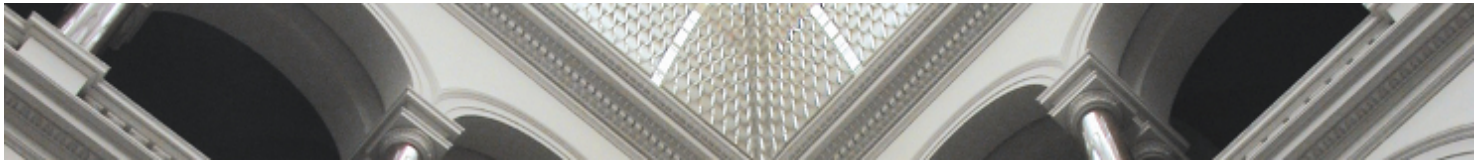
Technische Universität Berlin


Classical Methods: Regression & Regularization

Julius Hense
BIFOLD & Technische Universität Berlin - Machine Learning Group
BIFOLD AQTIVATE Workshop



Technische Universität Berlin

Agenda

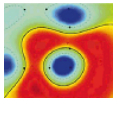
Covariance & Correlation

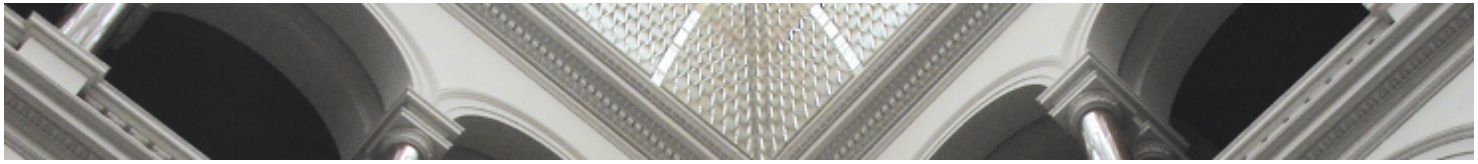
Linear Regression

Regularization, Ridge & Lasso

Logistic Regression

More Methods & Practical Considerations





Technische Universität Berlin

Agenda

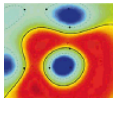
Covariance & Correlation

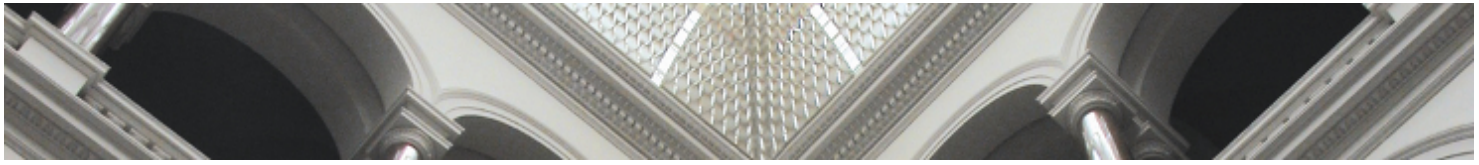
Linear Regression

Regularization, Ridge & Lasso

Logistic Regression

More Methods & Practical Considerations



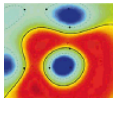


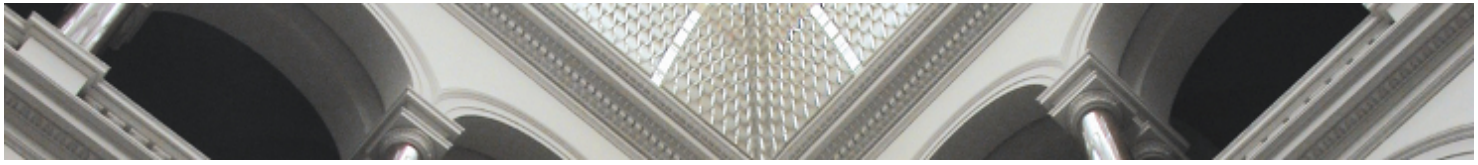
Covariance and Correlation

For two random variables X and Y, their **covariance** and **correlation** are defined as

$$\mathrm{Cov}(X, Y) := \mathrm{E}[(X - \mathrm{E}(X))(Y - \mathrm{E}(Y))]$$

$$\mathrm{Corr}(X, Y) := \frac{\mathrm{Cov}(X, Y)}{\sqrt{\mathrm{Var}(X)}\sqrt{\mathrm{Var}(Y)}}.$$





Technische Universität Berlin

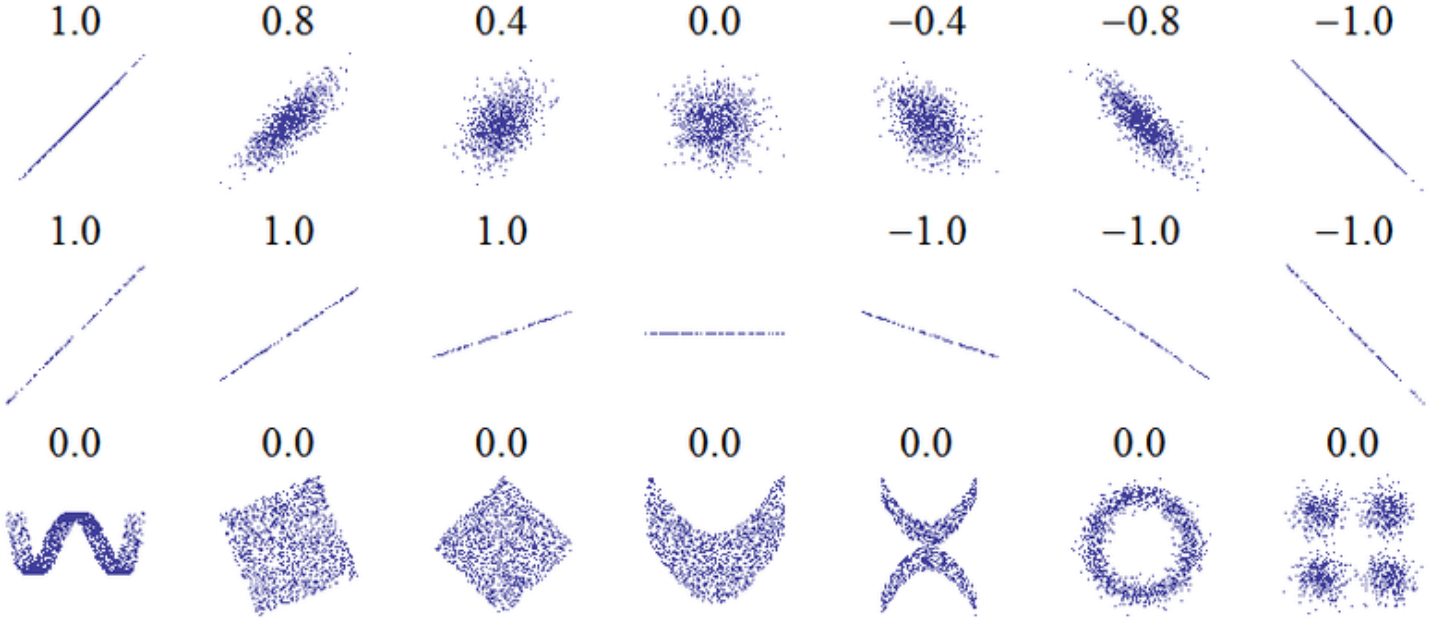
Covariance and Correlation

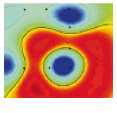
For two random variables X and Y, their **covariance** and **correlation** are defined as

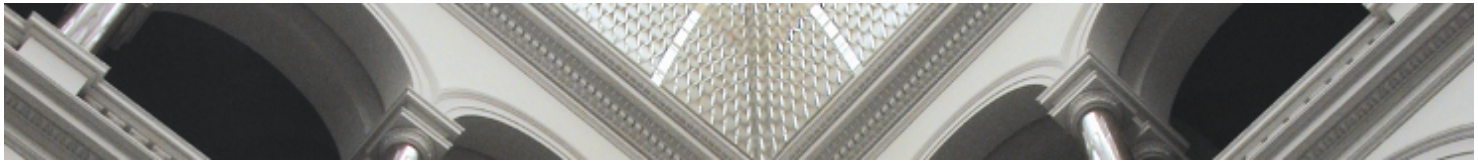
$$\mathrm{Cov}(X, Y) := \mathrm{E}[(X - \mathrm{E}(X))(Y - \mathrm{E}(Y))]$$

$$\mathrm{Corr}(X, Y) := \frac{\mathrm{Cov}(X, Y)}{\sqrt{\mathrm{Var}(X)}\sqrt{\mathrm{Var}(Y)}}.$$

Correlation measures the linear relationship between X and Y:







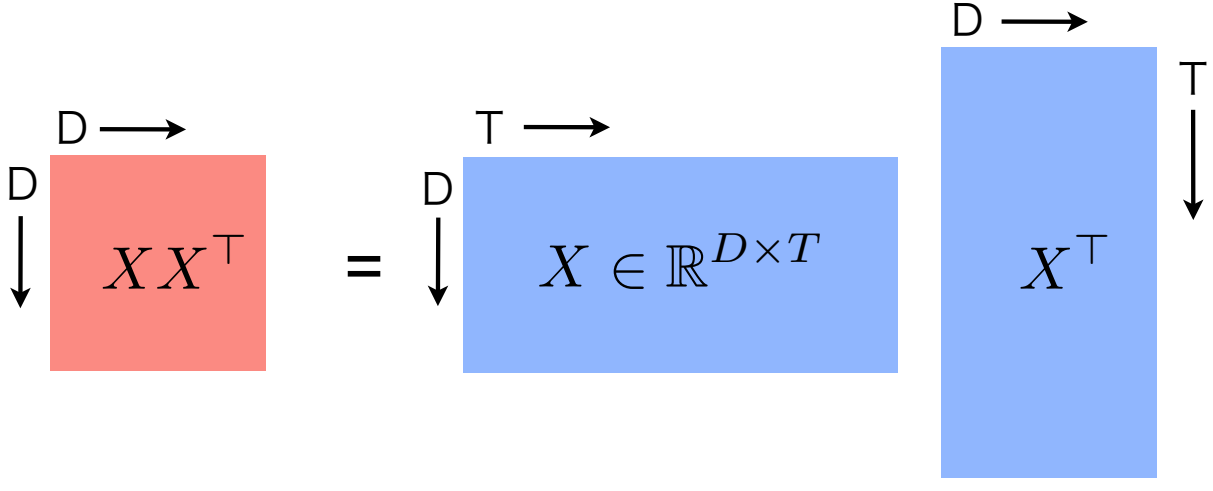
Technische Universität Berlin

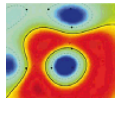
Covariance Matrices

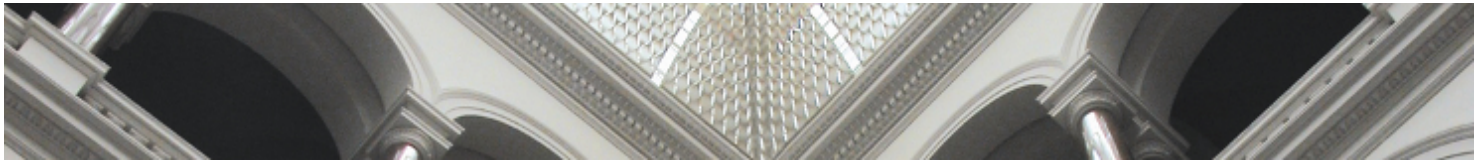
Given T data points $\mathbf{x}_t \in \mathbb{R}^D$ in a data matrix $X \in \mathbb{R}^{D \times T}$ the empirical estimate of the **covariance matrix** is defined as

$$S = \frac{1}{T} XX^\top \tag{1}$$

where we assume centered data, i.e. $\sum_{t=1}^{T} \mathbf{x}_t = \mathbf{0}$.

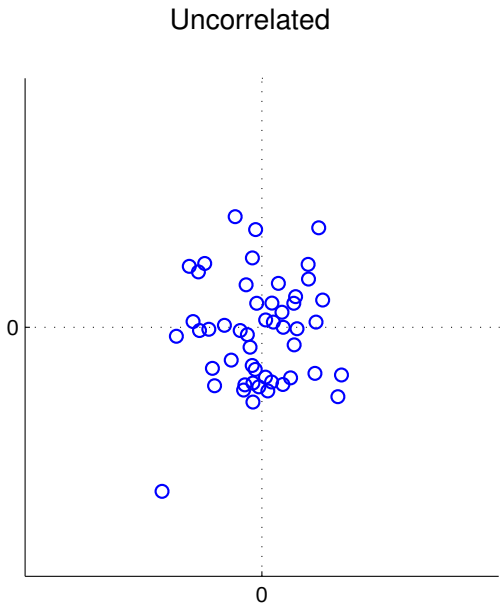




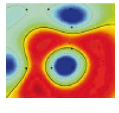


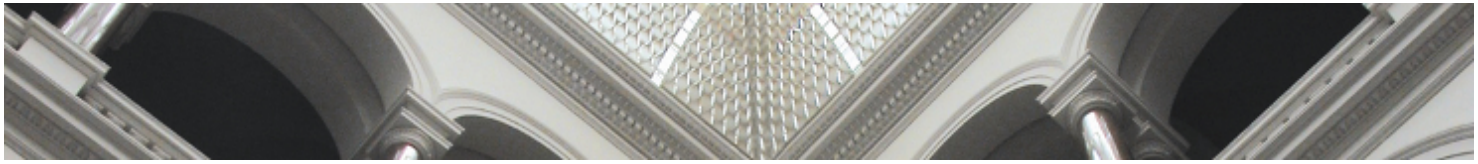
Technische Universität Berlin

Correlated Data: Examples



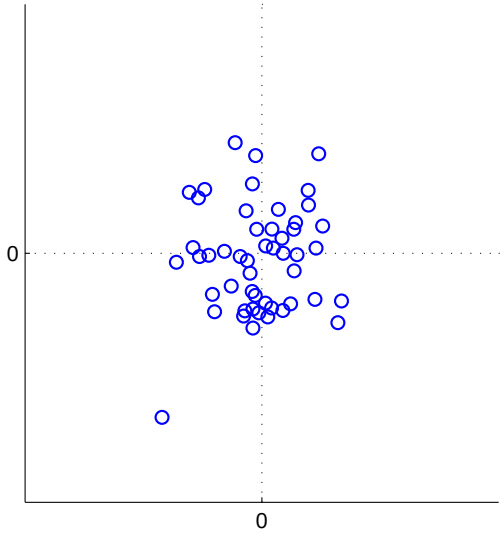
$$xx^\top = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$





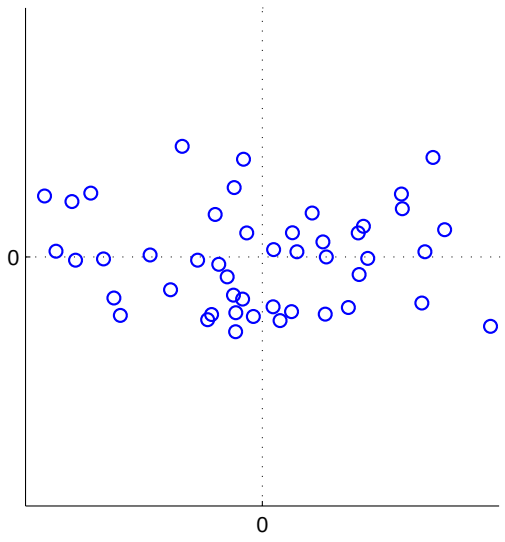
Correlated Data: Examples

Uncorrelated

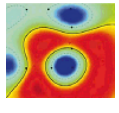


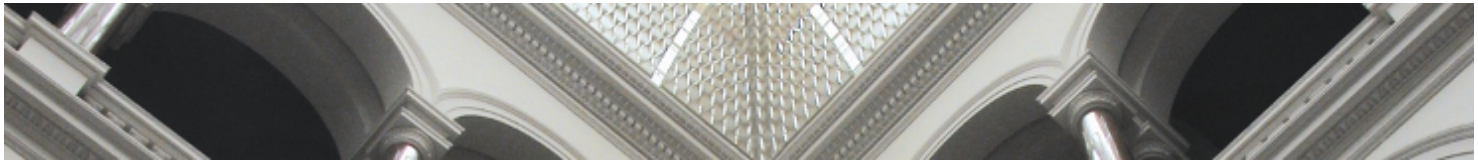
$$xx^\top = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Uncorrelated, scaled



$$xx^\top = \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix}$$

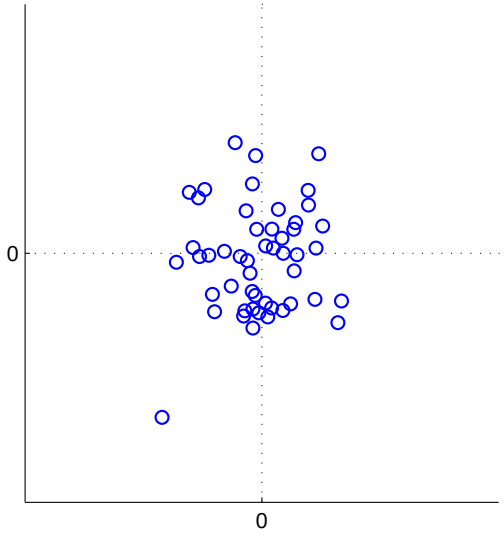




Technische Universität Berlin

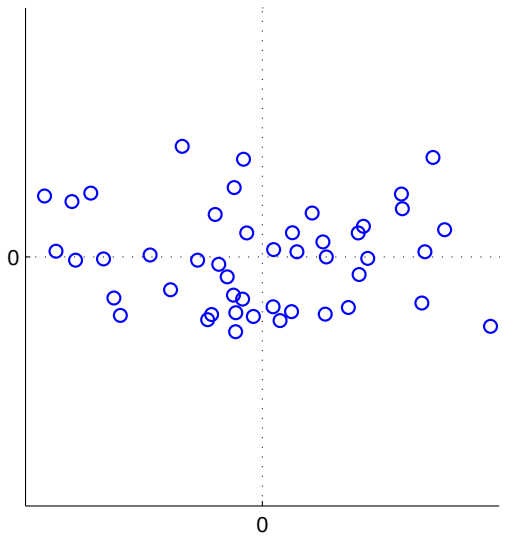
Correlated Data: Examples

Uncorrelated



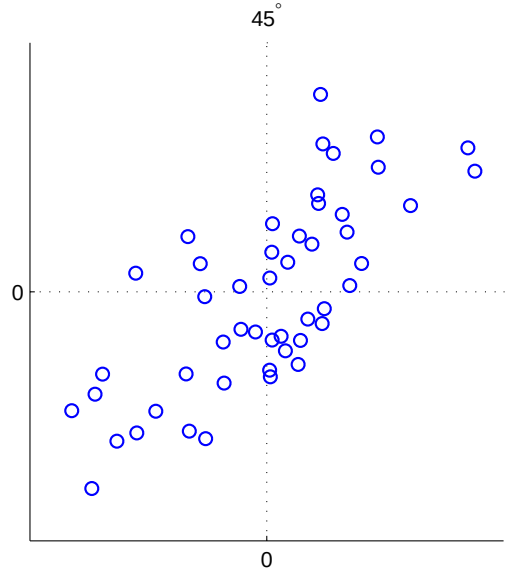
$$xx^\top = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Uncorrelated, scaled

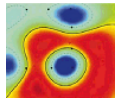


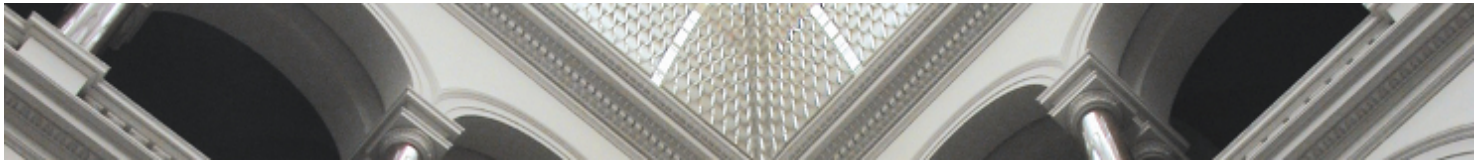
$$xx^\top = \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix}$$

Scaled, rotated by 45°



$$xx^\top = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$$





Technische Universität Berlin

Agenda

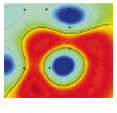
Covariance & Correlation

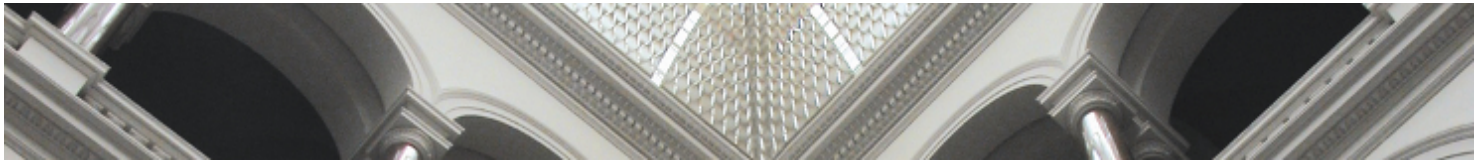
Linear Regression

Regularization, Ridge & Lasso

Logistic Regression

More Methods & Practical Considerations

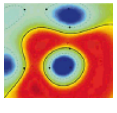


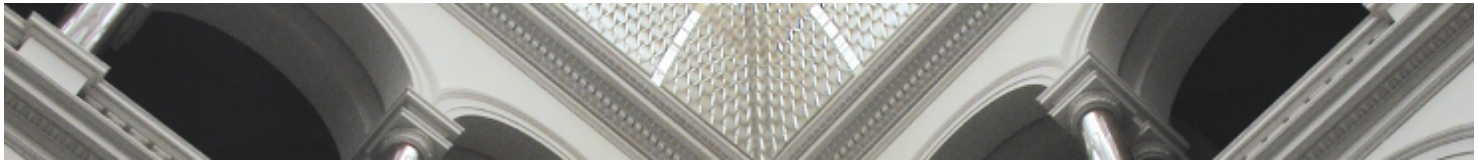


Technische Universität Berlin

Linear Regression: Example Applications

- **Predict** the price of a stock in 6 months based on company performance measures and economic data.
- **Predict** crop yield from environmental data.
- **Control** an artificial hand based on electronic activity measured on the arm.



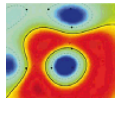


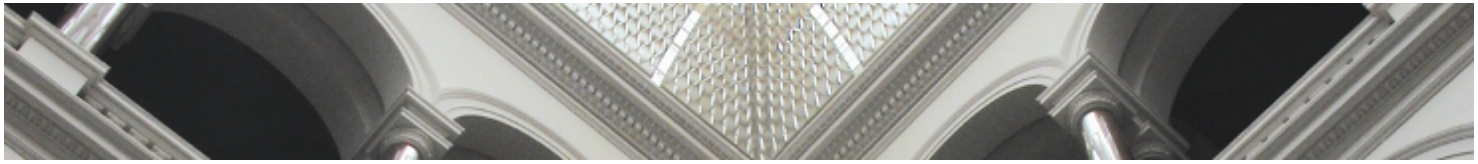
Technische Universität Berlin

Linear Regression: Example Applications

- **Predict** the price of a stock in 6 months based on company performance measures and economic data.
- **Predict** crop yield from environmental data.
- **Control** an artificial hand based on electronic activity measured on the arm.

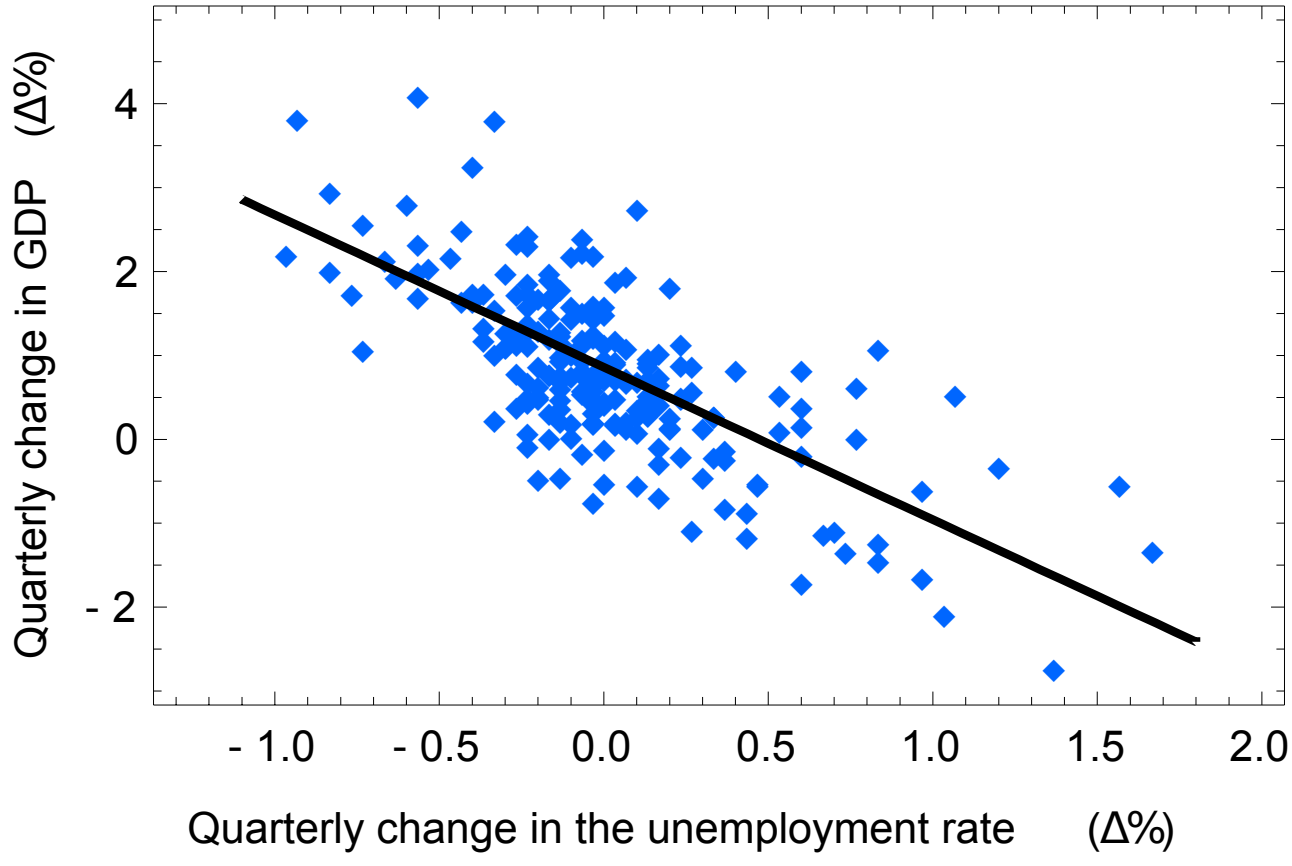
- Given data on product sales and TV, radio, and social media advertising budgets. Is there a **relationship** between advertising budget and sales? Which media contribute to sales? How accurately can we predict future sales?
- . . .

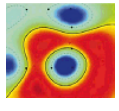


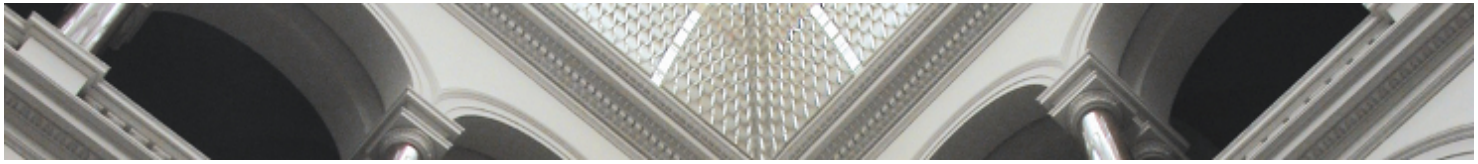


Technische Universität Berlin

Simple Linear Regression





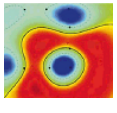


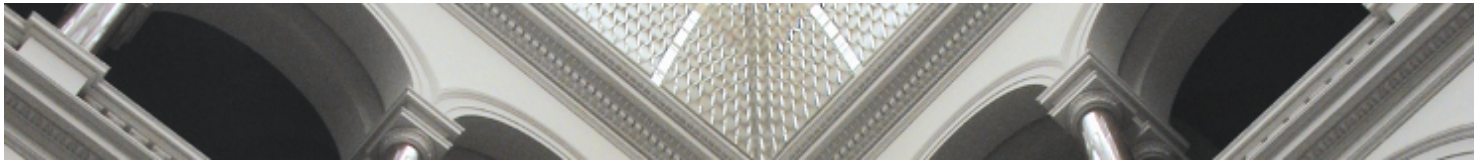
Technische Universität Berlin

Simple Linear Regression

Given data $x_1, \ldots x_T \in \mathbb{R}$ and labels $y_1, \ldots, y_T \in \mathbb{R}$, the goal is to predict y by a linear function of x

$$f(x_t) = \omega \cdot x_t$$



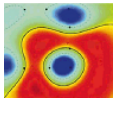


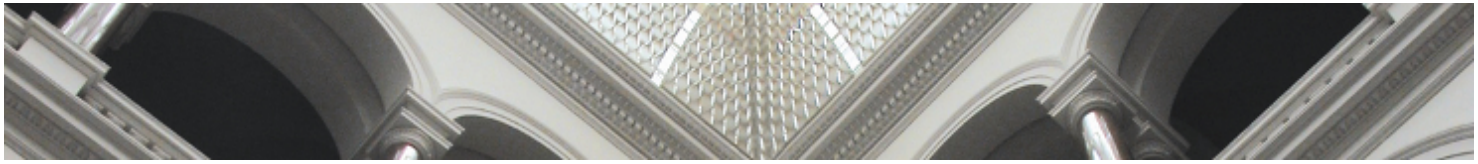
Simple Linear Regression

Given data $x_1, \ldots x_T \in \mathbb{R}$ and labels $y_1, \ldots, y_T \in \mathbb{R}$, the goal is to predict y by a linear function of x

$$f(x_t) = \omega \cdot x_t$$

$\Rightarrow$ Minimize **least-square error** to find the ”best fit ”



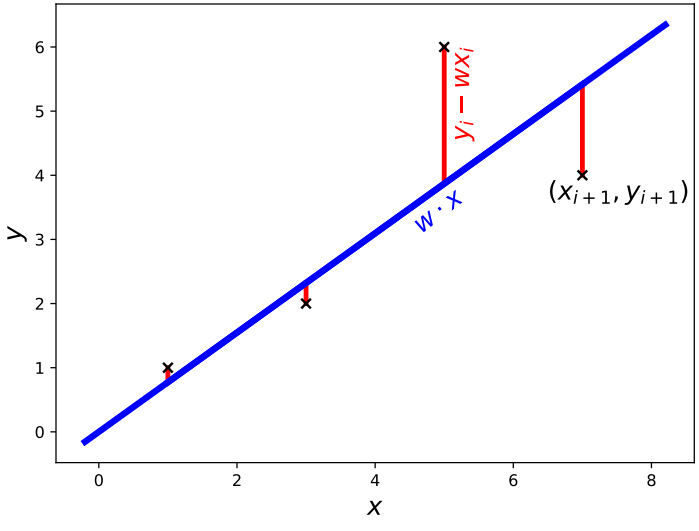


Simple Linear Regression

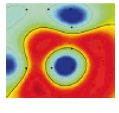
Given data $x_1, \ldots x_T \in \mathbb{R}$ and labels $y_1, \ldots, y_T \in \mathbb{R}$, the goal is to predict y by a linear function of x

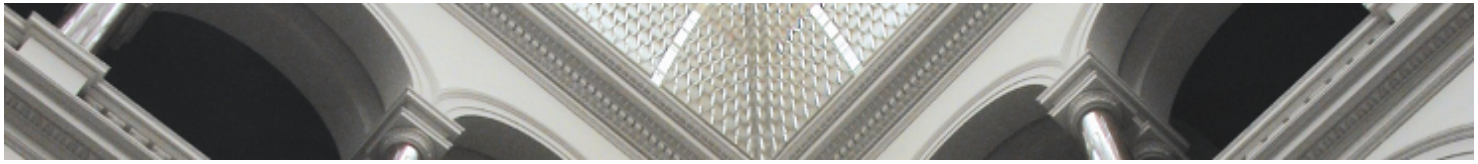
$$f(x_t) = \omega \cdot x_t$$

$\Rightarrow$ Minimize **least-square error** to find the ”best fit ”



$$\mathcal{E}(\omega) = \sum_{t=1}^{T} (y_t - \omega \cdot x_t)^2$$



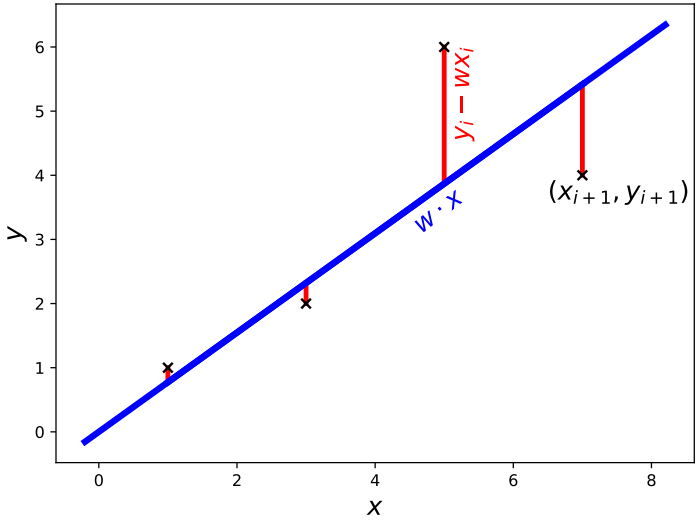


Simple Linear Regression

Given data $x_1, \ldots x_T \in \mathbb{R}$ and labels $y_1, \ldots, y_T \in \mathbb{R}$, the goal is to predict y by a linear function of x

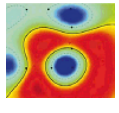
$$f(x_t) = \omega \cdot x_t$$

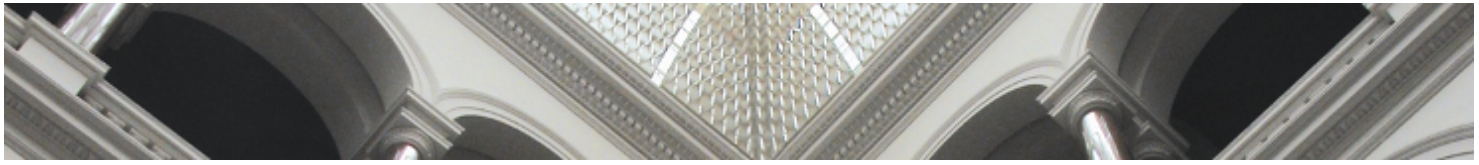
$\Rightarrow$ Minimize **least-square error** to find the ”best fit ”



$$\mathcal{E}(\omega) = \sum_{t=1}^{T} (y_t - \omega \cdot x_t)^2$$

- differentiable
- leads to analytic solution





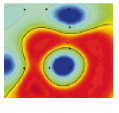
Technische Universität Berlin

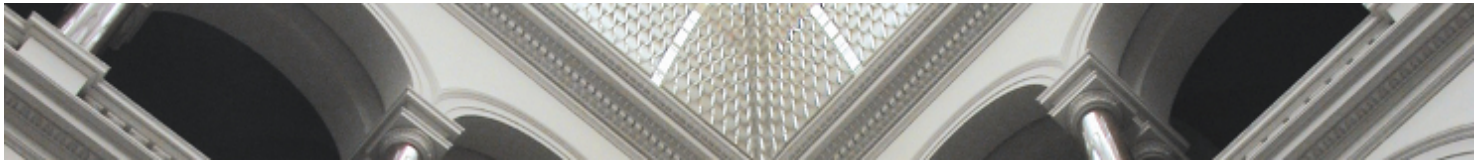
Simple Linear Regression

$$\mathcal{E}(\omega) = \sum_{t=1}^{T} (y_t - \omega \cdot x_t)^2$$

Compute the derivative w.r.t. ω

$$\frac{\partial \mathcal{E}(\omega)}{\partial \omega} = \sum_{t=1}^{T} 2(y_t - \omega \cdot x_t) \cdot (-x_t)$$





Technische Universität Berlin

Simple Linear Regression

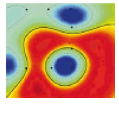
$$\mathcal{E}(\omega) = \sum_{t=1}^{T} (y_t - \omega \cdot x_t)^2$$

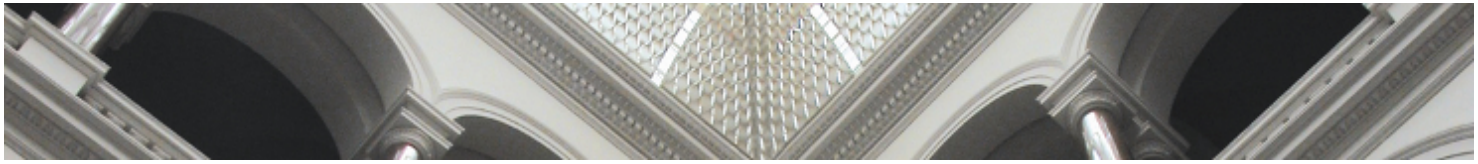
Compute the derivative w.r.t. ω

$$\frac{\partial \mathcal{E}(\omega)}{\partial \omega} = \sum_{t=1}^{T} 2(y_t - \omega \cdot x_t) \cdot (-x_t)$$

sets it to zero and solves for ω:

$$\sum_{t=1}^{T} 2(y_t - \omega \cdot x_t) \cdot (-x_t) = 0 \quad \Longrightarrow \quad \sum_{t=1}^{T} y_t x_t - \omega \sum_{t=1}^{T} x_t^2 = 0$$





Technische Universität Berlin

Simple Linear Regression

$$\mathcal{E}(\omega) = \sum_{t=1}^{T} (y_t - \omega \cdot x_t)^2$$

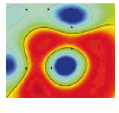
Compute the derivative w.r.t. ω

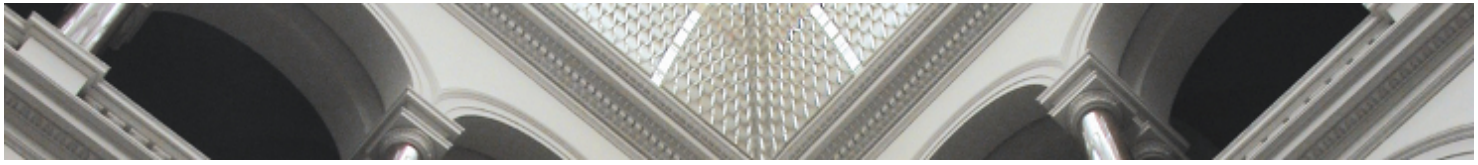
$$\frac{\partial \mathcal{E}(\omega)}{\partial \omega} = \sum_{t=1}^{T} 2(y_t - \omega \cdot x_t) \cdot (-x_t)$$

sets it to zero and solves for ω:

$$\sum_{t=1}^{T} 2(y_t - \omega \cdot x_t) \cdot (-x_t) = 0 \quad \Longrightarrow \quad \sum_{t=1}^{T} y_t x_t - \omega \sum_{t=1}^{T} x_t^2 = 0$$

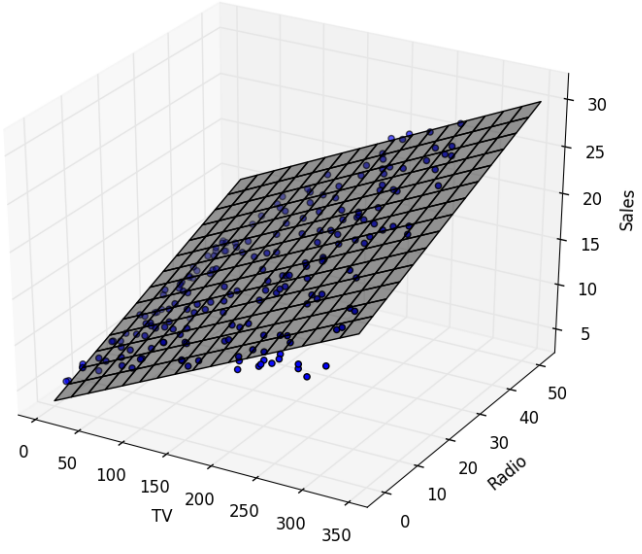
$$\Longrightarrow \quad \omega = \frac{\sum_{t=1}^{T} x_t y_t}{\sum_{t=1}^{T} x_t x_t}$$



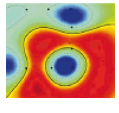


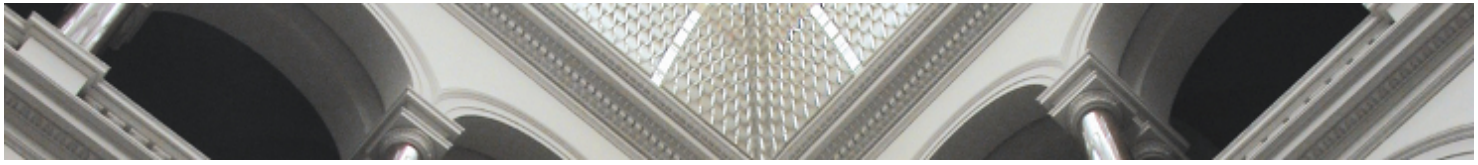
Technische Universität Berlin

Linear Regression



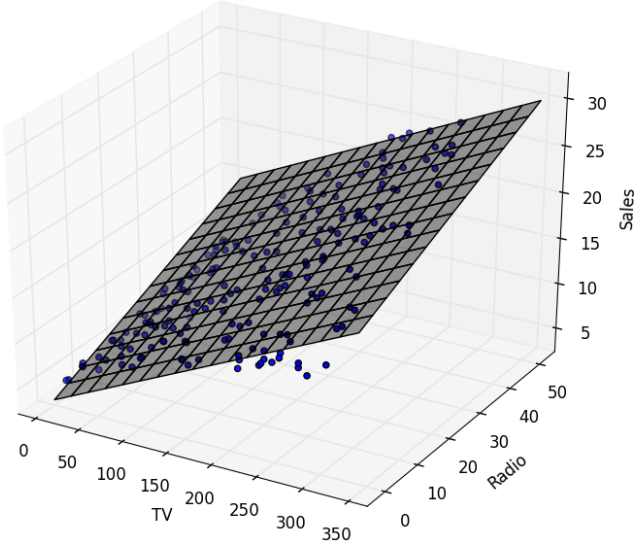
$$Y = w_1 \cdot X_1 + w_2 \cdot X_2$$





Technische Universität Berlin

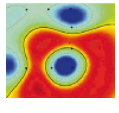
Linear Regression

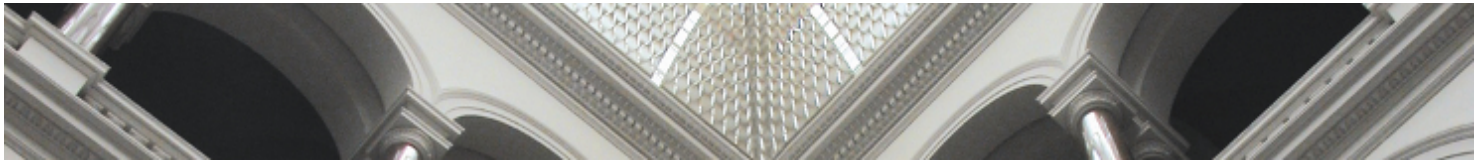


$$Y = w_1 \cdot X_1 + w_2 \cdot X_2$$

Target variable $y \in \mathbb{R}$ is modeled as a **linear combination** $\mathbf{w} \in \mathbb{R}^D$ of D features $\mathbf{x} \in \mathbb{R}^D$

$$y = \mathbf{w}^\top \mathbf{x}$$





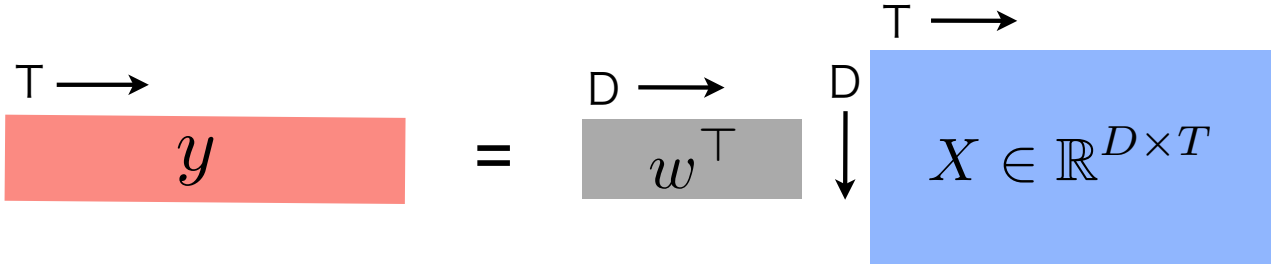
Technische Universität Berlin

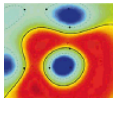
Linear Regression

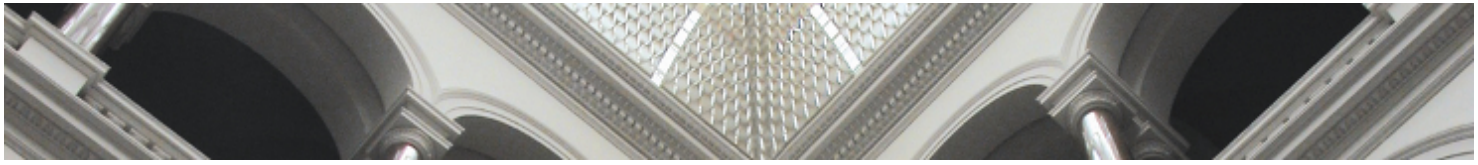
Let T be the number of samples, so $y \in \mathbb{R}^{1\times T}$ and $X \in \mathbb{R}^{D\times T}$.

The Linear Regression model in matrix notation then becomes

$$y = \mathbf{w}^\top X.$$







Technische Universität Berlin

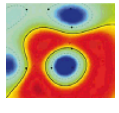
Linear Regression

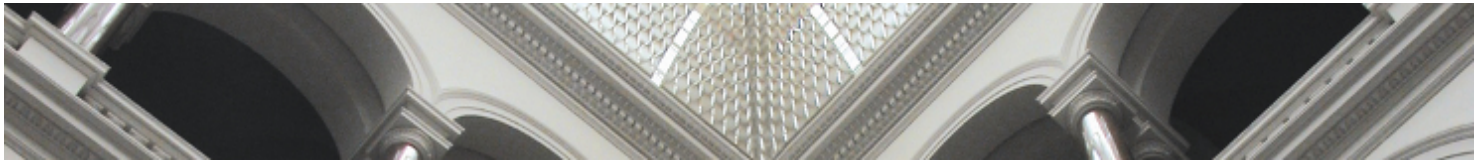
The most popular loss function to optimize **w**
is the **least-square error**

$$\mathcal{E}_{lsq}(\mathbf{w}) = \sum_{t=1}^{T} (y_t - \mathbf{w}^\top X_t)^2 \tag{2}$$

C.F. Gauß (1777-1855)

A.M. Legendre (1752-1833)



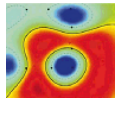


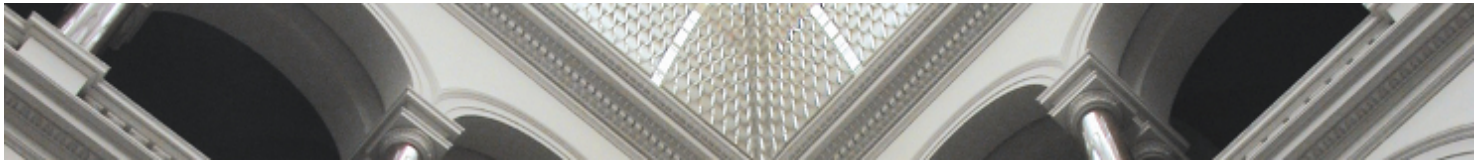
Technische Universität Berlin

Linear Regression

To minimize the least-squares loss function in eq. 2

$$\begin{aligned} \mathcal{E}_{lsq}(\mathbf{w}) &= \sum_{t=1}^{T} (y_t - \mathbf{w}^\top X_t)^2 \\ &= \|y - \mathbf{w}^\top X\|^2 \\ &= yy^\top - 2\mathbf{w}^\top Xy^\top + \mathbf{w}^\top XX^\top \mathbf{w} \end{aligned}$$





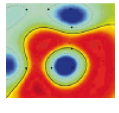
Technische Universität Berlin

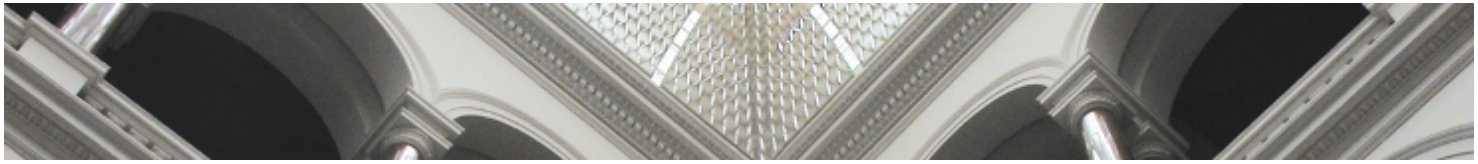
Linear Regression

To minimize the least-squares loss function in eq. 2

$$\begin{aligned} \mathcal{E}_{lsq}(\mathbf{w}) &= \sum_{t=1}^{T} (y_t - \mathbf{w}^\top X_t)^2 \\ &= \|y - \mathbf{w}^\top X\|^2 \\ &= yy^\top - 2\mathbf{w}^\top Xy^\top + \mathbf{w}^\top XX^\top \mathbf{w} \end{aligned}$$

We compute derivative w.r.t. $\mathbf{w}$





Technische Universität Berlin

Linear Regression

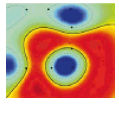
To minimize the least-squares loss function in eq. 2

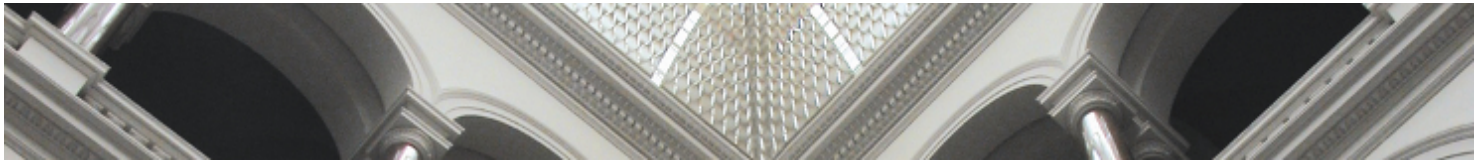
$$\begin{aligned}\mathcal{E}_{lsq}(\mathbf{w}) &= \sum_{t=1}^{T}(y_t - \mathbf{w}^\top X_t)^2 \\ &= \|y - \mathbf{w}^\top X\|^2 \\ &= yy^\top - 2\mathbf{w}^\top Xy^\top + \mathbf{w}^\top XX^\top \mathbf{w}\end{aligned}$$

We compute derivative w.r.t. $\mathbf{w}$

$$\frac{\partial \mathcal{E}_{lsq}(\mathbf{w})}{\partial \mathbf{w}} = -2Xy^\top + 2XX^\top \mathbf{w}$$

set it to zero and solve for $\mathbf{w}$





Technische Universität Berlin

Linear Regression

To minimize the least-squares loss function in eq. 2

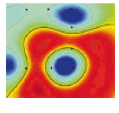
$$\begin{aligned}\mathcal{E}_{lsq}(\mathbf{w}) &= \sum_{t=1}^{T}(y_t - \mathbf{w}^\top X_t)^2 \\ &= \|y - \mathbf{w}^\top X\|^2 \\ &= yy^\top - 2\mathbf{w}^\top Xy^\top + \mathbf{w}^\top XX^\top \mathbf{w}\end{aligned}$$

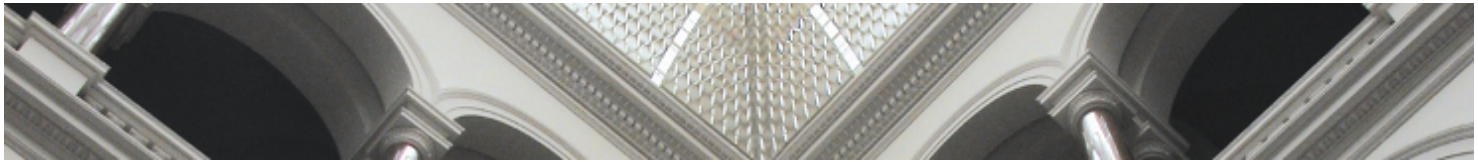
We compute derivative w.r.t. $\mathbf{w}$

$$\frac{\partial \mathcal{E}_{lsq}(\mathbf{w})}{\partial \mathbf{w}} = -2Xy^\top + 2XX^\top \mathbf{w}$$

set it to zero and solve for $\mathbf{w}$

$$\begin{aligned}-2Xy^\top + 2XX^\top \mathbf{w} &= 0 \\ XX^\top \mathbf{w} &= Xy^\top \\ \mathbf{w} &= (XX^\top)^{-1} Xy^\top\end{aligned} \tag{3}$$



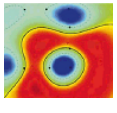


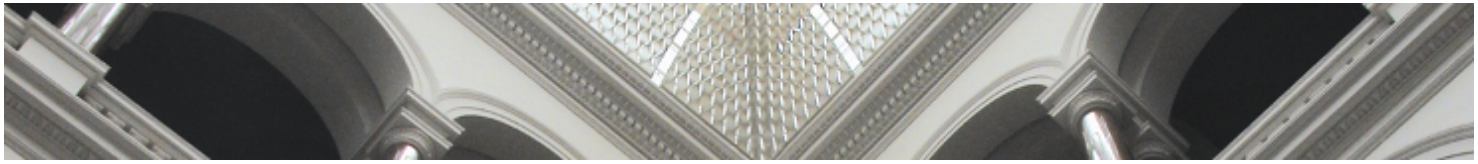
Technische Universität Berlin

Linear Regression

For a new data point $\mathbf{z} \in \mathbb{R}^D$, we have

$$\hat{\mathbf{y}} = \mathbf{w}^T \cdot \mathbf{z} = \left(XX^\top\right)^{-1} Xy^\top \cdot \mathbf{z}$$





Technische Universität Berlin

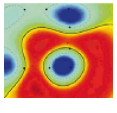
Linear Regression for Vector Labels

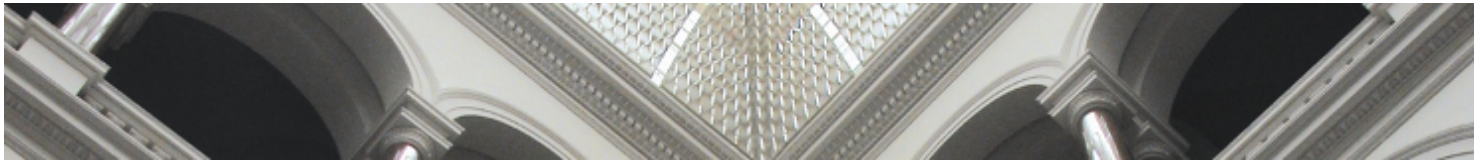
We now want to predict vector-valued labels $y \in \mathbb{R}^M$

For a measurement $X \in \mathbb{R}^{D \times T}$, $Y \in \mathbb{R}^{M \times T}$ the model is

$$Y = W^\top X$$

where $W^\top \in \mathbb{R}^{M \times D}$ is a **linear mapping** from data to labels.





Technische Universität Berlin

Linear Regression for Vector Labels

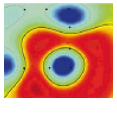
Given Data $X \in \mathbb{R}^{D \times T}$ and labels $Y \in \mathbb{R}^{M \times T}$, the error function for multiple linear regression is

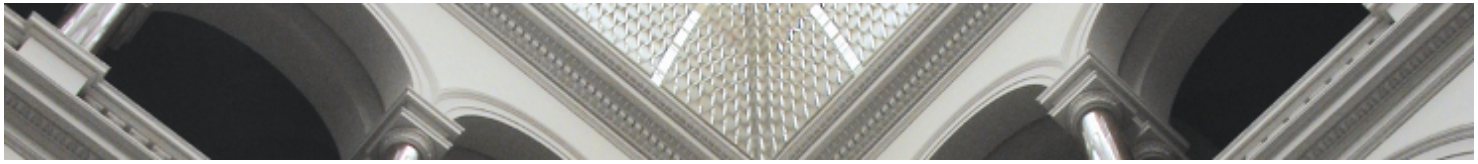
$$\mathcal{E}_{MLR}(W) = ||Y - W^\top X||_F \tag{4}$$

where $||.||_F$ denotes the Frobenius norm

Eq. 4 is minimized by (see also eq. 3)

$$W = (XX^\top)^{-1} XY^\top$$





Technische Universität Berlin

Agenda

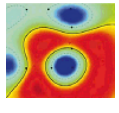
Covariance & Correlation

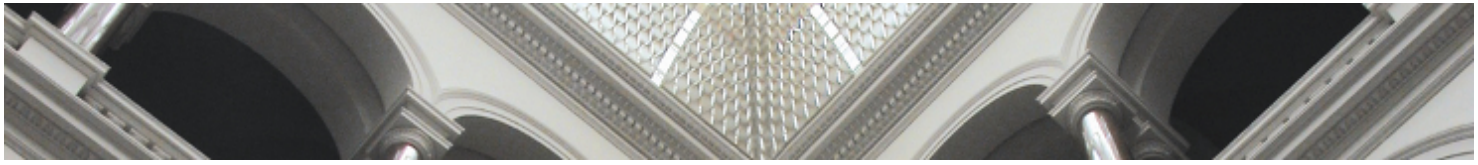
Linear Regression

Regularization, Ridge & Lasso

Logistic Regression

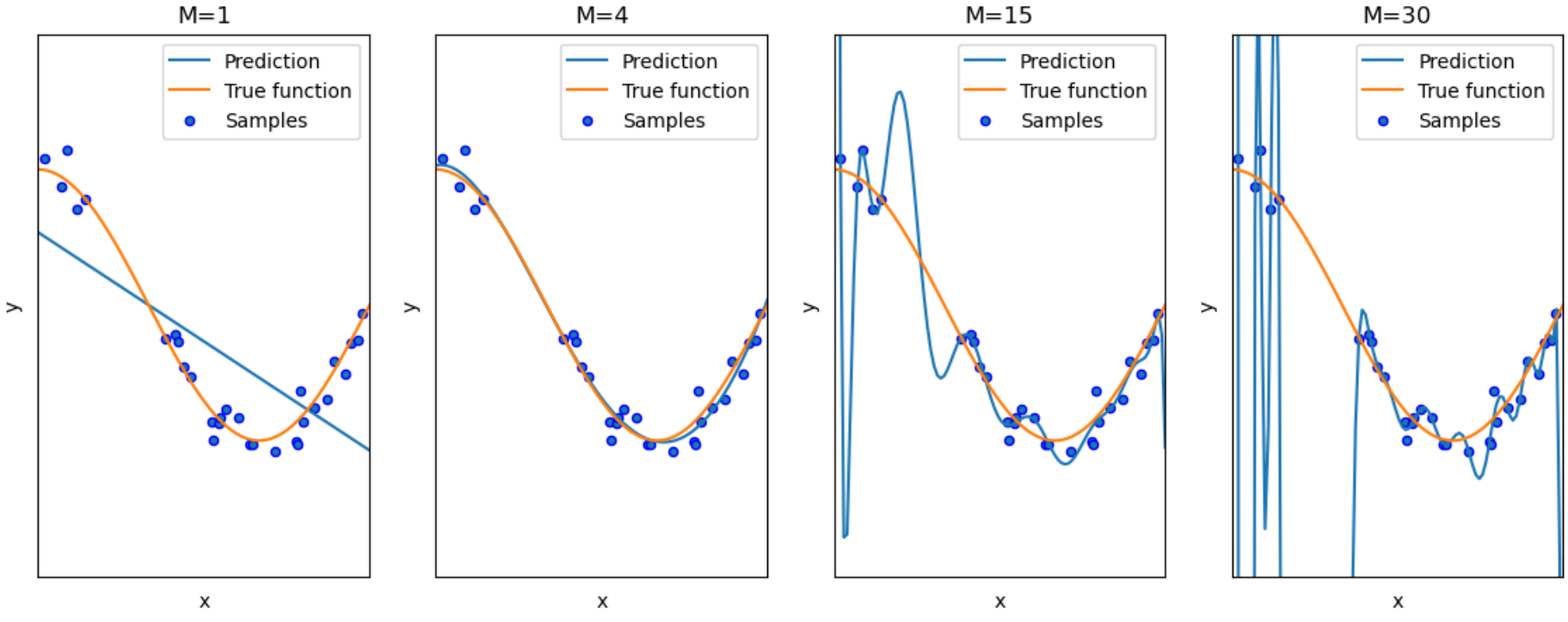
More Methods & Practical Considerations



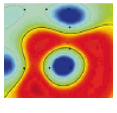


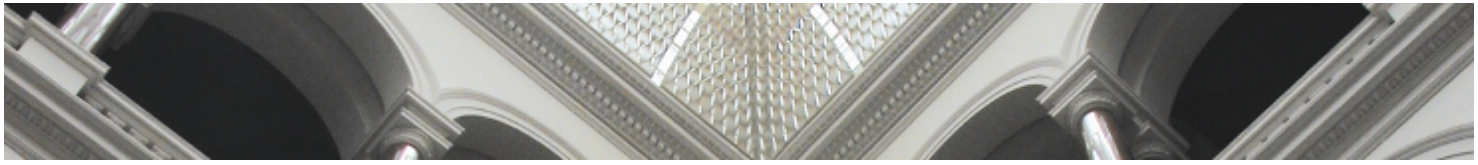
Technische Universität Berlin

Example: Polynomial Regression



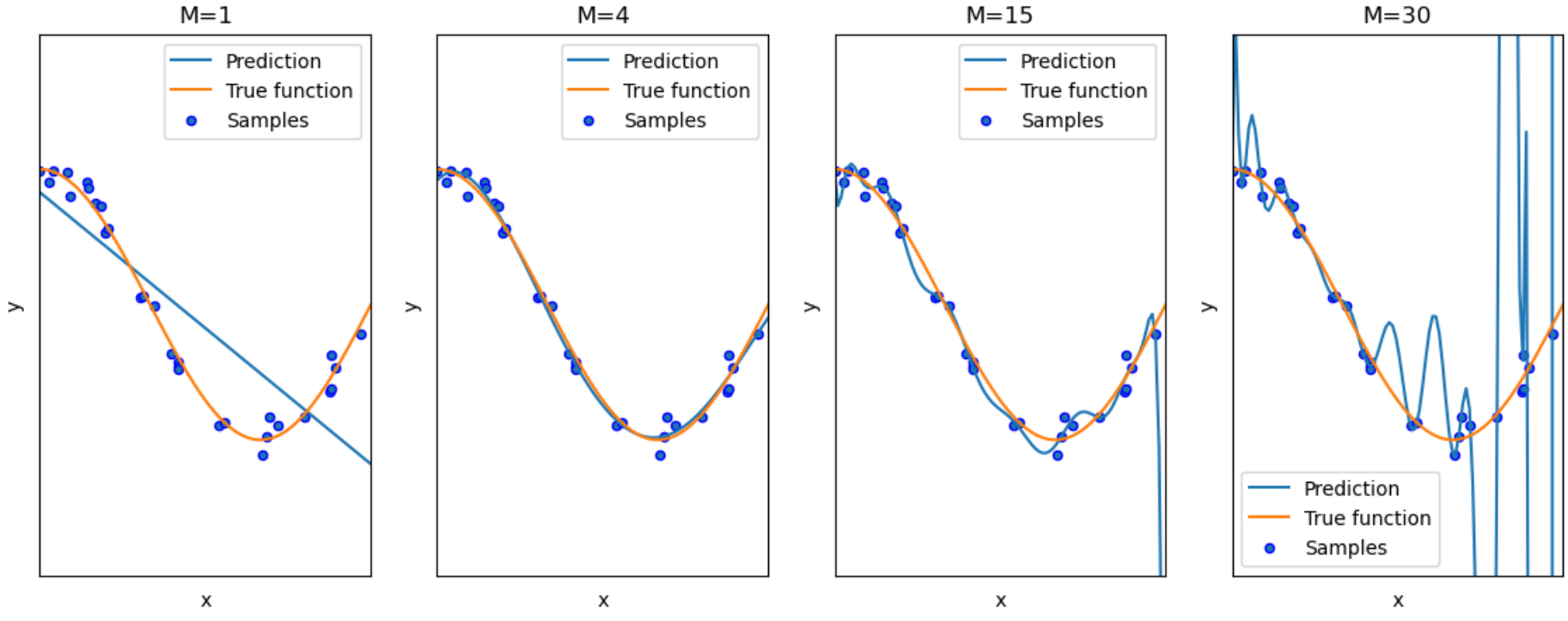
$$f(x) = w_0 + w_1 \cdot x^1 + \ldots + w_M \cdot x^M$$



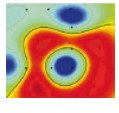


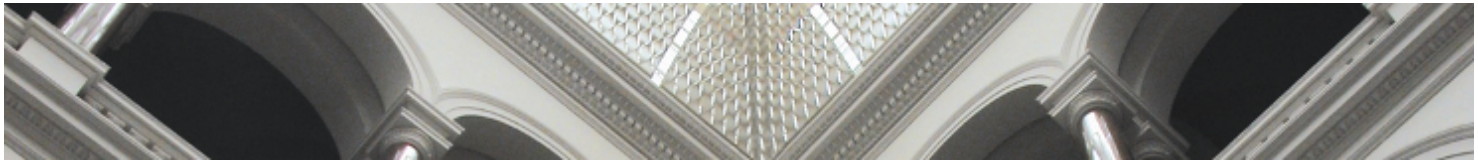
Technische Universität Berlin

Example: Polynomial Regression (2)



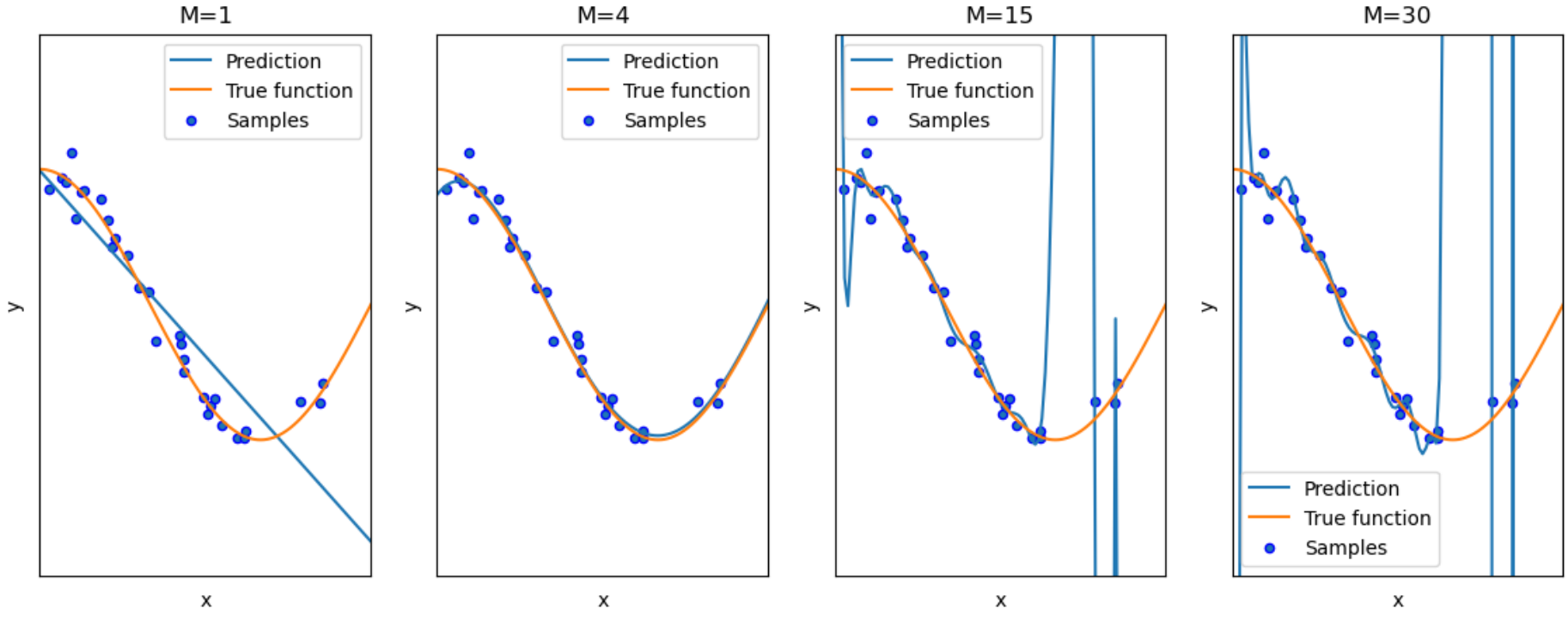
$$f(x) = w_0 + w_1 \cdot x^1 + \ldots + w_M \cdot x^M$$



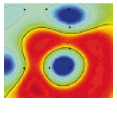


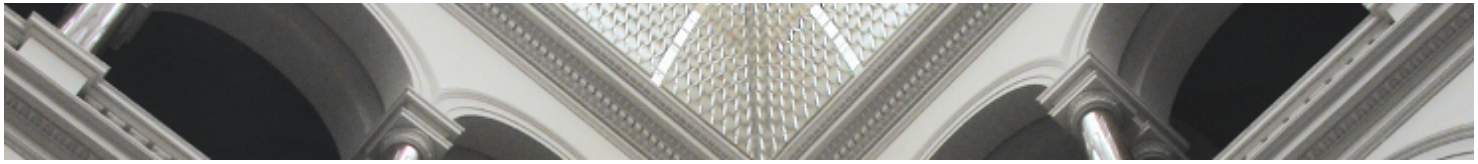
Technische Universität Berlin

Example: Polynomial Regression (3)



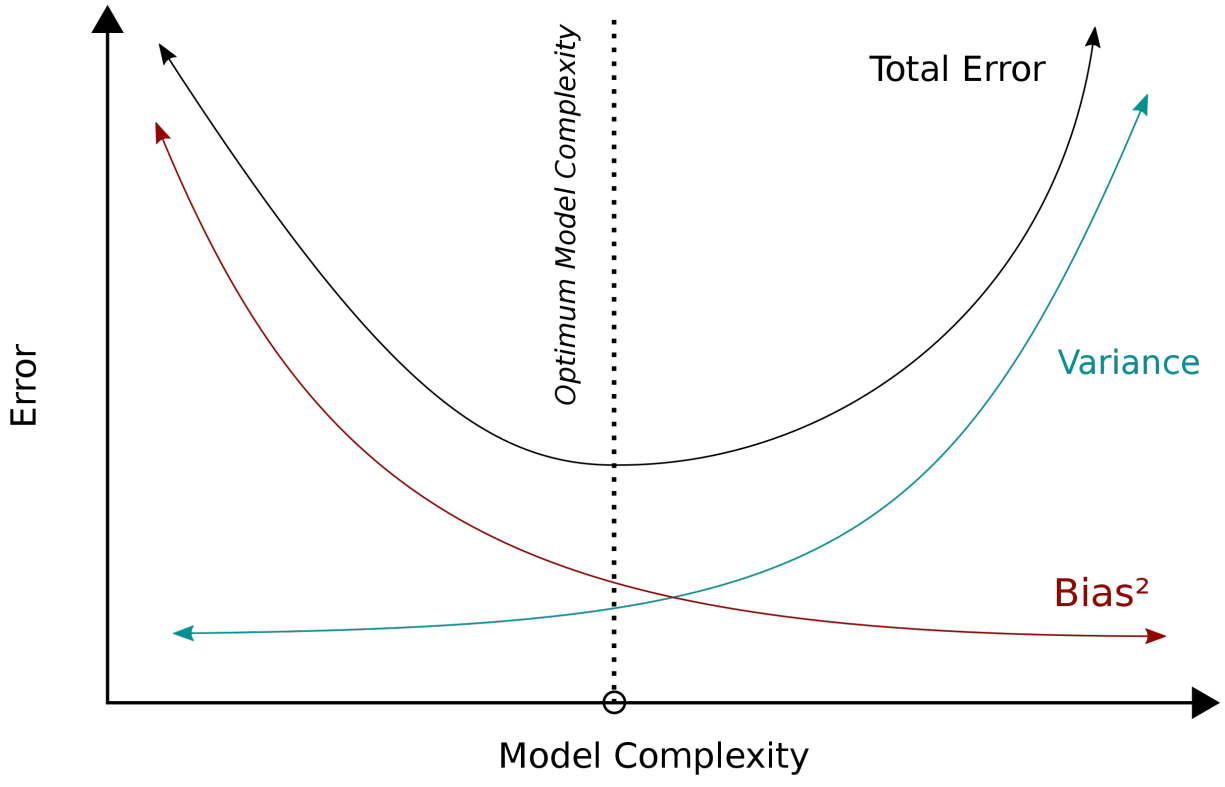
$$f(x) = w_0 + w_1 \cdot x^1 + \ldots + w_M \cdot x^M$$

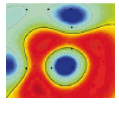


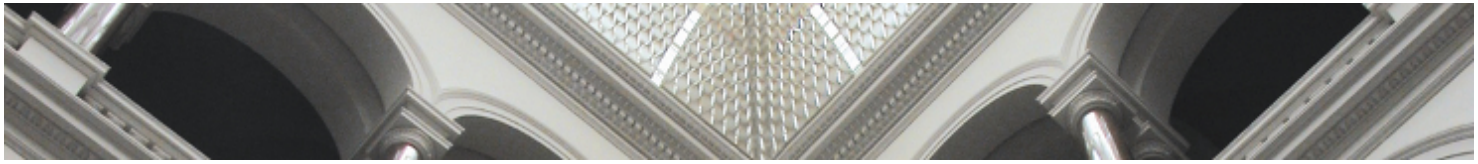


Technische Universität Berlin

The Bias-Variance Tradeoff







Technische Universität Berlin

A Statistical Perspective on Linear Regression

Linear Model:

$$y = \mathbf{w}^\top \cdot x + \epsilon$$

Linear Regression: estimates

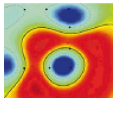
$$\hat{\mathbf{w}} = (XX^\top)^{-1}Xy$$

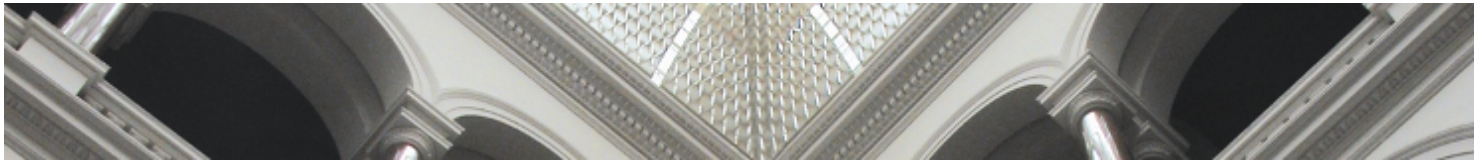
from given data X, y.

$\hat{\mathbf{w}}$ is a function of the data and thus itself a random variable

Desirable properties:

- **Unbiased**: The estimator's expected value is the true function.
- **Small variance**
- **Robust**: Not overly affected by outliers.

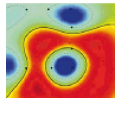


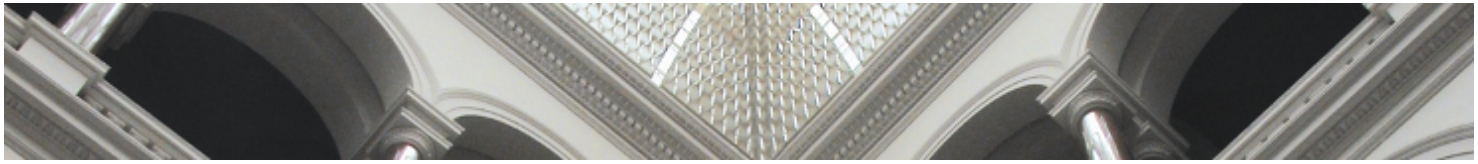


Gauss-Markov-Theorem

Good news:

Under the model assumption $y = \mathbf{w}^\top \cdot x + \epsilon$ with uncorrelated noise ϵ, our ordinary least squares estimator $\hat{\mathbf{w}} = (XX^\top)^{-1}Xy$ is the Best Linear Unbiased Estimator (BLUE), i.e. the minimum variance unbiased estimator that is linear in the y.



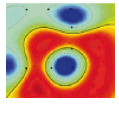


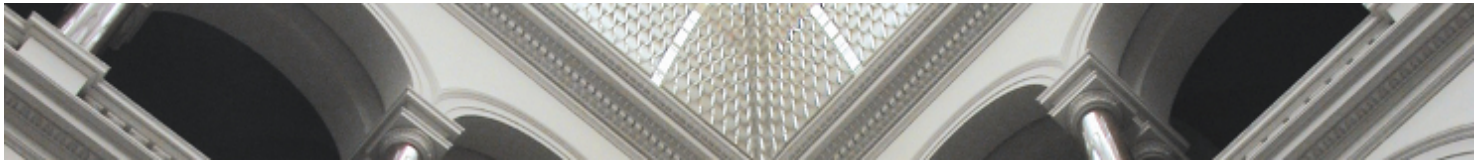
Gauss-Markov-Theorem

Good news:

Under the model assumption $y = \mathbf{w}^\top \cdot x + \epsilon$ with uncorrelated noise ϵ, our ordinary least squares estimator $\hat{\mathbf{w}} = (XX^\top)^{-1}Xy$ is the Best Linear Unbiased Estimator (BLUE), i.e. the minimum variance unbiased estimator that is linear in the y.

But: in some cases biased estimators with lower variance might be more suitable





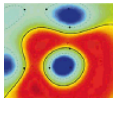
Technische Universität Berlin

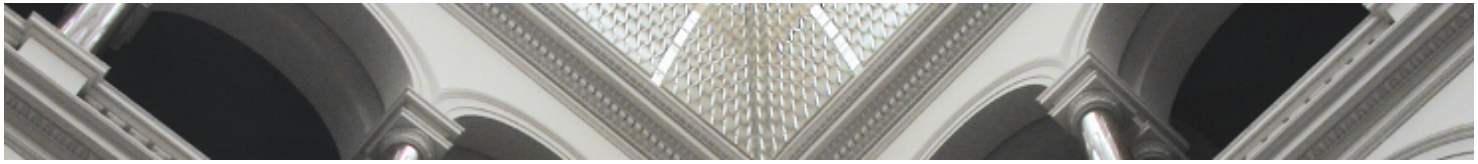
Ridge Regression

Often it is important to **control the complexity** the solution **w**.

This is done by constraining the norm of **w**,

$$\mathcal{E}_{RR}(\mathbf{w}) = ||y - \mathbf{w}^\top X||^2 + \lambda ||\mathbf{w}||^2$$





Ridge Regression

Often it is important to **control the complexity** the solution **w**.

This is done by constraining the norm of **w**,

$$\mathcal{E}_{RR}(\mathbf{w}) = ||y - \mathbf{w}^\top X||^2 + \lambda ||\mathbf{w}||^2$$

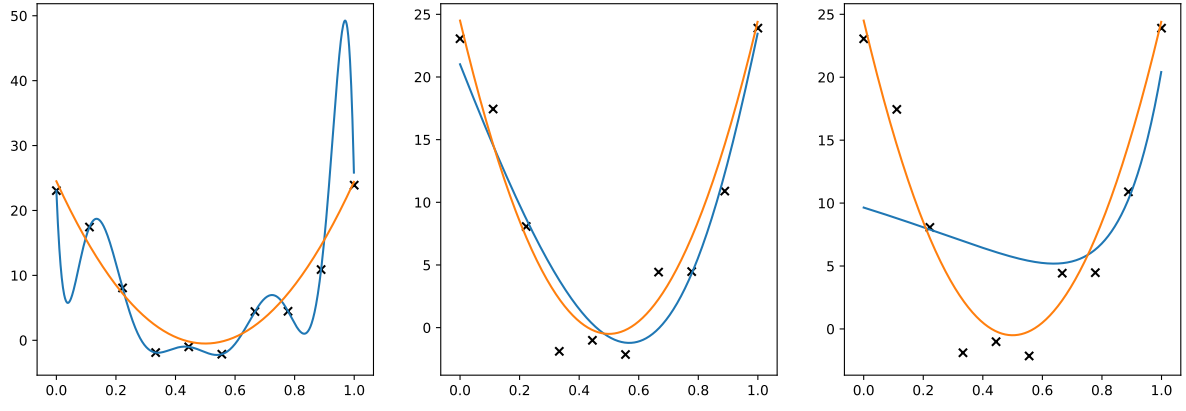
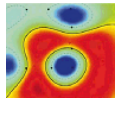
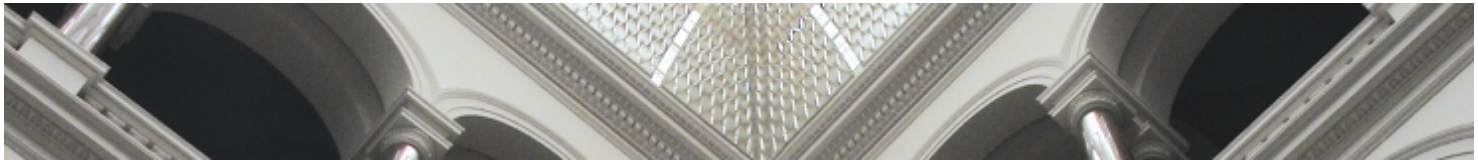


Figure: Orange: true function, blue: predicted function. Left: $\lambda = 0$, center: $\lambda = 0.01$, right: $\lambda = 1.0$





Technische Universität Berlin

Ridge Regression

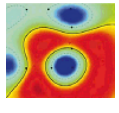
Computing the derivative w.r.t. w yields

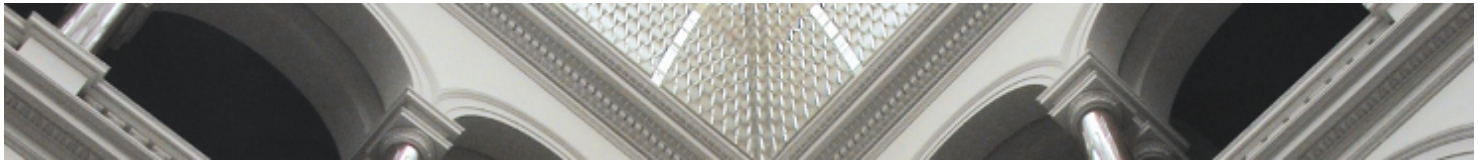
$$\frac{\partial \mathcal{E}_{RR}(\mathbf{w})}{\partial \mathbf{w}} = -2Xy^\top + 2XX^\top \mathbf{w} + \lambda 2\mathbf{w}.$$

Setting the gradient to zero and rearranging terms the optimal $\mathbf{w}$ is

$$\begin{aligned} 2XX^\top \mathbf{w} + \lambda 2\mathbf{w} &= 2Xy^\top \\ (XX^\top + \lambda I)\mathbf{w} &= Xy^\top \\ \mathbf{w} &= (XX^\top + \lambda I)^{-1} Xy^\top \end{aligned}$$

$\Rightarrow$ Biased estimator, but smaller variance





Lasso Regression

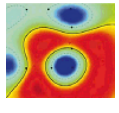
Ridge regression loss:

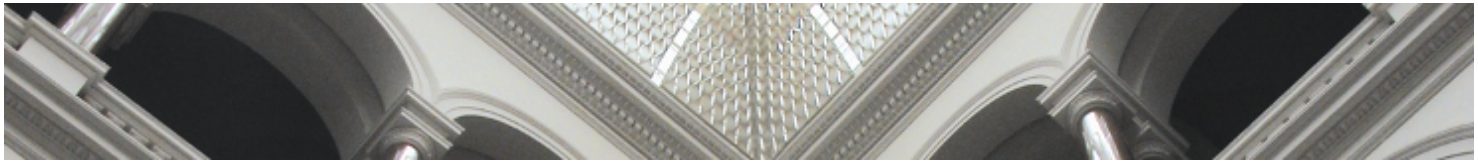
$$\mathcal{E}_{RR}(\mathbf{w}) = ||y - \mathbf{w}^\top X||^2 + \lambda ||\mathbf{w}||^2$$

Potential issue: Ridge regression will almost never set a coefficient w to exactly zero.

Therefore, it cannot perform **feature selection** in the linear model.

Variable selection can be desirable for **interpretability** (especially if the number of features is large).





Technische Universität Berlin

Lasso Regression

The **lasso** objective is defined as

$$\mathcal{E}_{lasso}(\mathbf{w}) = ||y - \mathbf{w}^\top X||^2 + \lambda ||\mathbf{w}||_1$$

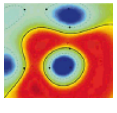
$$\mathcal{E}_{lasso}(\mathbf{w}) = ||y - \mathbf{w}^\top X||^2 + \lambda \sum_{i=1}^{D} |w_i|$$

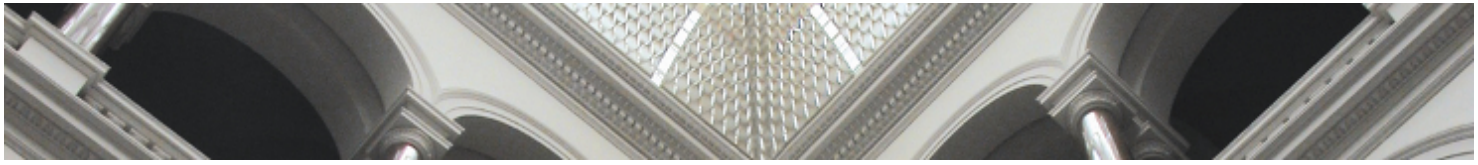
Even though Ridge and Lasso Regression look similar, their solutions behave very differently.

“Lasso”: Least Absolute Selection and Shrinkage Operator.

Generally speaking:

- The bias increases as λ (amount of shrinkage) increases.
- The variance decreases as λ (amount of shrinkage) increases.

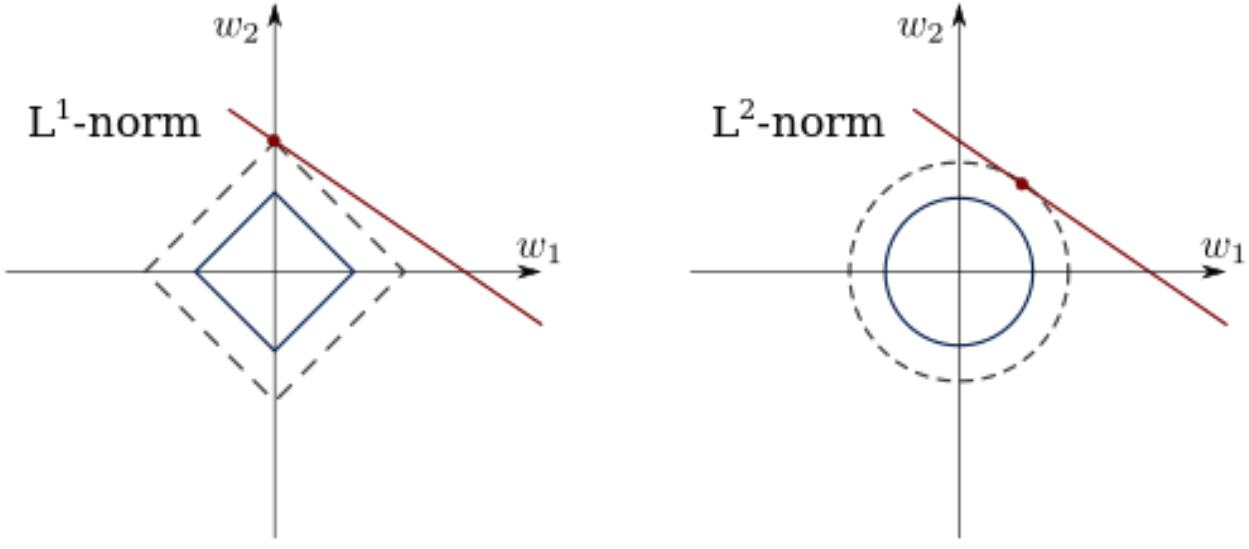




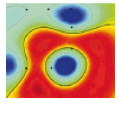
Technische Universität Berlin

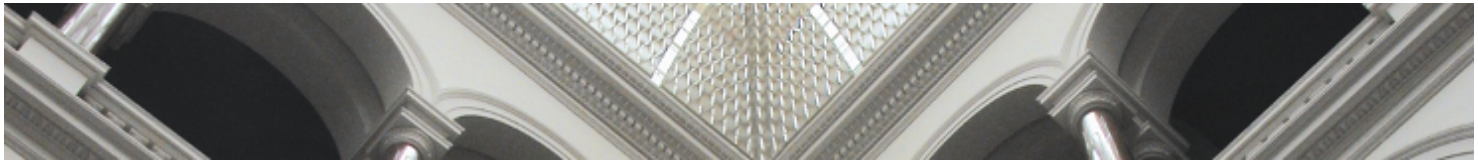
Regularization

Lasso regularization (L^1-norm) leads to sparsity.



Nicoguaro, CC BY 4.0 (https://creativecommons.org/licenses/by/4.0), via Wikimedia Commons





Technische Universität Berlin

Agenda

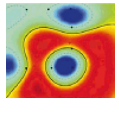
Covariance & Correlation

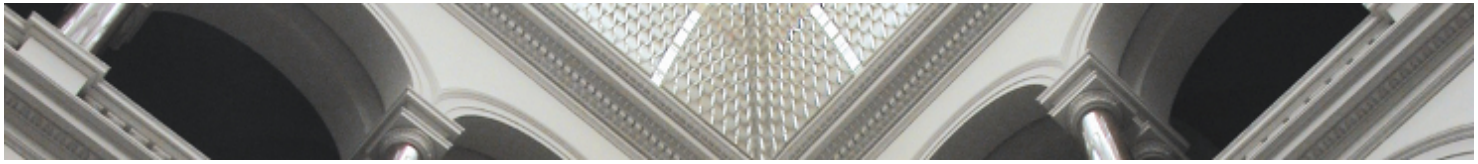
Linear Regression

Regularization, Ridge & Lasso

Logistic Regression

More Methods & Practical Considerations



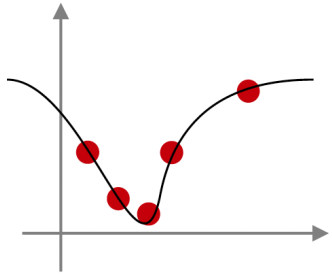


Technische Universität Berlin

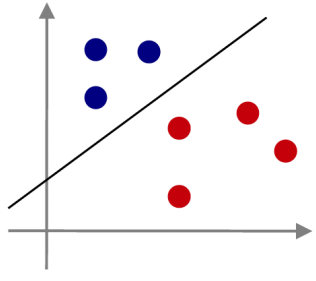
Binary Classification

Given data $x_1, \ldots x_T \in \mathbb{R}$ and labels $y_1, \ldots, y_T \in \{\mathbf{0}, \mathbf{1}\}$, the goal is to predict y from x.

Regression

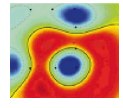


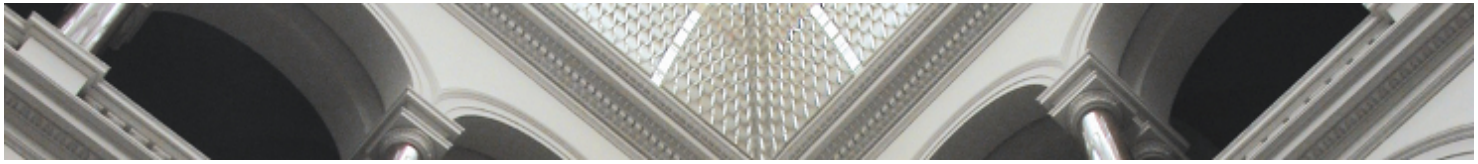
Classification



Examples of binary classification tasks:

- Given various patient data, will the patient **survive** a disease?
- Determine whether an email is **spam** or not.
- ...



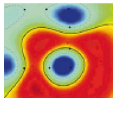


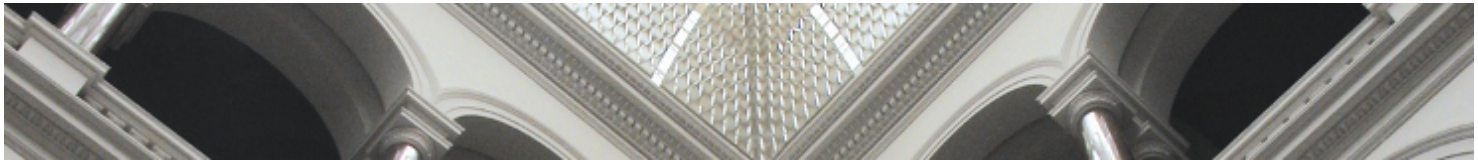
Logistic Regression

Given data $x_1, \ldots x_T \in \mathbb{R}$ and labels $y_1, \ldots, y_T \in \{\mathbf{0}, \mathbf{1}\}$, the goal is to predict y from x.

We previously developed a linear regression model:

$$y = \mathbf{w}^\top x.$$





Logistic Regression

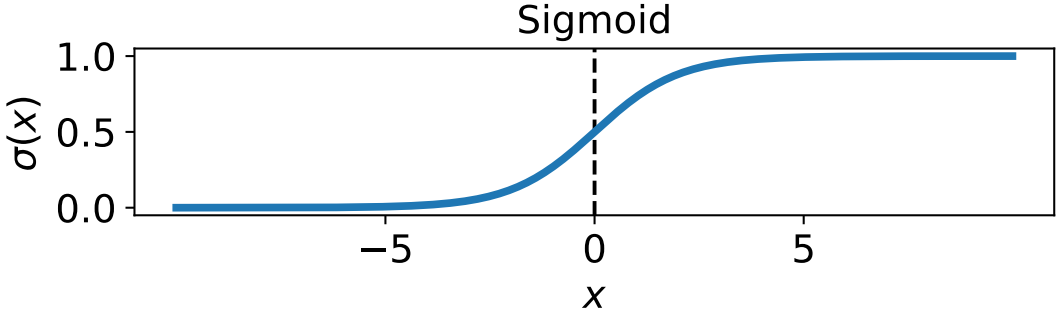
Given data $x_1, \ldots x_T \in \mathbb{R}$ and labels $y_1, \ldots, y_T \in \{\mathbf{0}, \mathbf{1}\}$, the goal is to predict y from x.

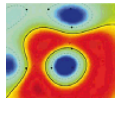
We previously developed a linear regression model:

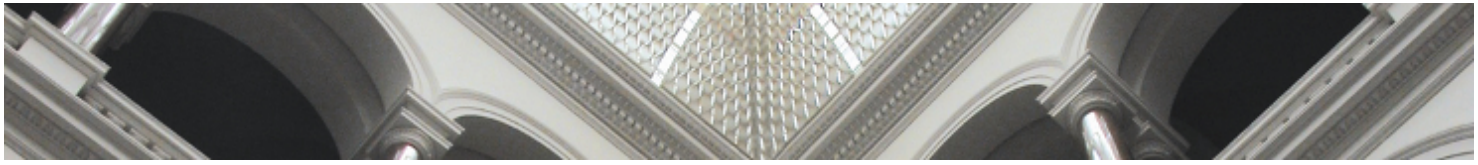
$$y = \mathbf{w}^\top X.$$

Instead, we can apply a simple adaptation: ”linear regression model + logit function”.

$$y = \sigma(\mathbf{w}^\top X), \text{ with } \sigma(x) = \frac{1}{1 + exp(-x)}$$



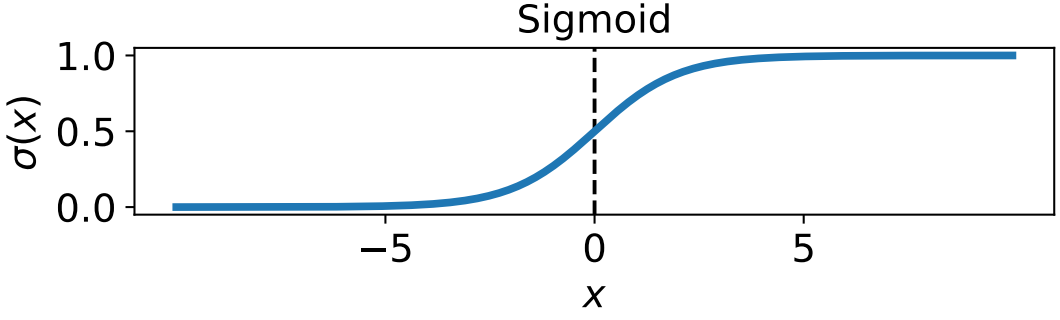




Technische Universität Berlin

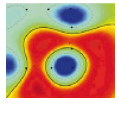
Logistic Regression

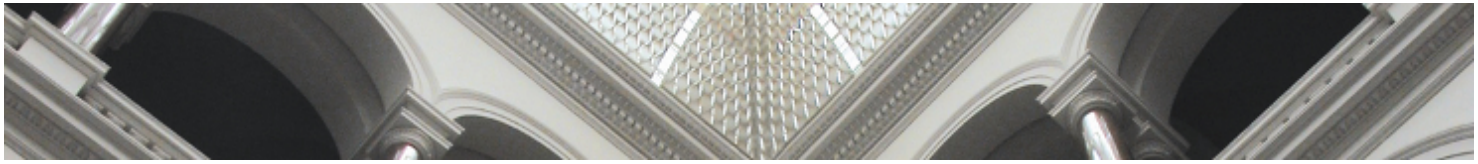
$$y = \sigma(\mathbf{w}^\top X), \text{ with } \sigma(x) = \frac{1}{1 + exp(-x)}$$



Characteristics

- We can interpret the prediction as the probability (or odds) of an event.
- We can set a threshold to make binary predictions, e.g., iff $\sigma(\mathbf{w}^\top X) > 0.5$, then the sample class is predicted to be 1 / positive.
- We can adapt the method to predict categorical variables with more than two classes (how?).





Technische Universität Berlin

Agenda

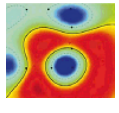
Covariance & Correlation

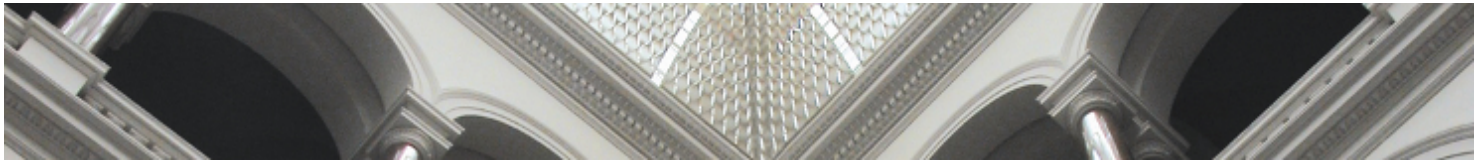
Linear Regression

Regularization, Ridge & Lasso

Logistic Regression

More Methods & Practical Considerations





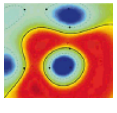
Many More Methods...

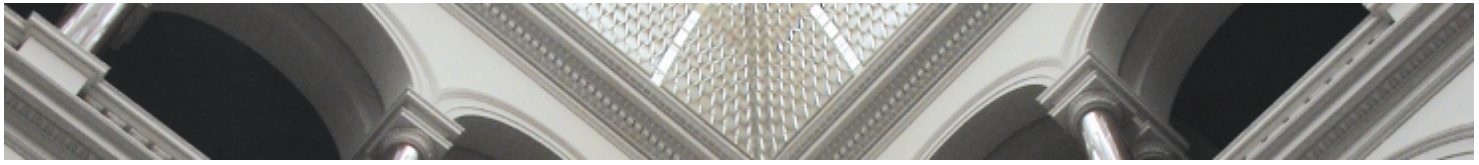
Many more classical machine learning methods exist.

Well-established implementations and documentation are available in popular Python packages.

Prominent Examples

- Nearest Neighbors & Nearest Centroid
- Linear Discriminant Analysis (LDA)
- Decision Tree / Random Forest
- Gradient Boosting
- Support Vector Machine (SVM)
- Principal Component Analysis (PCA)
- ...





Technische Universität Berlin

Software Packages

scikit learn Install User Guide API Examples Community More Go

scikit-learn

Machine Learning in Python

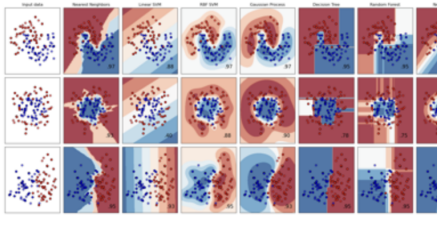
Getting Started | Release Highlights for 1.4 | GitHub

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: Gradient boosting, nearest neighbors, random forest, logistic regression, and more...

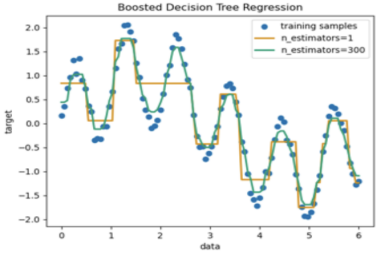


Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: Gradient boosting, nearest neighbors, random forest, ridge, and more...

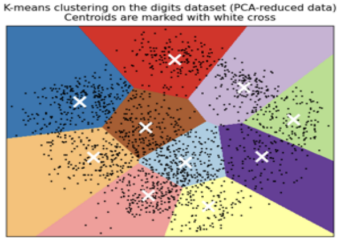


Examples

Clustering

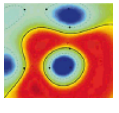
Automatic grouping of similar objects into sets.

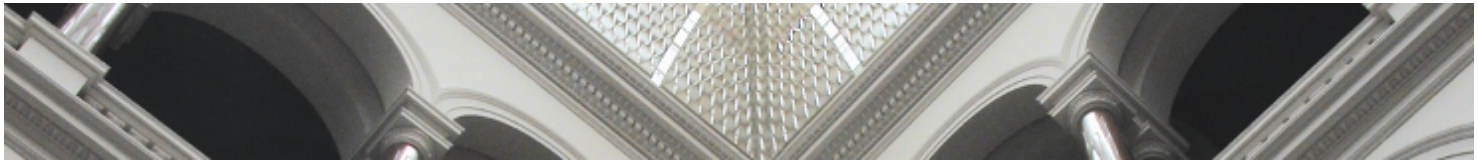
Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, HDBSCAN, hierarchical clustering, and more...



Examples

`https://scikit-learn.org`, accessed 10/02/2024.





Technische Universität Berlin

Thank you!