



Universität Stuttgart

KI – Institute for Artificial Intelligence

Analytic Computing

Machine Learning

4 Linear Regression

Prof. Dr. Steffen Staab

Nadeen Fatallah

Daniel Frank

Akram Sadat Hosseini

Rodrigo Lopez

Osama Mohamed

Yi Wang

Tim Schneider



<https://www.ki.uni-stuttgart.de/>

Current state of exam planning

Exam date

- Thursday August 14, 12.00hrs

Learning objectives

- Delineate interpolation and regression
- Know and be able to work with multiple regression
 - scalar notation
 - vector notation
 - Matrix notation
- Reminder: multi-dimensional derivatives
- Non-linear inputs
- Quality of linear models
- Regularization of linear models

Interpolation

Linear interpolation

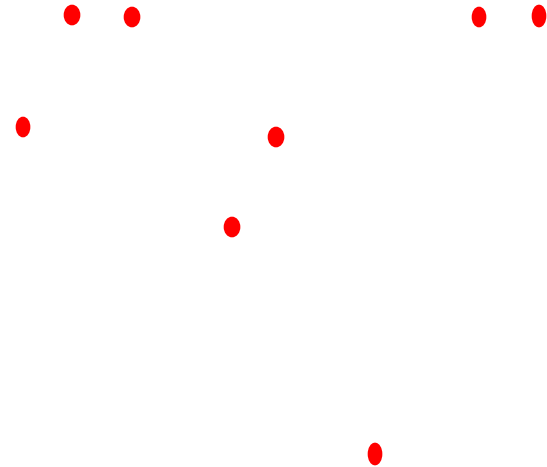
- Given: $\{(x_1, y_1), (x_2, y_2)\}$
- Linear interpolation:
 - Polynomial of degree 1



Example interpolation with Polynom of degree 7

- Given: $\{(x_1, y_1), \dots, (x_n, y_n)\}$
- There is a polynom of degree $n - 1$ that hits all data points

$$\hat{f}(x) = \sum_{i=1}^n y_i \prod_{k=1, k \neq i}^n \frac{x - x_k}{x_i - x_k}$$

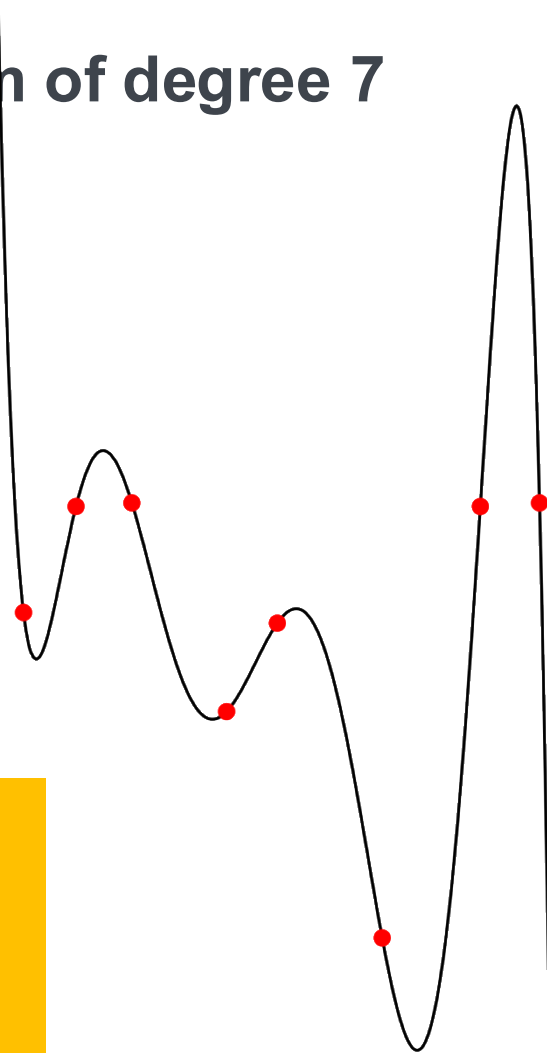


Example interpolation with Polynom of degree 7

- Given: $\{(x_1, y_1), \dots, (x_n, y_n)\}$
- There is a polynom of degree $n - 1$ that hits all data points

$$\hat{f}(x) = \sum_{i=1}^n y_i \prod_{k=1, k \neq i}^n \frac{x - x_k}{x_i - x_k}$$

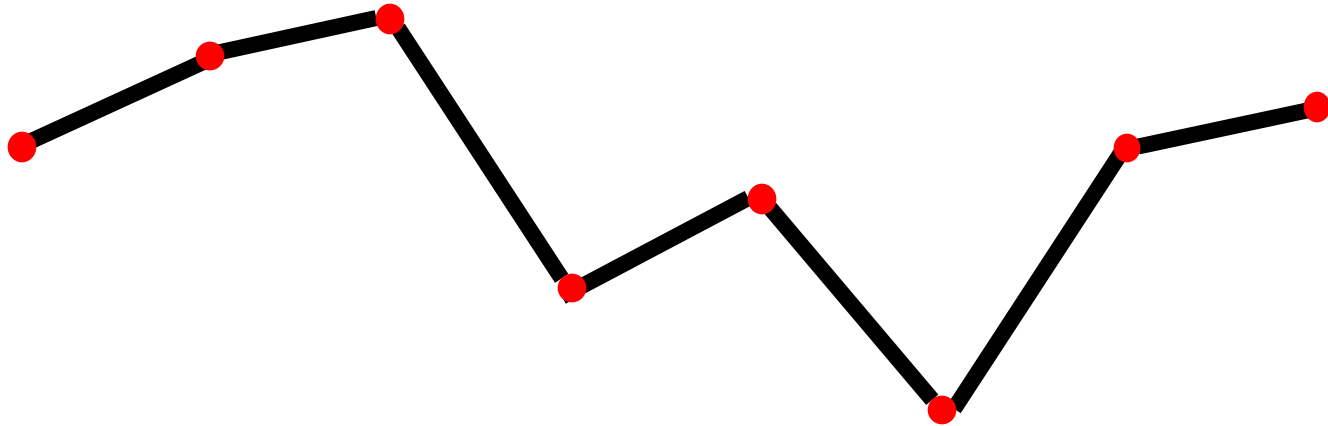
- Severe issues:
 - Oscillation, Instability
 - Most often implausible for generalizing to unobserved data



Piecewise linear interpolation

- Given: $\{(x_1, y_1), \dots, (x_n, y_n)\}$
- Interpolating curve:
 - Let $\hat{f}_i(x)$ be defined in $[x_i, x_{i+1}]$ for $i \in \{1, \dots, n-1\}$
 - $\hat{f}(x) = \hat{f}_i(x)$, if there is an i such that $\hat{f}_i(x)$ is defined
 - $\hat{f}_i(x_i) = y_i, \hat{f}_i(x_{i+1}) = y_{i+1}$

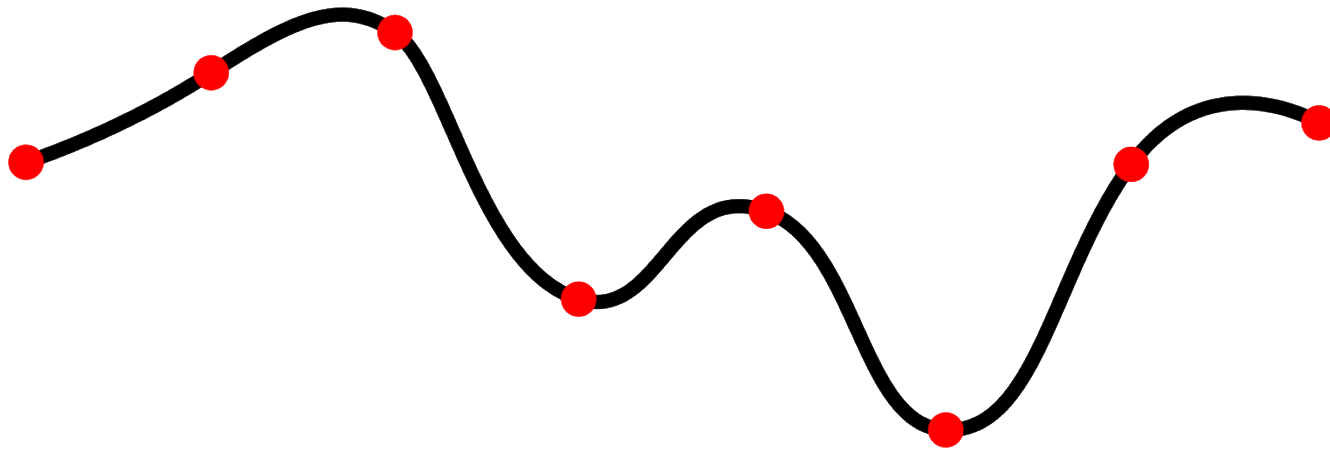
- Issues:
 - Not smooth
 - Sensitive to outliers



Example for Spline Interpolation

- Given: $\{(x_1, y_1), \dots, (x_n, y_n)\}$
- Interpolating curve:
 - Let $\hat{f}_i(x)$ be defined in $[x_i, x_{i+1}]$ for $i \in \{1, \dots, n-1\}$
 - $\hat{f}(x) = \hat{f}_i(x)$, if there is an i such that $\hat{f}_i(x)$ is defined
 - $\hat{f}_i(x_i) = y_i, \hat{f}_i(x_{i+1}) = y_{i+1}$
 - $\forall i \in \{2, \dots, n-1\} : \lim_{\epsilon \rightarrow 0} \hat{f}'(x_i - \epsilon) = \lim_{\epsilon \rightarrow 0} \hat{f}'(x_i + \epsilon)$

- Issues:
 - Depending on degree of spline, not smooth wrt higher derivation
 - For a lot of data points ($10^3 \dots 10^{10}$):
overly complex model
 - overfitting to outliers



Further interpolation methods

- Other base functions
 - trigonometric interpolation (Fourier)
 - logarithmic interpolation
 - ...
 - Gaussian processes

Regression

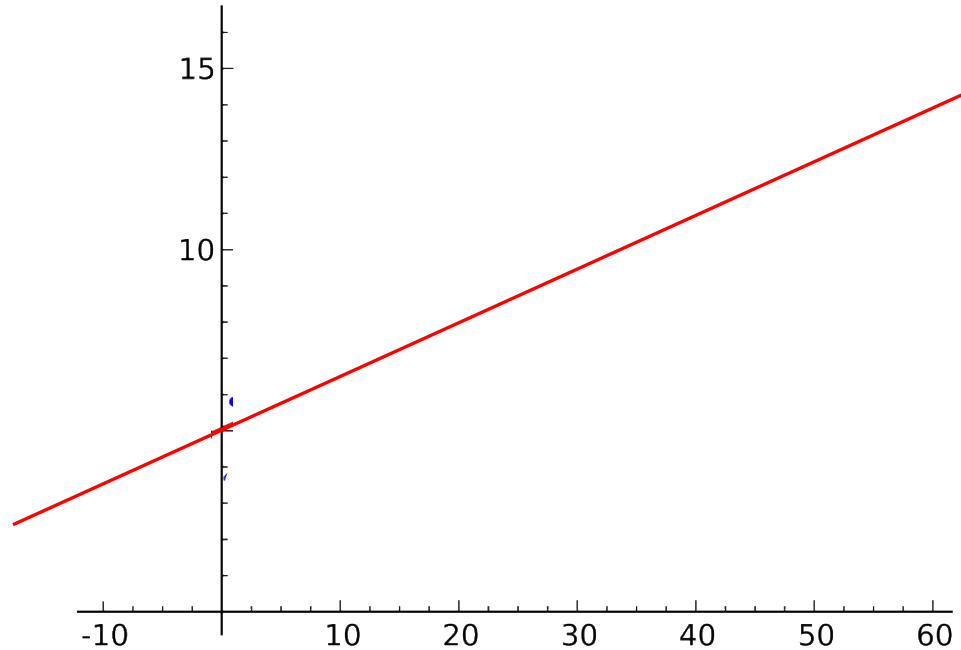
Regression instead of interpolation

- Handle noise
- Acquire simpler model
 - simple in terms of making predictions
 - not „simple“ in terms of finding it

Linear regression

Assumptions

- Linear function generates data
$$f(x) = b_1x + b_0$$



Linear regression

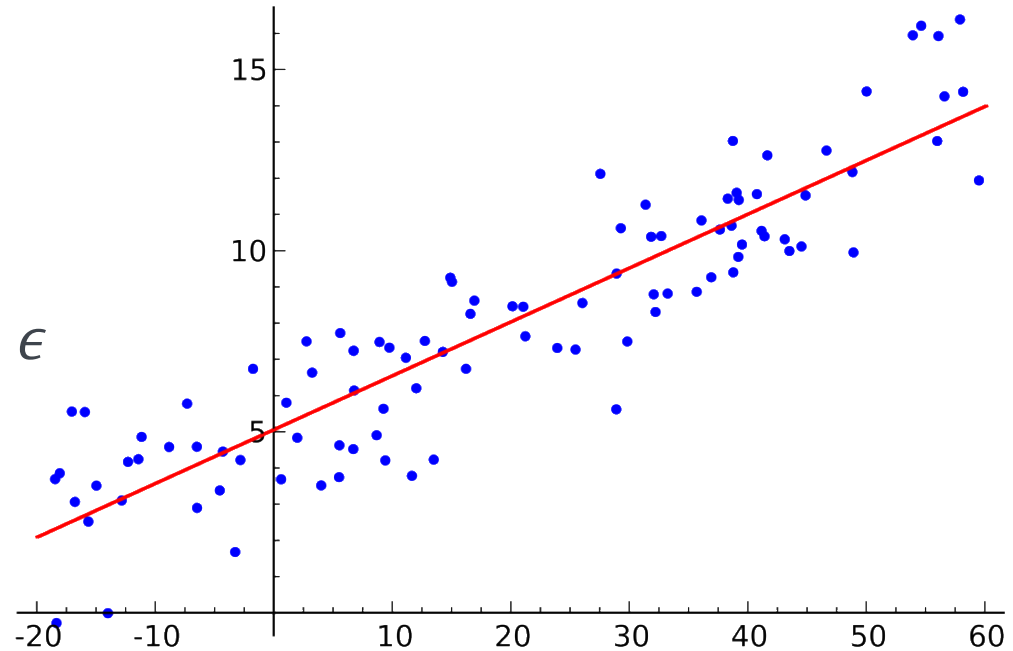
Assumptions

- Linear function generates data

$$f(x) = b_1x + b_0$$

- Additionally there is noise ϵ

$$f(x) = b_1x + b_0 + \epsilon$$



Linear regression variants

- Simple linear regression
 - for each observation i :
 - one predictor variable x_i , one response variable y_i
- Multiple linear regression
 - for each observation i :
 - several predictor variables $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,d})$, one response variable y_i
- Multivariate linear regression
 - for each observation i :
 - several predictor variables $\mathbf{x} = (x_{i,1}, \dots, x_{i,d})$,
several response variables $\mathbf{y} = (y_{i,1}, \dots, y_{i,k})$

Multiple linear regression model for multiple observations

Given is one **response variable** (y_i) per observation i

Up to some random errors (ϵ_i)

it is a linear function of several **predictor variables** ($x_{i,j}$).

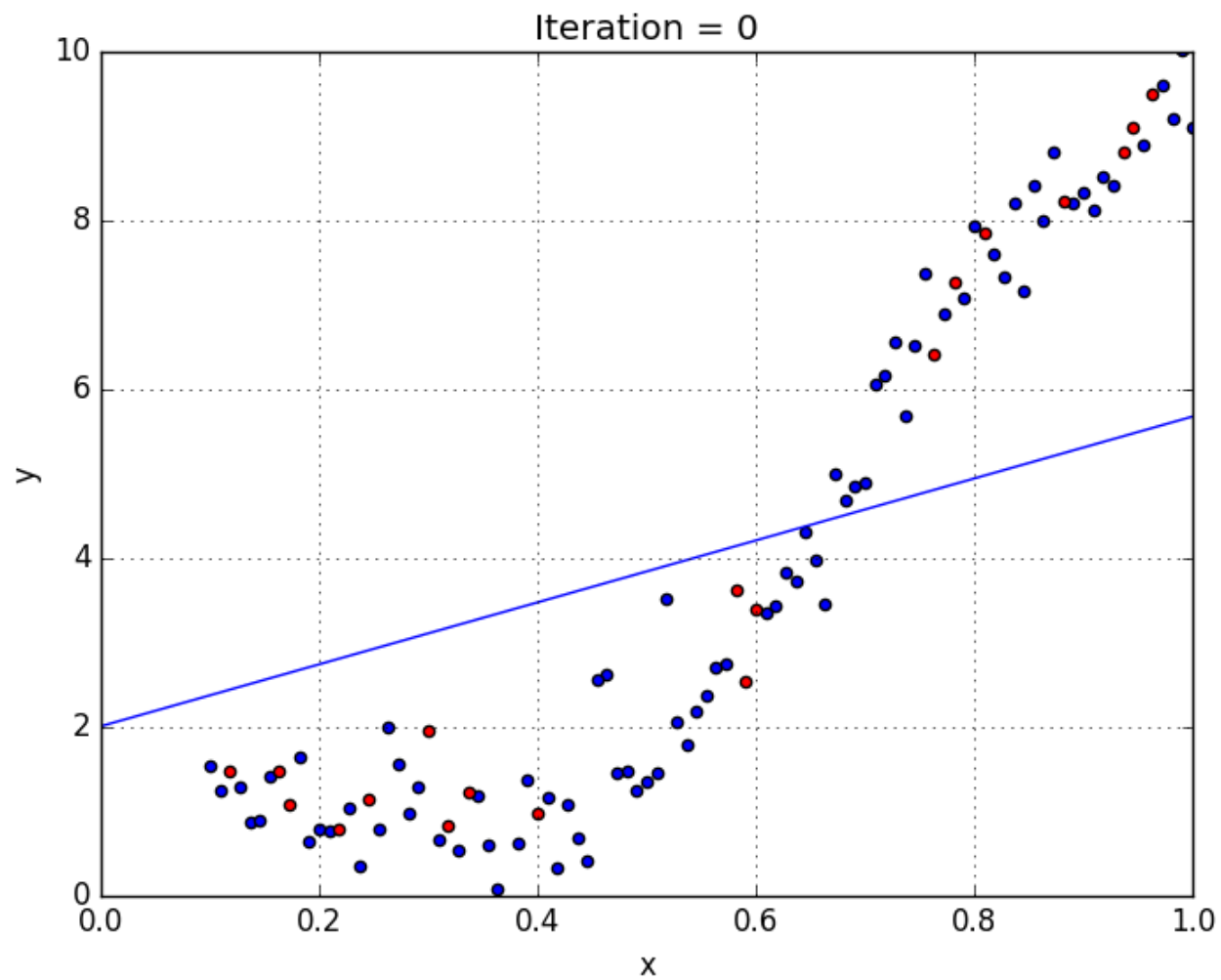
The linear function involves unknown parameters (β_1, \dots, β_d).

The goal is

1. to estimate these parameters,
2. to study their relevance, and
3. to estimate the error variance

Errors are also called **residuals**,

Predictor variables are also called
independent variables
or **input variables**



Multiple linear regression model


Given n observations in data set $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\} \subseteq \mathbb{R}^{d+1}$


Scalar representation:


$$y_i = \beta_0 + \beta_1 x_{i,1} + \dots + \beta_d x_{i,d} + \varepsilon_i, \quad (i = 1, \dots, n)$$

Vector representation


$$y_i = \bar{x}_i^T \beta + \varepsilon_i$$



$$\bar{x}_i = (1, x_{i,1}, \dots, x_{i,d})^T \in \mathbb{R}^{d+1}, \quad (i = 1, \dots, n)$$






$$\beta = (\beta_0, \beta_1, \dots, \beta_d)^T \in \mathbb{R}^{d+1}$$


$$\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)^T \in \mathbb{R}^n$$

Multiple regression model – matrix representation


$$y = X\beta + \varepsilon$$


$$X = \begin{pmatrix} \bar{x}_1^T \\ \vdots \\ \bar{x}_n^T \end{pmatrix}$$


$$\bar{x}_i = (1, x_{i,1}, \dots, x_{i,d})^T \in \mathbb{R}^{d+1}, \quad (i = 1, \dots, n)$$

$$y = (y_1, \dots, y_n)^T \in \mathbb{R}^n$$

$$\beta = (\beta_0, \beta_1, \dots, \beta_d)^T \in \mathbb{R}^{d+1}$$

$$\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)^T \in \mathbb{R}^n$$

Multiple regression model: optimization objective

Ansatz:

$$y = X\beta + \epsilon$$

Determine \hat{f} by finding minimal ϵ such that

$$\hat{f}(\bar{x}) = \bar{x}^T \beta$$

Typically, we just write

$$f(x) = x^T \beta$$

is an „optimal“ approximation of observed data points X .

Minimize $\epsilon^T \epsilon$

Multi-dimensional derivatives

Optimizing loss function L^{ls} (sum of squares)

$$\rightarrow L^{ls}(\beta) =$$

$$\rightarrow = ||\varepsilon(\beta)||^2 =$$

$$\rightarrow = \sum_{i=1}^n (y_i - \bar{x}_i^T \beta)^2 =$$

$$\rightarrow = (y - X\beta)^T (y - X\beta) =$$

$$\rightarrow = ||y - X\beta||^2$$

Compute first derivation:

$$\rightarrow \frac{\partial L^{ls}(\beta)}{\partial \beta} = -2(y - X\beta)^T X$$

Aside 1: Derivation rules

➡ Exponentiation $\frac{\partial x^n}{\partial x} = n \cdot x^{n-1}, \quad n \neq 0$

➡ Sum $\frac{\partial(g(x) + h(x))}{\partial x} = \frac{\partial g(x)}{\partial x} + \frac{\partial h(x)}{\partial x}$

➡ Product rule $\frac{\partial(g(x) \cdot h(x))}{\partial x} = \frac{\partial g(x)}{\partial x} \cdot h(x) + g(x) \cdot \frac{\partial h(x)}{\partial x}$

➡ Chain rule $\frac{\partial g(h(x))}{\partial x} = \frac{\partial g(h(x))}{\partial(h(x))} \cdot \frac{\partial h(x)}{\partial x}$
One dimension: $(g(h(x)))' = \dot{g}(h(x)) \cdot \dot{h}(x)$

Aside 2: Partial derivatives

Follows the rules just given

Furthermore

$$\frac{\partial(\partial f(x,y))}{\partial y \partial x} = \frac{\partial(\partial f(x,y))}{\partial x \partial y}$$

Aside 2: Partial derivatives

Follows the rules just given

Furthermore

$$\frac{\partial(\partial f(x,y))}{\partial y \partial x} = \frac{\partial(\partial f(x,y))}{\partial x \partial y}$$

For example

$$\frac{\partial(2xy+3yz+5zx)}{\partial x} = 2y + 5z$$

$$\frac{\partial(\partial(2xy+3yz+5zx))}{\partial y \partial x} = \frac{\partial(2y+5z)}{\partial y} = 2$$

Determine minimum

$$\rightarrow \frac{\partial L^{\text{ls}}(\beta)}{\partial \beta} = -2(y - X\beta)^T X = \mathbf{0}_d^T$$

$$\begin{matrix} \Leftrightarrow \\ \rightarrow \end{matrix} \mathbf{0}_d^T = X^T X \beta - X^T y$$

$$\begin{matrix} \Leftrightarrow \\ \rightarrow \end{matrix} \hat{\beta}^{\text{ls}} = (X^T X)^{-1} X^T y$$

- This method for finding the minimum is not used directly, because computing the inverse is both inefficient and computationally instable
- Rather, use stochastic gradient descent to find where $X^T X \beta - X^T y$ is zero in all its components

Beyond linear input

„Linear regression“ – linear in what?

- Linear in the input: no
- Linear in the effect of parameters: yes!
- Linear regression on non-linear features is very powerful:
 - polynomials
 - piece-wise
 - spline-basis
 - kernels

Features are object
descriptions derived
from the original
input data

Non-linear features

- Replace the inputs
by some **non-linear features**

$$x_i \in \mathbb{R}^d$$

$$\phi(x_i) \in \mathbb{R}^k$$

Non-linear features

- Replace the inputs
by some **non-linear features**

$$x_i \in \mathbb{R}^d$$

$$\phi(x_i) \in \mathbb{R}^k$$

such that

$$y = \sum_{j=1}^k \phi_j(x) \beta_j + \varepsilon = \phi(x)^T \beta + \varepsilon$$

Non-linear features

- Replace the inputs
by some **non-linear features**

$$x_i \in \mathbb{R}^d$$

$$\phi(x_i) \in \mathbb{R}^k$$

such that

$$y = \sum_{j=1}^k \phi_j(x) \beta_j + \varepsilon = \phi(x)^T \beta + \varepsilon$$

$$X = \begin{pmatrix} \phi(x_1)^T \\ \vdots \\ \phi(x_n)^T \end{pmatrix}$$

- Features are formed through an arbitrary set of basis functions
- Any function *linear in β* can be written as $f(x) = \phi(x)^T \beta$ for some ϕ

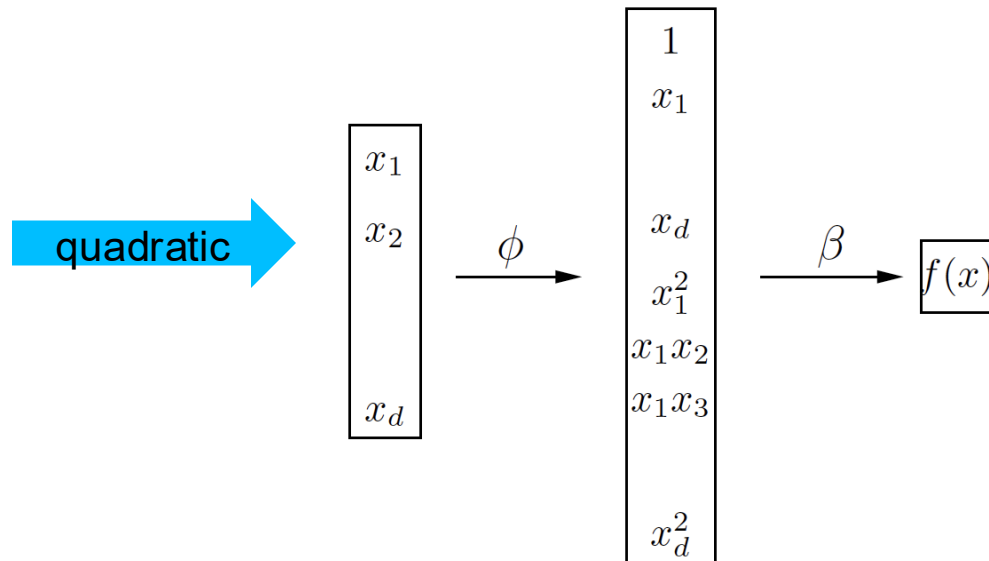
Example: Polynomial features

➡ Linear: $\phi(x) = (1, x_1, \dots, x_d) \in \mathbb{R}^{1+d}$

➡ Quadratic: $\phi(x) = (1, x_1, \dots, x_d, x_1^2, x_1x_2, x_1x_3, \dots, x_d^2) \in \mathbb{R}^{1+d+\frac{d(d+1)}{2}}$

➡ Cubic: $\phi(x) = (\dots, x_1^3, x_1^2x_2, x_1^2x_3, \dots, x_d^3) \in \mathbb{R}^{1+d+\frac{d(d+1)}{2}+\frac{d(d+1)(d+2)}{6}}$

$$x \qquad \phi(x) \qquad f(x) = \phi(x)^\top \beta$$



Example: Piece-wise features (in 1D)

➡ Piece-wise constant: $\phi_j(x) = [\xi_j < x \leq \xi_{j+1}]$

➡ Piece-wise linear: $\phi_j(x) = (1, x)^\top [\xi_j < x \leq \xi_{j+1}]$

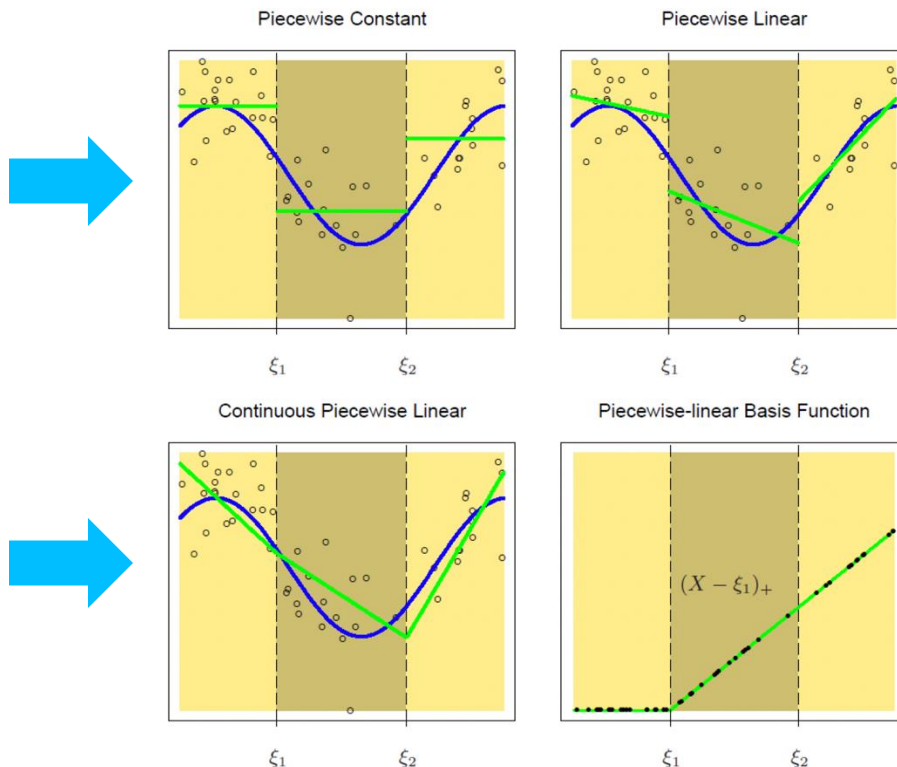
➡ Continuous piece-wise linear: $\phi_j(x) = [x - \xi_j]_+$ (and $\phi_0(x) = x$)

where „ $[condition]$ “ evaluates to

- 1 if *condition* is true and
- 0 otherwise

where „ $[term]_+$ “ evaluates to

- *term*, if *term* > 0 and
- 0, otherwise



(slide by Marc Toussaint 2019)

Example: Radial Basis Functions (RBF)

Given a set of centers $\{c_j\}_{j=1}^k$, define

Gaussian RBF $\phi_j(x) = b(x, c_j) = e^{-\frac{1}{2}\|x-c_j\|^2} \in [0, 1]$

Each $\phi_j(x)$ measures similarity with the center c_j

Any function $\varphi(x)$
which can be
rewritten as $\tilde{\varphi}(\|x-c\|)$
is a radial function

Example: Radial Basis Functions (RBF)

Given a set of centers $\{c_j\}_{j=1}^k$, define

Gaussian RBF $\phi_j(x) = b(x, c_j) = e^{-\frac{1}{2}\|x-c_j\|^2} \in [0, 1]$

Each $\phi_j(x)$ measures similarity with the center c_j

Special case:

use all training inputs $\{x_i\}_{i=1}^n$ as centers

$$\phi(x) = \begin{pmatrix} 1 \\ b(x, x_1) \\ \vdots \\ b(x, x_n) \end{pmatrix} \quad (n + 1 \text{ dim})$$

Any function $\varphi(x)$
which can be
rewritten as $\tilde{\varphi}(\|x-c\|)$
is a radial function

Question : Modeling an example for multiple regression

People studied payment of managers of corporations, for example:

- Herbert A. Simon, The compensation of Executive, Sociometry, March 1957.
- Elizabeth Keating and Peter Frumkin. What drives Nonprofit Executive Compensation? In: The Nonprofit Quarterly, Nov 30, 2018. (<https://nonprofitquarterly.org/what-drives-nonprofit-executive-compensation-2/>)
- Two factors that were considered (among others) were
 - size of organization (e.g. equivalents of full-time employees) and
 - amount of free cash flow (e.g. measured in US dollar).
- Assume you have corresponding data for one hundred managers and want to do an analysis by multiple regression.
 - Question A: How do you define $\Phi(x)$?
 - Question B: Why do you define it this way?
 - Question C: What can you read from β_1 and β_2 , correspondingly?

Answer for Question 4:

Modeling an example for multiple regression

a) Suggested: $\phi(x_i)^T = (s_i \quad \log(s_i) \quad c_i \quad \log(c_i))^T \in \mathbb{R}^4$

b) Herbert Simon (1957) found that payment of managers is mostly determined by size of organizations (s_i). However, a top manager of a company with 100,000 employees will usually not earn linearly more than a manager of a company with 10 employees. Sublinear growth might be modeled with a logarithmic component.

Same rationale might apply to free cash flow (c_i).

Maybe there is still a linear component for both.

c) β_1 and β_2 may indicate which component is how important, relatively speaking.



EXPERT OPINION

Contact Editor: Brian Brannon, bbrannon@computer.org

The Unreasonable Effectiveness of Data

Alon Halevy, Peter Norvig, and Fernando Pereira, Google

Eugene Wigner's article "The Unreasonable Effectiveness of Mathematics in the Natural Sciences"¹ examines why so much of physics can be neatly explained with simple mathematical formulas

such as $f = ma$ or $e = mc^2$. Meanwhile, sciences that involve human beings rather than elementary particles have proven more resistant to elegant mathematics. Economists suffer from physics envy over their inability to neatly model human behavior. An informal, incomplete grammar of the English language runs over 1,700 pages.² Perhaps when it comes to natural language processing and related fields, we're doomed to complex theories that will never have the elegance of physics equations. But if that's so, we should stop acting as if our goal is to author extremely elegant theories, and instead embrace complexity and make use of the best ally we have: the unreasonable effectiveness of data.

One of us, as an undergraduate at Brown University, remembers the excitement of having access to the Brown Corpus, containing one million English words.³ Since then, our field has seen several notable corpora that are about 100 times larger, and in 2006, Google released a trillion-word corpus with frequency counts for all sequences up to five words long.⁴ In some ways this corpus is a step backwards from the Brown Corpus: it's taken from unfiltered Web pages and thus contains incomplete sentences, spelling errors, grammatical errors, and all sorts of other errors. It's not annotated with carefully hand-corrected part-of-speech tags. But the fact that it's a million times larger than the Brown Corpus outweighs these

behavior. So, this corpus could serve as the basis of a complete model for certain tasks—if only we knew how to extract the model from the data.

Learning from Text at Web Scale

The biggest successes in natural-language-related machine learning have been statistical speech recognition and statistical machine translation. The reason for these successes is not that these tasks are easier than other tasks; they are in fact much harder than tasks such as document classification that extract just a few bits of information from each document. The reason is that translation is a natural task routinely done every day for a real human need (think of the operations of the European Union or of news agencies). The same is true of speech transcription (think of closed-caption broadcasts). In other words, a large training set of the input-output behavior that we seek to automate is available to us *in the wild*. In contrast, traditional natural language processing problems such as document classification, part-of-speech tagging, named-entity recognition, or parsing are not routine tasks, so they have no large corpus available in the wild. Instead, a corpus for these tasks requires skilled human annotation. Such annotation is not only slow and expensive to acquire but also difficult for experts to agree on, being bedeviled by many of the difficulties we discuss later in relation to the Semantic Web. The first lesson of Web-scale learning is to use available large-scale data rather than hoping for annotated data that isn't available. For instance, we find that useful semantic relationships can be automatically learned from the statistics of search queries and the

Machine learning and statistics before 2000

- find the function that best models the problem

Machine learning & big data

- use a simple model, but let the data itself determine the model in great detail

Their example:

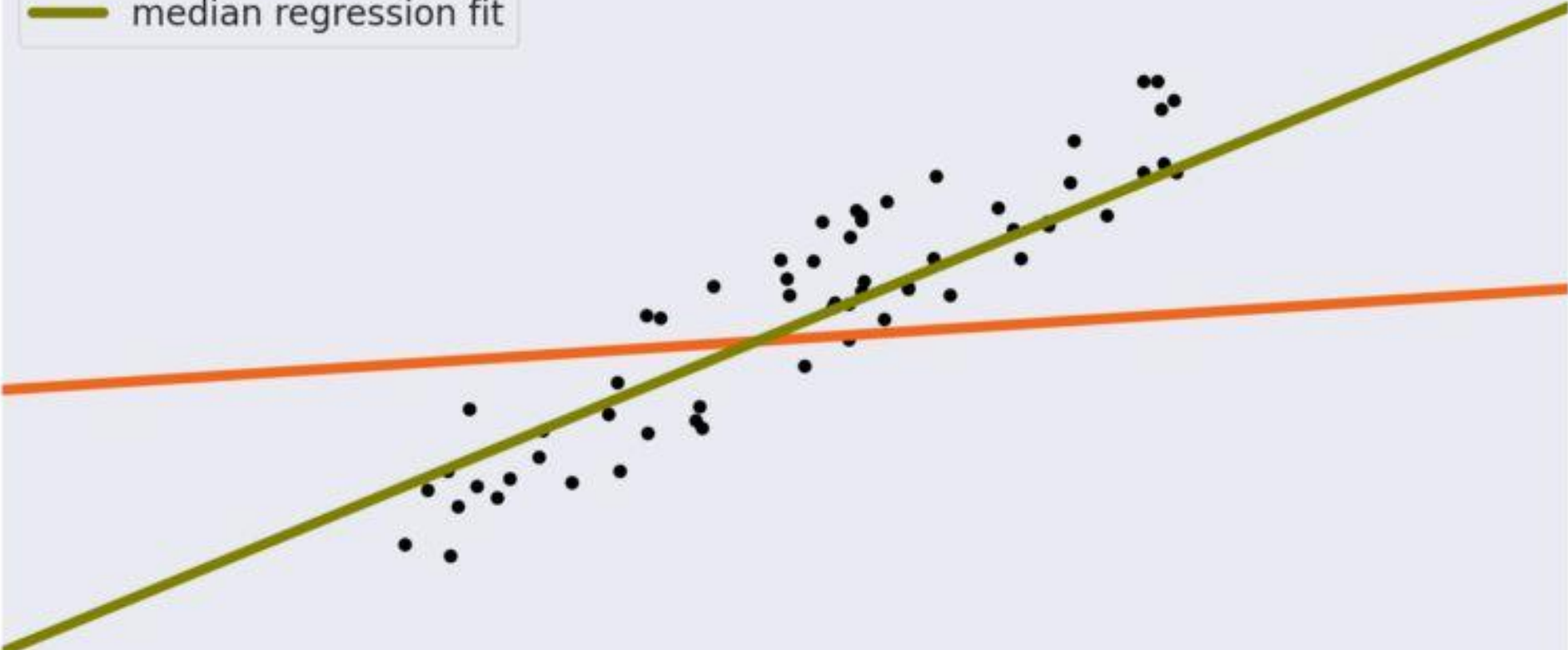
- text – most words are rare
- but there is sooo much text now
- big data is sooo effective (not always!)

Halevy, Norvig, Pereira. The unreasonable effectiveness of data. In: *IEEE Intelligent Systems*, March/Apr 2009.

Model quality of linear regression

Mean vs Median

- Mean regression fit: Loss $\ell = \epsilon^T \epsilon$
- Median regression fit: Loss $\ell = \sum_i |\epsilon_i|$



outliers:

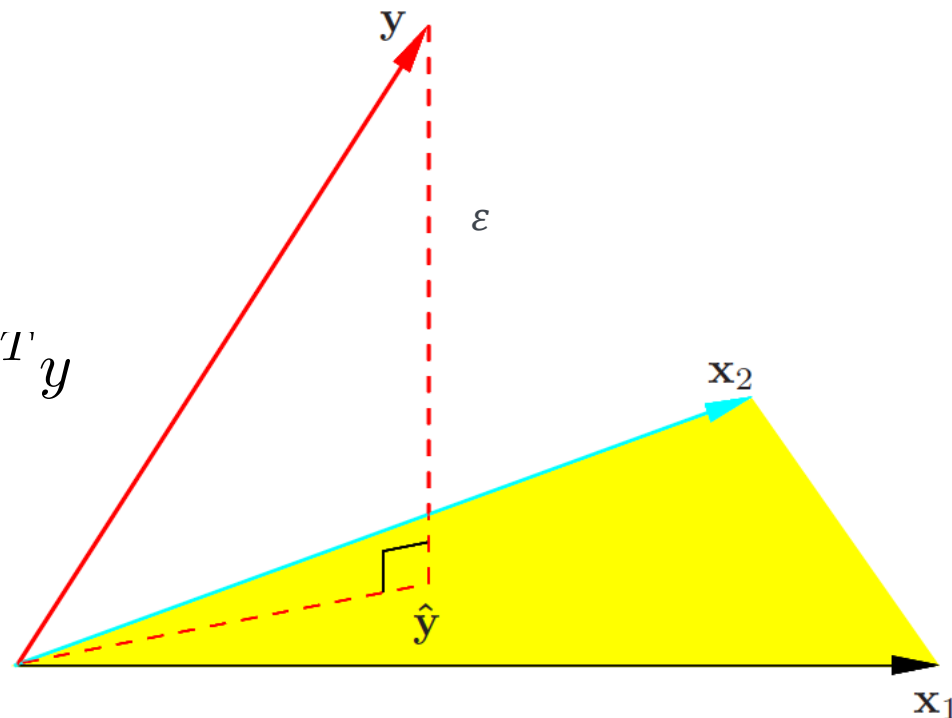
-
-
-
-

Outliers wrt y-axis vs outliers wrt x-axis

https://scikit-learn.org/stable/auto_examples/linear_model/plot_theilsen.html

The Hat matrix

$$\hat{y} = X\beta = X(X^T X)^{-1} X^T y$$



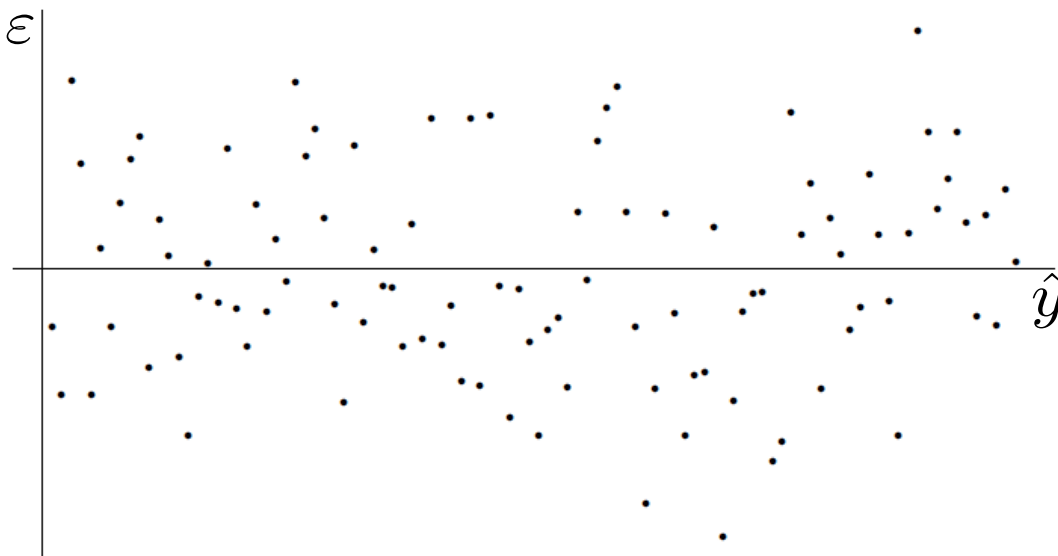
Residuals ε_i

$$\varepsilon_i = y_i - \hat{y}_i = y_i - x_i\hat{\beta}$$

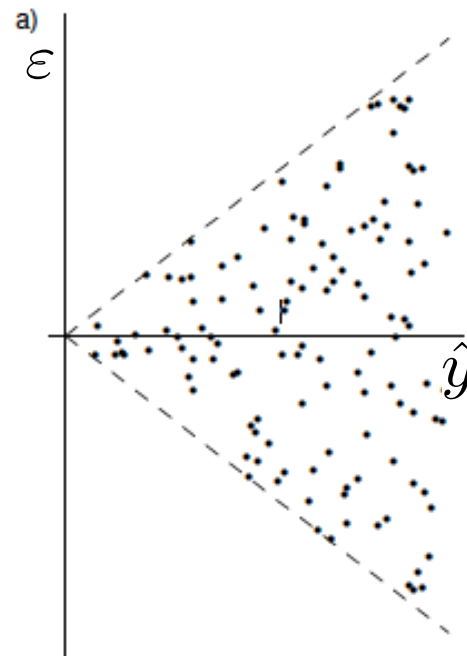
Fitted values

Observed values

Residuals should fluctuate randomly against \hat{y}

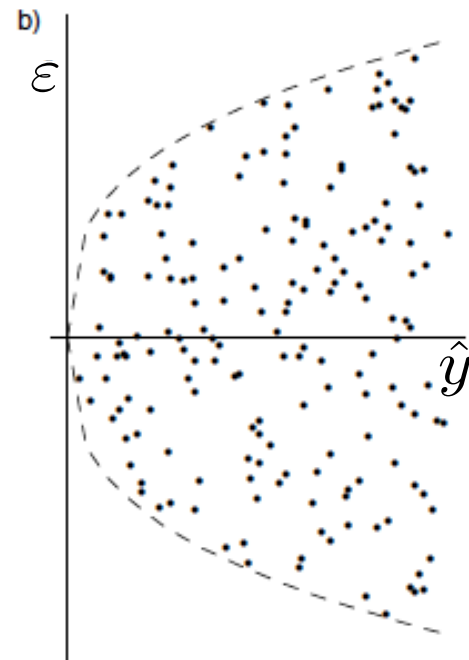


Tukey-Anscombe Plot: Example 1



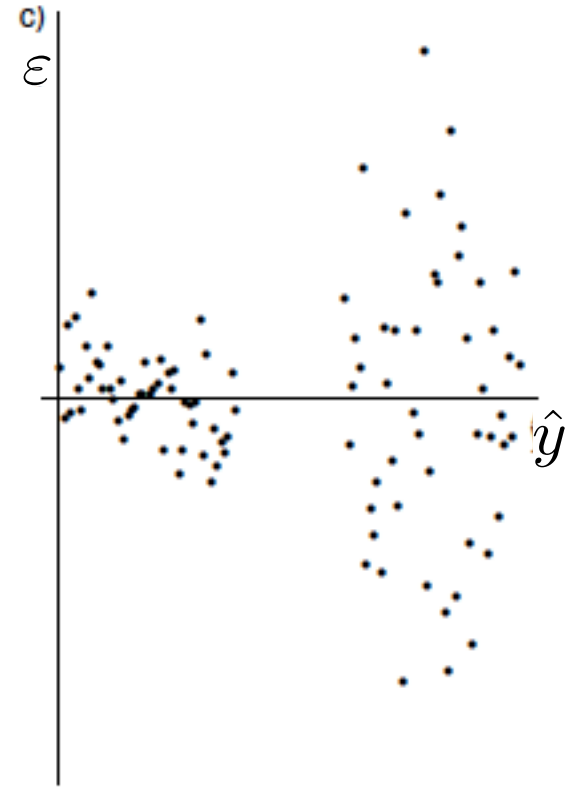
Residuals grow linearly with \hat{y}

Tukey-Anscombe Plot: Example 2



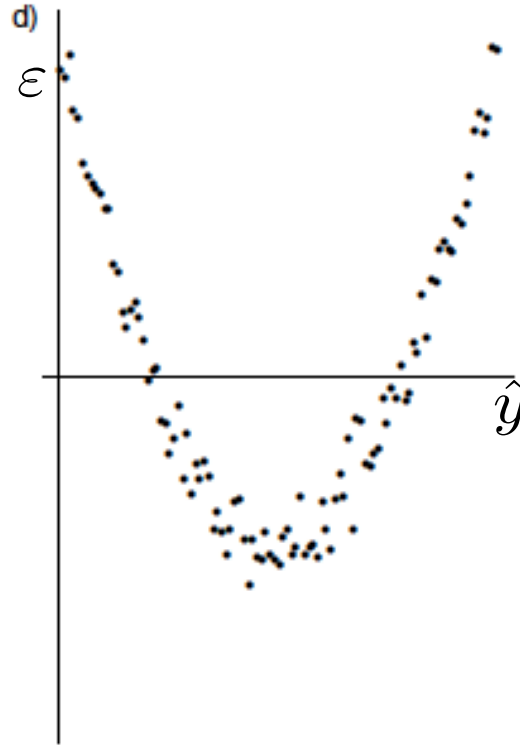
Residuals grow non-linearly with \hat{y}

Tukey-Anscombe Plot: Example 3



Two groups with different variances

Tukey-Anscombe Plot: Example 4



Missing quadratic term in formula

Investigating model assumptions

Assumptions for the Linear Model

1. Residuals $\{\epsilon_1 \dots \epsilon_n\}$ are jointly normally distributed with $E(\epsilon_i) = 0, \text{Var}(\epsilon_i) = \sigma^2$ („linear regression equation is correct“)
2. The variance of the errors σ^2 is constant for all ϵ_i („homoscedasticity“)
3. All x_i can be observed perfectly („not noisy“).
4. The errors are uncorrelated: $\text{Cov}(\epsilon_i, \epsilon_j) = 0$ for all $i \neq j$

Implication:

- Because $y = x\beta + \epsilon$ and x and β of the underlying function are exact, all y_i are distributed with $\text{var}(y_i) = \sigma^2$

If residuals $\{\epsilon_1 \dots \epsilon_n\}$ do not have the mean $E(\epsilon_i) = 0$,
we need another model.

For violations of other assumptions generalizations of the described method exist.

Goals of linear regression analysis

1. **Good fit:** Fitting a (hyper-)plane over predictor variables to explain response variables such that errors are small.
2. **Good parameter estimates:**
describe change of response when varying some predictor variable(s).
3. **Good prediction:** predict a new response as a function of new predictor variables.
4. **Uncertainties and significance**
 - Confidence intervals & statistical tests

Not the same!!!!



Development of a good model: Using methods for 1-4, we may change parts of an initial model to come up with a better model

Goals of linear regression analysis

1. **Good fit:** Fitting a (hyper-)plane over predictor variables to explain response variables such that errors are small.
2. **Good parameter estimates:**
describe change of response when varying some predictor variable(s).
3. **Good prediction:** predict a new response as a function of new predictor variables.
4. **Uncertainties and significance**
 - Confidence intervals & statistical tests

Estimator variance

- Because $\epsilon \sim N(0, \sigma^2)$, we conclude $Var(y) = \sigma^2 \mathbf{I}_n$ $y \sim N(X\beta, \sigma^2 I)$

A useful multivariate theorem:

Suppose u is a vector from a multivariate normal distribution $u \sim N(\mu, \Sigma)$

and c is a vector and D is a matrix and v is defined by $v = c + Du$

Then $v \sim N(c + D\mu, D\Sigma D^T)$

- Because $\hat{\beta}^{ls} = (X^T X)^{-1} X^T y$
- We can apply the theorem, setting
 $u = y, \mu = X\beta, \Sigma = \sigma^2 \mathbf{I}, v = \hat{\beta}, c = \mathbf{0}, D = (X^T X)^{-1} X^T$
- Then $\hat{\beta}$ is normally distributed with mean $(X^T X)^{-1} (X^T X) \beta = \beta$
and covariance

$$((X^T X)^{-1} X^T) \sigma^2 \mathbf{I} ((X^T X)^{-1} X^T)^T = \sigma^2 (X^T X)^{-1}$$

Estimator variance

- $\hat{\beta} \sim N(\beta, \sigma^2(X^T X)^{-1})$

implies that the variance gets smaller,
the more data (X) we have,
around the correct mean

Goals of linear regression analysis

1. **Good fit:** Fitting a (hyper-)plane over predictor variables to explain response variables such that errors are small.
2. **Good parameter estimates:** describe change of response when varying some predictor variable(s).
3. **Good prediction:** predict a new response as a function of new predictor variables.
4. **Uncertainties and significance**
 - Confidence intervals & statistical tests

2 Good parameter estimates: describe change of response when varying some predictor variable(s).

- Linear regression with one predictor variable:

$$y = \beta_0 + \beta_1 x$$

- Standardize

$$x^* = \frac{x - \bar{x}}{\sigma_x}$$

$$y^* = \frac{y - \bar{y}}{\sigma_y}$$

$$\bar{x}^* = 0 = \bar{y}^*$$

$$\sigma_x^* = 1 = \sigma_y^*$$

$$\beta_0^* = \bar{y}^* - \beta_1^* \bar{x}^* = 0$$

- Correlation $\rho_{x,y} = \beta_1^* = \frac{1}{n} \sum_{i=1}^n x_i^* y_i^* = \beta_1 \frac{\sigma_x}{\sigma_y}$

Goals of linear regression analysis

1. **Good fit:** Fitting a (hyper-)plane over predictor variables to explain response variables such that errors are small.
2. **Good parameter estimates:** describe change of response when varying some predictor variable(s).
3. **Good prediction:** predict a new response as a function of new predictor variables.
4. **Uncertainties and significance**
 - Confidence intervals & statistical tests

4 Uncertainties and significance

Confidence intervals & statistical tests

Is variable X_j relevant?

Null-hypothesis $H_{0,j} : \beta_j = 0$

Alternative hypothesis: $H_{A,j} : \beta_j \neq 0$

$$\frac{\hat{\beta}_j}{\sqrt{\sigma^2 (X^T X)^{-1}_{jj}}} \sim N(0, 1) \quad \text{under the Null-hypothesis}$$

Studentt T-test statistics

$$T_j = \frac{\hat{\beta}_j}{\sqrt{\hat{\sigma}^2 (X^T X)^{-1}_{jj}}} \sim t_{n-p} \quad \text{under the null-hypothesis } H_{0,j}$$

Problem arises with this test
if predictor variables are correlated!!!!
(then use ANOVA on joint hypotheses)

Goals of linear regression analysis

1. **Good fit:** Fitting a (hyper-)plane over predictor variables to explain response variables such that errors are small.
2. **Good parameter estimates:** describe change of response when varying some predictor variable(s).
3. **Good prediction:** predict a new response as a function of new predictor variables.
4. **Uncertainties and significance**
 - Confidence intervals & statistical tests

Bias-Variance Trade-off

Overfitting & generalization of linear regression models

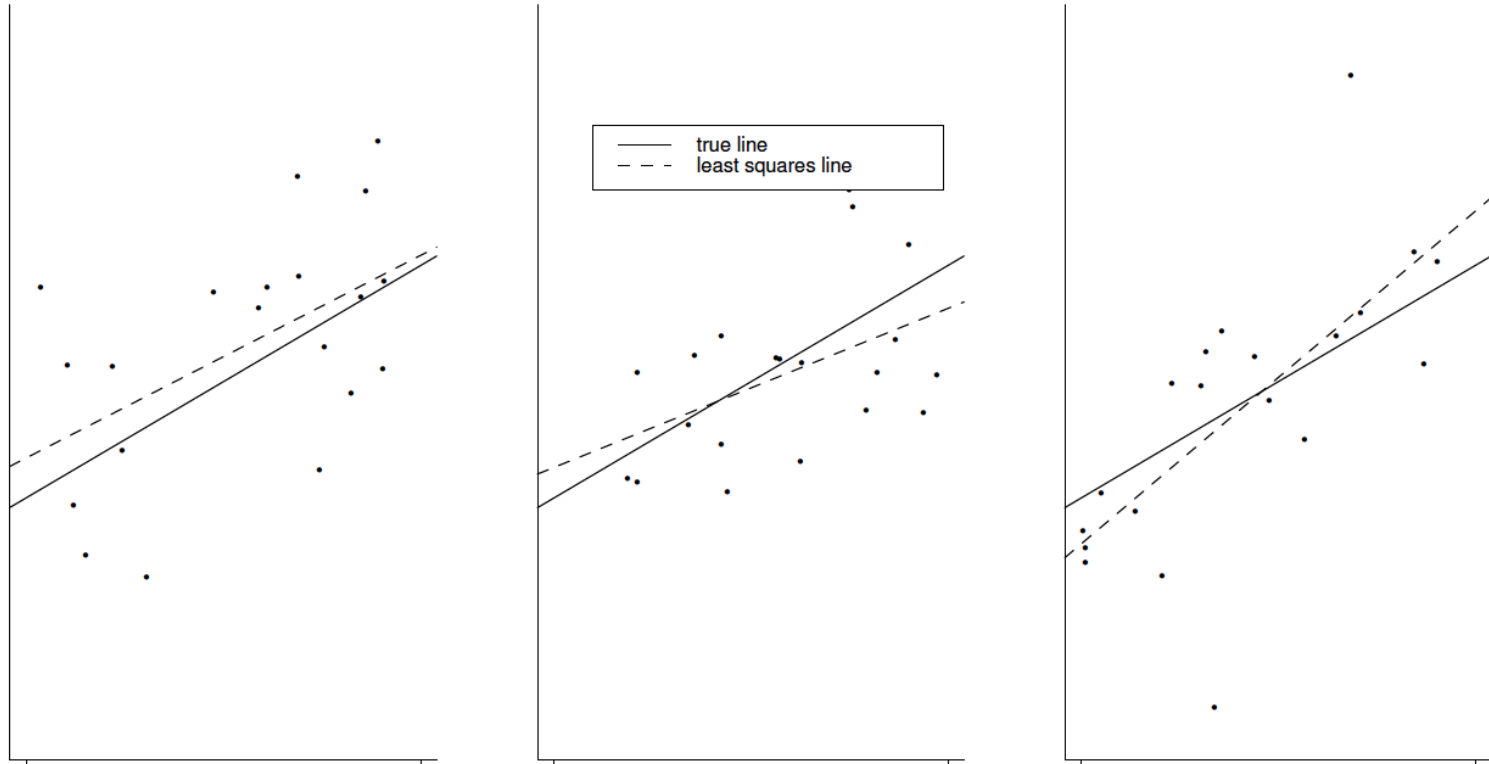
- The model overfits to the data—and may generalize badly

Estimator variance:

- When you repeat the experiment (keeping the underlying function fixed), the regression always returns a different model estimate
- **bias** and **variance** are error measures based on *average performance across possible training sets*

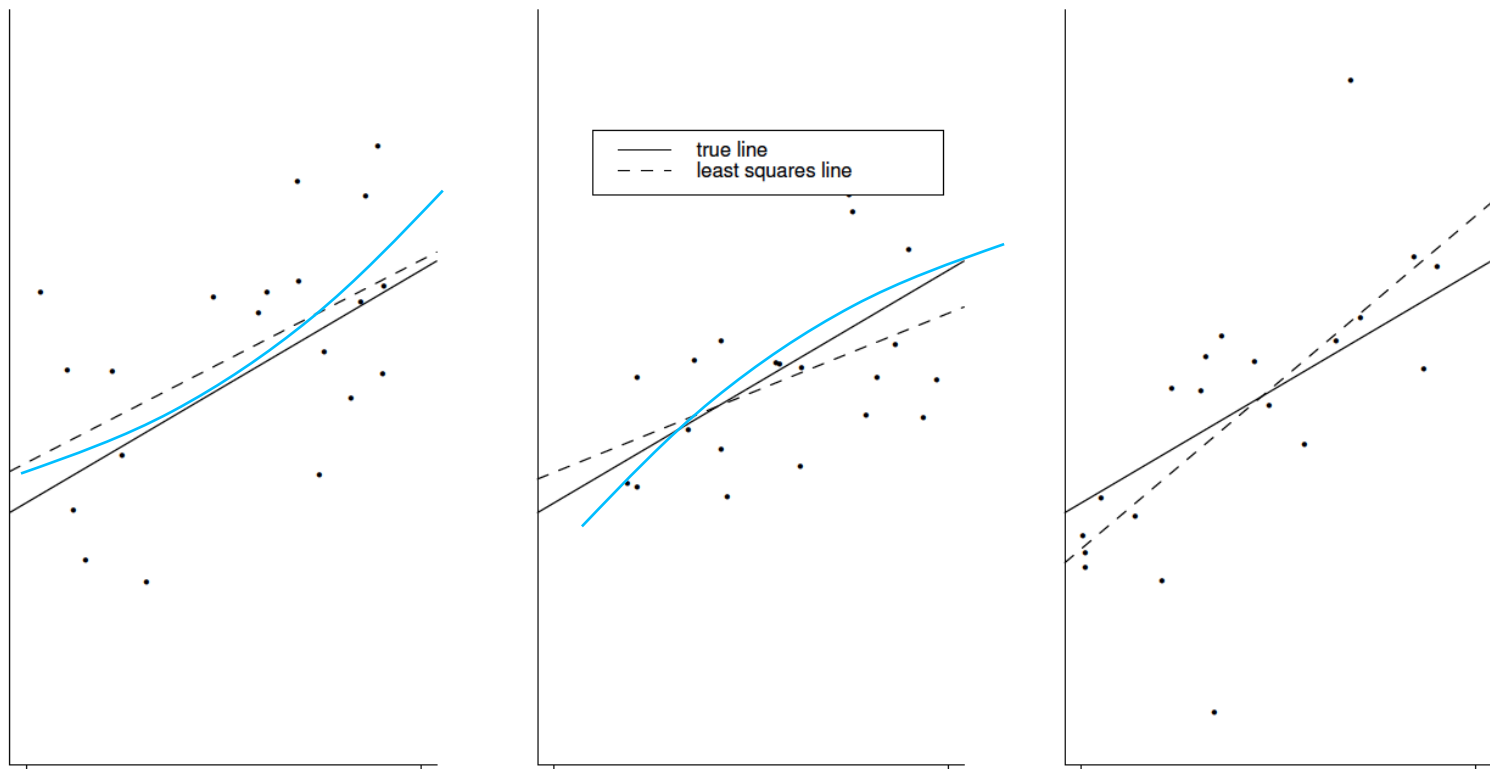
Sampling from the underlying noisy function

- When you repeat the experiment (keeping the underlying function fixed), the regression always returns a different model estimate



Sampling from the underlying noisy function

- Now with a more expressive **quadratic model**
- More expressive model will **always** reduce training error, but will fit to randomness in the training data



Background for investigating dependence on D

Notational

- D stands for training dataset
- $\hat{f} = \hat{f}(x; D), f = f(x)$

Background for investigating dependence on D

Notational

- D stands for training dataset
- $\hat{f} = \hat{f}(x; D), f = f(x)$

General statistics knowledge

- $\text{Var}[X] = \text{E}[X^2] - \text{E}[X]^2 \Leftrightarrow \text{E}[X^2] = \text{Var}[X] + \text{E}[X]^2$

Background for investigating dependence on D

Notational

- D stands for training dataset
- $\hat{f} = \hat{f}(x; D), f = f(x)$

General statistics knowledge

- $\text{Var}[X] = \text{E}[X^2] - \text{E}[X]^2 \Leftrightarrow \text{E}[X^2] = \text{Var}[X] + \text{E}[X]^2$

Background knowledge about residuals (modeling assumptions)

- $\text{E}[\varepsilon] = 0$
- $\text{E}[\varepsilon^2] = \text{Var}[\varepsilon] + \text{E}[\varepsilon]^2 = \sigma^2 + 0^2 = \sigma^2$

Background for investigating dependence on D

Notational

- D stands for training dataset
- $\hat{f} = \hat{f}(x; D), f = f(x)$

General statistics knowledge

- $\text{Var}[X] = \text{E}[X^2] - \text{E}[X]^2 \Leftrightarrow \text{E}[X^2] = \text{Var}[X] + \text{E}[X]^2$

Background knowledge about residuals (modeling assumptions)

- $\text{E}[\varepsilon] = 0$
- $\text{E}[\varepsilon^2] = \text{Var}[\varepsilon] + \text{E}[\varepsilon]^2 = \sigma^2 + 0^2 = \sigma^2$

Background knowledge about true function f

- f is true, error-free function, independent of D : $\text{E}[f] = f$
- $y = f + \varepsilon$
- $\text{E}[y] = \text{E}[f + \varepsilon] = \text{E}[f] = f$
- $\text{Var}[y] = \text{E}[(y - \text{E}[y])^2] = \text{E}[(y - f)^2] = \text{E}[(f + \varepsilon - f)^2] = \text{E}[\varepsilon^2] = \text{Var}[\varepsilon] + \text{E}[\varepsilon]^2 = \sigma^2 + 0^2 = \sigma^2$

Expected squared error when varying D

$$\begin{aligned} \mathbf{E}[(y - \hat{f})^2] &= \\ &= \mathbf{E}[y^2 + \hat{f}^2 - 2y\hat{f}] = \end{aligned}$$

Expected squared error when varying D

$$\begin{aligned} \mathbb{E}[(y - \hat{f})^2] &= \\ &= \mathbb{E}[y^2 + \hat{f}^2 - 2y\hat{f}] = \\ &= \mathbb{E}[y^2] + \mathbb{E}[\hat{f}^2] - \mathbb{E}[2y\hat{f}] = \\ &= \text{Var}[y] + \mathbb{E}[y]^2 + \text{Var}[\hat{f}] + \mathbb{E}[\hat{f}]^2 - 2f\mathbb{E}[\hat{f}] = \end{aligned}$$

Expected squared error when varying D

$$\begin{aligned} \mathbf{E}[(y - \hat{f})^2] &= \\ &= \mathbf{E}[y^2 + \hat{f}^2 - 2y\hat{f}] = \\ &= \mathbf{E}[y^2] + \mathbf{E}[\hat{f}^2] - \mathbf{E}[2y\hat{f}] = \\ &= \text{Var}[y] + \mathbf{E}[y]^2 + \text{Var}[\hat{f}] + \mathbf{E}[\hat{f}]^2 - 2f\mathbf{E}[\hat{f}] = \\ &= \text{Var}[y] + \mathbf{E}[y]^2 + \text{Var}[\hat{f}] + \mathbf{E}[\hat{f}]^2 - 2f\mathbf{E}[\hat{f}] = \\ &= \text{Var}[y] + \text{Var}[\hat{f}] + (f - \mathbf{E}[\hat{f}])^2 = \end{aligned}$$

Expected squared error when varying D

$$\begin{aligned} \mathbf{E}[(y - \hat{f})^2] &= \\ &= \mathbf{E}[y^2 + \hat{f}^2 - 2y\hat{f}] = \\ &= \mathbf{E}[y^2] + \mathbf{E}[\hat{f}^2] - \mathbf{E}[2y\hat{f}] = \\ &= \text{Var}[y] + \mathbf{E}[y]^2 + \text{Var}[\hat{f}] + \mathbf{E}[\hat{f}]^2 - 2f\mathbf{E}[\hat{f}] = \\ &= \text{Var}[y] + \mathbf{E}[y]^2 + \text{Var}[\hat{f}] + \mathbf{E}[\hat{f}]^2 - 2f\mathbf{E}[\hat{f}] = \\ &= \text{Var}[y] + \text{Var}[\hat{f}] + (f - \mathbf{E}[\hat{f}])^2 = \\ &= \text{Var}[y] + \text{Var}[\hat{f}] + \mathbf{E}[f - \hat{f}]^2 = \\ &= \sigma^2 + \text{Var}[\hat{f}] + \text{Bias}[\hat{f}]^2 \end{aligned}$$

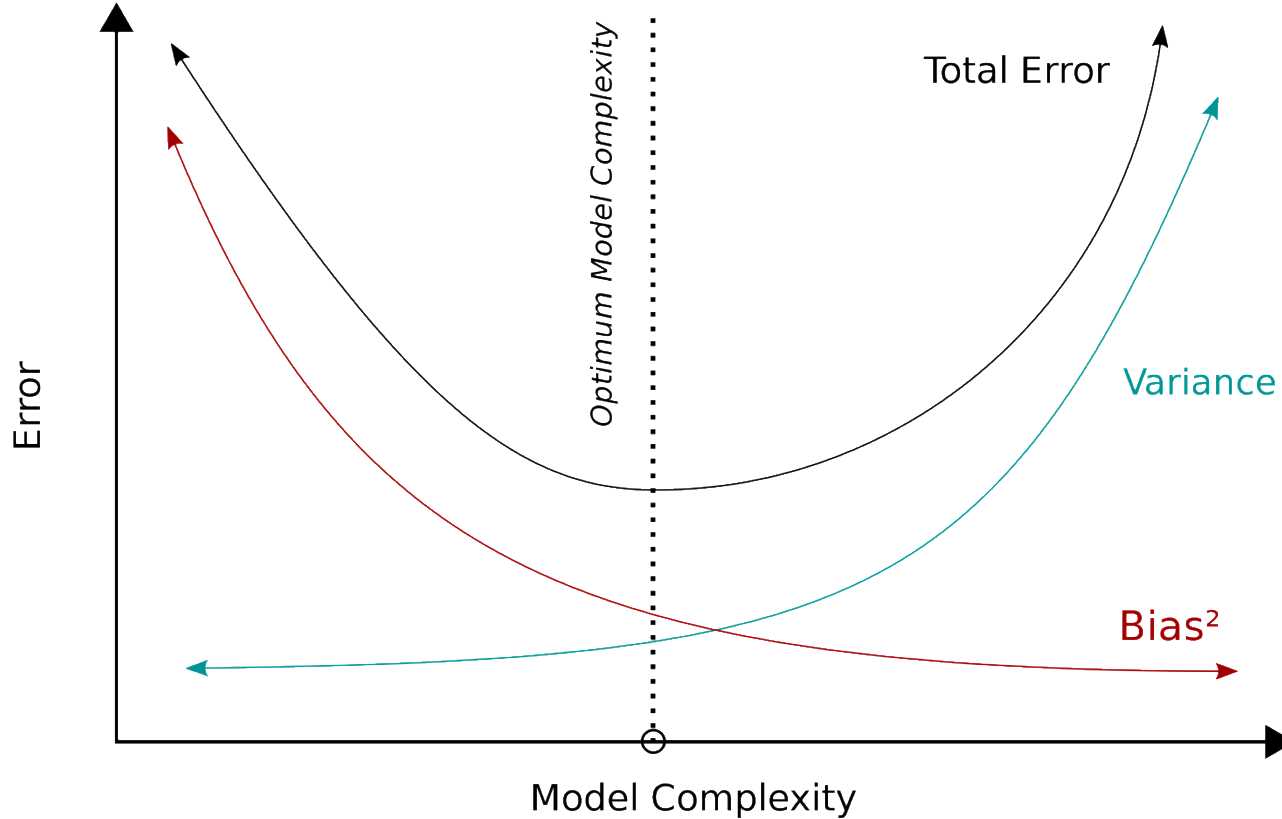
Expected squared error when varying D

$$\begin{aligned} \mathbf{E}[(y - \hat{f})^2] &= \\ &= \mathbf{E}[y^2 + \hat{f}^2 - 2y\hat{f}] = \\ &= \mathbf{E}[y^2] + \mathbf{E}[\hat{f}^2] - \mathbf{E}[2y\hat{f}] = \\ &= \mathbf{Var}[y] + \mathbf{E}[y]^2 + \mathbf{Var}[\hat{f}] + \mathbf{E}[\hat{f}]^2 - 2f\mathbf{E}[\hat{f}] = \\ &= \mathbf{Var}[y] + \mathbf{E}[y]^2 + \mathbf{Var}[\hat{f}] + \mathbf{E}[\hat{f}]^2 - 2f\mathbf{E}[\hat{f}] = \\ &= \mathbf{Var}[y] + \mathbf{Var}[\hat{f}] + (f - \mathbf{E}[\hat{f}])^2 = \\ &= \mathbf{Var}[y] + \mathbf{Var}[\hat{f}] + \mathbf{E}[f - \hat{f}]^2 = \\ &= \boldsymbol{\sigma}^2 + \mathbf{Var}[\hat{f}] + \mathbf{Bias}[\hat{f}]^2 \end{aligned}$$

Expected mean squared error:

$$\text{MSE} = \mathbf{E}_x\{\text{Bias}_D[\hat{f}(x; D)]^2 + \text{Var}_D[\hat{f}(x; D)]\} + \sigma^2$$

Bias-Variance Tradeoff for Different Models



Research in recent years: double descent

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal

Reconciling modern machine-learning practice and the classical bias–variance trade-off

PNAS 2019

<https://www.pnas.org/doi/pdf/10.1073/pnas.1903070116>

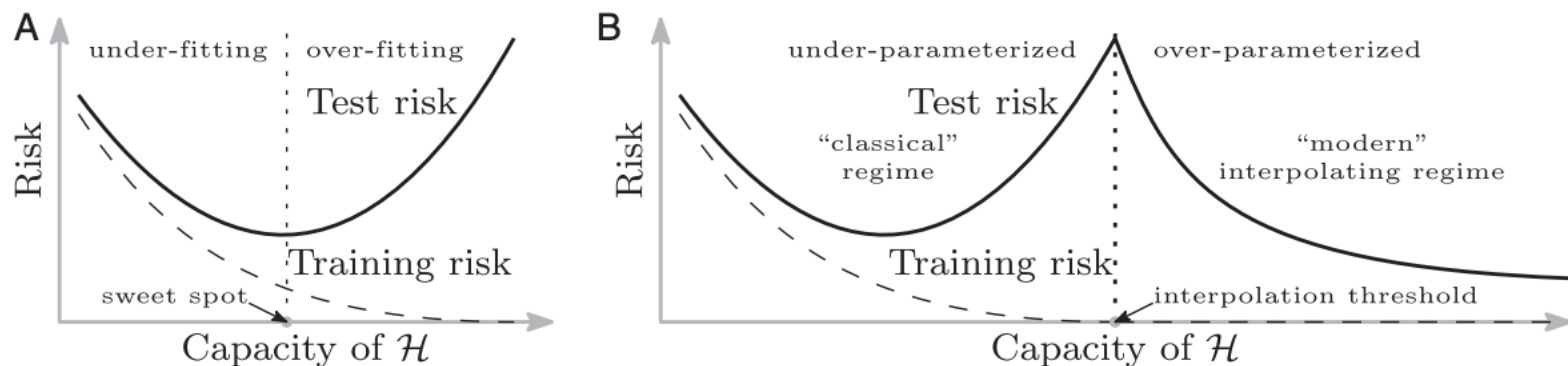
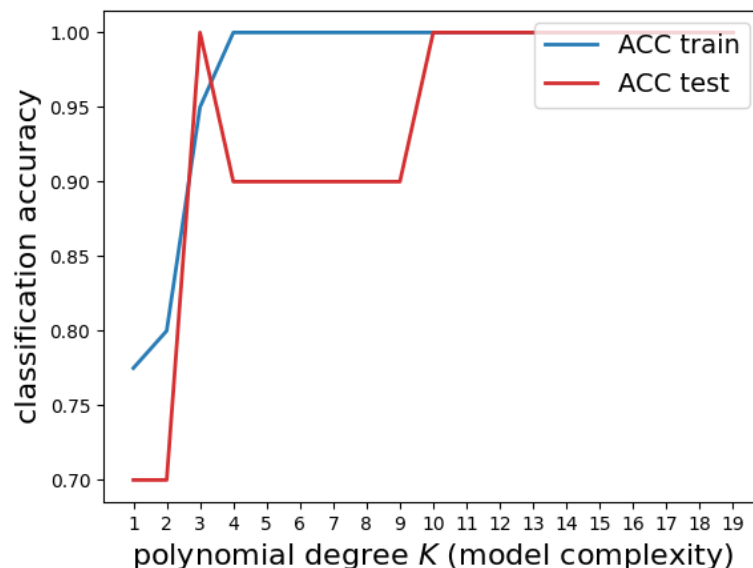
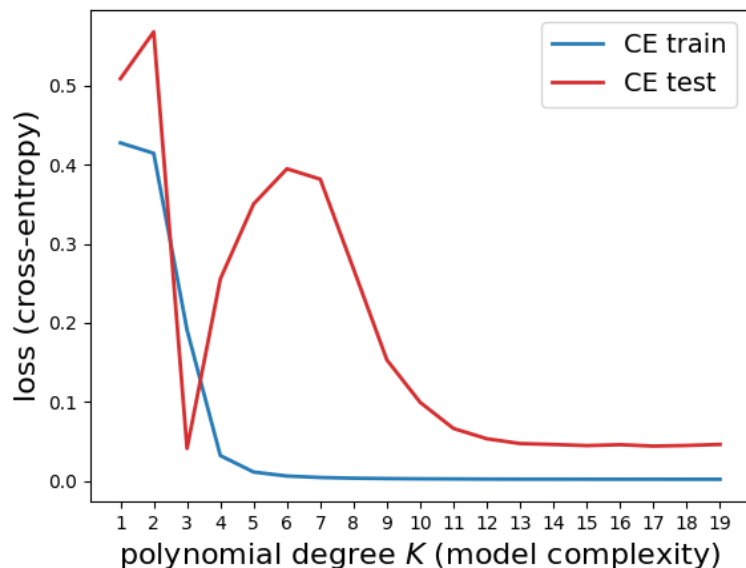


Fig. 1. Curves for training risk (dashed line) and test risk (solid line). (A) The classical U-shaped risk curve arising from the bias–variance trade-off. (B) The double-descent risk curve, which incorporates the U-shaped risk curve (i.e., the “classical” regime) together with the observed behavior from using high-capacity function classes (i.e., the “modern” interpolating regime), separated by the interpolation threshold. The predictors to the right of the interpolation threshold have zero training risk.

Research in recent years: double descent

Awkward example: logistic regression with polynomial features



- <https://twitter.com/ropeharz/status/1505337492807036944>

Reducing model complexity

Subset selection

Issues with (pure) least squares estimates:

1. Prediction accuracy: least squares estimates often have low bias but large variance.

- Prediction accuracy can sometimes be improved by shrinking or setting some coefficients to zero.
Increase bias to reduce the variance of the predicted values, and hence the overall prediction accuracy.

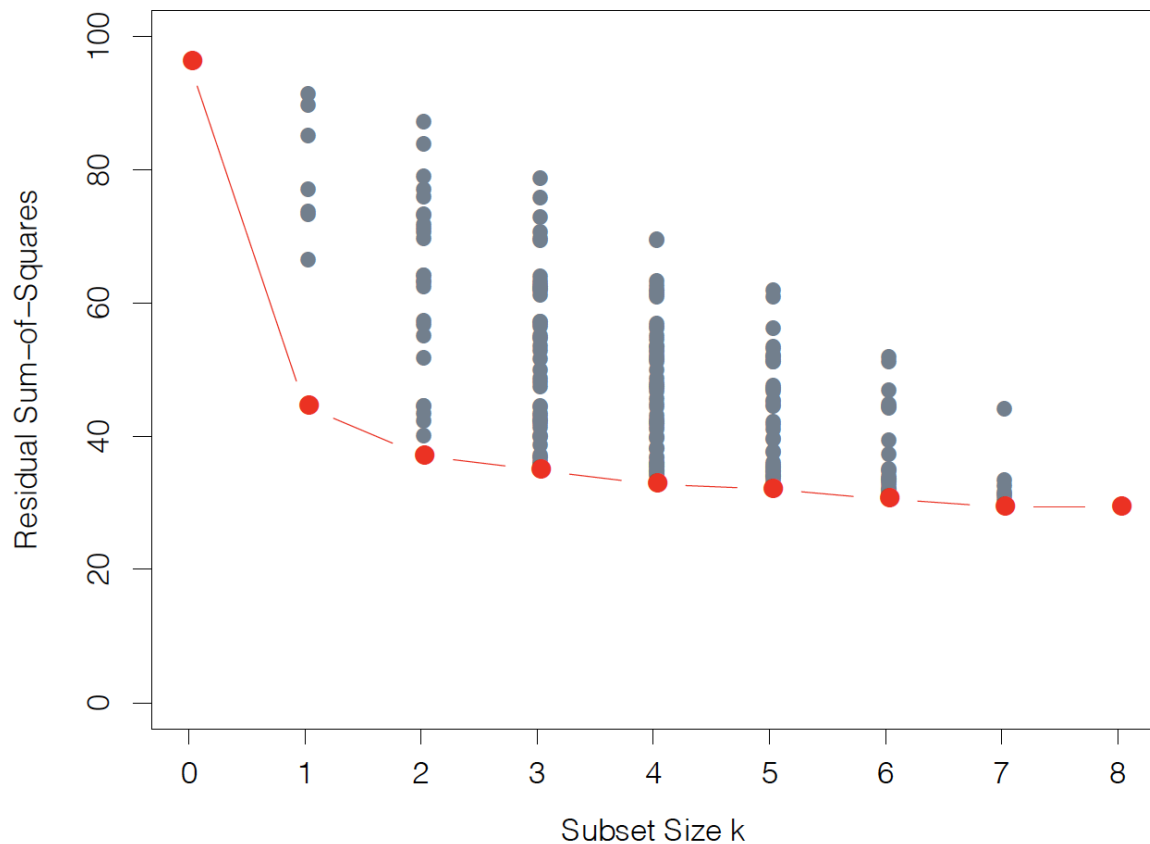
2. Interpretation: determine a smaller subset that exhibit the strongest effects.

- Understand the “big picture,” sacrifice some of the small details.

Growing the model

- Example: from 0 to 7 variables

- Larger model
 - higher variance
 - lower bias



(Hastie et al.,
page 58)

Subset selection

- fewer variables (x_i) allow for better interpretation
- binary choice: variable is either retained or discarded

Continuous Alternative: Shrinkage methods

Ridge Regression: L_2 -regularization

- Idea: constrain the size of β

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \sum_{I=1}^n \left(y_i - \beta_0 - \sum_{j=1}^d x_{i,j} \beta_j \right)^2$$

$$\sum_{j=1}^d \beta_j^2 \leq t$$

- can also be written with Lagrange multiplier λ as

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

- Note: The intercept β_0 is not regularized!
- Lagrange multiplier λ formalizes the inequality constraints

Regularized sum of squares

In matrix notation

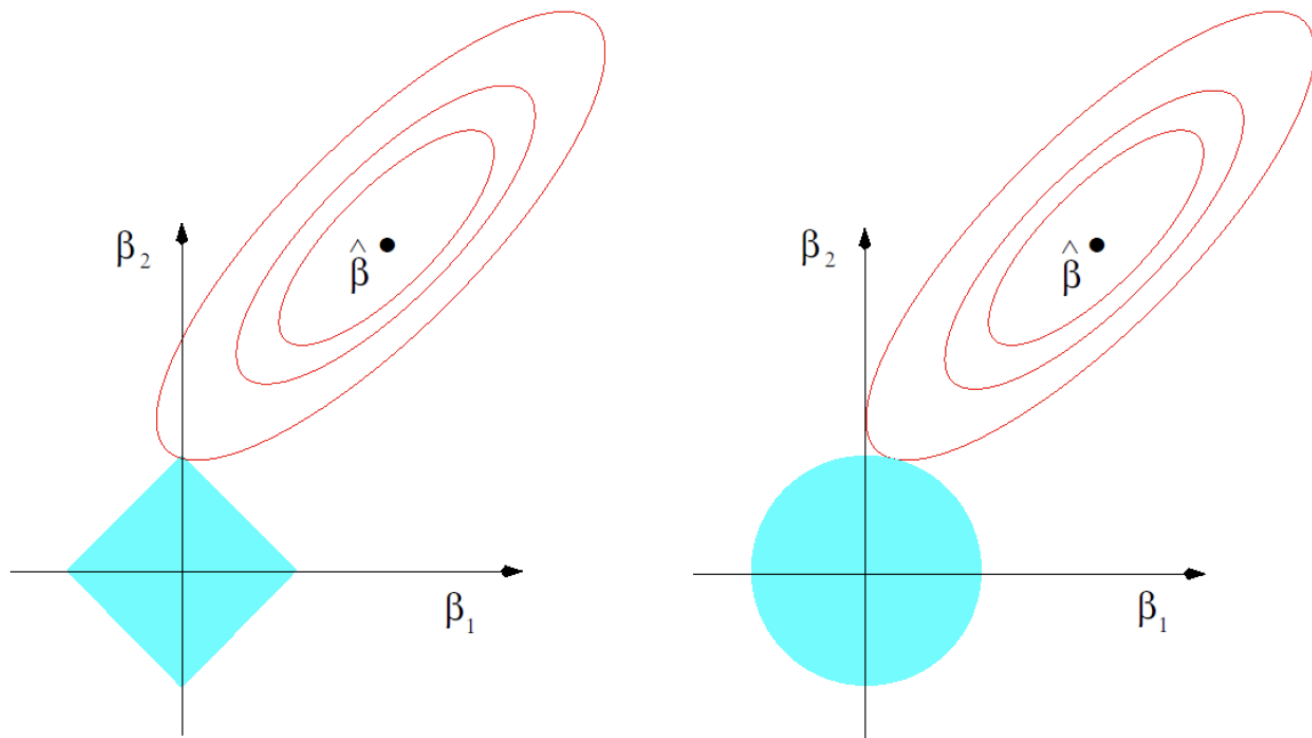
- $\text{RSS}(\lambda) = (y - X\beta)^T (y - X\beta) + \lambda \beta^T \beta$
- implies $\hat{\beta}^{\text{ridge}} = (X^T X + \lambda \mathbf{I})^{-1} X^T y$

Lasso: L_1 -regularization

- **beta lasso** $\beta^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^d x_{i,j} \beta_j \right)^2 + \lambda \sum_{j=1}^d |\beta_j| \right\}$

- No closed form optimization \rightarrow quadratic programming

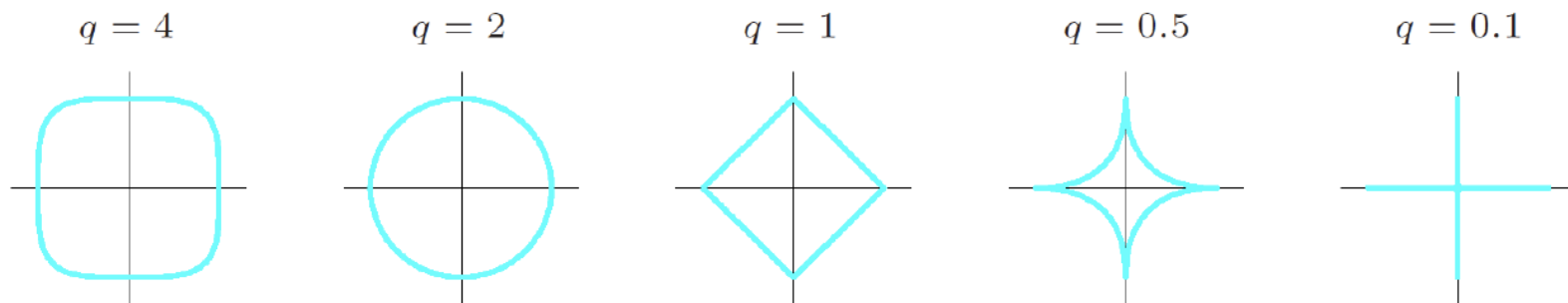
Lasso vs. Ridge:



- Lasso \rightarrow sparsity! feature selection!

(slide by Marc Toussaint 2019)

$$L^q(\beta) = \sum_{i=1}^n (y_i - \phi(x_i)^\top \beta)^2 + \lambda \sum_{j=2}^k |\beta_j|^q$$



- *Subset selection*: $q = 0$ (counting the number of $\beta_j \neq 0$)

Choosing λ : generalization error & cross validation

- $\lambda = 0$ will always have a lower *training* data error

We need to estimate the *generalization* error on test data

- **k -fold cross-validation:**



-
- 1: Partition data D in k equal sized subsets $D = \{D_1, \dots, D_k\}$
 - 2: **for** $i = 1, \dots, k$ **do**
 - 3: compute $\hat{\beta}_i$ on the training data $D \setminus D_i$ leaving out D_i
 - 4: compute the error $\ell_i = L^{\text{ls}}(\hat{\beta}_i, D_i)/|D_i|$ on the validation data D_i
 - 5: **end for**
 - 6: report mean squared error $\hat{\ell} = 1/k \sum_i \ell_i$ and variance $1/(k-1)[(\sum_i \ell_i^2) - k\hat{\ell}^2]$
-

- Choose λ for which $\hat{\ell}$ is smallest

(slide by Marc Toussaint 2019)

Summary

- **Representation:** choice of features $f(x) = \phi(x)^T \beta$
 - linear, polynomial
 - piece-wise linear
 - radial-basis functions, ...
- **Objective / loss function:** squared error + regularization
 - ridge, lasso
 - subset selection
- **Solver:** analytic (for ridge), quadratic program (for Lasso), stochastic gradient descent

References

1. Trevor Hastie, Robert Tibshirani, Jerome Friedman. The Elements of Statistical Learning. Data Mining, Inference, and Prediction. Springer Series in Statistics. Download from http://www.web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf



Universität Stuttgart
KI

Thank you!



Steffen Staab

E-Mail Steffen.staab@ki.uni-stuttgart.de

Telefon +49 (0) 711 685-88100

www.ki.uni-stuttgart.de/

Universität Stuttgart

Analytic Computing, KI

Universitätsstraße 32, 50569 Stuttgart