

Mathematics for Machine Learning

Prof Willie Brink

Applied Mathematics, Stellenbosch University

Lecture 8: Linear Regression

Contents of the module

Chapter 02: Linear Algebra

Chapter 03: Analytic Geometry

Chapter 04: Matrix Decompositions

Chapter 05: Vector Calculus

Chapter 06: Probability and Distributions

Chapter 07: Continuous Optimisation

Chapter 08: When Models Meet Data

Chapter 9: Linear Regression

Chapter 10: Dimensionality Reduction with Principal Component Analysis

Chapter 11: Density Estimation with Gaussian Mixture Models

Chapter 12: Classification with Support Vector Machines

9.1 Problem formulation

Assume we have a set of training inputs $\mathbf{x}_n \in \mathbb{R}^D$ and corresponding noisy observations $y_n = f(\mathbf{x}_n) + \epsilon$, 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.

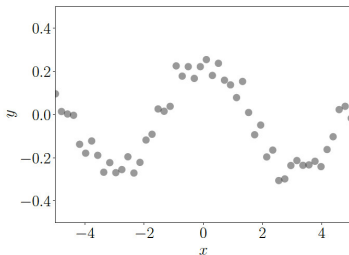
Let's assume a **linear** function $y = \mathbf{x}^T \boldsymbol{\theta} + \epsilon$, with $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Then

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

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 \mathbf{x} .



9.2 Parameter estimation

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

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

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

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

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

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

$$\boldsymbol{\theta}_{\text{ML}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

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

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

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, $\sigma_{\text{ML}}^2 = \frac{1}{N} \sum_{n=1}^N (y_n - \mathbf{x}_n^T \boldsymbol{\theta})^2$.

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

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

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

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

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

9.3 Bayesian linear regression

Bayesian linear regression takes the full posterior distribution over θ into account (instead of a point estimate).

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

To make predictions at input \mathbf{x}_* , we integrate θ out:

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

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

