Bettina Lory

Translational Neuromodelling: Exercise 1 Bayesian inference

1.1 Maximum Likelihood and Overfitting

a) polynomial model of order p:

$$y_n = \theta_0 + \theta_1 x_n + \theta_2 x_n^2 + \dots + \theta_p x_n^p + \varepsilon_n, \quad n = 1, \dots, N$$

ε_n: iid. Gaussian noise with $\mathcal{N}(0, \sigma^2)$

$\hookrightarrow \varepsilon_n = y_n - (\theta_0 + \theta_1 x_n + \theta_2 x_n^2 + \dots + \theta_p x_n^p)$ *1

probability density function:

$$f_{\mu=0,\sigma^2}(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\varepsilon_n^2}{2\sigma^2}\right) = (2\pi\sigma^2)^{-N/2} \exp\left(\frac{-1}{2\sigma^2} \sum_{n=1}^{N} \varepsilon_n^2\right) \text{ *2}$$

$$= L_\varepsilon(\mu, \sigma^2) \text{ (Likelihood function)}$$

Log-likelihood function:

$$\ell_\varepsilon(\mu, \sigma^2) = \ln L_\varepsilon(\mu, \sigma^2) \overset{*2}{=} -\frac{N}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{n=1}^{N} \varepsilon_n^2$$

$$\overset{*1}{=} -\frac{N}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - (\theta_0 + \theta_1 x_n + \dots + \theta_p x_n^p))^2$$

b) Maximum likelihood estimate

multivariant rewriting with

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \qquad X = \begin{pmatrix} x_1^0 & x_1^1 & \cdots & x_1^p \\ x_2^0 & x_2^1 & & x_2^p \\ \vdots & & \ddots & \vdots \\ x_N^0 & x_N^1 & \cdots & x_N^p \end{pmatrix} \qquad \beta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_p \end{pmatrix}$$

ML: $\hat{\beta} = \underset{\beta}{\operatorname{argmax}}\, \ell \longrightarrow \hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{n=1}^{N} (y_n - (\theta_0 + \theta_1 x_n + \dots + \theta_p x_n^p))^2$

because term is negative

$$\sum_{n=1}^{N} (y_n - (\theta_0 + \theta_1 x_n + \dots + \theta_p x_n^p))^2 \overset{\text{vectors not.}}{=} (y - X\beta)^T (y - X\beta)$$

$\hat{\beta}$ solves $\frac{\partial}{\partial \theta_j} (y - X\beta)^T (y - X\beta) \overset{!}{=} 0, \quad j = 0, \dots, p$

$$= \frac{\partial}{\partial \theta_j} \sum_{n=1}^{N} (y_n - (\theta_0 + \theta_1 x_n + \dots + \theta_p x_n^p))^2 = 2 \cdot \sum_{n=1}^{N} (y_n - (\theta_0 + \dots + \theta_p x_n^p))^2 \cdot (-x_n^j)$$

vector not

$$= -2 (X_{j\text{th column}})^T (y - X\beta) \overset{!}{=} 0$$

for all β:

$$-2 \begin{bmatrix} X_{1\text{th}}^T (y - X\beta) \\ X_{2\text{th}}^T (y - X\beta) \\ \vdots \\ X_{p\text{th}}^T (y - X\beta) \end{bmatrix} = -2 X^T (y - X\beta) \overset{!}{=} 0 \rightarrow \underline{\hat{\beta} = (X^T X)^{-1} X^T y}$$

$$X^T X \hat{\beta} = X^T y$$

c) quadratic model: $y_n = \theta_0 + \theta_1 x_n + \theta_2 x_n^2 + \varepsilon_n$

$\theta_0 = 0.3$, $\theta_1 = -0.1$, $\theta_2 = 0.5$ and $\sigma^2 = 0.001$

x goes from -0.5 to 0.2 in steps of 0.1

($x_1 = -0.5$, $x_2 = -0.4$, ...)

d) From Python calculations:

for P=1: $\theta_0 = 0.3155$ $\theta_1 = -0.2493$

for P=2: $\theta_0 = 0.30097$ $\theta_1 = -0.1041$ $\theta_2 = 0.4839$

for P=7: $\theta_0 = 0.3016$ $\theta_1 = -0.1192$ $\theta_2 = 0.4533$ ....

↳ The ML estimator for P=2 is closest to the true values of θ.
The higher order ($P=7$) cannot estimate θ better.

Using: $\log\text{-}l = -\frac{N}{2} \cdot \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \cdot (y - X\beta)^T \cdot (y - X\beta)$

P=1: log-l = 18.1436

P=2: log-l = 20.2782

P=7: log-l = 20.2795 ← biggest likelihood

likelihood grows with P.

e) For the second dataset but with the old estimates the log likelihood was:

P=1: 46.8753

P=2: 255.9787

P=7: -64.0972

For the second order estimate the likelihood for new data is the biggest and the higher order has a low likelihood, because it is overfit on the first dataset because of the high polinomials.

The P=2 is the best model, because the data is also a second order polynomial.

f) P=1: $\theta_0 = 0.3424$ $\theta_1 = -0.1001$

P=2: $\theta_0 = 0.3001$ $\theta_1 = -0.1001$ $\theta_2 = 0.4976$

P=7: $\theta_0 = 0.2998$ $\theta_1 = -0.0978$ $\theta_2 = 0.5176$ ...

The new estimates are more accurate (closer to the true values) than the old estimates.

The Log-likelihood for the new data under the new ML estimates are:

P=1: log-ℓ = 183.5815

P=2: log-ℓ = 255.9845

P=7: log-ℓ = 255.9848 ← biggest likelihood

Like for the first dataset with the first estimates, the likelihood increases with the order P → the estimates get overfit for the dataset it was trained on the higher P.