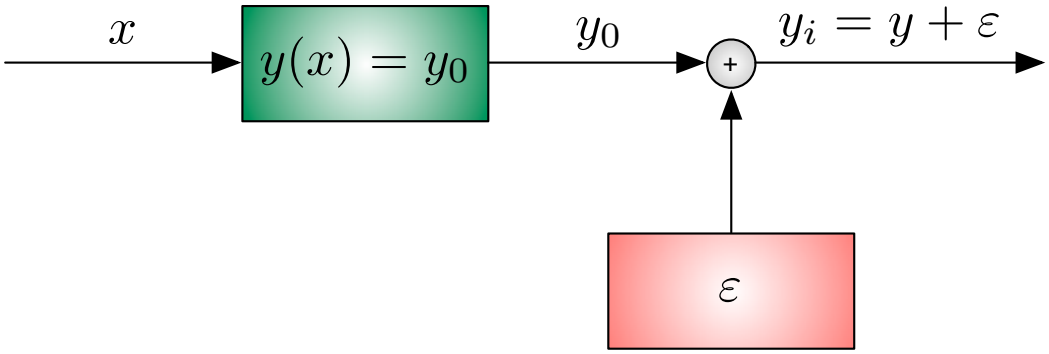
MTAT.03.227 Machine Learning

Linear models and polynomial regression

Sven Laur
University of Tartu

The simplest linear model

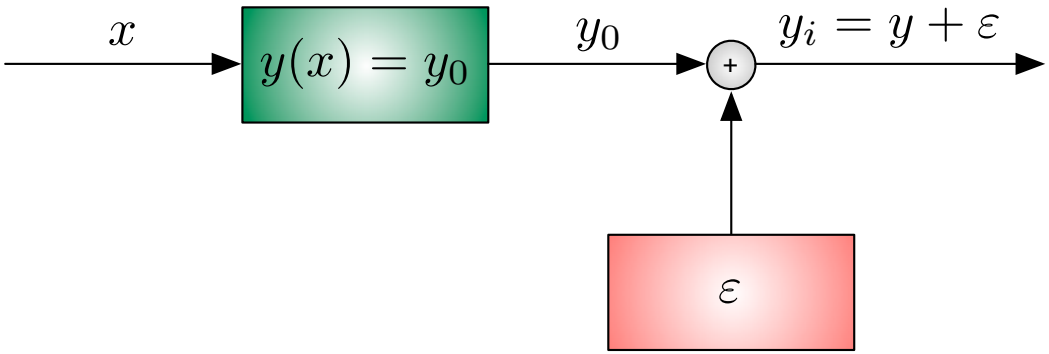


The true output $y(x)$ does not depend on the input. However, the additive noise ε corrupts observable values $y_1, \ldots, y_N$.

System identification task. Given enough observations $y_1, \ldots, y_N$ restore the output value y_0 as precisely as possible.

- ▷ The problem cannot be solved without making strong assumptions ε.
- ▷ Even with strong assumptions it is impossible to give guarantees about estimate $\hat{y}_{N+1}$ which hold with absolute certainty.

The simplest linear model



The true output $y(x)$ does not depend on the input. However, the additive noise ε corrupts observable values $y_1, \ldots, y_N$.

Prediction task. Given enough observations $y_1, \ldots, y_N$ predict the output of the next observation y_{N+1} as precisely as possible.

- ▷ The problem cannot be solved without making strong assumptions ε.
- ▷ Even with strong assumptions it is impossible to give guarantees about estimate $\hat{y}_0$ which hold with absolute certainty.

Solution for system identification

Under the assumption that the noise term is centred around zero, i.e., given enough samples:

$$\frac{1}{N} \cdot \sum_{i=1}^{N} \varepsilon_i \approx 0$$

we can reconstruct y_0 as

$$\hat{y}_0 = \frac{1}{N} \cdot \sum_{i=1}^{N} (y_0 + \varepsilon_i) = y_0 + \frac{1}{N} \cdot \sum_{i=1}^{N} \varepsilon_i \approx y_0 \ .$$

Solution for output prediction

Under the assumption that errors are independent form each other but still generated by the same stochastic procedure, we can predict

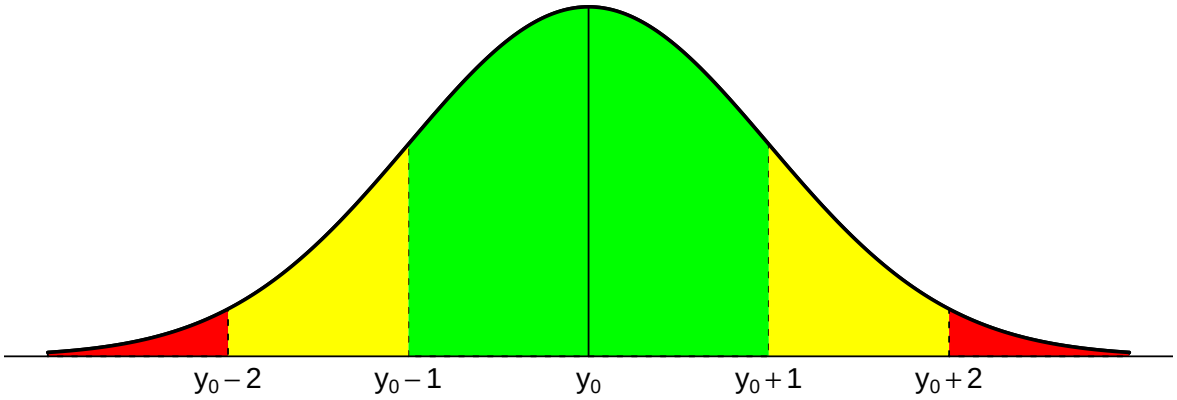
$$\hat{y}_{N+1} = \frac{1}{N} \cdot \sum_{i=1}^{N} y_i \ .$$

Note that the prediction is reasonable even if the error is biased, i.e.,

$$\frac{1}{N} \cdot \sum_{i=1}^{N} \varepsilon_i \ll 0 \qquad \text{or} \qquad \frac{1}{N} \cdot \sum_{i=1}^{N} \varepsilon_i \gg 0$$

even for large number of observations.

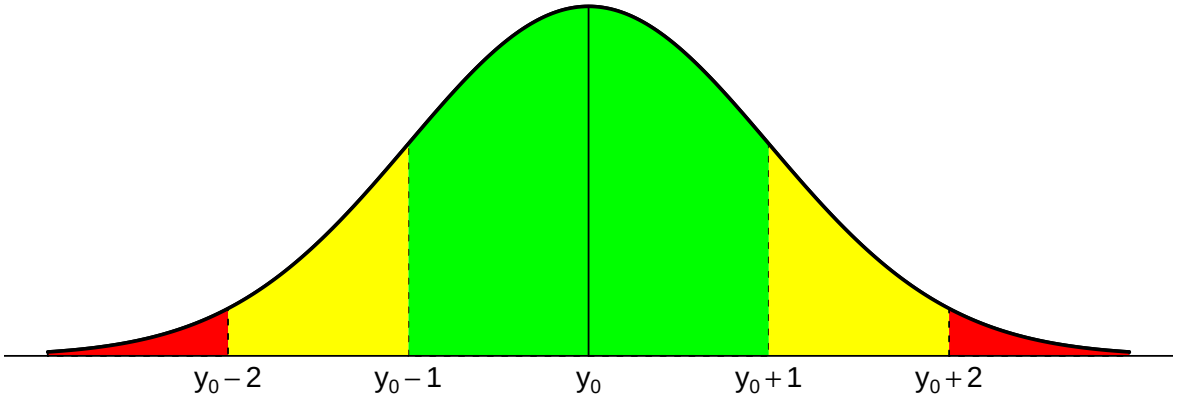
On the possibility of good approximation



For simplicity assume that we have only a single measurement y_1 and the error term ε_1 follows the normal distribution $\mathcal{N}(0,1)$.

- ▷ Then the error term ε_1 can be arbitrarily large on rare occasions.
- ▷ Given $\hat{y}_0$ we cannot provide any range where y_0 is guaranteed to be.
- ▷ It is impossible to give approximation bounds that always hold.

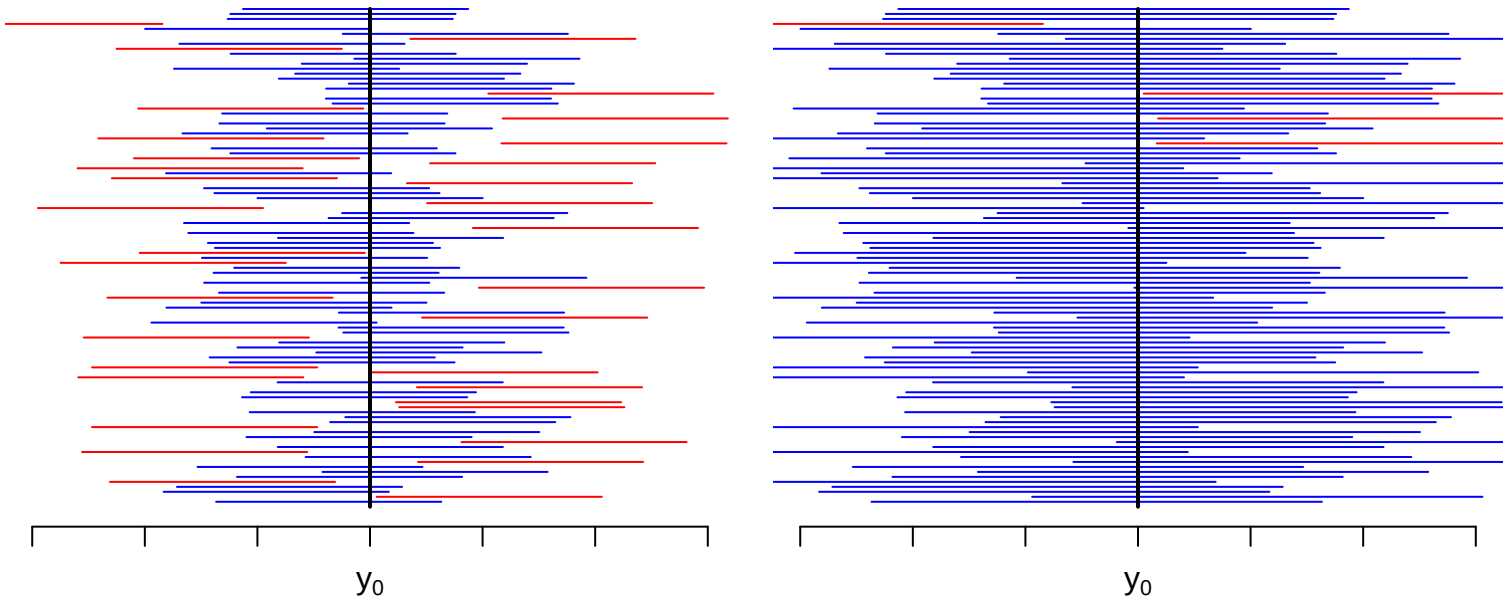
Confidence intervals



Second best alternative. If we report $y_0 \in [\hat{y}_0 - 1, \hat{y}_0 + 1]$, then we are correct in 68.3% cases if we could sample $\hat{y}_0$ several times.

- ▷ For the remaining 31.7% cases the error could be arbitrarily large.
- ▷ This claim says nothing about the particular measurement.
- ▷ By increasing the interval, we can reduce the fraction of failed runs.

Illustrative example



By increasing the length of the interval we increase the fraction of runs for which the true value of y_0 lies in the interval.

Prediction intervals

Even if we know the true value of y_0 we cannot predict y_i, since we do not know the additive noise term ε_i before the measurement.

▷ We cannot give upper and lower bounds for y_i which always hold.

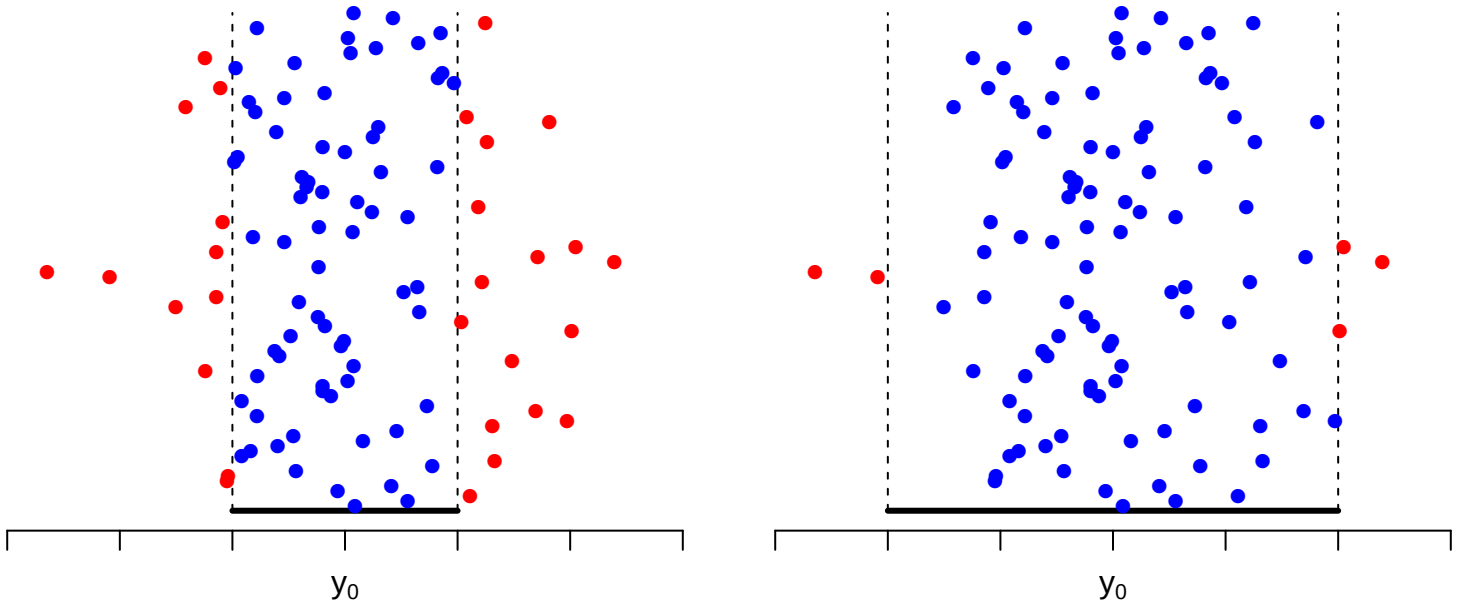
Instead, we can specify a prediction interval $[y_0 - \varepsilon, y_0 + \varepsilon]$ so that with probability 95% the resulting measurement y_i is in the range.

▷ Usually, the analysis is similar to confidence interval derivation.

Interpretation of prediction intervals is different from confidence intervals.

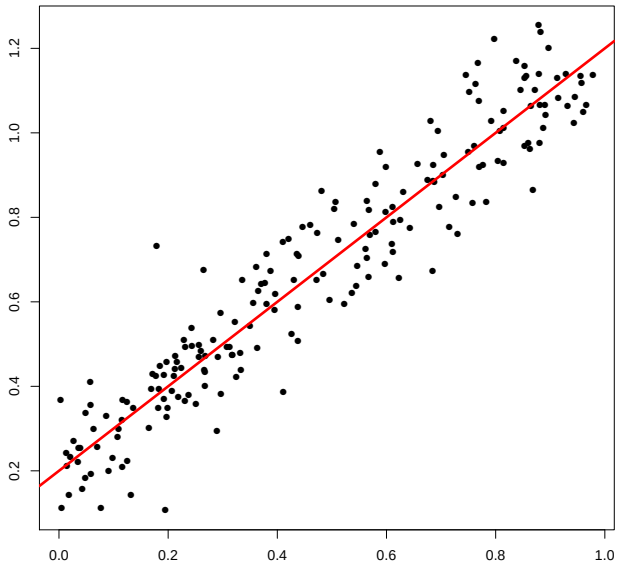
▷ The probability estimate holds for the particular interval.

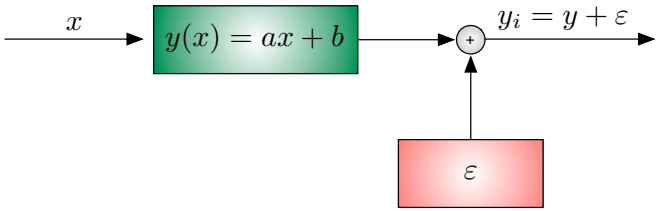
Illustrative example



By increasing the length of the interval we increase the fraction of future measurements which fall into interval.

Univariate linear regression problem





Regression task. Given a list of observations $(x_1, y_1), \ldots, (x_N, y_N)$ find a line $y = ax + b$ that approximates the correspondence in the data.

▷ The definition of approximation gives a rise to many methods.

Ordinary least squares regression

Means square error is a standard measure of goodness for the approximation

$$\textsc{Mse} = \frac{1}{N} \cdot \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 = \frac{1}{N} \cdot \sum_{i=1}^{N} (y_i - ax_i - b)^2$$

Ordinary least squares regression finds values a and b that minimise $\textsc{Mse}$.

The minimum is guaranteed to exist, since the following conditions hold:

- ▷ The function is bounded form below.
- ▷ The function is continuous wrt a and b

Further analysis indicates that there is a single minimum.

Towards closed-form solution

As the corresponding derivatives can be expressed as

$$\frac{\partial \textsc{Mse}}{\partial b} = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{\partial (y_i - ax_i - b)^2}{\partial b} = -\frac{2}{N} \cdot \sum_{i=1}^{N} (y_i - ax_i - b)$$

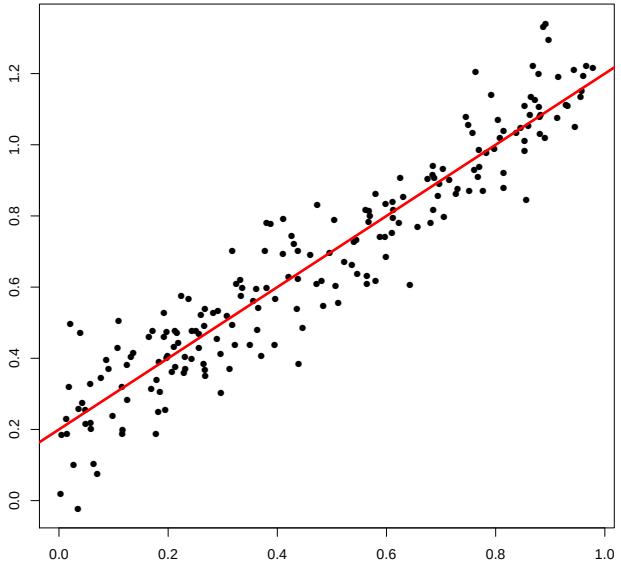
$$\frac{\partial \textsc{Mse}}{\partial a} = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{\partial (y_i - ax_i - b)^2}{\partial a} = -\frac{2}{N} \cdot \sum_{i=1}^{N} (y_i - ax_i - b)x_i$$

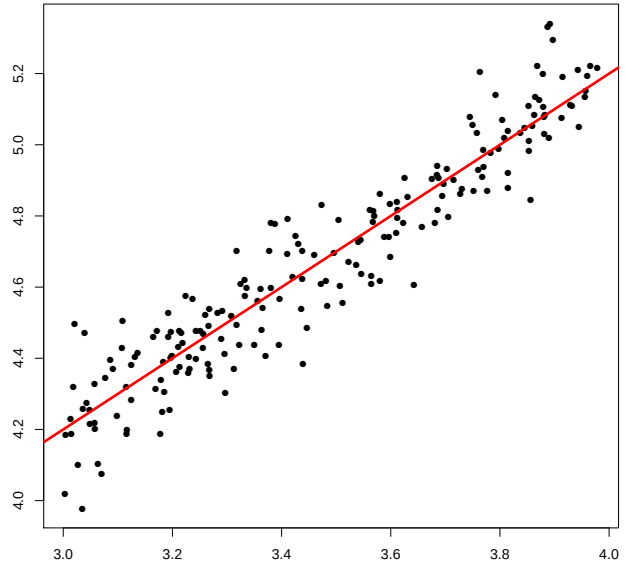
we arrive at system of linear equations

$$\begin{cases} \frac{\partial \textsc{Mse}}{\partial b} & = 0 \\ \frac{\partial \textsc{Mse}}{\partial a} & = 0 \end{cases} \quad \Longleftrightarrow \quad \begin{cases} \sum_{i=1}^{N} y_i & = a \cdot \sum_{i=1}^{N} x_i + b \cdot N \\ \sum_{i=1}^{N} y_i x_i & = a \cdot \sum_{i=1}^{N} x_i^2 + b \cdot \sum_{i=1}^{N} x_i \end{cases}$$

Simplifying data transformation

Observation. The best linear approximation does not change if we shift the origin of x and y axis. To be precise, the line as a *geometrical object* remains the same, while its *description* changes.





Simplifying data transformation

Thus, we can solve the system of linear equation for the centred data:

$$\sum_{i=1}^{N} x_i = 0 \qquad \text{and} \qquad \sum_{i=1}^{N} y_i = 0.$$

Hence we get

$$\begin{cases} 0 & = a \cdot 0 + b \cdot N \\ \sum\limits_{i=1}^{N} y_i x_i & = a \cdot \sum\limits_{i=1}^{N} x_i^2 + b \cdot 0 \end{cases} \iff \begin{cases} b & = 0 \\ a & = \dfrac{\sum\limits_{i=1}^{N} y_i x_i}{\sum\limits_{i=1}^{N} x_i^2} \end{cases}$$

Closed-form solution

Let us denote mean values by

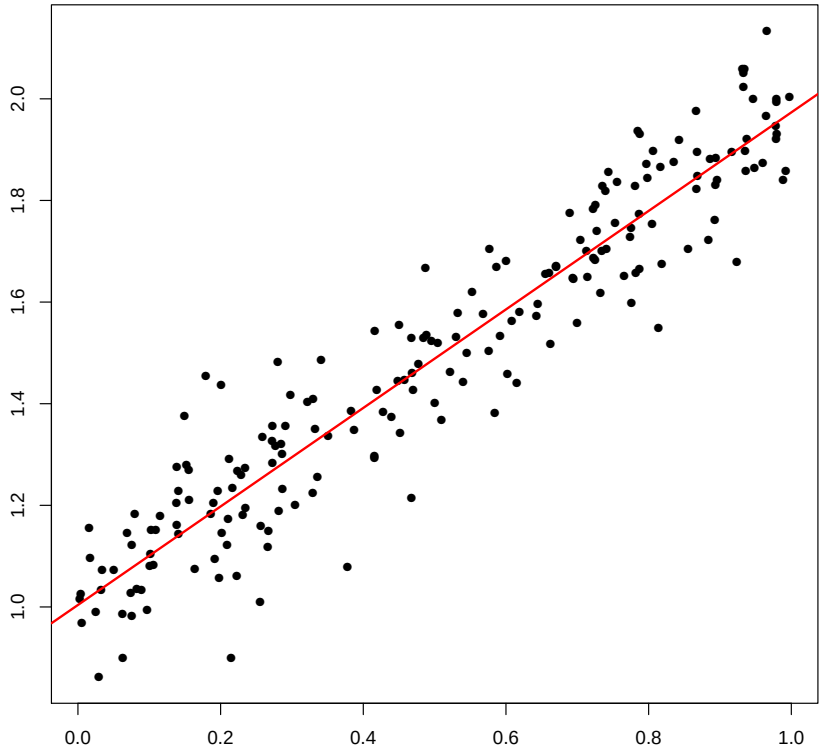
$$\overline{x} = \frac{1}{N} \cdot \sum_{i=1}^{N} x_i \qquad \text{and} \qquad \overline{y} = \frac{1}{N} \cdot \sum_{i=1}^{N} y_i$$

Then the closed form solution for the general case is

$$a = \frac{\sum_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{N} (x_i - \overline{x})^2}$$

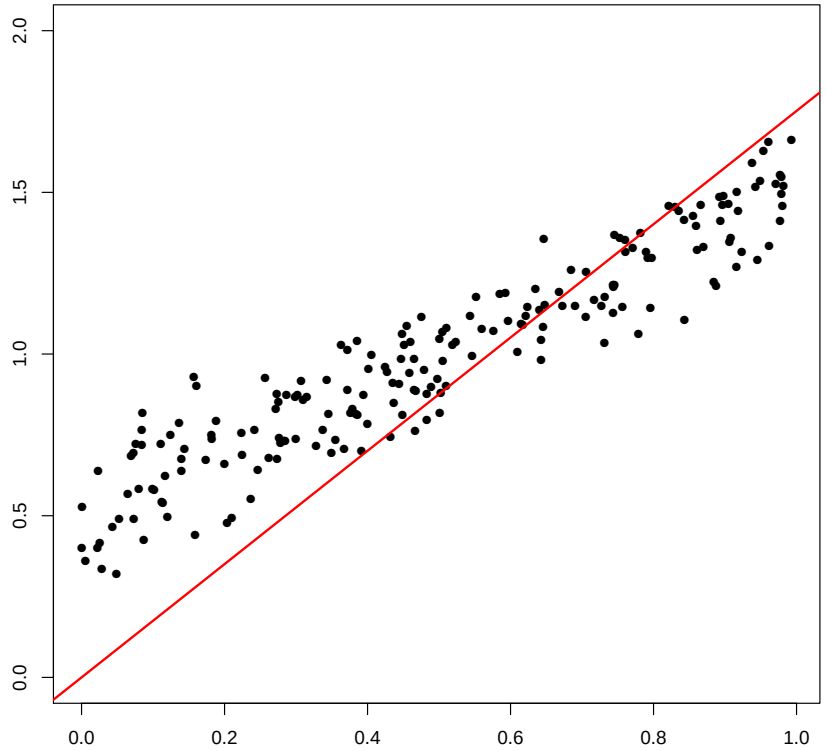
$$b = \overline{y} - a\overline{x}$$

Implementation in GNU R



▷ Fit linear model with intercept: `lm(y ~x + 1)`

Implementation in GNU R



- Fit linear model without intercept `lm(y ~x + 0)`

Multivariate linear regression problem

Each input $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ is a p-dimensional vector and we are looking for a model

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

that minimises the mean square error as before. We can express prediction values through a matrix equation

$$\begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_N \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & \cdots & x_{Np} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_p \end{pmatrix}$$

which is usually written in a compact form $\hat{\boldsymbol{y}} = X\boldsymbol{\beta}$.

Matrix formula for mean square error

By substituting $\hat{\boldsymbol{y}} = X\boldsymbol{\beta}$ into the formula

$$\textsc{Mse} = \frac{1}{N}(\boldsymbol{y} - \hat{\boldsymbol{y}})^T(\boldsymbol{y} - \hat{\boldsymbol{y}})$$

we obtain

$$\textsc{Mse} = \frac{1}{N}\left(\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T X\boldsymbol{\beta} + \boldsymbol{\beta}^T X^T X\boldsymbol{\beta}\right)$$

Thus we must choose a value for $\boldsymbol{\beta}$ that minimises

$$\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T X\boldsymbol{\beta} + \boldsymbol{\beta}^T X^T X\boldsymbol{\beta}$$

Closed-form solution

Again all partial derivatives of MSE wrt β_i must be zeroes. It is possible to take derivatives as before, but several steps can be combined if we use vector derivatives (gradient) and formulae from Matrix Cookbook:

$$\frac{\partial \text{MSE}}{\partial \boldsymbol{\beta}} = -2\boldsymbol{y}^T X + 2X^T X\boldsymbol{\beta} = \mathbf{0}$$

This leads to a linear equation

$$X^T X\boldsymbol{\beta} = X^T \boldsymbol{y}$$

with a unique solution

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

Implementation in GNU R

- Fit linear model with intercept: `lm(y ~ x1 + x2 + x3 + 1)`
- Fit linear model without intercept: `lm(y ~ x1 + x2 + x3 + 0)`

Adding new features

Linear regression is often too contained. For instance, we might try to seek a quadratic dependencies $y(x) = ax^2 + bx + c$.

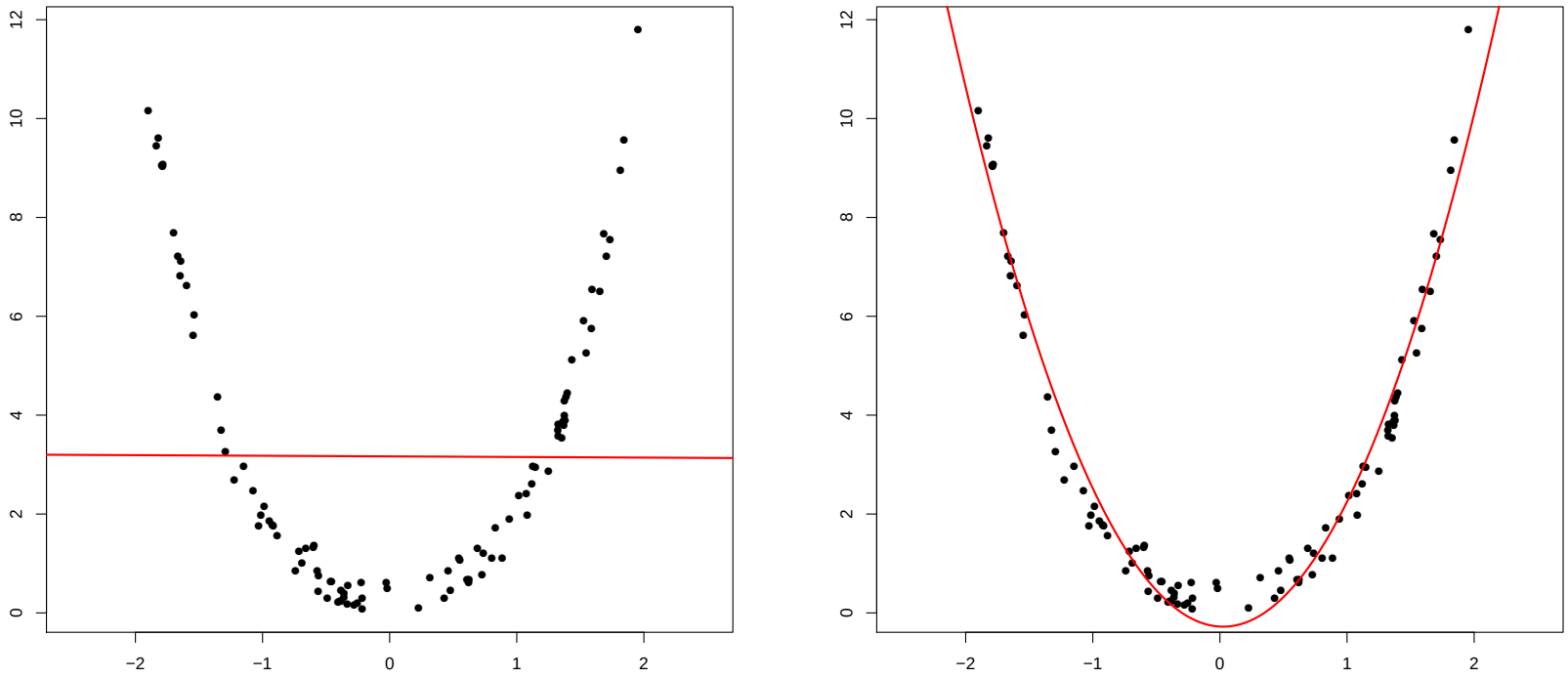
Such dependencies can be still found using linear regression

- ▷ First we must map inputs to new set of non-linear features. For quadratic regression, we must compute

$$\phi_1 = 1 \quad \phi_2 = x \quad \phi_3 = x^2$$

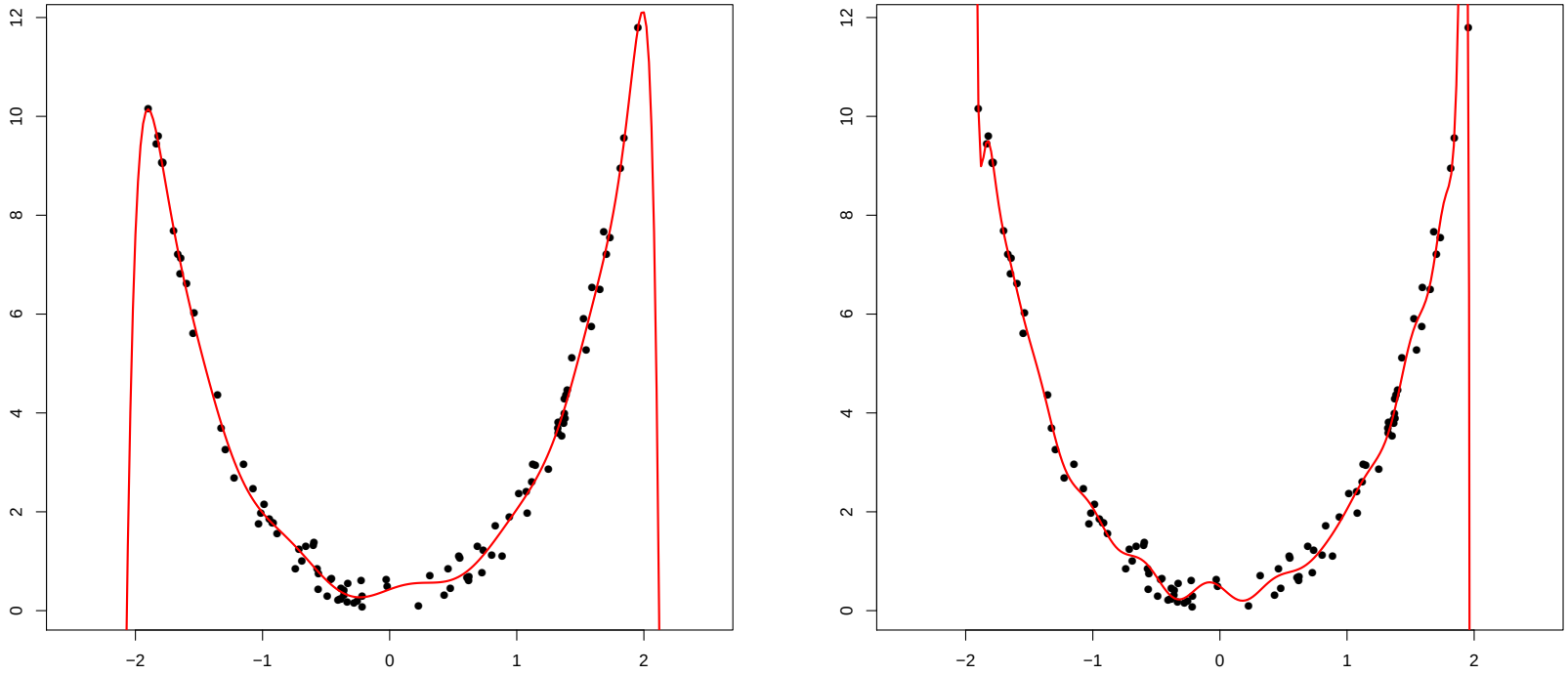
- ▷ Second, we must do a linear regression with the new set of features, i.e., look for the model $y = \beta_0 + \beta_1\phi_1 + \cdots + \beta_k\phi_k$
- ▷ The prediction $\hat{y}(x)$ can be found as $\beta_0 + \beta_1\phi_1(x) + \cdots + \beta_k\phi_k(x)$
- ▷ Functions $\phi_1, \ldots, \phi_k$ are sometimes called basis functions.

Examples of polynomial regression



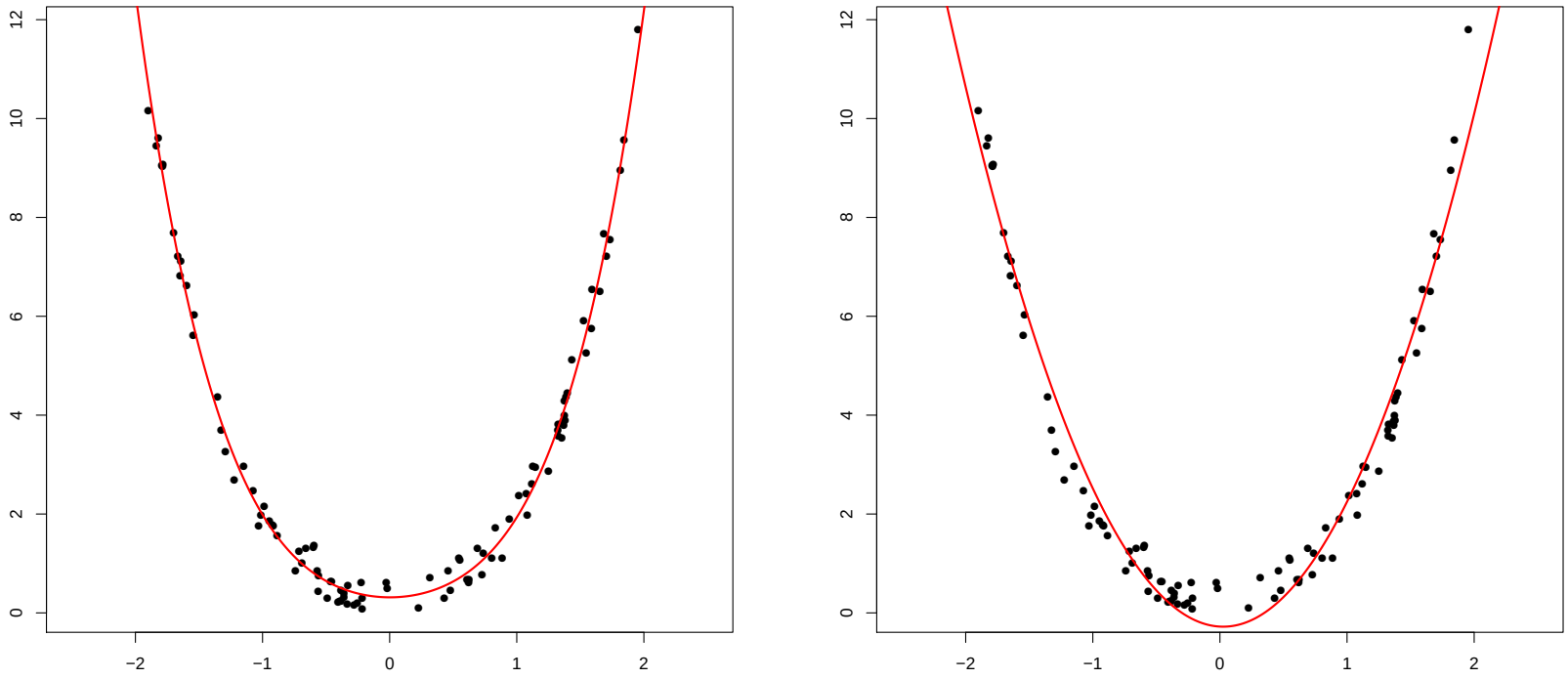
Linear and quadratic model clearly under-fit the data.

Examples of polynomial regression



High order polynomial models clearly over-fit the data. With high enough polynomial it is possible to fit all points.

Examples of polynomial regression



Hence, we need a tradeoff between flexibility to get optimal predictions.

Numerical stability

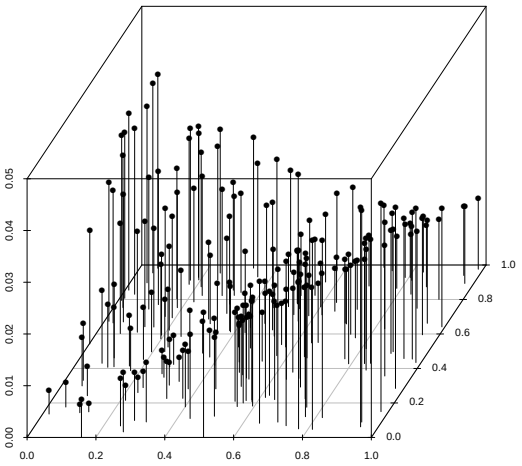
Although we have a closed form solution $\boldsymbol{\beta} = (X^T X)^{-1} X^T \boldsymbol{y}$ to linear regression problem, the resulting estimate might be unusable.

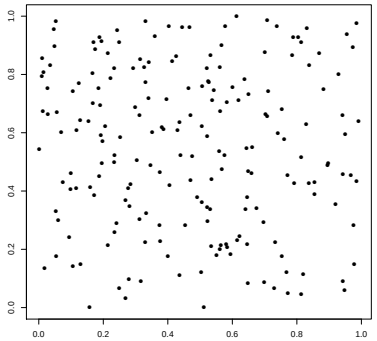
- ▷ The matrix $X^T X$ is non-invertible matrix
- ▷ Small measuring errors on the output produce large fluctuation of $\boldsymbol{\beta}$
- ▷ Small measuring errors on the inputs produce large fluctuations of $\boldsymbol{\beta}$
- ▷ All these instabilities depend on the input matrix X

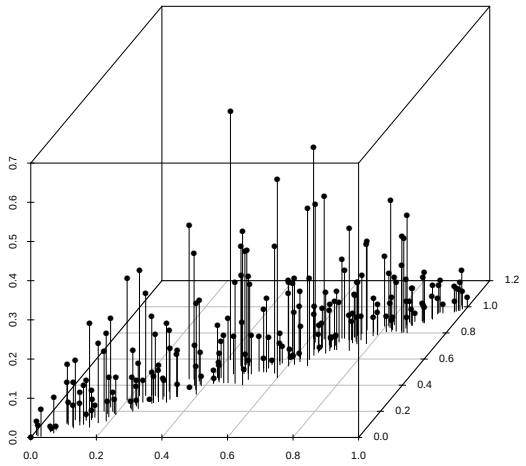
Fact. Matrix $X^T X$ is non-invertible if one feature can be expressed as linear combinations of others. More generally

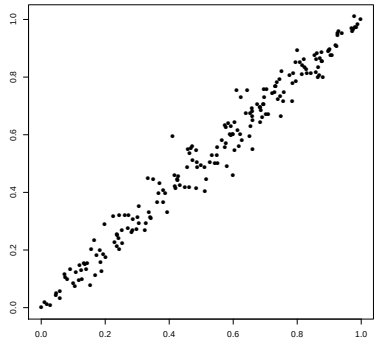
- ▷ Highly correlated features can cause stability problems
- ▷ Near-orthogonal features lead to matrix $X^T X$ which is very stable

Illustrative example about leverage









Concept of regularisation

Problem statement. If the experiment design is poor then individual points can have a huge leverage and thus corrupt the OLS solution.

- ▷ If we limit the maximal size of coefficients, points with high leverage have only a limited impact on the solution $\boldsymbol{\beta}$.
- ▷ However, we must solve a constrained optimisation task

$$\begin{aligned} &\textsc{Mse}(\boldsymbol{\beta}) \to \min \\ &\text{st.} \quad \|\boldsymbol{\beta}\| \leq c_0 \end{aligned}$$

- ▷ Regularisation methods differ mostly on what kind of restrictions are put on the coefficient vector $\boldsymbol{\beta}$.