Applied Data Science

Machine Learning
Lecture 2

John Tsitsiklis

February 7, 2022

Today's agenda

- Further topics on regression
  - what can go wrong
    heteroskedasticity
    mutlicollinearity
  - learning to predict versus learning a model
    endogeneity
  - using nonlinear features of the data
  - overfitting and regularization
    ridge regression
    sparse regression and Lasso

- Performance assessment, testing, and validation
  - training versus testing
  - cross-validation
  - bootstrap

FURTHER TOPICS ON REGRESSION

Quick reminder

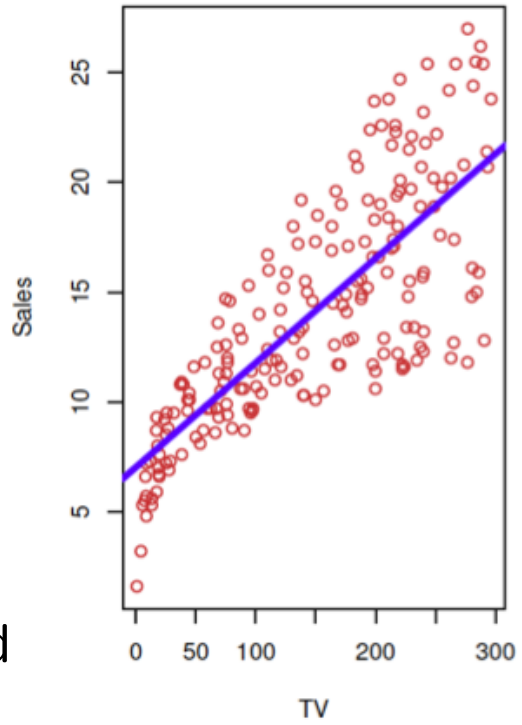
- Structural model: $Y_i = (\boldsymbol{\theta}^*)^T \mathbf{X}_i + W_i$
- Estimator: $\widehat{\boldsymbol{\Theta}} = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \mathbf{Y}$
- Predictor: $\widehat{Y} = \widehat{\boldsymbol{\Theta}}^T \mathbf{X}$
- Standard error σ_j: standard deviation of $\widehat{\Theta}_j$
- 95% confidence interval: $\text{CI} = [\widehat{\Theta}_j - 2\sigma_j, \widehat{\Theta}_j + 2\sigma_j]$
- Wald test: reject "$\theta_j^* = 0$" if $0 \notin \text{CI}$

Heteroskedasticity

- variance σ_i^2 of W_i changes with i
  (perhaps depends on $\mathbf{X}_i$)

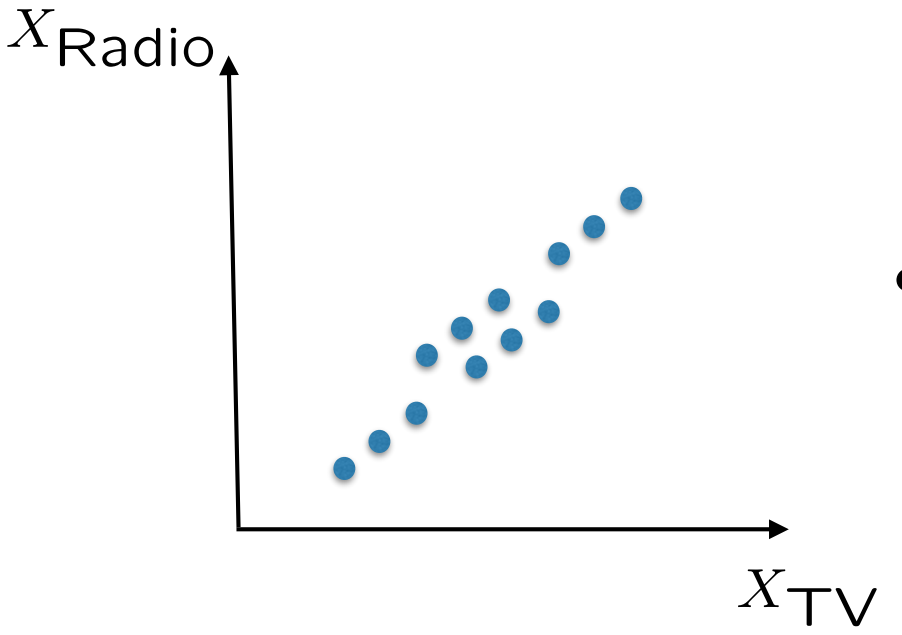
– **if** we knew the σ_i^2,
  could use a weighted least squares criterion

- formulas for standard errors, etc., do not hold



Multicollinearity

- vectors X (approximately) confined to a lower-dimensional set



- no way to tell relative effect of TV versus Radio

- matrix $\mathbb{X}$ does not have full rank (approximately)
- standard errors involve $(\mathbb{X}^T\mathbb{X})^{-1}$: infinite or huge

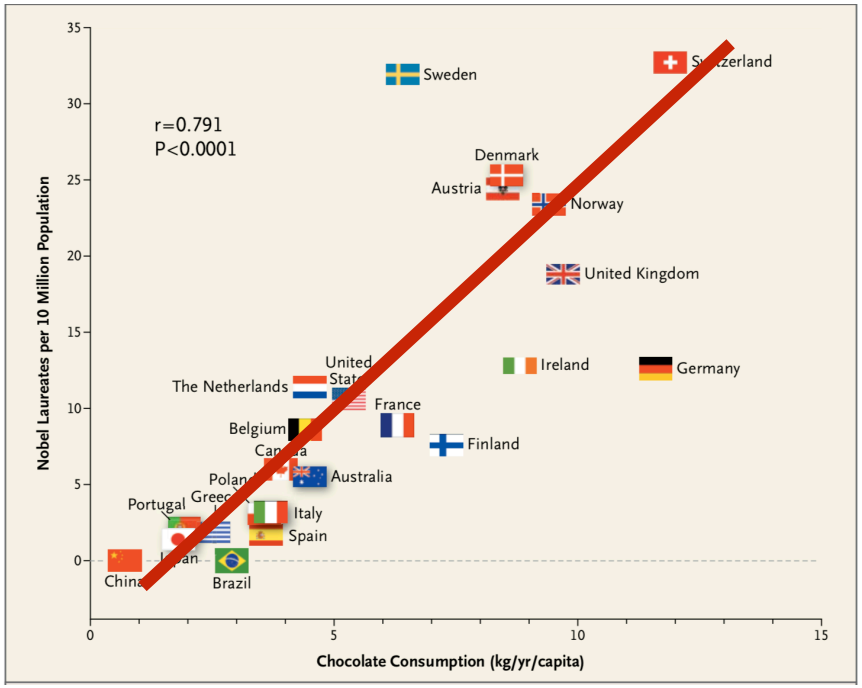
- What do to? Remove some variables

Prediction versus modeling

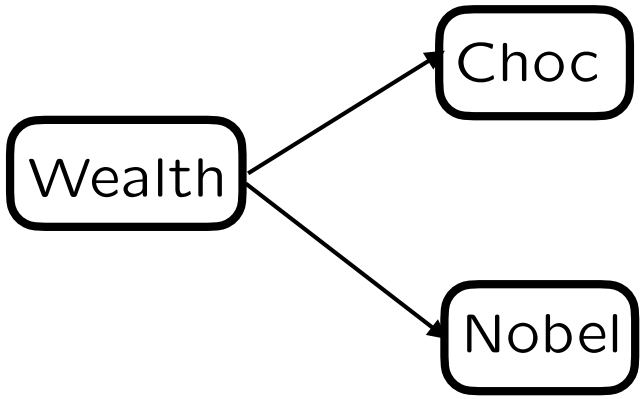
BUSINESS INSIDER

There's A Shocking Connection Between Eating More Chocolate And Winning The Nobel Prize

JOE WEISENTHAL
APR. 20, 2014, 11:10 AM



The NEW ENGLAND JOURNAL *of* MEDICINE

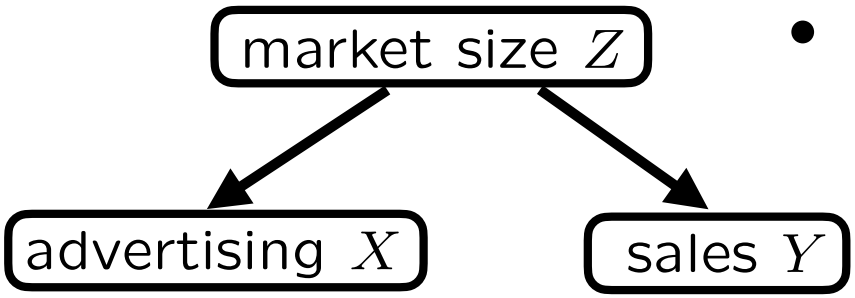


Perhaps useful for making predictions

Not enough for drawing conclusions about causality or structural model

Endogeneity: modeling is hard in the presence of latent variables

- Suppose that, for the purposes of policy-making,
  we want to assess the effect on Y,
  when **we** increase X by 1 unit.



- Suppose that Z determines everything:

  $X \approx Z$ and $Y \approx Z$

- Regression finds that X predicts Y very well

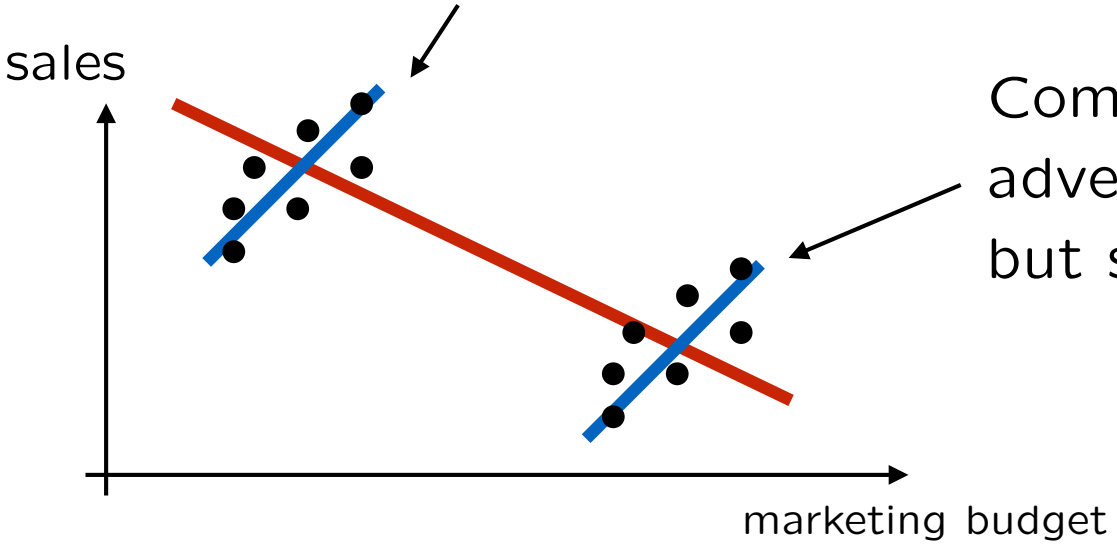
  but the estimate of the effect of X on Y will be wrong

  – a big issue in the social and bio/medical sciences

Possible endogeneity in advertising and sales

- Suppose there are two types of towns/markets

No competing stores/dealers,
advertise little,
strong sales no matter what



Competing stores/dealers,
advertise a lot,
but sales can never be too high

- Estimated regression model
  "shows" that more advertising
  results in fewer sales

Simpson's paradox

Endogeneity

- Common when there are relevant latent confounding variables not included in the model
- Common when dealing with time series data
- We can still get a "good" predictor
- But:
  - we do not get the right structural model
  - cannot answer "what if" questions
  - formulas for standard errors, etc., do not hold

Mitigating endogeneity: use more variables

$$\widehat{\text{Sales}} = \theta_0 + \theta_1 \cdot (\text{TV}) + \theta_2 \cdot (\text{Radio}) + \theta_3 \cdot (\text{NewsP})$$

$$+\theta_4 Z \qquad +\theta_5 U \qquad +\theta_6 V$$

Z = market size

$V = 0$: rural
$V = 1$: urban

$U = 0$: market with competitors
$U = 1$: market without competitors

- Note the use of categorical variables
- This example: 4 categories
  - encode as $C = 1, 2, 3, 4$ and use $+\theta_7 C$ instead of $+\theta_5 U + \theta_6 V$?

No!!!

A big question: what variables to include?

- Marketing example:

$$\widehat{\text{Sales}} = 12.35 + 0.055 \cdot (\text{NewsP})$$

"significant"

standard error $= 0.01$

- $\widehat{\text{Sales}} = 2.94 + 0.046 \cdot (\text{TV}) + 0.19 \cdot (\text{Radio}) - 0.001 \cdot (\text{NewsP})$

standard error $= 0.006$

supports the null hypothesis that $\theta_{\text{NewsP}} = 0$

once we take into account TV and Radio:
NewsP is "inconsequential"

Linear regression, with nonlinear features

- Why not include even more variables?
- New variables can be nonlinear functions of original variables

example: $\widehat{Y} = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \theta_3 \log X_2 + \theta_4 X_1 X_2$

original data vector: $\mathbf{X} = (1, X_1, X_2)$ predictors $\widehat{\boldsymbol{\theta}}^T \mathbf{X}$

augmented data vector: $\mathbf{X}_{\text{aug}} = (1, X_1, X_2, \log X_2, X_1 X_2)$

predictors $\widehat{\boldsymbol{\theta}}_{\text{aug}}^T \mathbf{X}_{\text{aug}}$

still linear in $\boldsymbol{\theta}_{\text{aug}}$

- In general: $\widehat{Y} = \sum_{j=1}^{k} \theta_j \phi_j(\mathbf{X}) = \boldsymbol{\theta}^T \phi(\mathbf{X})$

features

feature vector

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \left(Y_i - \boldsymbol{\theta}^T \phi(\mathbf{X}_i) \right)^2$$

Marketing example; try a multiplicative feature

- $\widehat{\text{Sales}} = 2.94 + 0.046 \cdot (\text{TV}) + 0.19 \cdot (\text{Radio}) - 0.001 \cdot (\text{NewsP})$

  $R^2 = 0.897$

$$\widehat{\text{Sales}} = 6.57 + 0.019 \cdot (\text{TV}) + 0.029 \cdot (\text{Radio}) + 0.001 \cdot (\text{TV}) \cdot (\text{Radio})$$

new feature

$R^2 = 0.968$

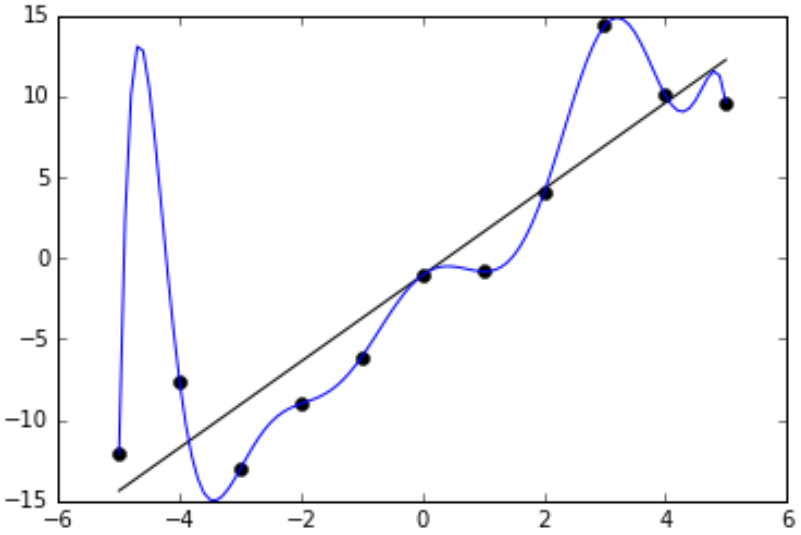
- Note: 0.001 looks small,

  but $(\text{TV}) \cdot (\text{Radio})$ is a relatively large number

- To ease interpretation:
  rescale the variables so that they have comparable ranges

Overfitting

- More variables, better R^2, "better" explanations
  - **but:** may be an illusion, fitting the noise



https://en.wikipedia.org/wiki/Overfitting

"With four parameters I can fit an elephant and with five I can make him wiggle his trunk"
(John von Neumann)

- Statistics texts have procedures and tests for adding removing variables
  - **but:** often rely on "standard" assumptions, possibly violated

- Two approaches (often used in combination)
  - regularization: penalize overfitting
  - data-driven methods, not relying on formulas

Regularization: Ridge regression

- "incentivize" parameters to stay small, rather than fitting the noise

$$\min_{\boldsymbol{\theta}} \left[\sum_{i=1}^{n} (Y_i - \boldsymbol{\theta}^T \mathbf{X}_i)^2 + \alpha \sum_{j=1}^{m} \theta_j^2 \right]$$

$\alpha \geq 0$: regularization hyperparameter

- Try different values of α

- minimize a quadratic: solve system of linear of equations

  - Can be interpreted as a Bayesian formulation: zero-mean normal prior on $\boldsymbol{\theta}$
    - α reflects the (inverse of the) variance of the prior
    - moves the parameter estimates closer to zero

Regularization: Lasso (sparsity enforcing)

- "incentivize" parameters to stay small, rather than fitting the noise

$$\min_{\boldsymbol{\theta}} \Big[\sum_{i=1}^{n} (Y_i - \boldsymbol{\theta}^T \mathbf{X}_i)^2 + \alpha \sum_{j=1}^{m} |\theta_j| \Big]$$

$\alpha \geq 0$: regularization hyperparameter

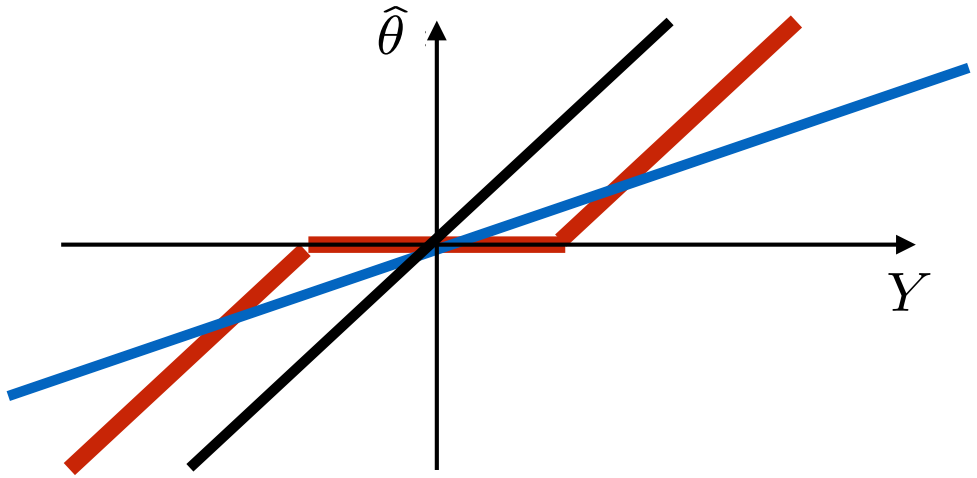
- Try different values of α
- Optimal solution tends to set many of the parameters to zero
- useful if we believe that the true vector θ is sparse, but do not know which entries are zero
  - larger $\alpha \longrightarrow$ more sparse solutions
- Convex optimization problem: fast solvers
- Strong theoretical guarantees (can discover true sparsity structure)

Toy example

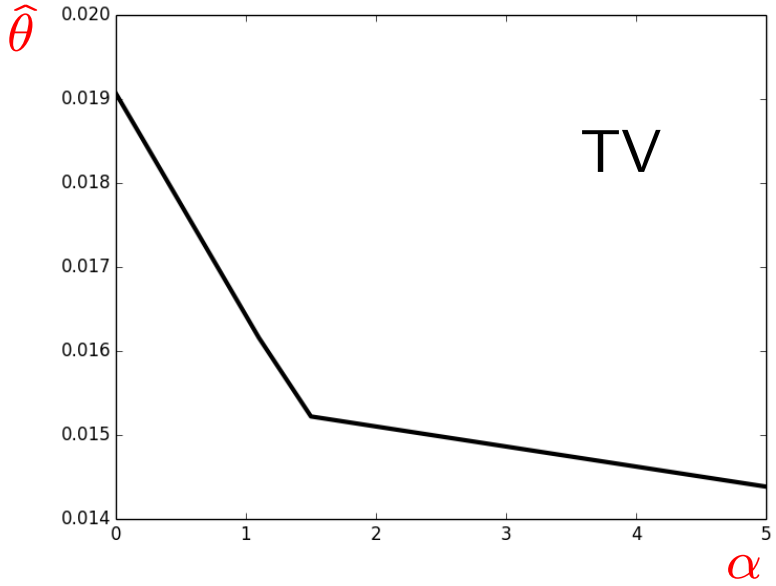
- True model: $Y = \theta^* + W$

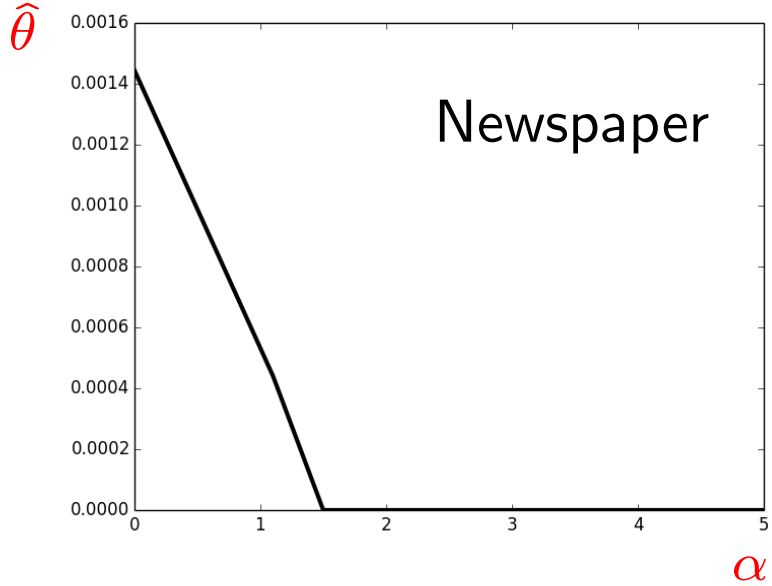
- Single data point Y
- Estimate θ^*

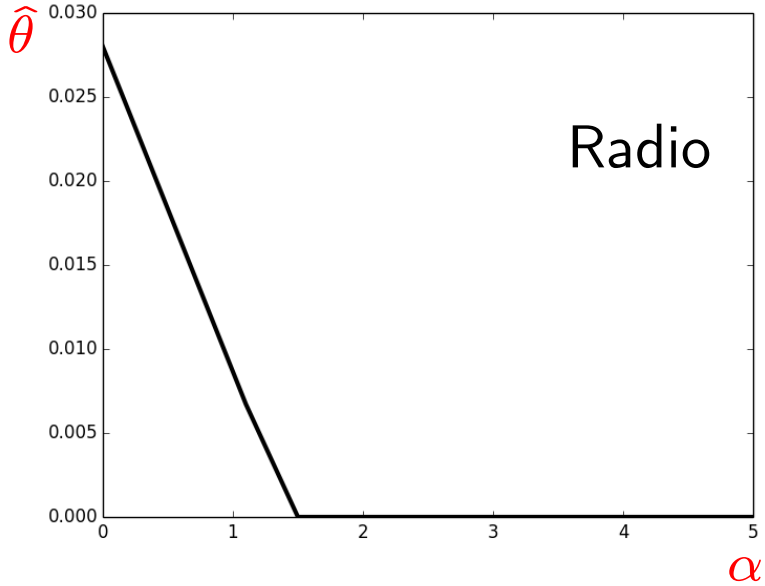
minimize	solution	
$(Y-\theta)^2$	$\widehat{\theta} = Y$	
$(Y-\theta)^2 + \theta^2$	$\widehat{\theta} = \frac{Y}{2}$	shrinkage
$(Y-\theta)^2 + \|\theta\|$		thresholding

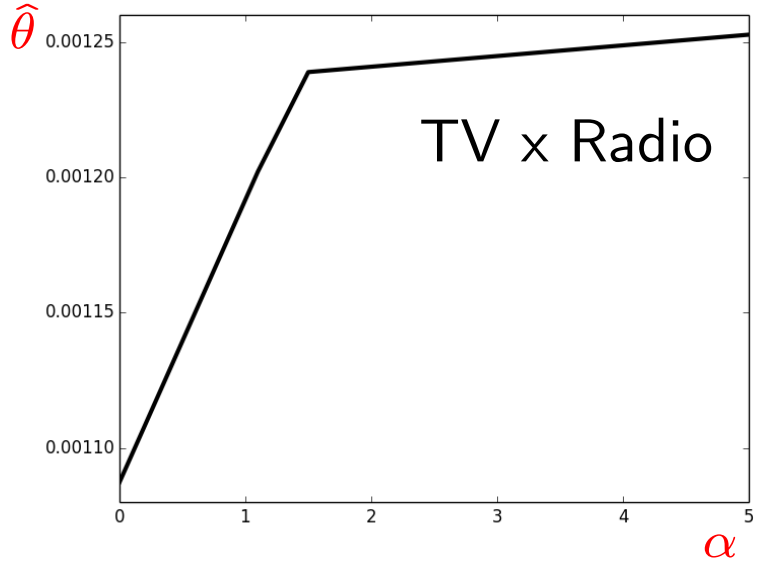


Marketing example: Lasso









PERFORMANCE ASSESSMENT

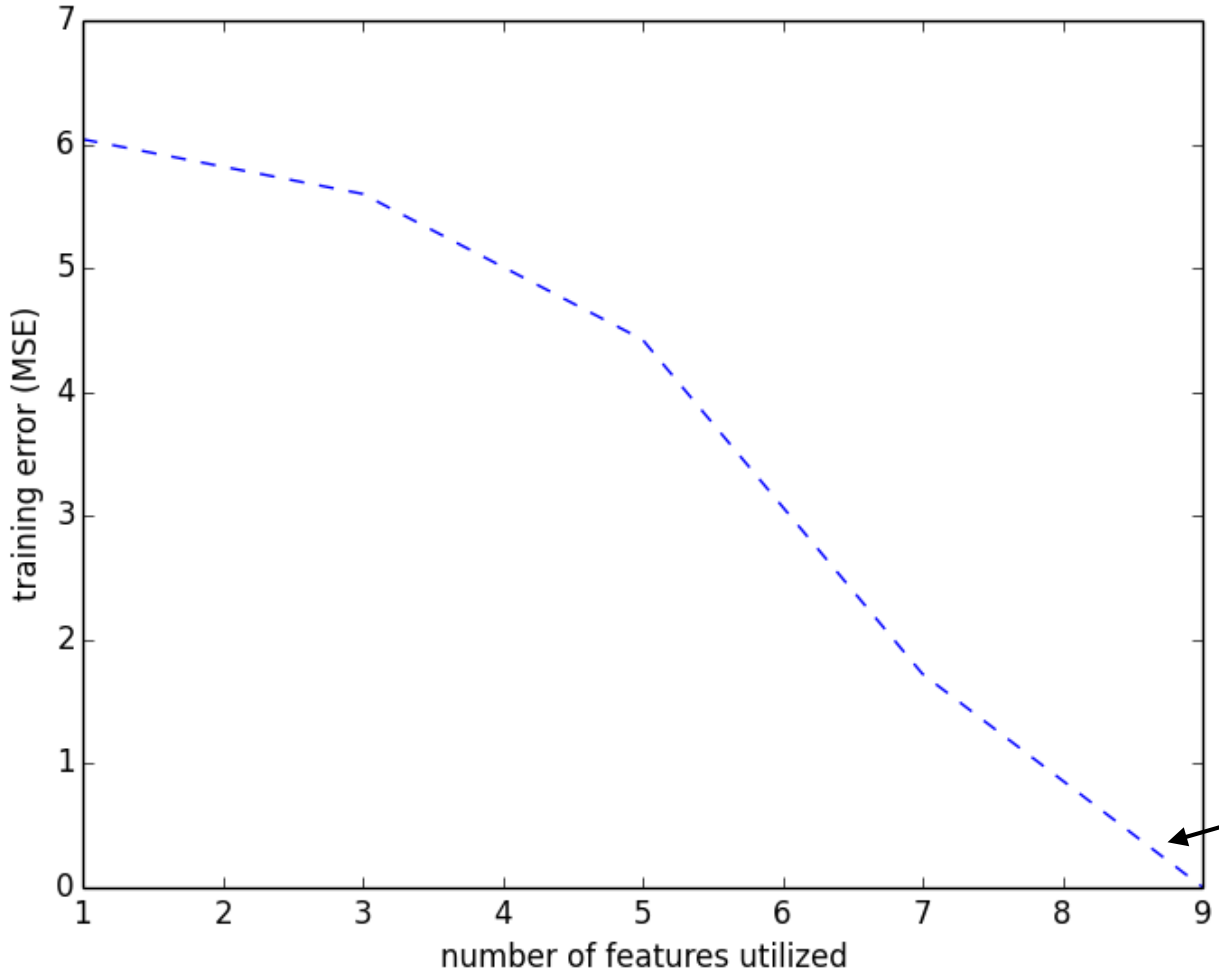
Too many "optimal" predictors to choose from

- Need systematic ways to:
  - assess standard errors
  - set hyperparameters
  - choose which variables/features to include
  - try less or more "complex" models
  - choose between different learning algorithms

- Interested in:
  - fitting and explaining existing data (e.g., a high R^2)
  - **generalization:** perform well on new data

- Especially when there are no applicable formulas

Prediction error drops with number of features

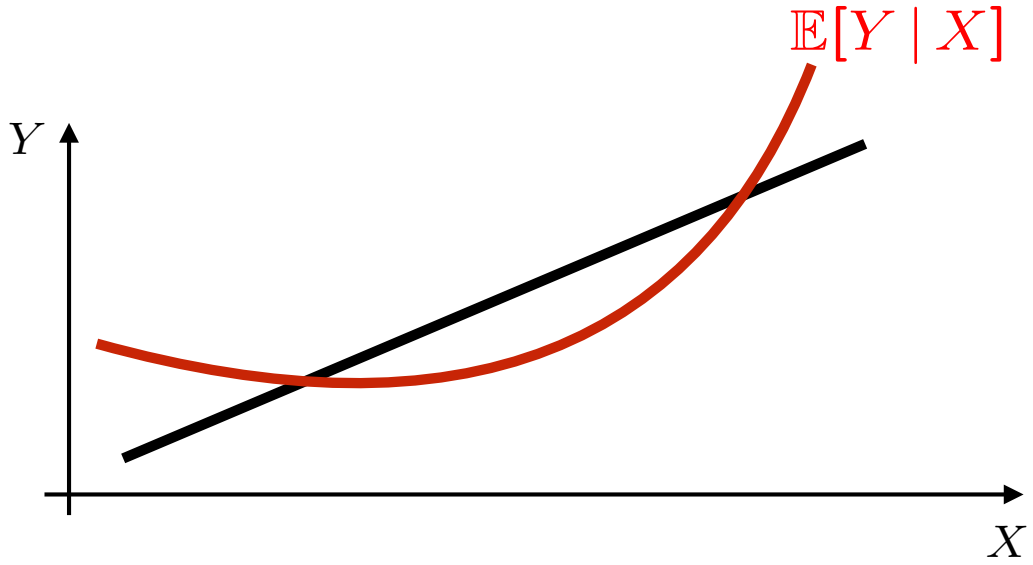


misleading?

on the data set used to train the model...

Bias-Variance tradeoff

- Too few features:
  - even with infinite data, $\widehat{\boldsymbol{\theta}}^T \mathbf{X} \neq \mathbb{E}[Y \mid \mathbf{X}]$ **(bias)**



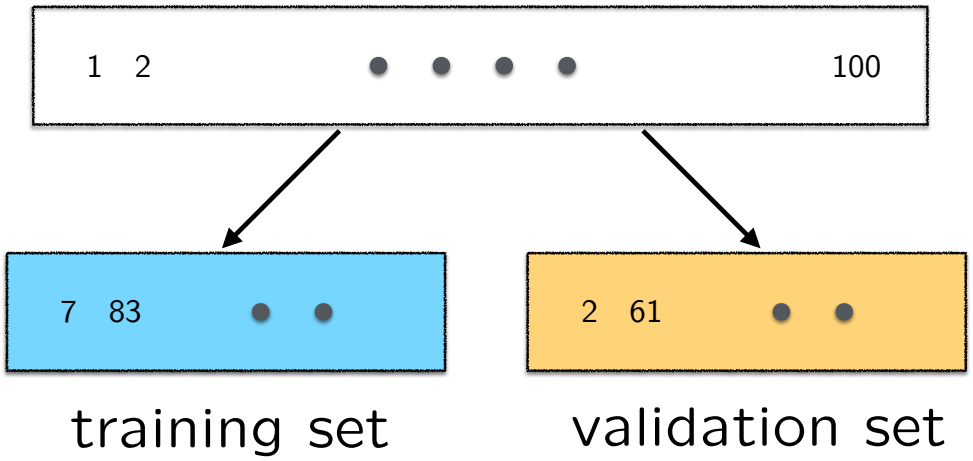
- Too many features
  - fewer "data per parameter"
  - parameter estimates are more "noisy" **(variance)**

ASSESSING PREDICTOR PERFORMANCE:

VALIDATION

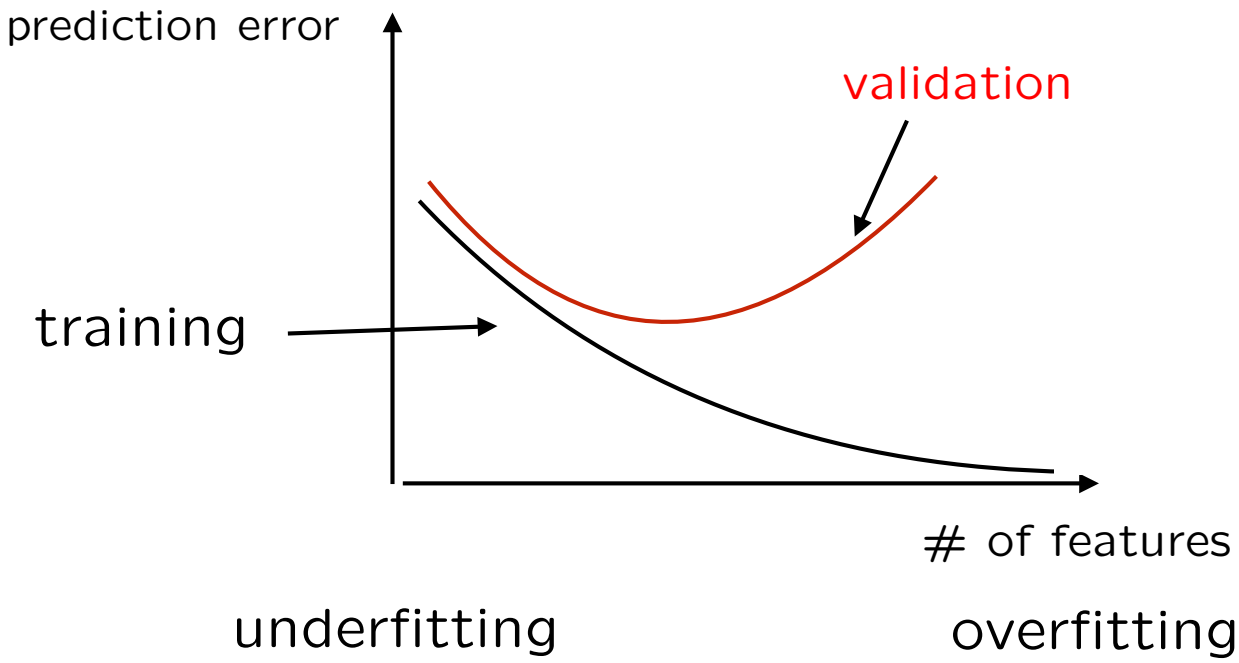
Use a validation set

- (Randomly) divide data into **training** and **validation** (hold out) set



- Use training set to fit the model
- Use validation set to assess performance on "new" data

Comparing different models



- Choose model structure that is best on validation set

– worry: maybe that model was lucky, overfitting the validation set

- Final evaluation on a third data set **(test set)**

- Similar procedure to find a "best" value of a hyperparameter

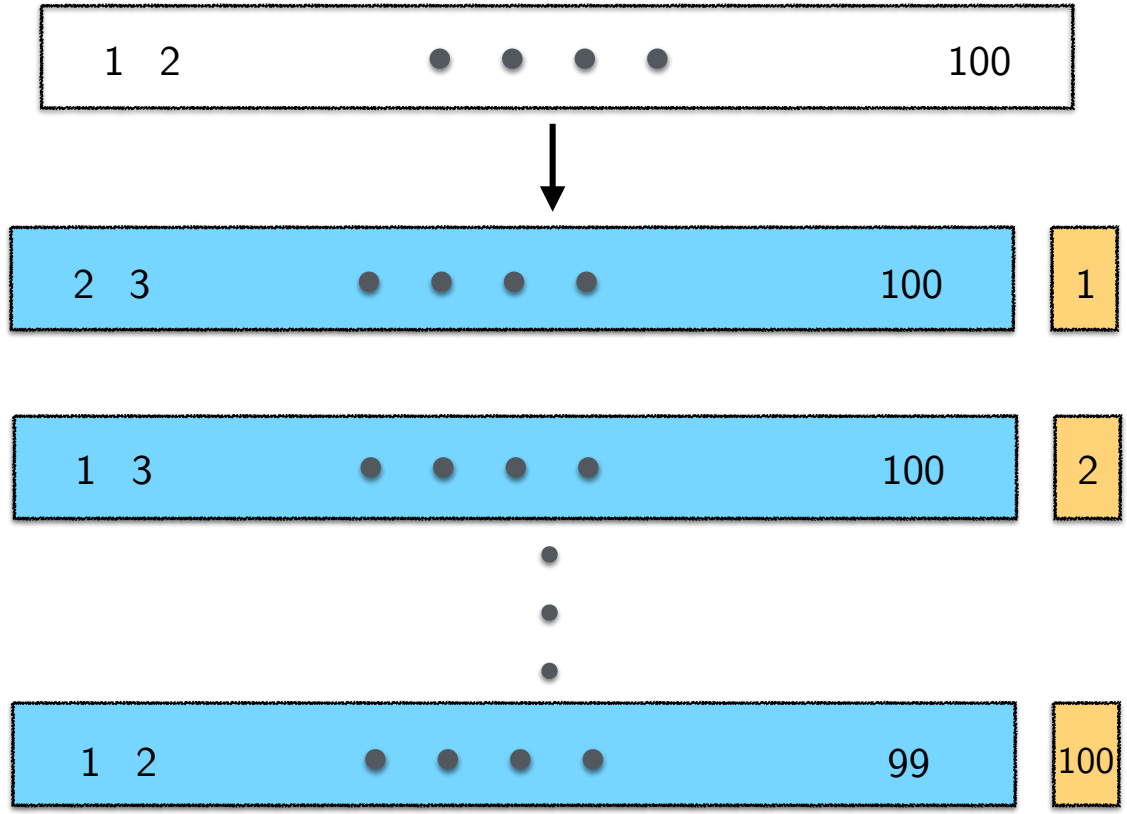
Drawbacks of this validation process

- Some data are "wasted" (not used for training)

- Error on validation set has significant randomness (depends on the choice of the random division)

Leave-One-Out Cross-Validation (LOOCV)

- Train on $n-1$ data points
- "Validate" on remaining data point
- Repeat for all n choices of left out data point
- Calculate mean square error, over the n repetitions



Leave-One-Out Cross-Validation (LOOCV) properties

- Advantages
  - no variability due to random choice of validation set
  - uses all data for training

- Drawbacks
  - have to train n times

    exception:

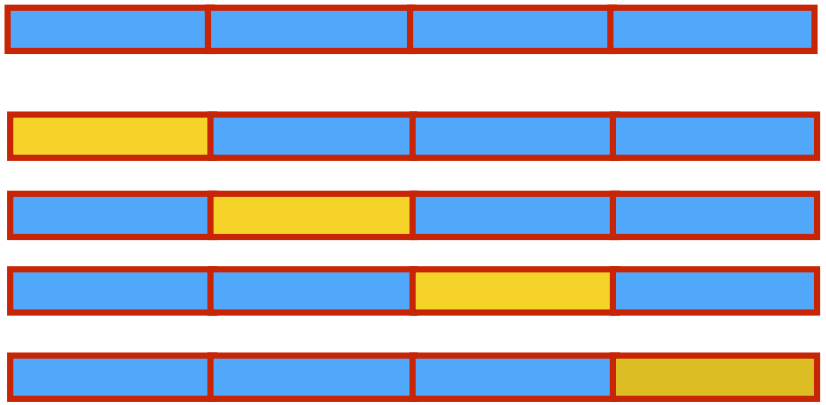
    linear regression,
    do not need to repeat
    all the computational work

  - the n prediction errors are highly dependent

    (not getting the advantage of n independent assessments)

k-fold Cross-Validation

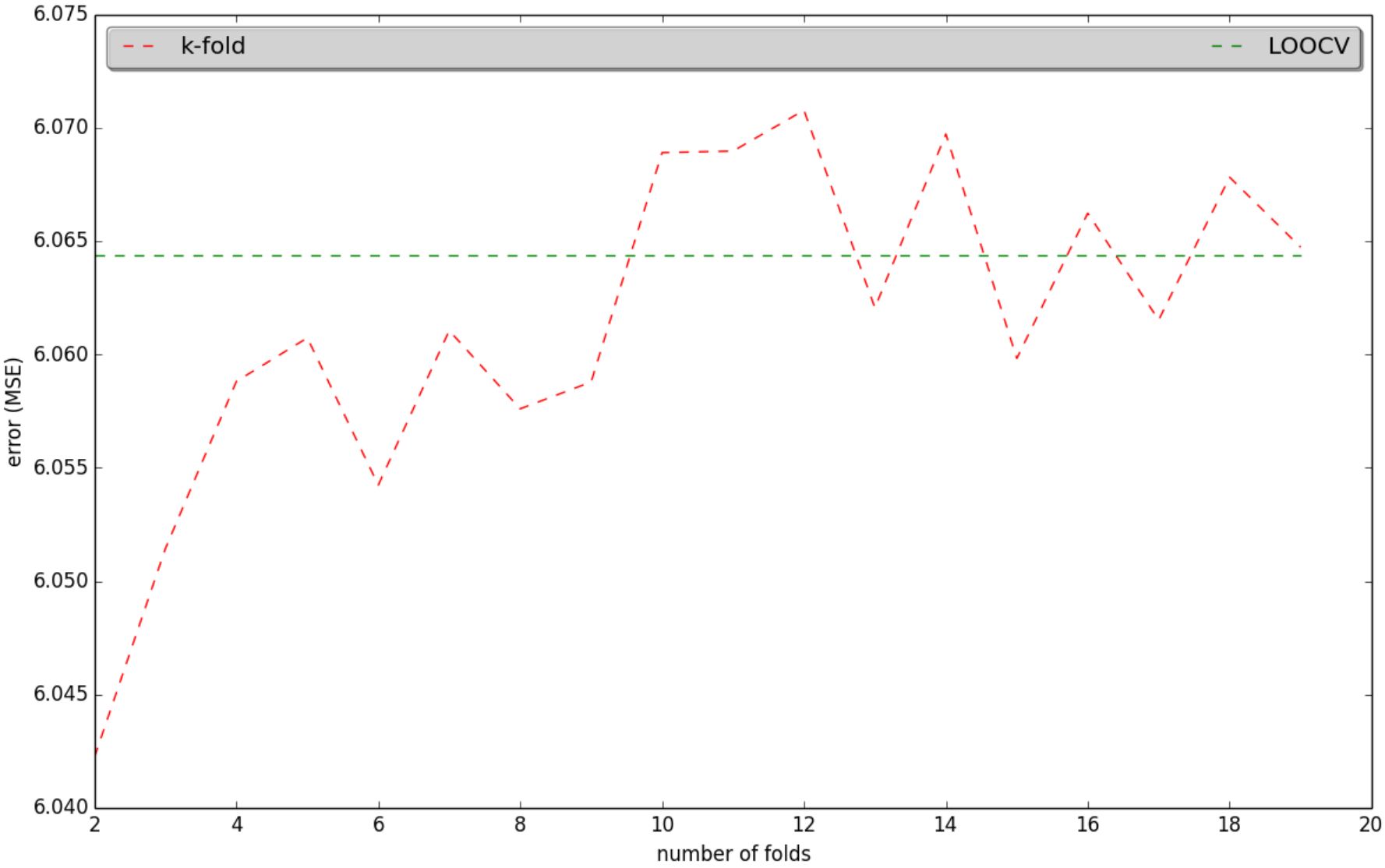
- A compromise: more "practical" version of LOOCV
- (Randomly) divide the data into k groups ("folds")



- For $i = 1, \ldots, k$:
  - keep ith fold as hold-out
  - train based on remaining $k - 1$ folds
  - evaluate (mean square) error on the hold-out fold: E_i
  - summary score: $$E = \frac{1}{k} \sum_{i=1}^{k} E_i$$

- For $k = n$: k-fold CV = LOOCV

k-fold Cross-Validation

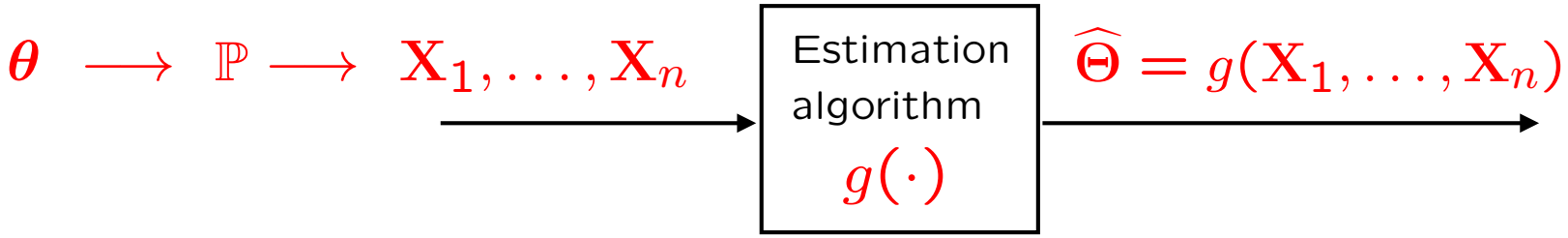


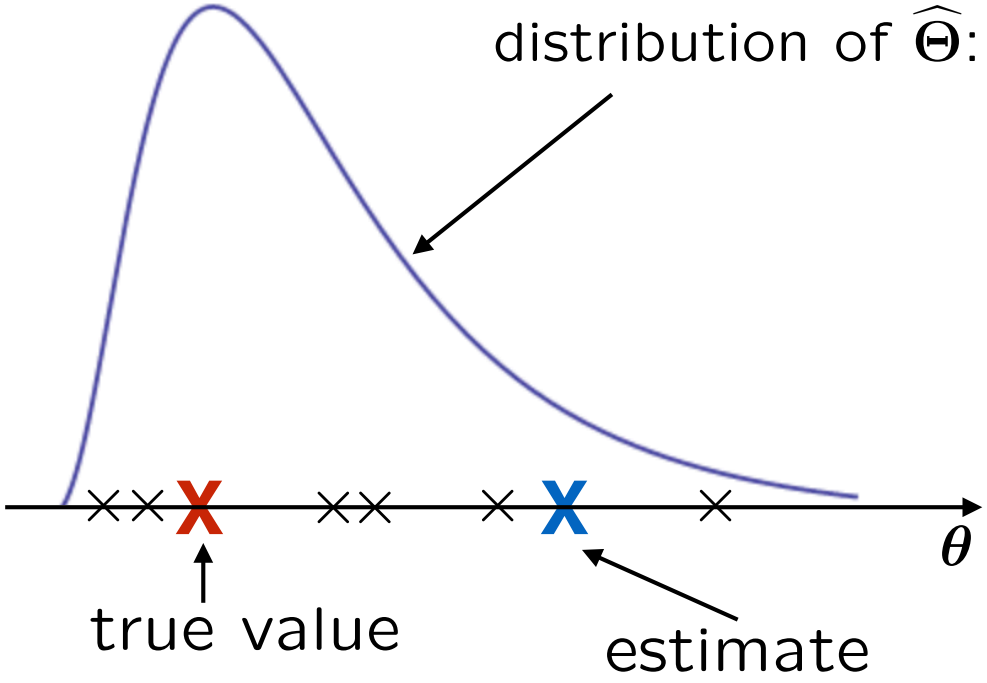
- In practice: $k = 5$ or 10

ASSESSING PARAMETER ESTIMATES:

BOOTSTRAP

Assessing parameter estimates



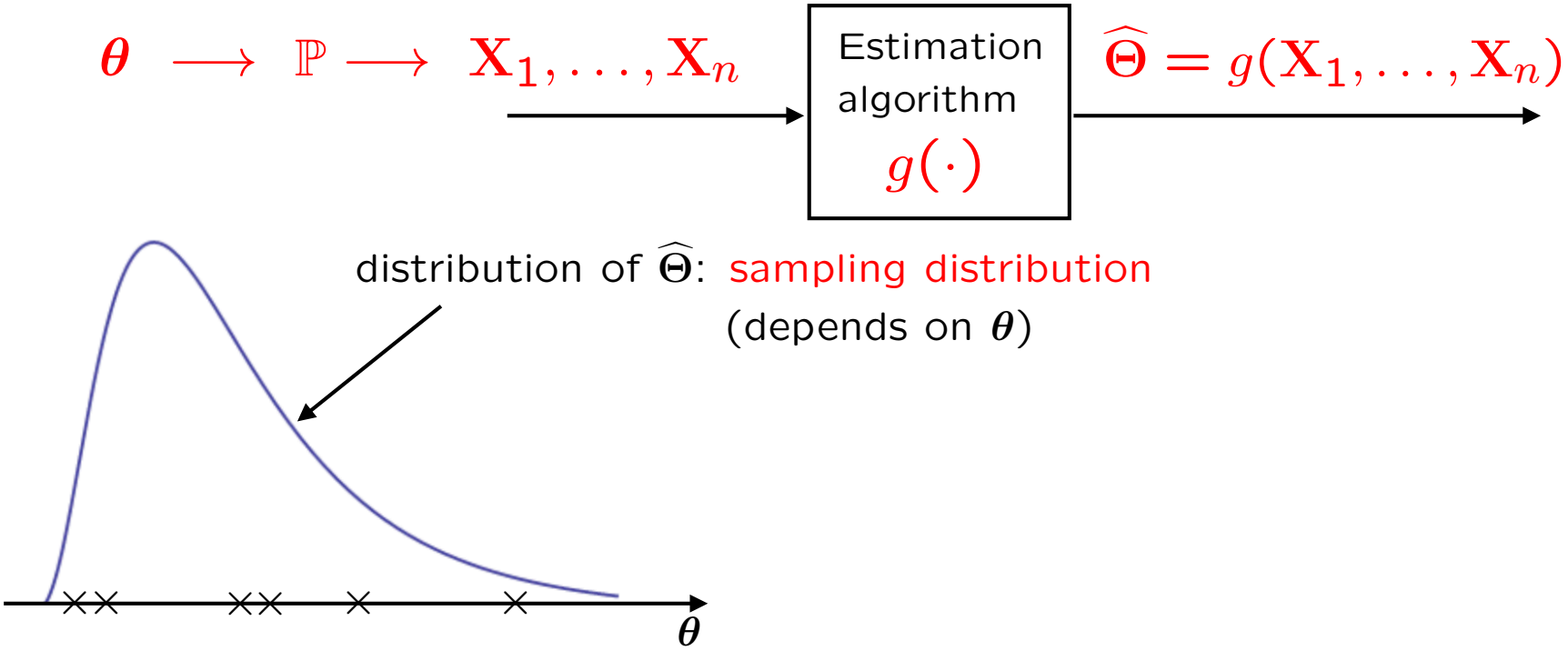


bias?
standard error?
shape of the tails?
confidence intervals?

in a data-driven manner

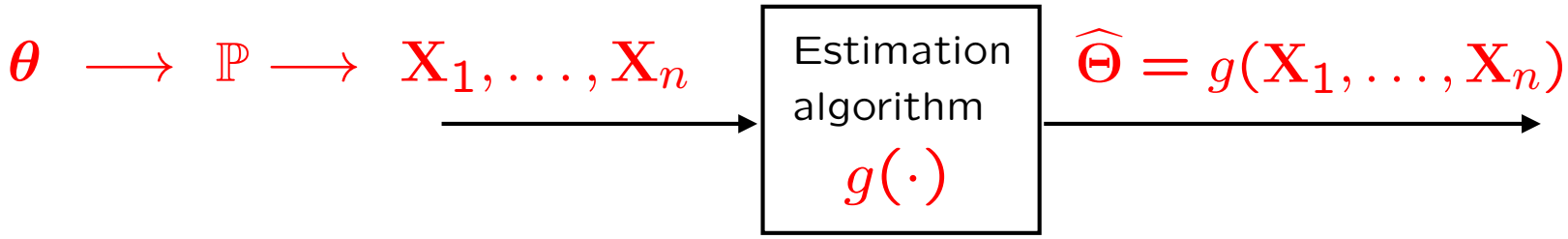
- It would have been easy if we had 100 data sets...
- Can we get 100 data sets by relying on the single data set available?

Approximating the sampling distribution



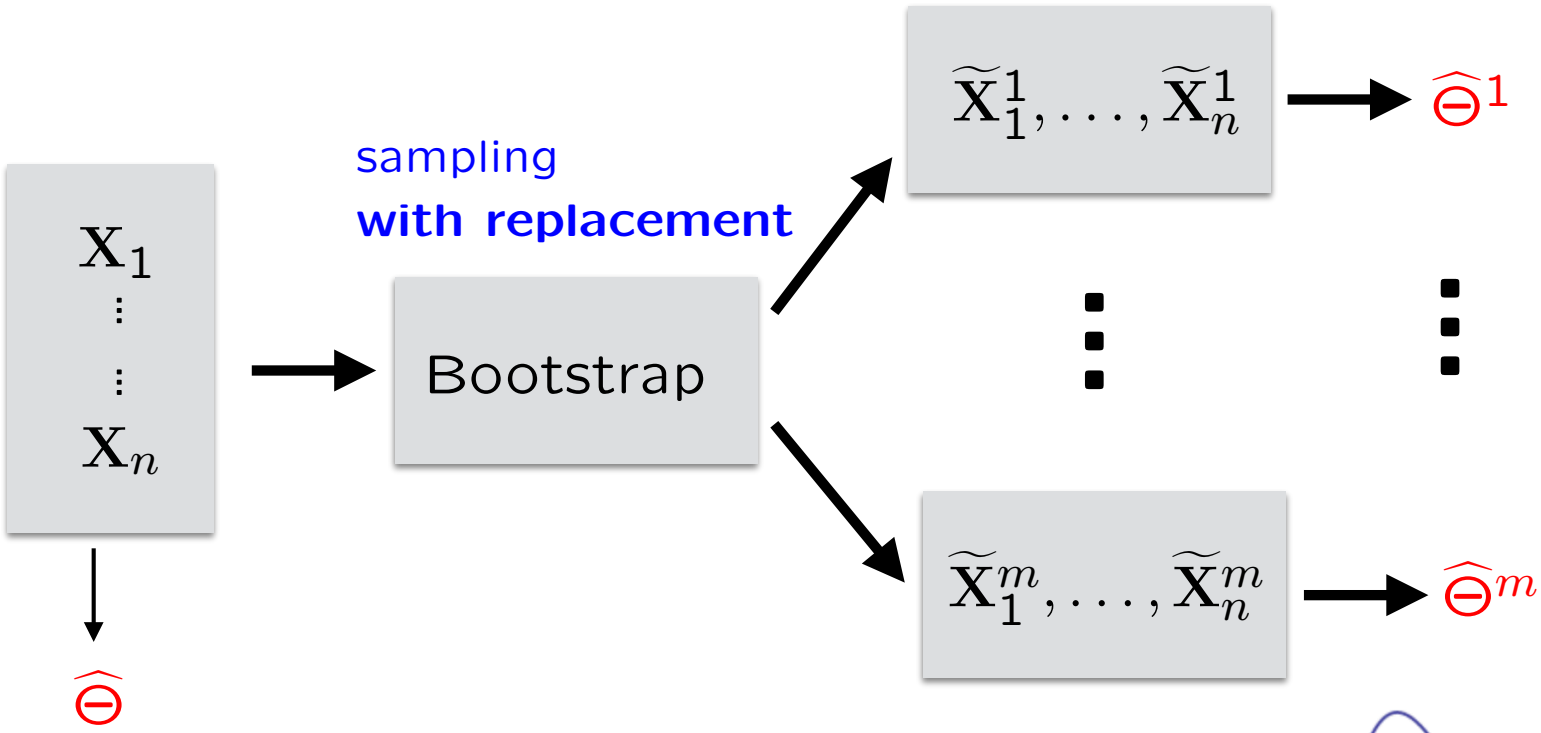
- Option 1: simulation
  - simulate new data sets, drawn from an approximation of $\mathbb{P}$
  - applicable when working with parametric models
  - true $\mathbb{P}^{\boldsymbol{\theta}}$ $\longrightarrow$ generate simulated data using $\mathbb{P}^{\hat{\boldsymbol{\theta}}}$

Data-driven bootstrap: the idea



- Know little or nothing about the structure of $\mathbb{P}$
- Want new samples from $\mathbb{P}$; but cannot simulate $\mathbb{P}$
- **Big idea:** the data are "representative" of $\mathbb{P}$
- Option 2: Resampling
  - generate new samples by sampling from the original data!

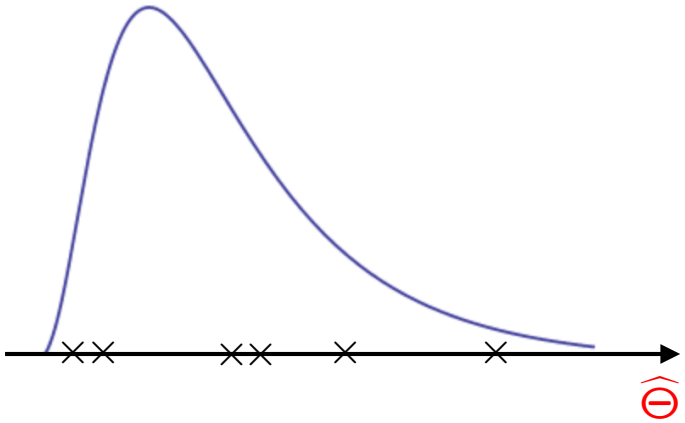
Data-driven bootstrap — in pictures



$$\widehat{\Theta}_{\text{ave}} = \frac{1}{m} \sum_{i=1}^{m} \widehat{\Theta}^i$$

$$\widehat{\text{Var}}(\widehat{\Theta}) = \frac{1}{m} \sum_{i=1}^{m} \left(\widehat{\Theta}^i - \widehat{\Theta}_{\text{ave}} \right)^2$$

$$\widehat{\text{se}}(\widehat{\Theta}) = \sqrt{\widehat{\text{Var}}(\widehat{\Theta})}$$



confidence intervals, etc.

Some key points on performance assessment

- Validation methods to:
  - assess prediction performance
  - compare models, methods, regularization parameters...

- Bootstrap to:
  - assess parameter estimates
  - calculate standard errors, etc.

- General idea: reuse the data set in clever ways