JÜLICH
FORSCHUNGSZENTRUM

Machine Learning and Material Science
1. Linear Regression

Steffen Brinckmann, Claas Hüter
IEK-2, Forschungszentrum Jülich GmbH

Linear Regression

Studying dependence of **one response variable** on one or more feature variables.

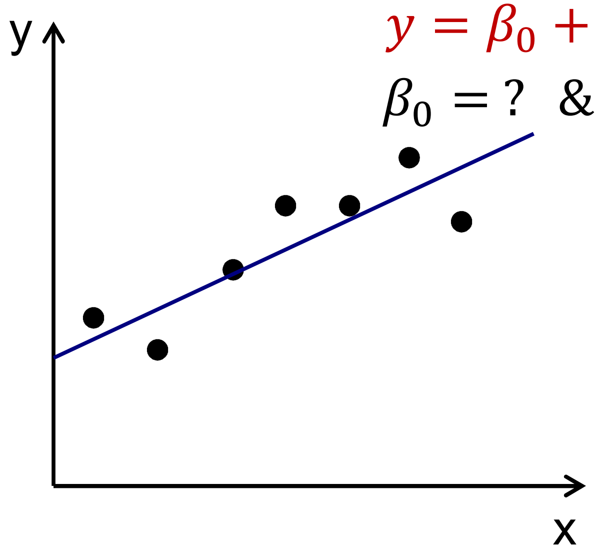
$$\boldsymbol{y} = \boldsymbol{\beta_0} + \boldsymbol{\beta_1 x_1} + \cdots + \boldsymbol{\beta_n x_n}$$

$\boldsymbol{y}$: response / target

$\boldsymbol{x_n}$: feature variables

$\boldsymbol{\beta_n}$: model coefficients which are learnt / trained

Simple Linear Regression: predict a line



$y = \beta_0 + \beta_1 \mathrm{x}$

$\beta_0 = ?$ & $\beta_1 = ?$

Ordinary Least Square error

$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

$$\sum_{i=1}^{n}(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2$$

By solving least square problem

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}, \qquad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

3

Linear Regression of multiple parameters: plane

$$\boldsymbol{y} = \boldsymbol{\beta_0} + \boldsymbol{\beta_1 x_1} + \cdots + \boldsymbol{\beta_n x_n}$$

$\boldsymbol{y}$: response / target

$\boldsymbol{x_n}$: feature variables

$\boldsymbol{\beta_n}$: model coefficients which are learnt / trained

Rewrite as: $\boldsymbol{y} = \boldsymbol{X\beta}$

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \qquad \boldsymbol{X} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ x_{1,1} & x_{1,2} & & x_{1,p} \\ \vdots & \vdots & & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{pmatrix} \qquad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_n \end{pmatrix}$$

Solve by:

- Matrix inversion & multiplication
- $Error = \sum_{i=1}^{n} \|\boldsymbol{y} - \widehat{\boldsymbol{y}}\|^2 = \sum_{i=1}^{n} \left\|\boldsymbol{y} - \widehat{\boldsymbol{X}}\widehat{\boldsymbol{\beta}}\ \right\|^2$

Linear Regression of multiple parameters: plane

Solve by:

- Matrix inversion & multiplication
  (inversion too expensive)

- $Error = \sum_{i=1}^{n} \left\| \boldsymbol{Y} - \widehat{\boldsymbol{Y}} \right\|^2 = \sum_{i=1}^{n} \left\| \boldsymbol{Y} - \widehat{\boldsymbol{X}}\widehat{\boldsymbol{\beta}} \right\|^2$

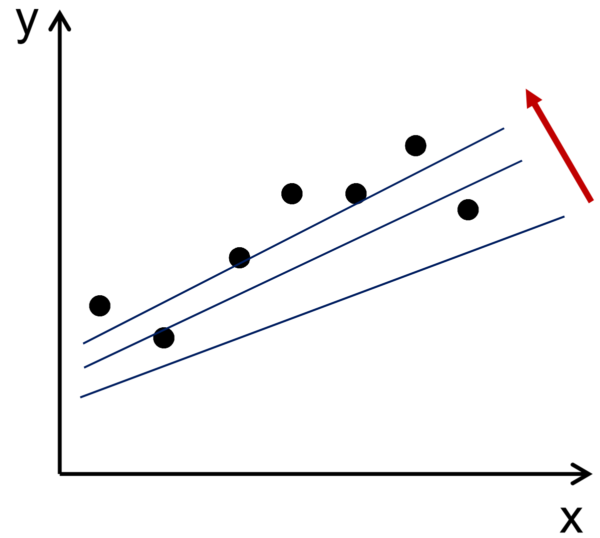
  $\text{Minimum of error} \rightarrow \frac{\boldsymbol{\partial Error}}{\boldsymbol{\partial \beta_i}} = \boldsymbol{0}$

  - standard least squares solver (for few variables)
  - (stochastic) gradient descent (generally)

Gradient descent method

also used Artificial Neural Network

Predicted function: $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i,1} + \cdots + \hat{\beta}_n x_{i,n}$



Error model or cost function:

- mean square error: $J(\hat{\beta}) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$
- relative mean square error
- …

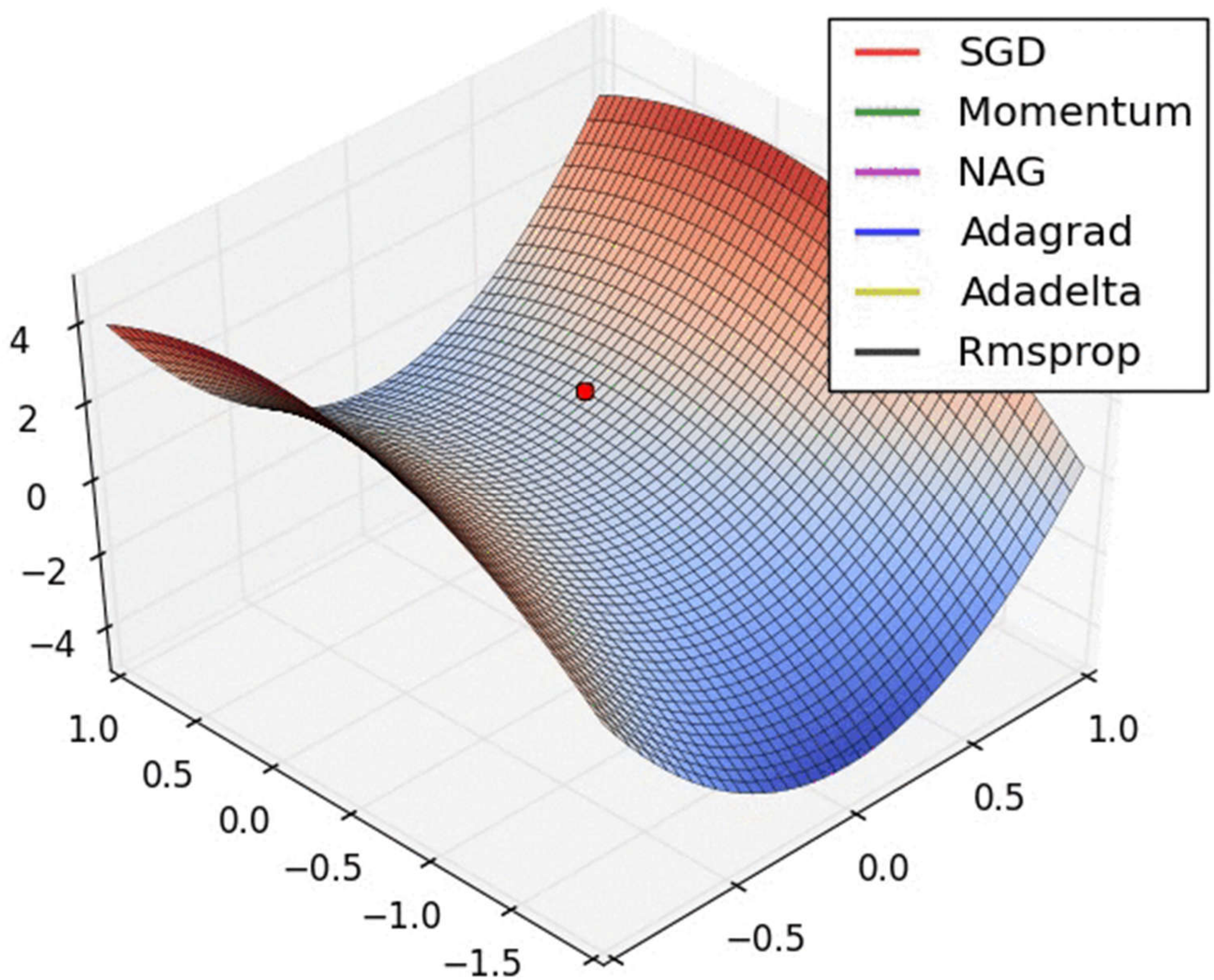
Search algorithm

- starts with 'initial guess' for $\hat{\beta}_i$
- repeatedly change $\hat{\beta}_i$ to minimize Jacobian $J(\hat{\beta})$

  $\hat{\beta}_i \coloneqq \hat{\beta}_i - \alpha \frac{\partial J(\hat{\beta})}{\partial \hat{\beta}_i}$ α: learning rate = how fast $\hat{\beta}_i$ changes **(DANGER)**

- stop when convergence

Gradient descent methods



Stanfort course

Linear Regression: ONE training vector

Ordinary Square Error: $J(\hat{\beta}) = (y - \hat{y})^2$

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_j x_j$$

$$\hat{\beta}_j \ := \hat{\beta}_j - \alpha \frac{\partial J(\hat{\beta})}{\partial \hat{\beta}_j} = \hat{\beta}_j + 2\alpha(y - \hat{y})x_j$$

magnitude of update is proportional to **error** $(y - \hat{y})$

If predicted $\hat{y}$ is close to target y,
there is little change in parameters

Linear Regression: multiple N training vectors

$$\hat{\beta}_j := \hat{\beta}_j + 2\alpha \sum_{i=1}^{N} (y_i - \hat{y}_i) x_{i,j}$$

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i,1} + \cdots + \hat{\beta}_n x_{i,n}$$

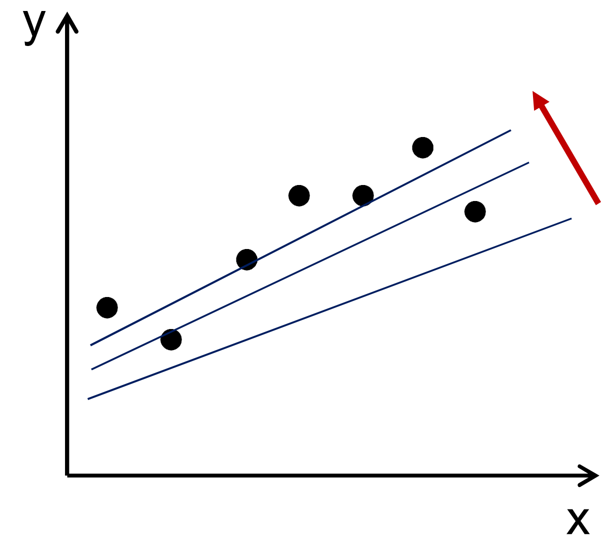
Batch gradient descent:
1. use entire training set
2. update $\hat{\beta}_j$
3. use entire training set

repeat until convergence

Stochastic gradient descent:
1. use random vector (one vector)
2. update $\hat{\beta}_j$
3. use random vector

Repeat until convergence

Error functions: How to determine error?

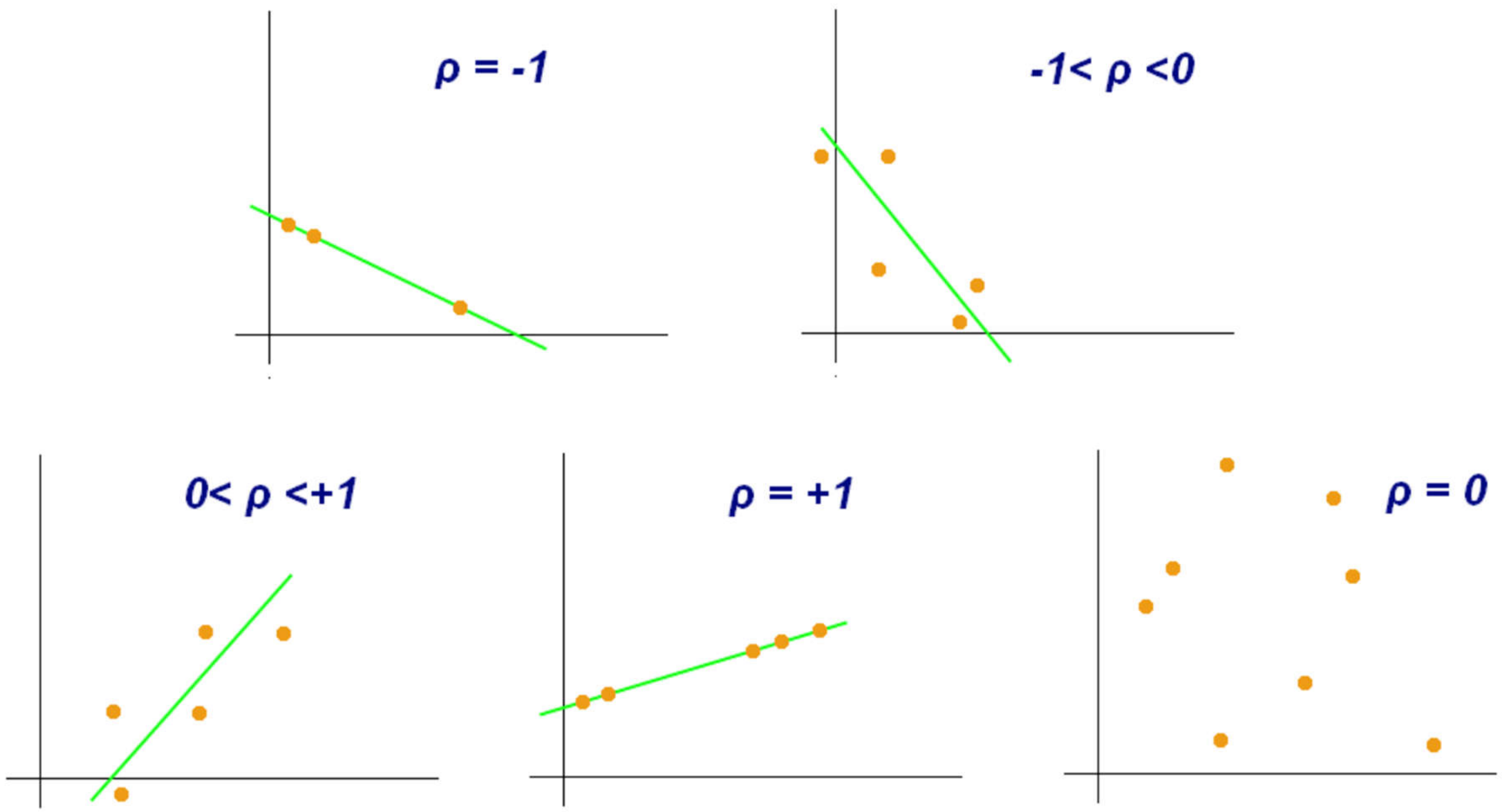


Many error measures exist:

- Ordinary Square Error: $J(\hat{\beta}) = \sum_{i=1}^{m}(y_i - \hat{y}_i)^2$
- Mean Square Error: $J(\hat{\beta}) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$
- Relative Mean Square error: $J(\hat{\beta}) = \frac{1}{m}\sum_{i=1}^{m}\frac{(y_i - \hat{y}_i)^2}{{y_i}^2}$
- Pearson correlation coefficient (be careful)
- R2 (be smart)
- p-value
- ….

Linear fit is good? Pearson correlation coefficient

$$\rho = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2}\sqrt{\sum(y_i - \bar{y})^2}}$$



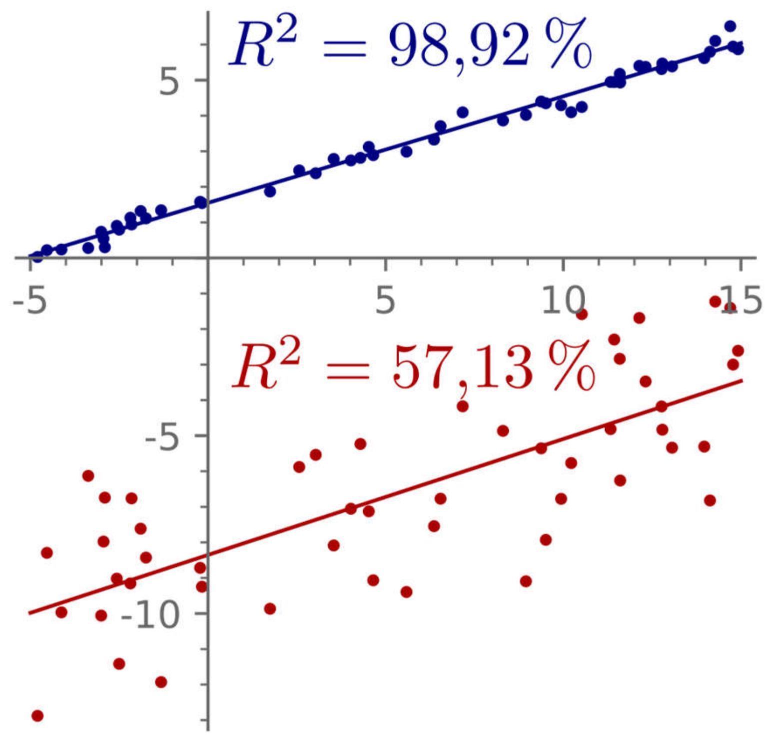
Variance in data explained by variance in model: R^2

Coefficient of determination $R^2 = \rho^2 = \frac{SS_{reg}}{SS_{tot}} = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2}$

Problem: increases with increasing number of parameters (Inflation of R^2)

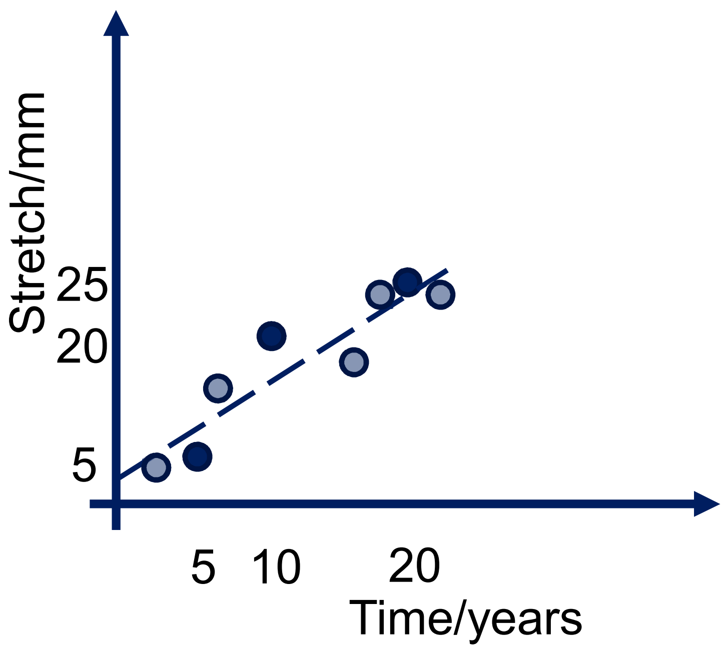
Unbiased R^2 only increases if new parameters have less error

$$\bar{R}^2 = 1 - 1(1 - R^2)\frac{n-1}{n-p-1}$$



wikipedia

How likely are those points measured by accident Significance of data (p-value)



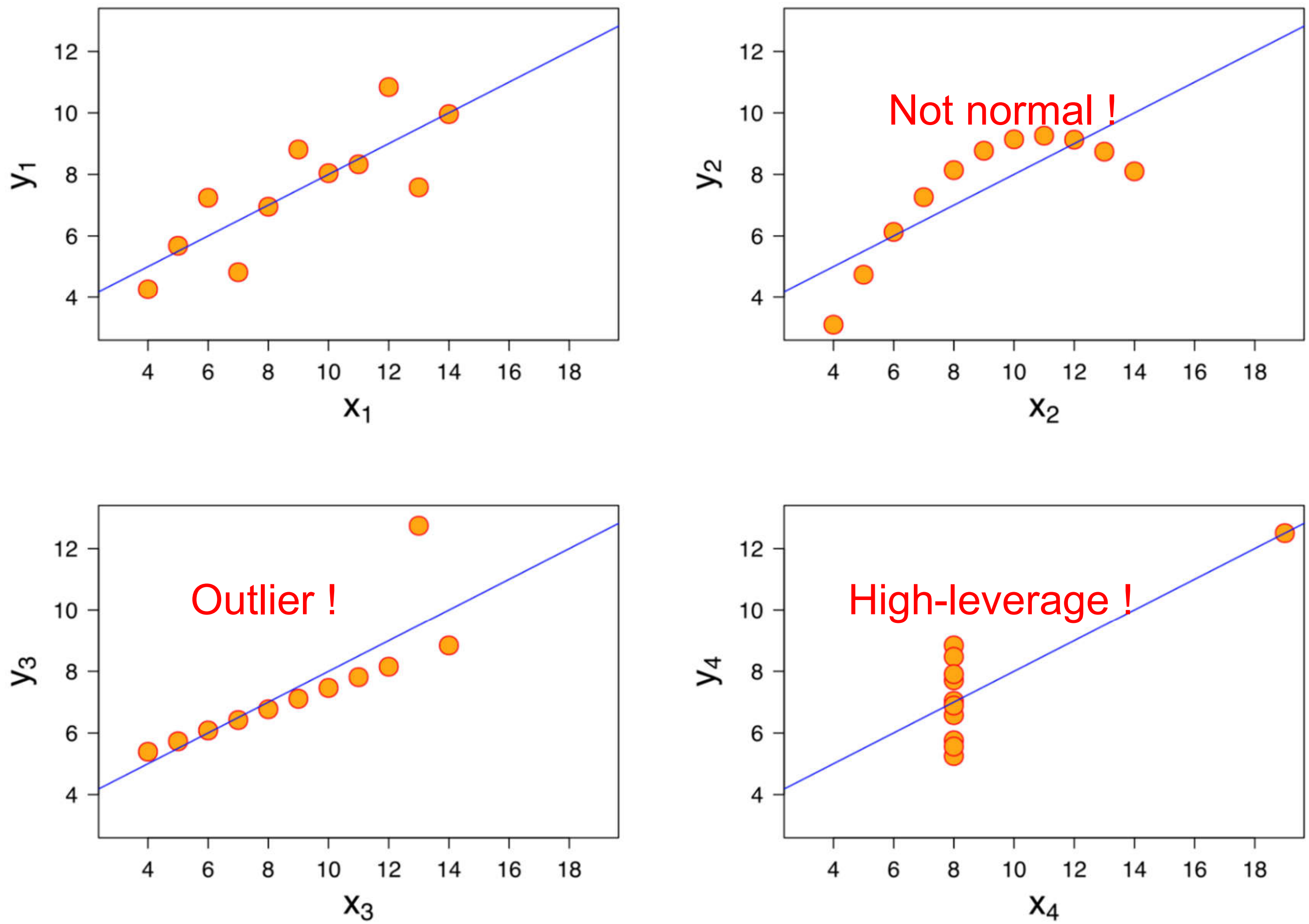
Null hypothesis: there is NO relation between x and y

p-value is probability that Null hypothesis is correct („it is just random noise“)

Goal of machine learning: small p-value
“There is a relation in the data”

Null hypothesis is rejected if $p<0.05 = 5\%$
„If $p<5\%$ one cannot say: random noise“
(Careful of opposite)

Problems: Anscombe‘s Quartet
Identical OLS, identical R^2



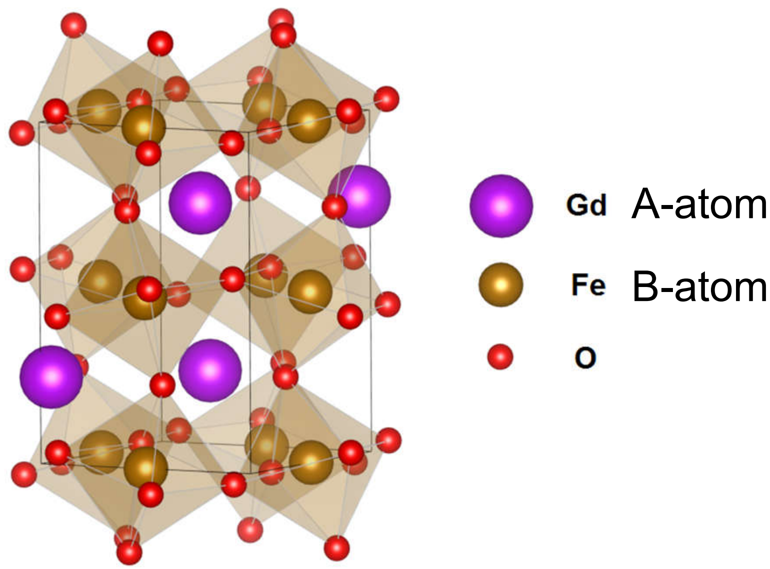
What makes a good linear regression?

- Check residual plots: overlooked systematic trends?

- Outliers / hidden cases

- Large data set – search for reference data

Example: lattice constant of $GdFeO_3$-type perovskite



Orthorhombically distorted perovskite (ABO_3)
($GdFeO_3$-type perovskite)

Predict lattice constant (a, b, c)

$$\boldsymbol{y} = \boldsymbol{\beta_0} + \boldsymbol{\beta_1 x_1} + \cdots + \boldsymbol{\beta_n x_n}$$

Response / Target: $y = a$ or b or c

Features x_n

- Two ionic radii (r_A & r_B)
- combinations

C. Li et al., J. Physics and Chemistry of Solids **64**:2147-2156, 2003.

Example: lattice constant of $GdFeO_3$-type perovskite

Features: $X = (r_A, r_B, t, r_A/t)$

Goldschmidt tolerance factor

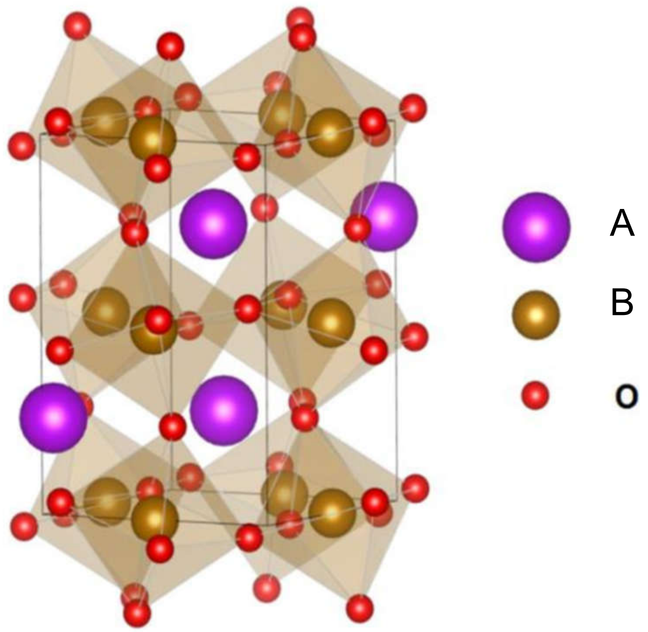
$$t = \frac{r_A + r_O}{\sqrt{2}(r_B + r_O)}$$

Lattice constant (a, b, c) of 161 $GdFeO_3$-type compounds from different databases

Divide into

157 compounds → training data

4 compounds → testing data



C. Li et al., J. Physics and Chemistry of Solids **64**:2147-2156, 2003.

Example: lattice constant of $GdFeO_3$-type perovskite

Linear equation to predict (a, b, c)

$$\boldsymbol{y} = \boldsymbol{\beta_0} + \boldsymbol{\beta_1 x_1} + \cdots + \boldsymbol{\beta_n x_n}$$

Features: $X = (r_A, r_B, t, r_A/t)$

The response: $y = a \text{ or } b \text{ or } c$

$$a = 17.2443 + 7.5013r_A - 3.2537r_B - 17.4019t - 2.1508\, r_A/t, \quad R^2 = 0.8851$$

$$b = -2.9248 - 0.1803r_A + 8.2772r_B + 10.7858t - 2.8652\, r_A/t, \quad R^2 = 0.9029$$

$$c = -1.4013 - 0.2280r_A + 4.2506r_B + 6.3082t - 0.7542\, r_A/t, \quad R^2 = 0.9029$$

R^2: coefficient of determination comparing calculated to actual values

C. Li et al., J. Physics and Chemistry of Solids **64**:2147-2156, 2003.

Example: contact area

Scratch surface with hard tip & measure contact area

Response/Target: contact area

Parameters:

- Radius of hard nanoindenter tip
- Hardness of material
- Normal force during scratching
- Temperature during experiment