Linear Regression

Linear regression p. 34

is a linear approach for modelling the relationship between a scalar response and one or more explanatory variables.

Basis Model approach p. 35 - 37

DATA
=
TRUE SIGNAL
+
NOISE

Linear Basis Function Model p. 38

$y_1 = \omega_0 \cdot \phi_0(1) + \omega_1 \cdot \phi_1(x_{i,1}) + \omega_2 \cdot \phi_2(x_{i,2}) + \ldots + \omega_j \cdot \mathrm{j}(x_{i,j})$

$y_i = \vec{\omega} \cdot \vec{\phi}(\vec{x})^T$

Vanilla linear regression model p. 37 - 38

$y_i = \omega_0 \cdot 1 + \omega_1 \cdot x_{i,1} + \omega_2 \cdot x_{i,2} + \cdots + \omega_j \cdot x_{i,j}$

$y_i = \vec{\omega} \cdot \vec{x}^T$

Homoscedasticity
is the property that each sample has the same variance in its noise as the others, regardless of the values contained in this sample.

Independence of errors
states that the noise (or error) of the samples is not correlated with each other.

Finding weights for ω_0 to ω_j p. 38 - 42

1. Use Log Likelihood function for a Gaussian distribution: $ln\, P(t|x, w, \beta)$

2. Derive 1. to get the score function with respect to the gradient ∇.

3. Results in: $-\nabla \sum_{i=1}^{n} \frac{(k_i - g(\vec{x}_i))^2}{2\sigma^2}$

4. Introduce the design matrix Φ:

$$\begin{bmatrix} \phi_0(x_{1,0}) & \cdots & \phi_j(x_{1,j}) \\ \vdots & \ddots & \vdots \\ \phi_0(x_{n,0}) & \cdots & \phi_j(x_{n,j}) \end{bmatrix}$$

5. This finally results in: $\vec{\omega} = \Phi^{\dagger}\vec{k}$

To calculate the weights, simply multiply the pseudo-inverse with the vector of targets.

As a rule of thumb, use the pseudo-inverse only for less than 10000 samples.

Regularization p. 48 - 52

Prevent the model from overfitting without reducing its complexity. Keep the weights of the model small, as large weights cause a high sensitivity.
Therefore, the error function is defined as follows: $E(\vec{\omega}) = E_D(\vec{\omega}) + \lambda \cdot E_\omega(\vec{\omega})$
With the error E_D between true values and predictions, the regularization coefficient E_ω and the regularization parameter λ.

L1 norm: $E_\omega(\vec{\omega}) = \frac{1}{2}\sum_{i=0}^{j} |\omega_i|$ → **Lasso regularization**

L2 norm: $E_\omega(\vec{\omega}) = \frac{1}{2}\sum_{i=0}^{j} \omega_i^2$ → **Quadratic regularization**

The regularization becomes stronger with increasing λ as this leads to smaller model weights.
L1 regularization *results in a rather sparse weight vector since some weights are set to zero.*
L2 regularization *primarily prevents the weights from becoming too large (due to the squaring).*

Bias Variance Decomposition p. 42 - 48

The error of the model should be decomposed into an error that arises from a mismatch between the model and the real data **(bias)** and an error that arises from the noise in the data **(variance)**.

1. Use the expected value of the squared error between true values and predictions **(L2 error function)**: $E[L2] = \frac{1}{N}\sum_{n=1}^{N} E[(t_n - y_n)^2]$

2. Expanding and reforming yields the expected value of the noise and the expected value of the squared error between the real function f and the predictions:

$$E[L2] = \frac{1}{N}\sum_{n=1}^{N}(E[\epsilon^2] + E[(f_n - y_n)^2])$$

3. Further expanding and reforming results in:

$$E[L2] = \frac{1}{N}\sum_{n=1}^{N}(E[\epsilon^2] + E[(f_n - E[y_n])^2] + E[(E[y_n] - y_n)^2])$$

So, the general expected value of the error of a model depends on the noise of the data, the squared bias and the variance of the model across different datasets.

Analysis: ↑ *model complexity:* ↓ *bias error term,* ↑ *variance error term*

Good fit: *The model is exactly as complex as it needs to be.*
Overfit: *The model is much too complex for the data.*
Underfit: *The model is not complex enough.*

Basis Functions p. 52 - 56

Global: *The function covers the whole interval of interest and thus the outlier changes the values of the basis function for every possible prediction.*

Local: *The function may be composed of multiple basis functions, each covering its respective interval. The outlier may change the basis function responsible for its interval, but the rest of the basis function (and thus the rest of the predictions) stays untouched.*

Relatively local: *The function is theoretically global, but the change it undergoes is so small that it behaves like a local function.*

Polynomial Basis Function *(global)*: $\phi_j(x_{i,j}) = x_{i,j}^j$

Gaussian basis functions *(relatively local)*: $\phi_j(x_{i,j}) = e^{\frac{(x_{i,j}^j - \mu_j)^2}{2\sigma^2}}$

Sigmoidal basis functions *(relatively local)*: $\phi_j(x_{i,j}) = \sigma\left(\frac{x_{i,j} - \mu_j}{s}\right)$ *where* $\sigma(z) = \frac{1}{1+e^{-z}}$

Periodic basis function *(global)*: $f(x) = f(x + nk), k \in \mathbb{N}$

Bin-based basis function *(local)*: $\phi_j(x_{i,j}) = \begin{cases} 1 & x^l \le x_{i,j} < x^r \\ 0 & else \end{cases}$