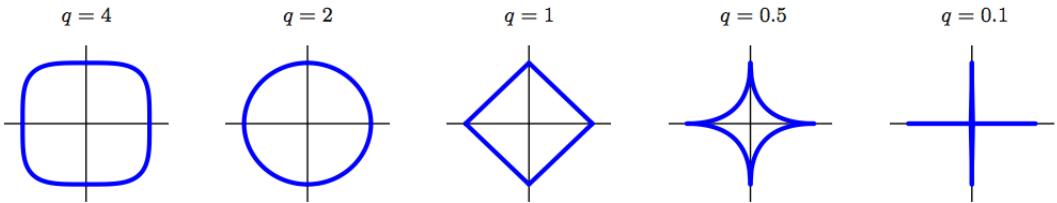
Now we add x_3 which is identical to x_2.

$$(\hat{\beta}_1, \gamma\hat{\beta}_2, (1-\gamma)\hat{\beta}_2)$$

produces the same fit and its norm is the same. So we have infinite number of solutions.

Different penalties

$$\min_{\beta_0,\beta_1} \left\{ \frac{1}{2N} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q \right\}$$



Constraint regions for different q-s. (source: SLS)

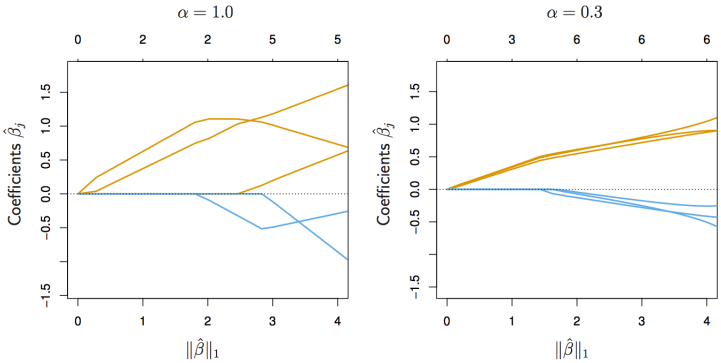
Elastic net

$$\min_{\beta_0,\beta_1} \left\{ \frac{1}{2N} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \underbrace{\lambda \left[\frac{1}{2}(1-\alpha)||\beta||_2^2 + \alpha||\beta||_1 \right]}_{\text{more sophisticated penalty}} \right\}$$

Takes the best out of the two worlds:

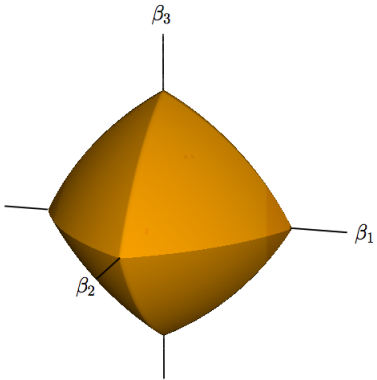
- ability to make some parameters exact zeroes as LASSO ($\alpha = 1$)
- ability to handle highly correlated data as ridge regression ($\alpha = 0$)
- we pay a price for that $\rightarrow$ the choice of α is needed

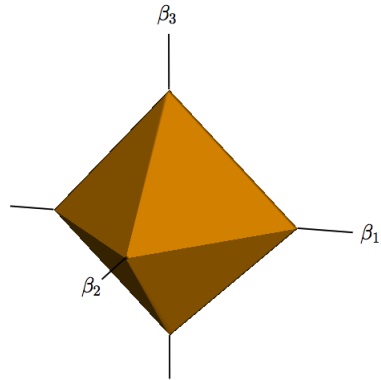
Elastic net



Left LASSO, right elastic net for highly correlated data. (source: SLS)

Elastic net





Left elastic net ball for $\alpha = 0.7$, right LASSO ball. (source: SLS)

Regression tree

Medical doctor asks a patient the following yes/no questions:

- Are you more than 30 years old?
- Is you diastolic pressure higher than 100?
- Is there anyone in your family with heart condition?
- Do you sport more than 60mins per week?

Based on these yes/no question MD can make a prognosis or suggest a suitable intervention.

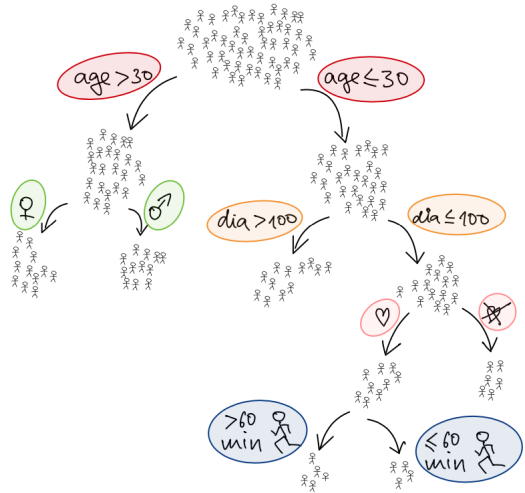
Yes/no questions are easy. They can be standardized and made into guidelines.

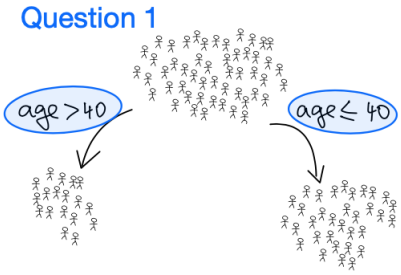
Morgan and Sonquist (1963), Breiman, Friedman, Olshen and Stone (1984)

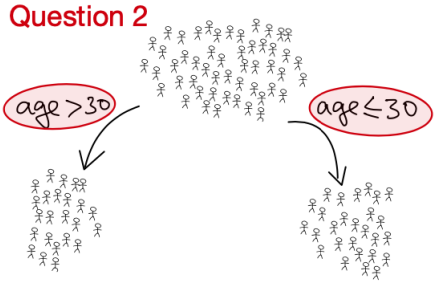
How to build such a tree?

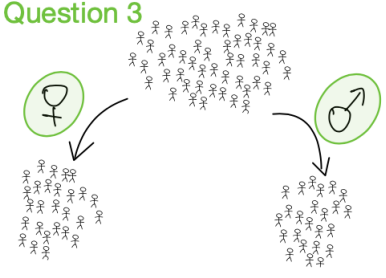
What is a good yes/no question?

Can we measure how good a question is?




Question 1
age > 40
age ≤ 40


Question 2
age > 30
age ≤ 30


Question 3

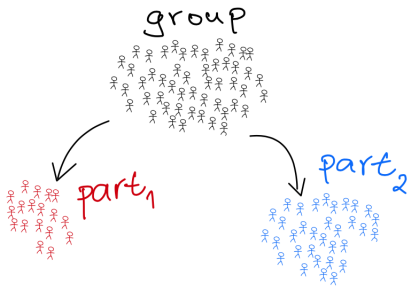

Question 4
dia > 120
dia ≤ 120

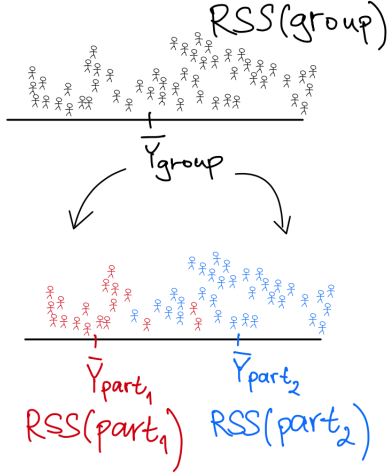
- Outcome Y
- Covariates $X_1, X_2, \ldots X_p$
- Measure of variablity: $RSS(group) = \sum_{i \in group} (Y_i - \bar{Y}_{group})^2$

Choose the partitioning $group = part_1 \cup part_2$ so that the overall measure of variability is minimized.

$$RSS(group) \rightarrow \Big(RSS(part_1) + RSS(part_2) \Big)$$

(Make the 2 groups very different.)


group
part1
part2

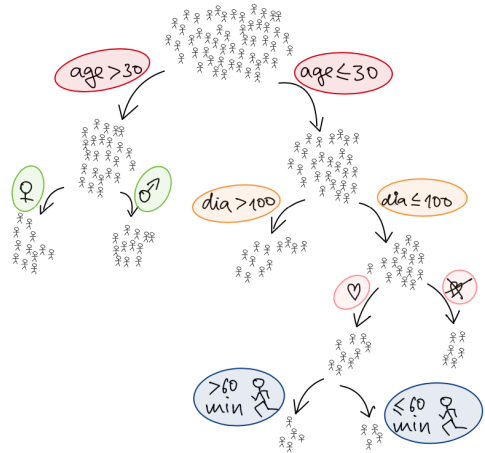

RSS(group)
$\bar{Y}_{group}$
$\bar{Y}_{part_1}$
$\bar{Y}_{part_2}$
RSS(part1)
RSS(part2)

- If all p variables are numerical. We have $p \cdot (n-1)$ different ways of how to partition the data.
- How big should the tree be?
  - Stop if the *RSS* improvement is small. (what is "small"?)
  - Prunning: Grow a large tree and cut the "leaves".
  - Choose tree that minimize $CC(tree) = \sum_{leaf_j} RSS(part_j) + \underbrace{\lambda(\#leaves)}_{\text{penalty for complexity}}$
  - How to choose penalty λ? So that we predict well! E.g. by cross-validation.

- If all p variables are numerical. We have $p \cdot (n-1)$ different ways of how to partition the data.
- How big should the tree be?
  - Stop if the *RSS* improvement is small. (what is "small"?)
  - Prunning: Grow a large tree and cut the "leaves".
  - Choose tree that minimize $CC(tree) = \sum_{leaf_j} RSS(part_j) + \underbrace{\lambda(\#leaves)}_{\text{penalty for complexity}}$
  - How to choose penalty λ? So that we predict well! E.g. by cross-validation.

It is a regression:

$$
\begin{aligned}
Y &= \beta_1 I(age > 30) \cdot I(gender = ♀) \\
&+ \beta_2 I(age > 30) \cdot I(gender = ♂) \\
&+ \beta_3 I(age \leq 30) \cdot I(dia > 100) \\
&+ \beta_4 I(age \leq 30) \cdot I(dia \leq 100) \\
&+ \beta_5 I(age \leq 30) \cdot I(dia \leq 100) \cdot I(♡) \cdot I(sport > 60) \\
&+ \beta_6 I(age \leq 30) \cdot I(dia \leq 100) \cdot I(♡) \cdot I(sport \leq 60) \\
&+ \beta_7 I(age \leq 30) \cdot I(dia \leq 100) \cdot I(\not{♡}) \\
&+ \varepsilon
\end{aligned}
$$



Prediction in a leaf is a simple average.

PROS

- Trees are visually appealing and easy to understand.
- No parametric structure is imposed
- Ability to capture complex interactions

CONS

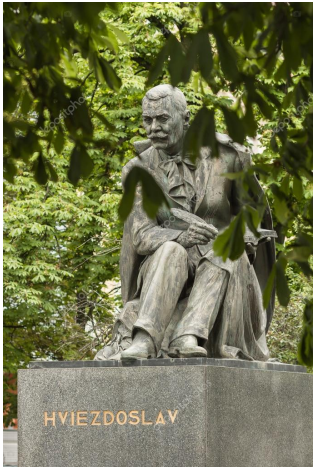
- Sensitive if observations have values around the cutting points
- Variance is high. Small change in the data can results in a large change of estimated model.

Random forrest

Pozdravujem vás, lesy, hory,
z tej duše pozdravujem vás!

I greet you, forests, mountains, I greet you from that soul!

P. O. Hviezdoslav



P.O.H. in a forrest.

Random forrest

High variability in a single tree?

We use bootstrap to grow many trees.

Then we average across the predictions across multiple trees.

Or we may give higher weight to the trees that predict better (**Bagging**)

Random forrest.

Random forrest - many similar trees

Not all that random, right?

Random forrest

Trees are somewhat different because they are based on different bootstrapped samples.

We may explicitly make the tree even more different.

We only choose a subset of regressors. Say randomly pick $\sqrt{p}$ of them instead of all p.

How many trees? Enough so that the prediction error does not change.

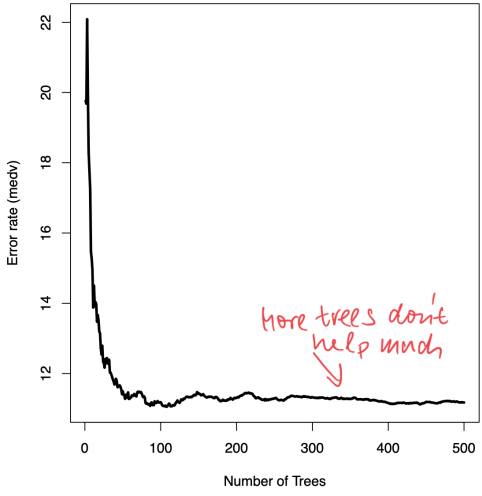
Random forrest.

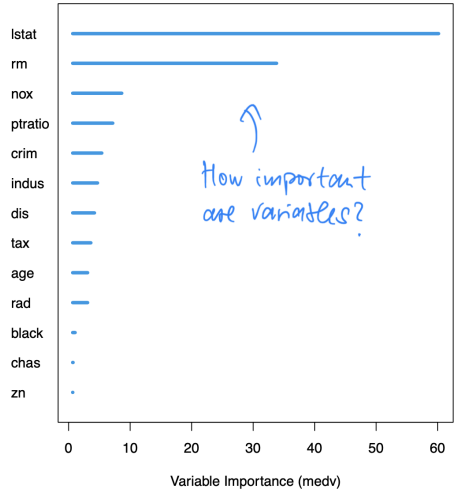
Random forrest - many different trees

Effect of a variable

How to quantify an effect of a particular variable on outcome in such a complicated object as random forrest?

- Partial dependence - set the variable at a particular value for all the observations and look at the mean predictions.
- Partial effects - Fix all the other variables at mean values and look at the mean predictions.
- Importance - Use out-of-bag data to access change in MSE after you randomly permute a predictor.
- Minimal depth - How early or late is the variable used for a split? Average the depth of the first split.
- SHAP values (will not cover this)


Error rate (medv)
22
20
18
16
14
12
0
100
200
300
400
500
Number of Trees
More trees don't help much


lstat
rm
nox
ptratio
crim
indus
dis
tax
age
rad
black
chas
zn
How important are variables?
0
10
20
30
40
50
60
Variable Importance (medv)

Classification using trees and random forrest

- So far, we looked at continuous outcome variable.
- The presented ideas are often used for classification (categorical variable).
- The measure of group difference is not RSS in this case (e.g. deviance, entropy, Gini index).
- The prediction in the leaf is a proportion.

Why random forrest?

Random forrest is considered as one of the best off-the-shelf predictor/classifier.

Especially if you don't know much of the problem.

It is relatively easy to compute.

Interpretability is not the nicest.

(*) Machine learning and causality

Prediction is nice, but economists often care more about the underlying mechanism more.

While ML gives us many great prediction tools, we are often interested in a certain variable of interest.

Having a lot of information we need to cope with high dimensionality of covariates. But sample size is small.

Lot of information is great, it could make our Selection on observables assumption more credible!

Motivation (cont'd)

More information is desirable. Traditional models are not feasible.

It helps with

- statistical precision - reduces uncertainty
- identification - treated and non-treated job-seekers are more comparable

Also, we wish to have flexible model specification.

Can ML algorithms help??

Can we make use of the great predictive capabilities of ML algorithms for improving the estimation of parameters of interest?

This is an area of active research. Here we will discuss one important paper on **DOUBLE MACHINE LEARNING**

Chernozhukov, Victor, et al. ”Double/debiased machine learning for treatment and structural parameters.” The Econometrics Journal 21.1 (2018): C1-C68.

Double Machine Learning framework

Example: Consider the following partially linear model. θ is the parameter of interest. $g(X)$ and $m(X)$ are some flexible functions, not of interest

$$\begin{aligned} Y &= \theta D + g(X) + U, & E[U|D,X] = 0 \\ D &= m(X) + V, & E[V|X] = 0 \end{aligned}$$

Split the data into two parts

- Use the first one to get $\hat{g}$ by some ML algorithm (LASSO, RF)
- Use the second portion of data to get $\hat{\theta}$ from regressing $Y - \hat{g}(X)$ on D

$\hat{\theta}_1$ is based on $E[\psi_1] = 0$ where $\psi_1 = D(Y - g(X) - \theta D)$

How does this naive estimator $\hat{\theta}_1$ behave?

$$\sqrt{n}(\hat{\theta}_1 - \theta) = \underbrace{\left(\frac{1}{n}\sum_i D_i^2\right)^{-1} \frac{1}{\sqrt{n}}\sum_i D_i U_i}_{\text{Nicely behaved, approx. Gaussian}} + \underbrace{\left(\frac{1}{n}\sum_i D_i^2\right)^{-1} \frac{1}{\sqrt{n}}\sum_i D_i(g(X_i) - \hat{g}(X_i))}_{\text{In general divergent.}}$$

Why?

$$\left(\frac{1}{n}\sum_i D_i^2\right)^{-1} \frac{1}{\sqrt{n}}\sum_i D_i(g(X_i) - \hat{g}(X_i)) = \left(E[D_i^2]\right)^{-1} \frac{1}{\sqrt{n}}\sum_i \underbrace{m_i(X_i)}_{\neq 0} \underbrace{(g(X_i) - \hat{g}(X_i))}_{\text{more slowly than } \sqrt{n}} + \underbrace{o_P(1)}_{\to_P 0}$$

So it leads to a regularization bias.

Double Machine Learning framework

Now we do something else.

Instead of $\psi_1 = D(Y - g(X) - \theta D)$ we will base our estimation on different moment conditions:

$$\psi_2 = V(Y - g(X) - \theta D) = (D - m(X)) \cdot (Y - g(X) - \theta D)$$
$$\psi_3 = V(Y - g(X) - \theta V) = (D - m(X)) \cdot (Y - g(X) - \theta (D - m(X)))$$

These moment conditions are somewhat more ”clever” as the problematic regularization bias part will converge to zero under mild conditions.

$\hat{\theta}_2$ based on ψ_2

Split the data into two parts

- Use the first one to get $\hat{g}$ and $\hat{m}$ by some ML algorithm (LASSO, RF)
- Use the second portion of data to get $\hat{V} = D - \hat{m}(X)$ and use this to get $\hat{\theta}_2$ .....

$$\sqrt{n}(\hat{\theta}_2 - \theta) = \underbrace{a^*}_{\text{Nicely behaved, approx. Gaussian}} + \underbrace{b^*}_{\text{Regularization bias}} + \underbrace{c^*}_{\text{Overfitting bias}}$$

- Regularization bias : $b^* = \left(\frac{1}{n}\sum_i D_i^2\right)^{-1} \frac{1}{\sqrt{n}} \sum_i (m(X_i) - \hat{m}(X_i))(g(X_i) - \hat{g}(X_i))$
- Overfitting bias: Sample splitting takes care of this.

Regularization bias : $b^* = \left(\frac{1}{n}\sum_i D_i^2\right)^{-1} \frac{1}{\sqrt{n}} \sum_i (m(X_i) - \hat{m}(X_i)) \cdot (g(X_i) - \hat{g}(X_i))$

$\hat{g}$ and $\hat{m}$ no longer need to converge at the rate $n^{-1/2}$

It is sufficient if they both converge at the rate $n^{-1/4}$ and this is much much easier for the ML algorithms.

$\hat{\theta}_3$ based on ψ_3

Split the data into two parts

- Use the first one to get $\hat{g}$ and $\hat{m}$ by some ML algorithm (LASSO, RF)
- Use the second portion of data to get $\hat{V} = D - \hat{m}(X)$ and $\hat{W} = Y - \hat{m}(X)$ and use this to get $\hat{\theta}_3$ via regressing $\hat{W}$ on $\hat{V}$

This is, in fact ortogonalization. We project both D and Y onto space spanned by X. By means of Frisch-Waugh-Lowell theorem we recover the coefficient of D.

Similar decomposition can be shown. Regularization bias also includes cross product $(m(X_i) - \hat{m}(X_i)) \cdot (g(X_i) - \hat{g}(X_i))$

What makes ψ_2 and ψ_3 different from ψ_1 ???

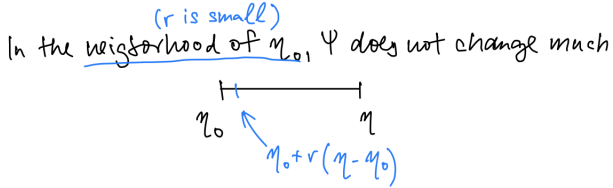
Regularization bias vanishes under mild conditions.

In other words, ψ_2 and ψ_3 are both locally insensitive to some mild perturbations of $\hat{m}, \hat{g}$ around m, g.

Neyman-orthogonality

This local insensitiveness has a name: **Neyman-orthogonality**.

- ψ is a moment condtion
- θ is the parameter of interest (target parameter), θ_0 is the true one
- η is the nuisance parameter, θ_0 is the true one
- W denotes data



$$\frac{\partial}{\partial r} E[\psi(W; \theta_0, \eta_0 + r(\eta - \eta_0))]\Big|_{r=0} = 0$$

Neyman-orthogonality of ψ_2

We will verify that ψ_2 satisfy the Neyman-orthogonality condition, while ψ_1 does not.
Notation

- $\eta = (m, g)$ is the vector of nuisance parameters, $\theta_0 = (m_0, g_0)$ is the true one
- $\eta_r = \eta_0 + r(\eta - \eta_0)$.

Neyman-orthogonality of ψ_2

$$
\begin{aligned}
\psi_2(W;\theta_0,\eta_r) &= (D-m_0(X)-r(m(X)-m_0(X)))\cdot(Y-g_0(X)-r(g(x)-g_0(X))-D\theta_0) \\
&= (D-m_0(X))\cdot(Y-g_0(X)-D\theta_0)+ \\
&\quad -r(D-m_0(X))\cdot(g(x)-g_0(X)) \\
&\quad -r(m(X)-m_0(X))\cdot(Y-g_0(X)-D\theta_0) \\
&\quad +r^2(m(X)-m_0(X))\cdot(g(x)-g_0(X)) \\
\frac{\partial}{\partial r}E[\psi_2(W;\theta_0,\eta_r)] &= -E[(D-m_0(X))\cdot(g(x)-g_0(X))] \\
&\quad -E[(m(X)-m_0(X))\cdot(Y-g_0(X)-D\theta_0)] \\
&\quad +2\cdot r\cdot E[(m(X)-m_0(X))\cdot(g(x)-g_0(X))] \\
\frac{\partial}{\partial r}E[\psi_2(W;\theta_0,\eta_r)]\Big|_{r=0} &= -E[(D-m_0(X))\cdot(g(x)-g_0(X))] \\
&\quad -E[(m(X)-m_0(X))\cdot(Y-g_0(X)-D\theta_0)]
\end{aligned}
$$

Neyman-orthogonality of ψ_2

$$\begin{aligned}\frac{\partial}{\partial r} E[\psi(W;\theta_0,\eta_r)]\Big|_{r=0} &= -E[(D-m_0(X))\cdot(g(x)-g_0(X))] \\ &\quad -E[(m(X)-m_0(X))\cdot(Y-g_0(X)-D\theta_0)] \\ &= 0\end{aligned}$$

because

$$E[(D-m_0(X))\cdot(g(x)-g_0(X))] = E[(g(x)-g_0(X))\cdot\underbrace{E[D-m_0(X)|X]}_{E[V|X]=0}] = 0$$

$$E[(m(X)-m_0(X))\cdot(Y-g_0(X)-D\theta_0)] = E[(m(X)-m_0(X))\cdot\underbrace{E[Y-g_0(X)-D\theta_0|X]}_{E[U|X,D]=0}] = 0$$

and hence ψ_2 is indeed Neyman-orthogonal.

Neyman-orthogonality of ψ_3

Follows similarly as ψ_2 but ithe derivation is a little bit longer.

Neyman-orthogonality of ψ_1 ???

$$
\begin{aligned}
\psi_1(W;\theta_0,\eta_r) &= D\cdot(Y-g_0(X)-r(g(x)-g_0(X))-D\theta_0) \\
\frac{\partial}{\partial r}E[\psi_2(W;\theta_0,\eta_r)] &= -E[D\cdot(g(x)-g_0(X))] \\
\frac{\partial}{\partial r}E[\psi(W;\theta_0,\eta_r)]\Big|_{r=0} &= -E[D\cdot(g(x)-g_0(X))] \\
&\neq 0
\end{aligned}
$$

There is nothing we could do to use $E[U|X,D]=0$ and $E[V|X]=0$ to make this term equal to zero.

Overfitting bias

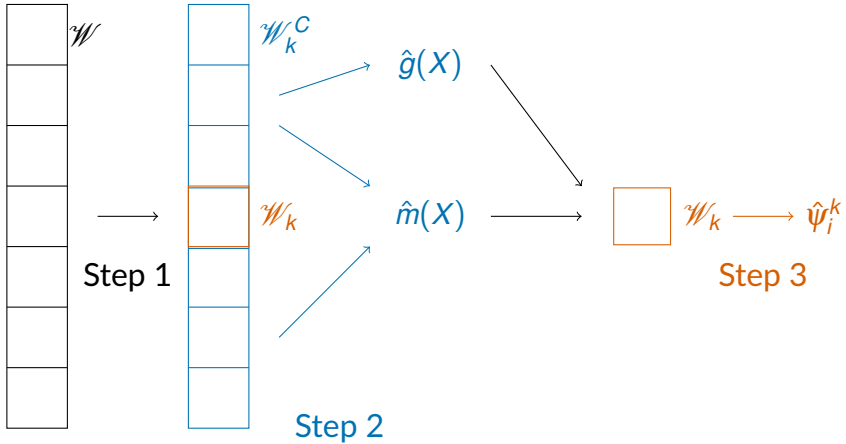
$$\sqrt{n}(\hat{\theta}_2 - \theta) = \underbrace{a^*}_{\text{Nicely behaved, approx. Gaussian}} + \underbrace{b^*}_{\text{Regularization bias}} + \underbrace{c^*}_{\text{Overfitting bias}}$$

Overfitting bias may arise from the fact that the same data is used for both estimation of nuisance functions and target parameter.

We can split the data. Randomly split data into two parts. Use one for nuisance parameter estimation, the other one for target. $\rightarrow$ But then we loose many observations.

How to fix this? Swap the roles of the two data parts and then average across them!

Sample splitting for dealing with overfitting bias



$$\hat{\psi} = \frac{1}{n}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\hat{\psi}_i^k$$

Step 4

DML wrap-up (1)

There are different ways how one can estimate θ. We saw three: $\hat{\theta}_1$, $\hat{\theta}_2$ and $\hat{\theta}_3$.

These three estimators are based on three different moment condition functions: ψ_1, ψ_2 and ψ_3.

While ψ_1 was locally sensitive to some small changes in the η, the other two ψ_2 and ψ_3 were not.

This allows us to get rid of the regularization bias.

Sample-splitting removes the overfitting bias.

DML wrap-up (2)

- Estimator $\hat{\theta}$ based on Neyman-orthogonal moment function ψ
- Apply sample splitting
- Nuisance parameter estimators are ”good enough” (e.g. converge at rate at least $n^{-1/4}$ - so that the regularization bias. vanishes)

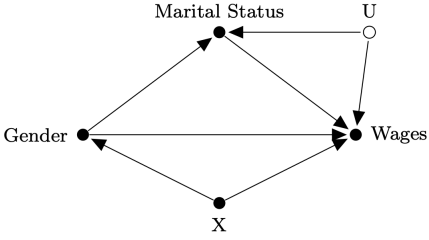
We get that (Theorem 1 in Chernozhukov et al. 2019)

$$\sqrt{n}(\hat{\theta} - \theta) \to N(0, \sigma^2)$$

Asymptotically normally distributed estimator that is $\sqrt{n}$ consistent.

Limitations - Kitchen sink regression

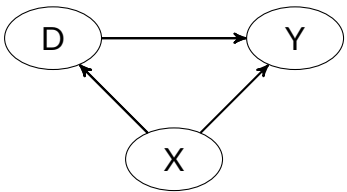
[proper source should be cited here]



Hünermund, Beyers and Caspi (2021)

Hünermund, Paul, Beyers Louw, and Itamar Caspi. ”Double Machine Learning and Bad Controls–A Cautionary Tale.” arXiv preprint arXiv:2108.11294 (2021).

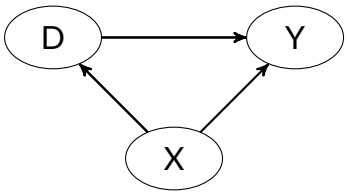
DML and policy evaluation



Notation:

- $Y(d)$: (Potential) outcome as function of treatment d
- Y - observed outcome
- D - observed treatment
- X - observed covariates

DML and policy evaluation



Object of interest:

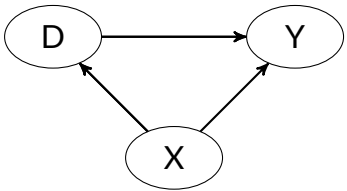
$$\Delta = E[Y(1) - Y(0)]$$

Indentifying assumptions:

1) Conditional independence of D:
$\{Y(1), Y(0)\} \perp D \mid X$

2) Common support:
$\Pr(D = d | X = x) > 0$

DML and policy evaluation



Moment function:

$$\begin{aligned} \psi(W;\theta_0,\eta) &= \frac{I\{D=d\}\cdot[Y_2-\mu(d,X)]}{p(X)} + \mu(d,X) - \theta_0. \\ E\Big[\psi(W;\theta_0,\eta)\Big] &= E\Big[Y(d)\Big] - \theta_0 = 0 \end{aligned}$$

Data: $W = (Y, D, X)$

Nuisance functions: $\eta = (p, \mu)$

- $p(X) \equiv \Pr(D = d|X)$
- $\mu(D, X) \equiv E[Y|D, X]$

Bang, Heejung, and James M. Robins. "Doubly robust estimation in missing data and causal inference models." Biometrics 61.4 (2005): 962-973.

DML applications

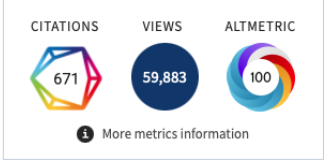
There are **many**:

Double/debiased **machine learning** for treatment and structural parameters [HTML] oup.com

V Chernozhukov, D Chetverikov, M Demirer, E Duflo… - 2018 - academic.oup.com

… To estimate η 0 , we consider the use of statistical or **machine learning** (ML) methods, which are … We call the resulting set of methods **double** or debiased ML (DML). We verify that DML …

☆ Save 99 Cite Cited by 1391 Related articles All 24 versions



Most read

Double/debiased machine learning for treatment and structural parameters

DML provides a framework for developing estimators that:

- can handle high-dimensional data
- make use of predictive powers of ML
- are well behaved under mild conditions

Thank you for your attention!

References

- Why ML is becoming interesting in Economics by Varian, Hal R. "Big data: New tricks for econometrics." Journal of Economic Perspectives 28.2 (2014): 3-28.
- More recent JEP article on ML: Mullainathan, Sendhil, and Jann Spiess. "Machine learning: an applied econometric approach." Journal of Economic Perspectives 31.2 (2017): 87-106.
- THE most standard and by far the best book: James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer. Made free by the authors `https://www.statlearning.com` Free online course based on this book: `https://www.dataschool.io/15-hours-of-expert-machine-learning-videos/`
- More comprehensive, but somewhat less accessible book: [ESL] Friedman, Jerome H. The elements of statistical learning: Data mining, inference, and prediction. Springer open, 2017.
- Beautiful exposition of the essential of Bias-Variance trade-off `https://daviddalpiaz.github.io/r4sl/biasvariance-tradeoff.html`
- LASSO: [SLS] Hastie, Trevor, Robert Tibshirani, and Martin Wainwright. Statistical learning with sparsity: the lasso and generalizations. Chapman and Hall/CRC, 2019. `https://web.stanford.edu/~hastie/StatLearnSparsity/`
- LASSO: If you only have 1hour, read this: Sparsity and the Lasso (Statistical Machine Learning, Spring 2015) Ryan Tibshirani (with Larry Wasserman) `http://www.stat.cmu.edu/~larry/=sml/sparsity.pdf`
- "Tree" idea: Morgan, James N., and John A. Sonquist. "Problems in the analysis of survey data, and a proposal." Journal of the American statistical association 58.302 (1963): 415-434.
- Standard reference book on trees: Breiman, Leo, et al. Classification and regression trees. Routledge, 2017.
- SHAP values - original paper: Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions." Proceedings of the 31st international conference on neural information processing systems. 2017.
- SHAP values - digestible intro: `https://towardsdatascience.com/explain-your-model-with-the-shap-values-bc36aac4de3d`
- LIME values (tool for explaining similar to SHAP) - digestible intro: `https://medium.com/dataman-in-ai/explain-your-model-with-lime-5a1a5867b423`

References

- Double machine learning framework: Chernozhukov, Victor, et al. "Double/debiased machine learning for treatment and structural parameters." The Econometrics Journal 21.1 (2018): C1-C68.
- Somewhat accessible intro to DML: https://towardsdatascience.com/double-machine-learning-for-causal-inference-78e0c6111f9d
- DML video by one of the authors of DML https://www.youtube.com/watch?v=eHOjmyoPCFU
- DoubleML package in R https://cran.r-project.org/web/packages/DoubleML/DoubleML.pdf
- Bach, Philipp, et al. "DoubleML–An Object-Oriented Implementation of Double Machine Learning in R." arXiv preprint arXiv:2103.09603 (2021).