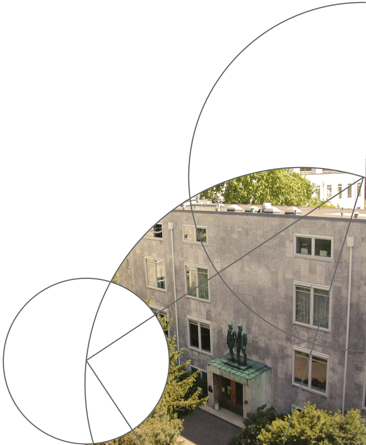
Faculty of Science

Linear Regression

Machine Learning

Christian Igel
Department of Computer Science



Outline

Outline

Derivative

Let $f : \mathbb{R} \to \mathbb{R}$ be a real-valued function. If the limit

$$f'(x) = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon}$$

exists, the function f is differentiable and f' is its derivative.

Differentiation rules

Let f and g be differentiable functions.

Sum. If $h(x) = f(x) + g(x)$ then h is differentiable and $h'(x) = f'(x) + g'(x)$.

Chain. If $h(x) = f(g(x))$ then h is differentiable and $h'(x) = f'(g(x))g'(x)$.

Product. If $h(x) = f(x)g(x)$ then h is differentiable and $h'(x) = f'(x)g(x) + f(x)g'(x)$.

Power. $f(x) = x^r$ with $r \in \mathbb{R}$ has derivative $f'(x) = rx^{r-1}$.

Exponential. For $f(x) = e^x$ we have $f'(x) = e^x$.

Logarithm. For $f(x) = \ln x$ we have $f'(x) = 1/x$.

Important special cases

Constant. Product rule implies $h'(x) = af'(x)$ for $h(x) = af(x)$ with $a \in \mathbb{R}$

Quotient. Product and power give for

$$h(x) = \frac{f(x)}{g(x)}$$

with non-vanishing g (i.e., g is never zero)

$$h'(x) = -\frac{f'(x)g(x) - f(x)g'(x)}{[g(x)]^2} \quad .$$

Reciprocal. Special case, we have $g'(x) = -1/x^2$ for $g(x) = 1/x$ and, more general, if $h(x) = \frac{1}{f(x)}$ for f non-vanishing, then

$$h'(x) = -\frac{f'(x)}{[f(x)]^2} \quad .$$

Partial derivative

Let $f : \mathbb{R}^d \to \mathbb{R}$ depend on d variables $(x_1, \dots, x_d)^\mathsf{T} \in \mathbb{R}^d$. The partial derivative of f with respect to x_i at a point $\boldsymbol{x} \in \mathbb{R}^d$ is defined as

$$\frac{\partial}{\partial x_i} f(\boldsymbol{x}) = \lim_{\epsilon \to 0} \frac{f(\boldsymbol{x} + \epsilon \boldsymbol{e}_i) - f(\boldsymbol{x})}{\epsilon}$$

with $\boldsymbol{e}_i$ being the ith unit vector (i.e., a vector where the ith component is one and all other components are zero) given that the limit exists. For $d = 1$ we have

$$\frac{\partial}{\partial x} f(x) = f'(x) \ .$$

Gradient

- *Rate of change* (*directional derivative*) of $f : \mathbb{R}^d \rightarrow \mathbb{R}$ at a point $\boldsymbol{x} \in \mathbb{R}^d$ when moving in the direction $\boldsymbol{u} \in \mathbb{R}^d$, $\|\boldsymbol{u}\| = 1$, is defined as:

$$\nabla_{\boldsymbol{u}} f(\boldsymbol{x}) = \lim_{\epsilon \rightarrow 0} \frac{f(\boldsymbol{x} + \epsilon \boldsymbol{u}) - f(\boldsymbol{x})}{\epsilon}$$

- The *gradient*

$$\nabla f(\boldsymbol{x}) = \left(\frac{\partial f(\boldsymbol{x})}{\partial x_1}, \frac{\partial f(\boldsymbol{x})}{\partial x_2}, \dots, \frac{\partial f(\boldsymbol{x})}{\partial x_d} \right)^{\mathsf{T}}$$

points in the direction $\nabla f(\boldsymbol{x}) / \|\nabla f(\boldsymbol{x})\|$ giving maximum rate of change $\|\nabla f(\boldsymbol{x})\|$.

Chain rule

The *chain rule* for computing the derivative of a composition of two functions,

$$\frac{\partial f(g(x))}{\partial x} = f'(g(x))g'(x)$$

with $f'(x) = \frac{\partial f(x)}{\partial x}$ and $g'(x) = \frac{\partial g(x)}{\partial x}$, can be extended to:

$$\frac{\partial f(g_1(x), g_2(x), \dots, g_d(x))}{\partial x} = \sum_{i=1}^{d} \frac{\partial f(g_1(x), \dots, g_d(x))}{\partial g_i(x)} \frac{\partial g_i(x)}{\partial x}$$

Outline

The linear regression problem

- The goal is to predict a target variable $\boldsymbol{y} \in \mathbb{R}^m$ based on given data $\boldsymbol{x} \in \mathcal{X} = \mathbb{R}^d$. For simplicity, let $m = 1$.
- We build affine linear predictive models

$$y = f(\boldsymbol{x}) = \sum_{i=1}^{d} w_i x_i + b = \boldsymbol{w}^\mathsf{T}\boldsymbol{x} + b$$

  with $\boldsymbol{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$.
- For convenience, we define $\tilde{\boldsymbol{x}}^\mathsf{T} = (x_1, \dots, x_d, 1)$ and $\tilde{\boldsymbol{w}}^\mathsf{T} = (w_1, \dots, w_d, b)$ and consider the equivalent formulation

$$y = f(\tilde{\boldsymbol{x}}) = \sum_{i=1}^{d+1} \tilde{w}_i \tilde{x}_i = \tilde{\boldsymbol{w}}^\mathsf{T}\tilde{\boldsymbol{x}} \ .$$

Data matrix

Training data $S = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\}$ is gathered in data matrix (x_{ij} is the jth component of the ith training pattern, i.e., $[\boldsymbol{x}_i]_j$)

$$\tilde{\boldsymbol{X}} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} & 1 \\ x_{21} & x_{22} & \dots & x_{2d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nd} & 1 \end{pmatrix}$$

Let's omit the tilde to keep notation uncluttered.

Formalizing the goal

- We want to minimize the empirical risk under the squared loss $\ell(y, \hat{y}) = (y - \hat{y})^2$:

$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i) = \frac{1}{n}\sum_{i=1}^{n} \left(y_i - \boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i\right)^2$$

  This quantity is also known as mean-squared-error (MSE).

- Optimal parameters are given by

$$\boldsymbol{w}^\star = \left(\boldsymbol{X}^\mathsf{T}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\mathsf{T}\boldsymbol{y} \ ,$$

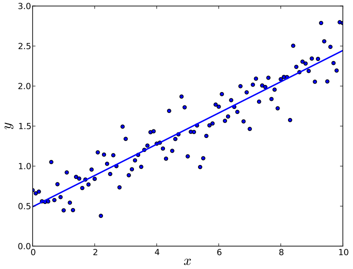
  where $\boldsymbol{y} = (y_1, \dots, y_n)^\mathsf{T}$.

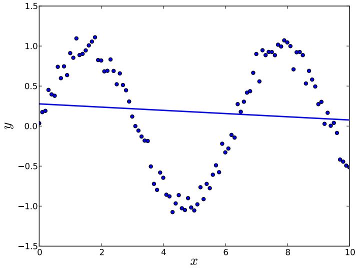
Linear regression algorithm

Algorithm 1: linear regression

```
Input: $S = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\} \in (\mathbb{R}^d \times \mathbb{R})^n$
Output: affine linear model
1 $\boldsymbol{y} = (y_1, \dots, y_n)^\mathsf{T}$
2 $\tilde{\boldsymbol{X}} = \begin{pmatrix} x_{11} & \dots & x_{1d} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ x_{n1} & \dots & x_{nd} & 1 \end{pmatrix}$
3 $(w_1, \dots, w_d, b)^\mathsf{T} = \left(\tilde{\boldsymbol{X}}^\mathsf{T} \tilde{\boldsymbol{X}}\right)^{-1} \tilde{\boldsymbol{X}}^\mathsf{T} \boldsymbol{y}$
Result: $(\boldsymbol{w}, b)$, defining the model $f(\boldsymbol{x}) = \boldsymbol{w}^\mathsf{T} \boldsymbol{x} + b$
```

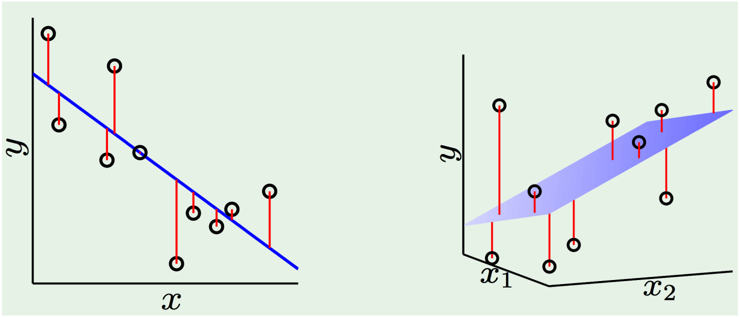
Examples





The right plot is an example of “underfitting”.

Visualizing the error



Abu-Mostafa, Magdon-Ismail, and Lin. *Learning from Data.* AMLbook, 2012

Outline

Solution using pseudo-inverse I

Setting the derivative

$$\frac{\partial}{\partial \boldsymbol{w}} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \boldsymbol{w}^{\mathsf{T}} \boldsymbol{x}_i\right)^2 = - \sum_{i=1}^{n} \left(y_i - \boldsymbol{w}^{\mathsf{T}} \boldsymbol{x}_i\right) \boldsymbol{x}_i^{\mathsf{T}}$$

of ($n/2$ times) the empirical risk to $\mathbf{0}$ yields

$$\mathbf{0}^{\mathsf{T}} = \sum_{i=1}^{n} y_i \boldsymbol{x}_i^{\mathsf{T}} - \boldsymbol{w}^{\mathsf{T}} \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathsf{T}} \ ,$$

which implies with $\boldsymbol{y} = (y_1, \dots, y_n)^{\mathsf{T}}$

$$\mathbf{0}^{\mathsf{T}} = \sum_{i=1}^{n} y_i \boldsymbol{x}_i^{\mathsf{T}} - \boldsymbol{w}^{\mathsf{T}} \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathsf{T}} = \boldsymbol{y}^{\mathsf{T}} \boldsymbol{X} - \boldsymbol{w}^{\mathsf{T}} \boldsymbol{X}^{\mathsf{T}} \boldsymbol{X} \ .$$

Solution using pseudo-inverse II

$$\mathbf{0}^\mathsf{T} = \boldsymbol{y}^\mathsf{T}\boldsymbol{X} - \boldsymbol{w}^\mathsf{T}\boldsymbol{X}^\mathsf{T}\boldsymbol{X}$$

implies

$$\boldsymbol{w}^\mathsf{T}\boldsymbol{X}^\mathsf{T}\boldsymbol{X} = \boldsymbol{y}^\mathsf{T}\boldsymbol{X}$$

and thus

$$\left(\boldsymbol{X}^\mathsf{T}\boldsymbol{X}\right)^\mathsf{T}\boldsymbol{w} = \boldsymbol{X}^\mathsf{T}\boldsymbol{y} \ .$$

Assuming $\boldsymbol{X}^\mathsf{T}\boldsymbol{X}$ has full rank, we get the maximum likelihood estimate:

$$\boldsymbol{w}^\star = \left(\boldsymbol{X}^\mathsf{T}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\mathsf{T}\boldsymbol{y}$$

$\boldsymbol{X}^\dagger = \left(\boldsymbol{X}^\mathsf{T}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\mathsf{T}$ is called *Moore-Penrose pseudo-inverse*.

In practice, one does not compute $\left(\boldsymbol{X}^\mathsf{T}\boldsymbol{X}\right)^{-1}$, but considers the $\boldsymbol{QR}$-decomposition of $\boldsymbol{X}$ and solves a system of linear equations.

Why sum-of-squares?

- Sum-of-squares error has the nice properties of
  - being differentiable and
  - penalizing large deviations from a target value more than proportionally stronger than small deviations.
- There are theoretical reasons ...

Likelihood function

- Likelihood (function) of the parameters $\boldsymbol{w}$ given training data S is the probability of observing S when the data is generated by h with parameters $\boldsymbol{w}$.
- Likelihood for i.i.d. S:

$$\prod_{i=1}^{N} P(Y = y_i \,|\, X = \boldsymbol{x}_i; h) \text{ or short } \prod_{i=1}^{N} P(y_i \,|\, \boldsymbol{x}_i)$$

- Negative log(arithmic) likelihood:

$$-\log \prod_{i=1}^{N} P(y_i \,|\, \boldsymbol{x}_i) = -\sum_{i=1}^{N} \log P(y_i \,|\, \boldsymbol{x}_i)$$

- Learning principle: Negative log-likelihood minimization

Sum-of-squares and Gaussian noise I

Consider a deterministic real-valued target functions perturbed by zero-mean additive Gaussian noise with variance σ^2, then we have

$$p(y \,|\, \boldsymbol{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-f(\boldsymbol{x}))^2}{2\sigma^2}} \quad .$$

Given $S = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\}$ the likelihood of the model $h(x) = \boldsymbol{w}^\mathsf{T}\boldsymbol{x}$ is

$$p((y_1, \dots, y_n)^\mathsf{T} \,|\, (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n), \boldsymbol{w}, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - \boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i)^2}{2\sigma^2}} \quad .$$

Sum-of-squares and Gaussian noise II

Let the logarithm turn products into sums:

$$\begin{aligned}\ln p(\boldsymbol{y} \,|\, \boldsymbol{w}) &= \ln \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - \boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i)^2}{2\sigma^2}} \\ &= \sum_{i=1}^{n} \ln \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - \boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i)^2}{2\sigma^2}} \\ &= \underbrace{-\frac{n}{2}\ln\sigma^2 - \frac{n}{2}\ln(2\pi)}_{\text{independent of } \boldsymbol{w}} - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i\right)^2\end{aligned}$$

Thus, maximizing the likelihood corresponds to minimizing MSE under the assumption of Gaussian noise.

Why Gaussian noise? Central limit theorem, maximum entropy principle

Outline

Non-linear transformations

Consider a non-linear transformation

$$\Phi : \mathcal{X} \to \mathcal{Z} \quad .$$

Then we can consider hypotheses

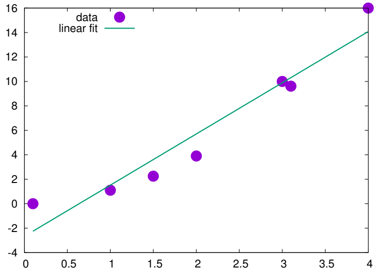
$$h(\boldsymbol{x}) = \tilde{h}(\Phi(\boldsymbol{x})) = \tilde{\boldsymbol{w}}^{\mathsf{T}} \Phi(\boldsymbol{x})$$

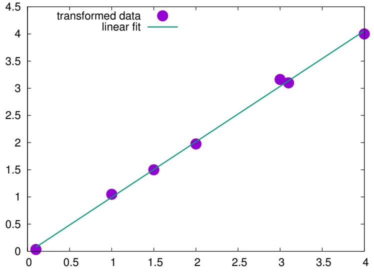
and do linear regression with the data matrix:

$$\begin{pmatrix} [\Phi(\boldsymbol{x}_1)]_1 & [\Phi(\boldsymbol{x}_1)]_2 & \dots & 1 \\ [\Phi(\boldsymbol{x}_2)]_1 & [\Phi(\boldsymbol{x}_2)]_2 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ [\Phi(\boldsymbol{x}_n)]_1 & [\Phi(\boldsymbol{x}_2)]_2 & \dots & 1 \end{pmatrix}$$

Example I

$$\Phi : \mathbb{R} \to \mathbb{R}$$
$$x \mapsto \sqrt{x}$$

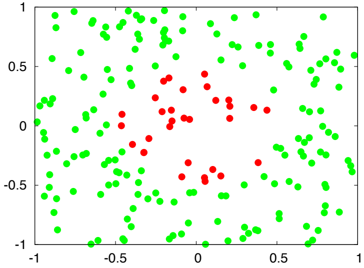




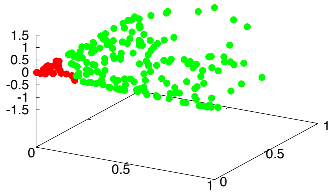
x $\sqrt{x}$

Example II

$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$
$$(x_1, x_2) \mapsto (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$



(x_1, x_2)



$(x_1^2, x_2^2, \sqrt{2}x_1x_2)$

Outline

Small sample problem

Consider the case $n \leq d$. Then linear regression always finds a model with zero empirical risk by solving the system of linear equations:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}$$

Even if the y_i are independent of the $\boldsymbol{x}_i$!

Overfitting

Overfitting occurs if the model faithfully reflects idiosyncrasies of the training data rather than the underlying process that has generated the data. This may happen if the model is too complex/flexible in relation to the amount of available information.

Formal concepts of model complexity and flexibility exist.

Outline

Summary

- Regression means building predictive models for real-valued target patterns.
- Linear regression is a baseline method for regression tasks.
- Too complex/flexible models can lead to overfitting.
- However, the affine linear models built by standard linear regression are not very flexible and rather tend to “underfit” the data. Therefore non-linear methods are needed.
- Non-linear transformations combined with linear regression lead to non-linear models.