يونيرسيتي بروني دارالسلام
UNIVERSITI BRUNEI DARUSSALAM

Regression modelling using I-priors

NUS Department of Statistics & Data Science Seminar

Haziq Jamil
Mathematical Sciences, Faculty of Science, UBD
https://haziqj.ml

Wednesday, 16 November 2022

Overview

Estimation

- Hyperparameters of the model
- Computational bottleneck

Hyperparameters of the model

$$
\begin{aligned}
y_i &= f_0(x_i) + \sum_{j=1}^{n} h_\lambda(x_i, x_j) w_j + \epsilon_i \\
(\epsilon_1, \ldots, \epsilon_n)^\top &\sim \mathrm{N}_n(0, \boldsymbol{\Psi}^{-1}) \\
(w_1, \ldots, w_n)^\top &\sim \mathrm{N}_n(0, \boldsymbol{\Psi})
\end{aligned}
\tag{1}
$$

A number of hyperparameters remain undetermined. Further assumptions:

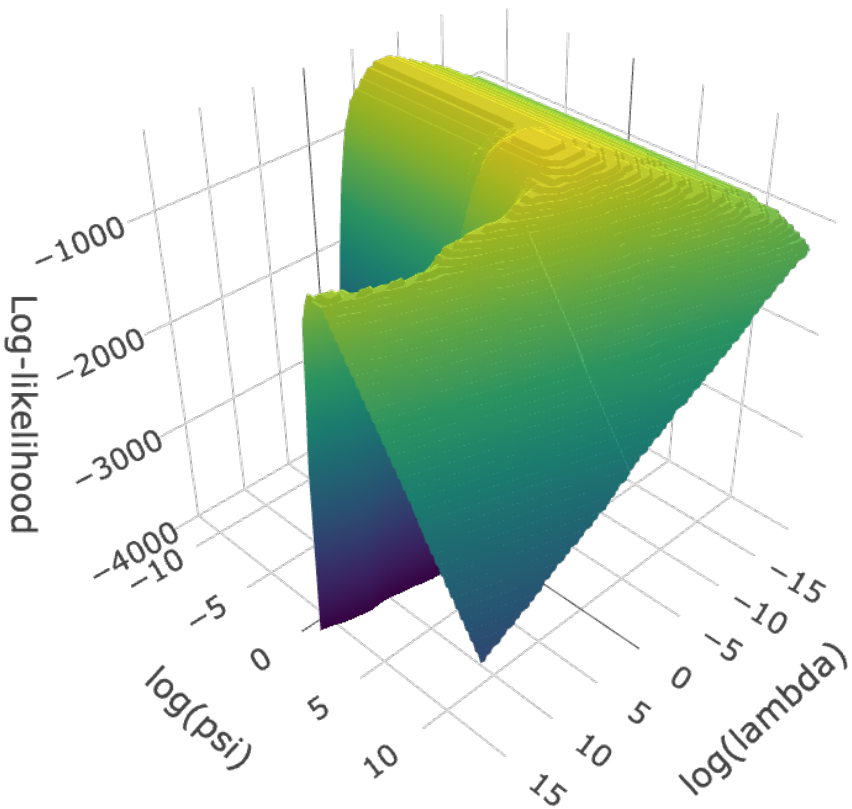
1. The error variance $\boldsymbol{\Psi}$ is known up to a low-dimensional parameter, e.g. $\boldsymbol{\Psi} = \psi \mathbf{I}_n$, $\psi > 0$ (iid errors).
2. Each RKHS $\mathcal{F}$ is defined by the kernel $h_\lambda = \lambda \tilde{h}$, where $\lambda \in \mathbb{R}$ is a scale[1] parameter.
3. Certain kernels also require tuning, e.g. the Hurst coefficient of the fBm or the lengthscale of the Gaussian. For now, assume fixed.

[1] This necessitates the use of reproducing kernel Krein spaces.

Marginal likelihood

The marginal log-likelihood of $(\lambda, \boldsymbol{\Psi})$ is

$$L(\lambda, \boldsymbol{\Psi} \mid \mathbf{y}) = \text{const.} - \frac{1}{2}\log|\mathbf{V}_y| - \frac{1}{2}(\mathbf{y}-\mathbf{f}_0)^\top \mathbf{V}_y^{-1}(\mathbf{y}-\mathbf{f}_0),$$



- Direct optimisation using e.g. conjugate gradients or Newton methods.
- Numerical stability issues–workaround: Cholesky or eigen decomposition.
- Prone to local optima.
- Possible to also optimise kernel hyperparameters.

EM algorithm

An alternative view of the model:

$$\mathbf{y} \mid \mathbf{w} \sim \mathrm{N}_n(\mathbf{f}_0 + \mathbf{H}_\lambda w, \boldsymbol{\Psi}^{-1})$$
$$\mathbf{w} \sim \mathrm{N}_n(\mathbf{0}, \boldsymbol{\Psi})$$

in which the $\mathbf{w}$ are "missing". The full data log-likelihood is

$$L(\lambda, \boldsymbol{\Psi} \mid \mathbf{y}, \mathbf{w}) = \text{const.} - \frac{1}{2}(\mathbf{y} - \mathbf{f}_0)^\top \boldsymbol{\Psi}(\mathbf{y} - \mathbf{f}_0) - \frac{1}{2}\operatorname{tr}\left(\mathbf{V}_y \mathbf{w}\mathbf{w}^\top\right) + (\mathbf{y} - \mathbf{f}_0)^\top \boldsymbol{\Psi}\mathbf{H}_\lambda \mathbf{w}$$

The E-step entails computing

$$Q_t(\lambda, \boldsymbol{\Psi}) = \mathrm{E}\left\{ L(\lambda, \boldsymbol{\Psi} \mid \mathbf{y}, \mathbf{w}) \;\middle|\; \mathbf{y}, \lambda^{(t)}, \boldsymbol{\Psi}^{(t)} \right\}$$

in which the following posterior quantities are needed

$$\hat{\mathbf{w}} := \mathrm{E}(\mathbf{w} \mid \mathbf{y}, \lambda, \boldsymbol{\Psi}) \quad \text{and} \quad \hat{\mathbf{W}} := \mathrm{E}(\mathbf{w}\mathbf{w}^\top \mid \mathbf{y}, \lambda, \boldsymbol{\Psi}) = \mathbf{V}_y^{-1} + \hat{\mathbf{w}}\hat{\mathbf{w}}^\top,$$

EM algorithm (cont.)

Let $\tilde{\mathbf{w}}^{(t)}$ and $\tilde{\mathbf{W}}^{(t)}$ be versions of $\hat{w}$ and $\hat{W}$ computed using $\lambda^{(t)}$ and $\boldsymbol{\Psi}^{(t)}$. The M-step entails solving

$$\frac{\partial Q_t}{\partial \lambda} = -\frac{1}{2}\operatorname{tr}\left(\frac{\partial \mathbf{V}_y}{\partial \lambda}\tilde{\mathbf{W}}^{(t)}\right) + (\mathbf{y}-\mathbf{f}_0)^\top \boldsymbol{\Psi}\frac{\partial \mathbf{H}_\lambda}{\partial \lambda}\tilde{\mathbf{w}}^{(t)} = 0$$

$$\frac{\partial Q_t}{\partial \psi} = -\frac{1}{2}\operatorname{tr}\left(\frac{\partial \mathbf{V}_y}{\partial \psi}\tilde{\mathbf{W}}^{(t)}\right) - \frac{1}{2}(\mathbf{y}-\mathbf{f}_0)^\top\left(\mathbf{y}-\mathbf{f}_0-2\mathbf{H}_\lambda\tilde{\mathbf{w}}^{(t)}\right) = 0$$

- This scheme admits a closed-form solution for ψ and (sometimes) for λ too (e.g. linear addition of kernels $h_\lambda = \lambda_1 h_1 + \cdots + \lambda_p h_p$).
- Sequential updating $\lambda^{(t)} \to \boldsymbol{\Psi}^{(t+1)} \to \lambda^{(t+1)} \to \cdots$ (expectation conditional maximisation, Meng and Rubin, 1993).
- Computationally unattractive for optimising kernel hyperparameters.

Computational bottleneck

In either estimation method, V_y^{-1} is computed and takes $O(n^3)$ time; storage requirements is $O(n^2)$.

Trick: low-rank matrix approximations. Suppose $H \approx QQ^\top$, where $Q \in \mathbb{R}^{n \times m}$ where $m \ll n$. Then, using the Woodbury matrix identity,

$$V_y^{-1} = (H\Psi H + \Psi^{-1})^{-1} \approx \Psi - \Psi Q\big((Q^\top \Psi Q)^{-1} + Q^\top \Psi Q\big)^{-1} Q^\top \Psi$$

is a much cheaper $O(nm^2)$ operation.