Probabilistic Solutions to Differential Equations and their Application to Riemannian Statistics — Supplementary Material —

Philipp Hennig **Søren Hauberg**
Max Planck Institute for Intelligent Systems, Spemannstraße 38, 72076 Tübingen, Germany
[philipp.hennig|soren.hauberg]@tue.mpg.de

1 Gaussian Process Posteriors

Equations (10), (12), (18) and (19) in the main paper are Gaussian process posterior distributions over the curve c arising from observations of various combinations of derivatives of c. These forms arise from the following general result.[1] Consider a Gaussian process prior distribution

$$p(c) = \mathcal{GP}(c; \mu, k) \tag{1}$$

over the function c, and observations y with the likelihood

$$p(y \,|\, c, A) = \mathcal{N}(y; Ac, \Lambda), \tag{2}$$

with a linear operator A. This includes the special cases of the selection operator $A = \delta(x - x_i)$ which selects function values $Ac = \int \delta(x - x_i)c(x)\mathrm{d}x = c(x_i)$, and the special case of derivative operators $\partial_x^n \delta(x - x_i)$ which give $Ac = \int \partial_x \delta(x - x_i)c(x)\mathrm{d}x = c^{(n)}(x_i)$. Then the posterior over any linear map Bc of the curve c (including $B = \delta(x - x_j)$, giving $Bc = c(x_j)$) is

$$p(Bc \,|\, y, A) = \mathcal{GP}(Bc; B\mu + BkA^\top(AkA^\top + \Lambda)^{-1}(y - A\mu), BkB - BkA^\top(AkA^\top + \Lambda)^{-1}AkB^\top). \tag{3}$$

And the marginal probability for y is

$$p(y \,|\, A) = \int p(y \,|\, c, A)p(c)\, dc = \mathcal{N}(y; A\mu, AkA^\top + \Lambda) \tag{4}$$

The classic example is that of the marginal posterior at $c(x_*)$ arising from noisy observations at $[c(x_1), \dots, c(x_N)]^\top$. This is the case of $B = \delta(x - x_*)$ and $A = [\delta(x - x_1), \dots, \delta(x - x_N)]^\top$, which gives

$$B\mu = \mu(x_*) \tag{5}$$

$$A\mu = [\mu(x_1), \dots \mu(x_N)]^\top \tag{6}$$

$$BkA^\top = \left[\iint \delta(a - x_*)k(a, b)\delta(b - x_i)\, da\, db\right]_{i=1,\dots,N} = [k(x_*, x_1), \dots k(x_*, x_N)] \tag{7}$$

$$AkA^\top = \begin{pmatrix} k(x_1, x_1) & \cdots & k(x_1, x_N) \\ \vdots & \ddots & \vdots \\ k(x_N, x_1) & \cdots & k(x_N, x_N) \end{pmatrix} \tag{8}$$

and so on. All the Gaussian forms in the paper are special cases with various combinations of A and B.

[1] Equation A.6 in C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, 2006

2 Covariance Functions

The models in the paper assume a squared-exponential (aka. radial basis function, Gaussian) covariance function between values of the function $\boldsymbol{f} : \mathbb{R} \to \mathbb{R}^N$, of the form

$$\operatorname{cov}(f_i(t), f_j(t')) = V_{ij} \exp\left(-\frac{(t-t')^2}{2\lambda^2}\right) =: V_{ij} k_{tt'} \tag{9}$$

The calculations require the covariance between various combinations of derivatives of the function. For clear notation, we'll use the operator $\partial := \partial/\partial t$, and the abbreviation $\delta_{tt'} := (t-t')/\lambda^2$

$$\operatorname{cov}(f_i(t), \dot{f}_j(t')) = V_{ij} k_{tt'} \partial^\top = V_{ij} \frac{t-t'}{\lambda^2} k_{tt'} = V_{ij} \delta_{tt'} k_{tt'} = -\operatorname{cov}(\dot{f}_i(t), f_j(t')) \tag{10}$$

$$\operatorname{cov}(\dot{f}_i(t), \dot{f}_j(t')) = V_{ij} \partial k_{tt'} \partial^\top = V_{ij} \left(\frac{1}{\lambda^2} - \left(\frac{t-t'}{\lambda^2}\right)^2\right) k_{tt'} = V_{ij} \left(\frac{1}{\lambda^2} - \delta_{tt'}^2\right) k_{tt'} \tag{11}$$

$$\operatorname{cov}(f_i(t), \ddot{f}_j(t')) = V_{ij} k_{tt'} \partial^\top \partial^\top = V_{ij} \left(\left(\frac{t-t'}{\lambda^2}\right)^2 - \frac{1}{\lambda^2}\right) k_{tt'} = V_{ij} \left(\delta_{tt'}^2 - \frac{1}{\lambda^2}\right) k_{tt'} = -\operatorname{cov}(\dot{f}_i(t), \dot{f}_j(t')) \tag{12}$$

$$\operatorname{cov}(\dot{f}_i(t), \ddot{f}_j(t')) = V_{ij} \partial k_{tt'} \partial^\top \partial^\top = V_{ij} \left(\frac{2}{\lambda^2}\frac{t-t'}{\lambda^2} - \frac{t-t'}{\lambda^2}\left(\left(\frac{t-t'}{\lambda^2}\right)^2 - \frac{1}{\lambda^2}\right)\right) k_{tt'} = V_{ij} \left(-\delta_{tt'}^3 + \frac{3}{\lambda^2}\delta_{tt'}\right) k_{tt'} \tag{13}$$

$$\operatorname{cov}(\ddot{f}_i(t), \ddot{f}_j(t')) = V_{ij} \partial\partial k_{tt'} \partial^\top \partial^\top = V_{ij} \left(\delta_{tt'}^4 - \frac{6}{\lambda^2}\delta_{tt'}^2 + \frac{3}{\lambda^4}\right) k_{tt'} \tag{14}$$

Of course, all those derivatives retain the Kronecker structure of the original kernel, because $\partial(V \otimes k) = V \otimes \partial k$.

3 Inferring Hyperparameters

Perhaps the most widely used way to learn hyperparameters for Gaussian process models it type-II maximum likelihood estimation, also known as evidence maximisation: The marginal probability for the observations y is $p(y \mid \lambda) = \int p(y \mid c) p(c \mid \lambda)\, dc = \mathcal{N}(y; \ddot{\mu}_T, \partial\partial k_{TT}(\lambda)\partial\partial + \Lambda)$. Using the shorthand $G := (\partial\partial k_{TT}(\lambda)\partial\partial + \Lambda)$, its logarithm is

$$-2 \log p(y \mid \lambda) = (y - \ddot{\mu}_T)^\top G^{-1} (y - \ddot{\mu}_T) + \log|G| + N \log 2\pi \tag{15}$$

To optimise this expression with respect to the length scale λ, we use

$$-2\frac{\partial \log p(y \mid \lambda)}{\partial \lambda^2} = -(y - \ddot{\mu}_T)^\top G^{-1} \frac{\partial G}{\partial \lambda^2} G^{-1} (y - \ddot{\mu}_T) + \operatorname{tr}(G^{-1}\frac{\partial G}{\partial \lambda^2}). \tag{16}$$

From Equation (14), and using

$$\frac{\partial \delta_{tt'}}{\partial \lambda^2} = -\frac{\delta_{tt'}}{\lambda^2} \qquad\qquad \frac{\partial k_{tt'}}{\partial \lambda^2} = k_{tt'} \frac{\delta_{tt'}^2}{2} \tag{17}$$

we find

$$\frac{\partial G_{tt'}^{ij}}{\partial \lambda^2} = V_{ij} \left[\left(-\frac{4}{\lambda^2}\delta_{tt'}^4 + \frac{18}{\lambda^4}\delta_{tt'}^2 + \frac{6}{\lambda^6}\right) k_{tt'} + \partial\partial k_{tt'} \partial^\top \partial^\top \frac{\delta_{tt'}^2}{2}\right] \tag{18}$$

$$= V_{ij} \left(\frac{\delta^6}{2} - \frac{7}{\lambda^2}\delta_{tt'}^4 + \frac{39}{2\lambda^4}\delta_{tt'}^2 + \frac{6}{\lambda^6}\right) k_{tt'} \tag{19}$$

It is also easy to evaluate the second derivative, giving

$$-2\frac{\partial^2 \log p(y \mid \lambda)}{(\partial \lambda^2)^2} = 2(y - \ddot{\mu}_T)^\top G^{-1} \frac{\partial G}{\partial \lambda^2} G^{-1} \frac{\partial G}{\partial \lambda^2} G^{-1} (y - \ddot{\mu}_T) - \operatorname{tr}\left[\frac{\partial G}{\partial \lambda^2} G^{-1} \frac{\partial G}{\partial \lambda^2} G^{-1}\right] \tag{20}$$

$$- (y - \ddot{\mu}_T)^\top G^{-1} \frac{\partial^2 G}{(\partial \lambda^2)^2} G^{-1} (y - \ddot{\mu}_T) + \operatorname{tr}\left[G^{-1} \frac{\partial^2 G}{(\partial \lambda^2)^2}\right] \tag{21}$$

$$\text{where} \qquad \frac{\partial^2 G_{tt'}^{ij}}{(\partial \lambda^2)^2} = V_{ij} \left(-\frac{3}{\lambda^2}\delta_{tt'}^6 + \frac{35}{\lambda^4}\delta_{tt'}^4 - \frac{78}{\lambda^6}\delta_{tt'}^2 + \frac{18}{\lambda^8}\right) k_{tt'} + \frac{\delta^2}{2}\frac{\partial G_{tt'}^{ij}}{\partial \lambda^2} \tag{22}$$

This allows constructing a Newton-Raphson optimisation scheme for the length scale of the algorithm.