Mathematics for Machine Learning

Prof Willie Brink

Applied Mathematics, Stellenbosch University

Lecture 8: Linear Regression

Contents of the module

9.1 Problem formulation

Assume we have a set of training inputs $\boldsymbol{x}_n \in \mathbb{R}^D$ and corresponding noisy observations $y_n = f(\boldsymbol{x}_n) + \epsilon$, where ϵ is an i.i.d. random variable that describes noise.

The task is to find f that models the training data **and** generalises well to new data.

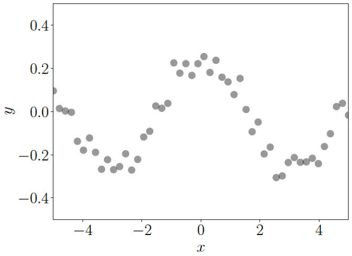
Let's assume a linear function $y = \boldsymbol{x}^T\boldsymbol{\theta} + \epsilon$, with $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Then

$$p(y \mid \boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(y \mid \boldsymbol{x}^T\boldsymbol{\theta}, \sigma^2)$$

where $\boldsymbol{\theta} \in \mathbb{R}^D$ are the parameters we seek.

We'll assume the noise variance σ^2 is known.

Once we have optimal parameters $\boldsymbol{\theta}^*$, we can predict y for any input $\boldsymbol{x}$.



9.2 Parameter estimation

We note that y_i and y_j are conditionally independent given their inputs, so that

$$p(\mathcal{Y} \,|\, \mathcal{X}, \boldsymbol{\theta}) = \prod_{n=1}^{N} \mathcal{N}(y_n \,|\, \boldsymbol{x}_n^\top \boldsymbol{\theta}, \sigma^2) \quad \text{with } \mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \text{ and } \mathcal{Y} = \{y_1, \ldots, y_N\}$$

Maximum likelihood estimation : $\theta_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} p(\mathcal{Y} \,|\, \mathcal{X}, \theta)$

We consider the negative log-likelihood, with $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^\top$ and $\boldsymbol{y} = [y_1, \ldots, y_N]^\top$:

$$\mathcal{L}(\boldsymbol{\theta}) = -\log p(\mathcal{Y} \,|\, \mathcal{X}, \boldsymbol{\theta}) = -\sum_{n=1}^{N} \left[-\frac{1}{2\sigma^2} (y_n - \boldsymbol{x}_n^\top \boldsymbol{\theta})^2 + \text{const} \right] = \frac{1}{2\sigma^2} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|^2 + \text{const}$$

We compute the gradient of $\mathcal{L}$ with respect to $\boldsymbol{\theta}$, set it to 0, and solve for $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}_{\mathrm{ML}} = \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

Note: we can fit higher order polynomials using linear regression. If our training set is $\mathcal{X} = \{x_1, \ldots, x_N\}$, we may define $\boldsymbol{X}$ as shown on the right.

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_1 & \cdots & x_1^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & \cdots & x_N^p \end{bmatrix}$$

If the noise variance is unknown, we can also use MLE, by finding $\partial \mathcal{L} / \partial \sigma^2$, setting it to 0, and solving for σ^2. In this way, $\sigma^2_{\mathrm{ML}} = \frac{1}{N} \sum_{n=1}^{N} (y_n - \boldsymbol{x}_n^\top \boldsymbol{\theta})^2$.

Unfortunately, MLE is prone to overfitting when the number of parameters is high.

Maximum a posteriori estimation : $\theta_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \,|\, \mathcal{X}, \mathcal{Y})$

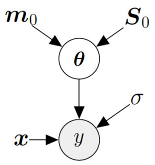
To mitigate overfitting, we place a (conjugate) Gaussian prior $\boldsymbol{\theta}$: $p(\boldsymbol{\theta}) = \mathcal{N}\mathbf{0}, b^2 \boldsymbol{I})$

Differentiate the negative log-posterior w.r.t. $\boldsymbol{\theta}$, set it to 0, and solve for $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}_{\mathrm{MAP}} = \left(\boldsymbol{X}^\top \boldsymbol{X} + \frac{\sigma^2}{b^2} \boldsymbol{I} \right)^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

9.3 Bayesian linear regression

Bayesian linear regression takes the full posterior distribution over $\boldsymbol{\theta}$ into account (instead of a point estimate).



Our model: $p(y, \boldsymbol{\theta} | \boldsymbol{x}) = p(y | \boldsymbol{x}, \boldsymbol{\theta})\, p(\boldsymbol{\theta}) = \mathcal{N}(y \mid \boldsymbol{x}^T\boldsymbol{\theta}, \sigma^2) \underbrace{\mathcal{N}(\boldsymbol{m}_0, \boldsymbol{S}_0)}_{\text{conjugate prior}}$

To make predictions at input $\boldsymbol{x}_*$, we integrate $\boldsymbol{\theta}$ out:

$$\begin{aligned} p(y_* | \boldsymbol{x}_*) &= \int p(y_* | \boldsymbol{x}_*, \boldsymbol{\theta})\, p(\boldsymbol{\theta})\, d\boldsymbol{\theta} \\ &= \mathcal{N}\left(\boldsymbol{x}_*^\top \boldsymbol{m}_0,\ \boldsymbol{x}_*^\top \boldsymbol{S}_0 \boldsymbol{x}_* + \sigma^2\right) \end{aligned}$$

When we have the parameter posterior $p(\boldsymbol{\theta} | \mathcal{X}, \mathcal{Y})$, we can replace the prior $p(\boldsymbol{\theta})$ in the above with it.