Linear Regression

Training Data

We have N input, output pairs $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_N, y_N)\}$ where $\boldsymbol{x}_n \in \mathbb{R}^D$ where D is the dimension and $y_n \in \mathbb{R}$.

Hence we can write $\mathcal{D}$ as a matrix.

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{ND} \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}.$$

Each row i is the data point $\boldsymbol{x}_i$ and each column is refered to as a feature dimension.

Assumptions

Underlying function f is linear, so that

$$f_\theta(\boldsymbol{x}) = \theta^T \boldsymbol{x}, \text{where } \theta^T \in \mathbb{R}^D.$$

Observation y is a noisy version of f.

$$f_\theta(\boldsymbol{x}_i) \approx y$$

What we need

We want to prove that θ^T is a good approximation for $\boldsymbol{y}$.

Linear Regression

Linear regression means linear in the parameters, not in the input data.

Objective Function

In this case, our objective function is the error function we want to minimise. We use the *L2 loss function.*

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} (y_i - f_\theta(\boldsymbol{x}_i))^2 = \sum_{i=1}^{N} (y_i - \theta^T \boldsymbol{x}_i).$$

It is easy to check that

$$L(\theta) = \frac{1}{N} (\boldsymbol{y} - X\theta)^T (\boldsymbol{y} - X\theta).$$

Minimal Solution

We now prove that the minimal solution of L is $(X^T X)^{-1} X^T \boldsymbol{y}$. We first derive L with respect to θ.

$$L(\theta) = \frac{1}{N}\|\boldsymbol{y} - X\theta\|^2$$

Then, using chain rule,

$$\begin{aligned}\frac{\partial}{\partial\theta}L(\theta) &= \frac{2}{N}(\boldsymbol{y} - X\theta)\cdot(-X)\\ &= \frac{-2}{N}X^T(\boldsymbol{y} - X\theta)\end{aligned}$$

Then, setting it to 0,

$$\begin{aligned}\frac{-2}{N}X^T(\boldsymbol{y} - X\theta) = 0\\ \Rightarrow X^T(\boldsymbol{y} - X\theta) = 0\\ \Rightarrow X^T\boldsymbol{y} - X^TX\theta = 0\\ \Rightarrow \theta = \left(X^TX\right)^{-1}X^T\boldsymbol{y}.\end{aligned}$$

With Features

To provide more flexibility to the model, we may apply a nonlinear transformation on the data. (We only require the objective function to be linear in the parameters θ.) Hence, we define a matrix

$$\Phi(\boldsymbol{x}) = \begin{bmatrix}\phi_1(x_1) & \phi_2(x_1) & \ldots & \phi_D(x_1)\\ \phi_1(x_2) & \phi_2(x_2) & \ldots & \phi_D(x_2)\\ \vdots & \vdots & \ddots & \vdots\\ \phi_1(x_N) & \phi_2(x_N) & \ldots & \phi_D(x_N)\end{bmatrix}$$

where each ϕ defines a feature function. One example of this is $\phi_i = x^{i-1}$.

Then if we define $f_\theta(x) = \theta\Phi(\boldsymbol{x})$ then we have the closed-form solution $\left(\Phi^T\Phi\right)^{-1}\Phi^T\boldsymbol{y}$.

Regularisation

The idea is to penalise the error function for having larger amplitude solutions.

$$L_\lambda(\theta) = L(\theta) + \lambda\|\theta\|_p^p$$

Hyperparameters

Examples of hyperparameters include

- Degree of polynomial regression
- Number of kernels in kernel methods
- Regularisation parameter