Bridging the gap between data-driven and mechanistic modelling: the latent force model approach

Mauricio A. Álvarez

Centre for AI Fundamentals

The University of Manchester

Contents

Gaussian process

- **Definition.**

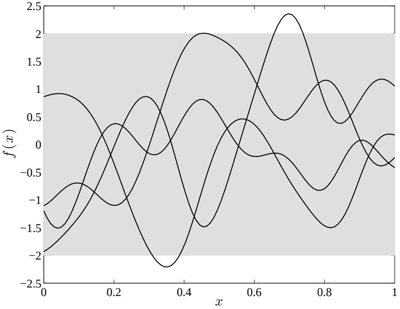
  A Gaussian process (GP) is an stochastic process with the property that any finite number of random variables taken from the process follows a joint Gaussian distribution.

- The GP is specified as

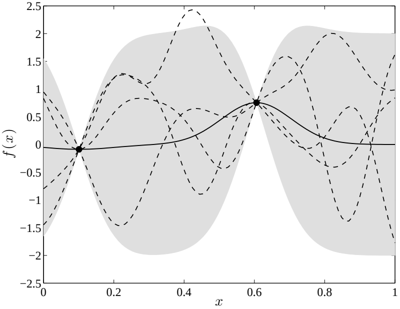
$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')).$$

- Common use: prior over functions in Bayesian modelling.

Bayesian modelling



Prior



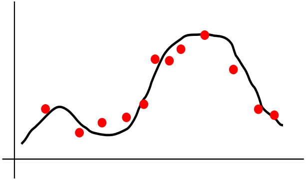
Posterior

Contents

Valid covariance functions for multiple outputs (I)

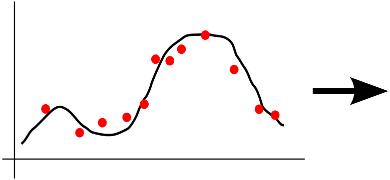
Gaussian process priors for multiple outputs have been thoroughly studied in the spatial analysis and geostatistics literature

Valid covariance functions for multiple outputs (II)



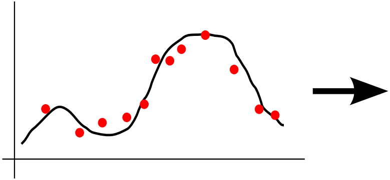
$\mathcal{D}^1 = \{(\mathbf{x}_i^1, y_i^1) | i = 1, \ldots, N_1\}$

Valid covariance functions for multiple outputs (II)



$\mathcal{D}^1 = \{(\mathbf{x}_i^1, y_i^1) | i = 1, \ldots, N_1\}$

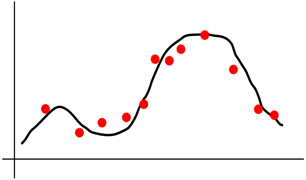
Valid covariance functions for multiple outputs (II)



$k^1(\mathbf{x}_i, \mathbf{x}_j)$

$\mathcal{D}^1 = \{(\mathbf{x}_i^1, y_i^1) | i = 1, \ldots, N_1\}$

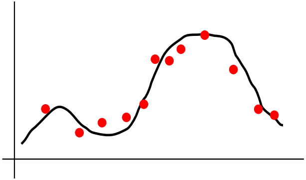
Valid covariance functions for multiple outputs (II)



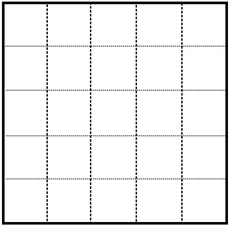
$k^1(\mathbf{x}_i, \mathbf{x}_j)$

$\mathcal{D}^1 = \{(\mathbf{x}_i^1, y_i^1) | i = 1, \ldots, N_1\}$

Valid covariance functions for multiple outputs (II)

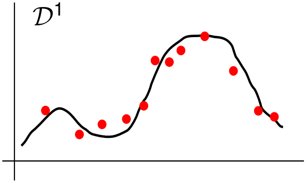


$\longrightarrow k^1(\mathbf{x}_i, \mathbf{x}_j) \longrightarrow \mathbf{K}^1 =$

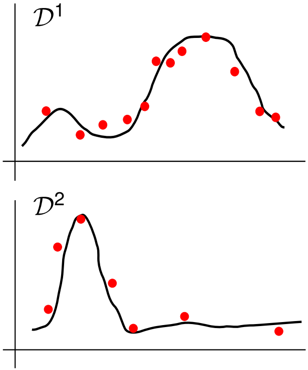


$$\mathcal{D}^1 = \{(\mathbf{x}_i^1, y_i^1) | i = 1, \ldots, N_1\}$$

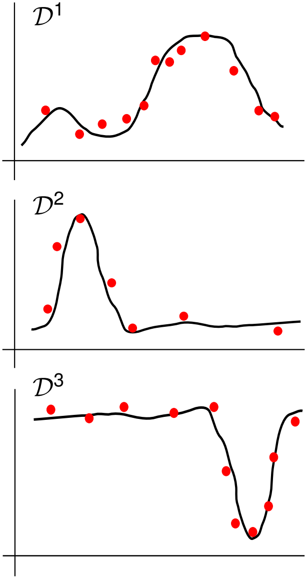
Valid covariance functions for multiple outputs (II)



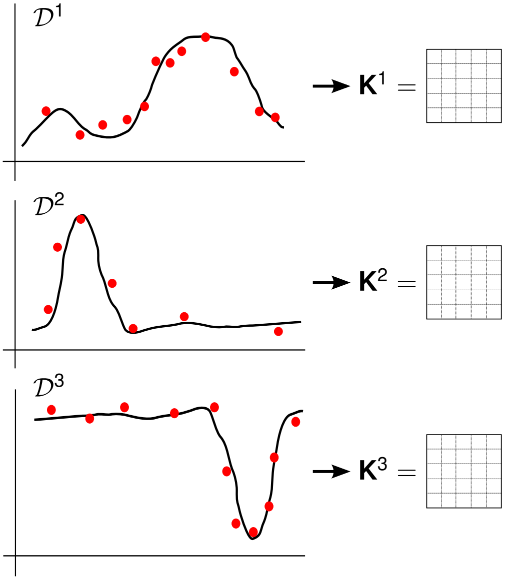
Valid covariance functions for multiple outputs (II)



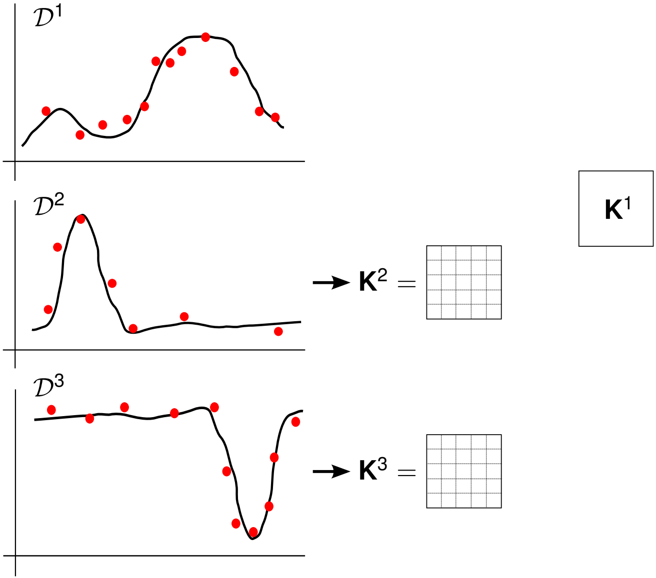
Valid covariance functions for multiple outputs (II)



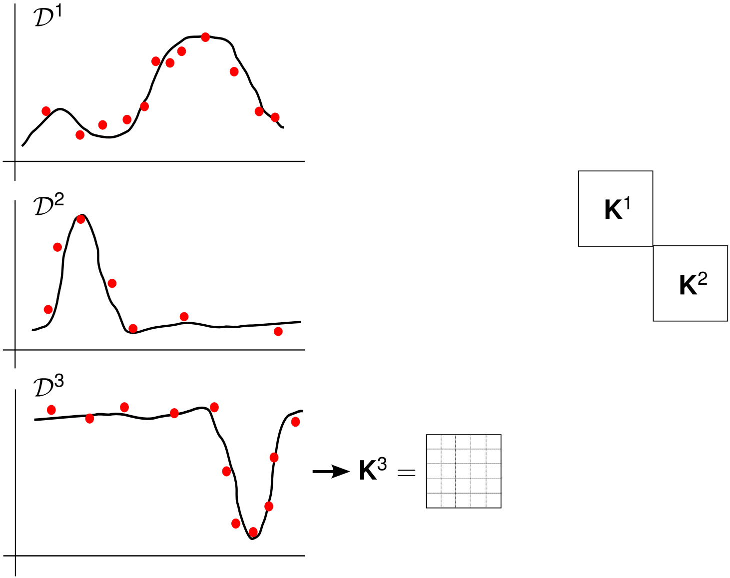
Valid covariance functions for multiple outputs (II)



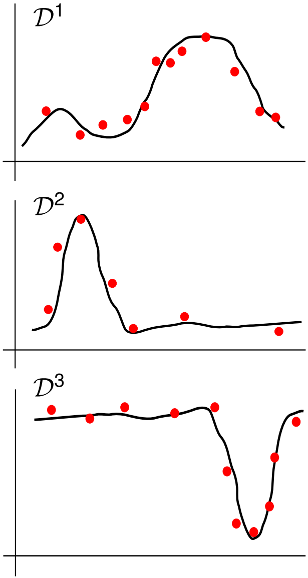
Valid covariance functions for multiple outputs (II)

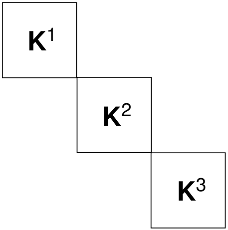


Valid covariance functions for multiple outputs (II)

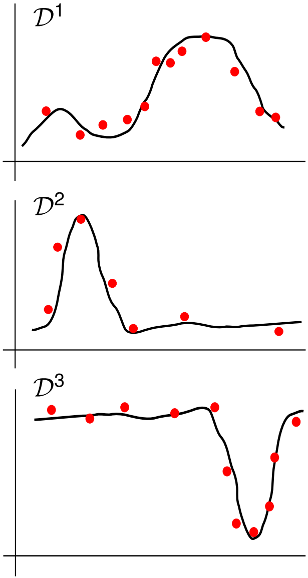


Valid covariance functions for multiple outputs (II)



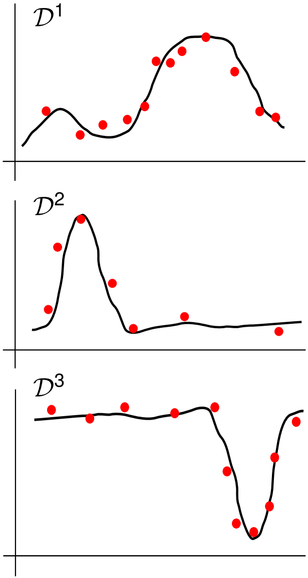


Valid covariance functions for multiple outputs (II)



$$\mathbf{K} = \begin{bmatrix} \mathbf{K}^1 & & \\ & \mathbf{K}^2 & \\ & & \mathbf{K}^3 \end{bmatrix}$$

Valid covariance functions for multiple outputs (II)



Joint covariance

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}^1 & ? & ? \\ ? & \mathbf{K}^2 & ? \\ ? & ? & \mathbf{K}^3 \end{bmatrix}$$

K be a valid covariance matrix

Contents

Building new kernels

- Covariance function: convolution integrals between Green's functions associated with differential operators, and covariance functions associated with latent functions.

- Back in 2009, we named this way of building covariance functions for GPs as latent force models (LFM).

- An LFM is a Gaussian process with a covariance function inspired by a differential operator.

Process convolutions

- Consider a set of functions $\{f_d(\mathbf{x})\}_{d=1}^{D}$.
- Each function can be expressed as

$$f_d(\mathbf{x}) = \int_{\mathcal{X}} G_d(\mathbf{x}-\mathbf{z})u(\mathbf{z})\mathrm{d}\mathbf{z} = G_d(\mathbf{x}) * u(\mathbf{x}).$$

- If $u(\mathbf{x})$ is a GP, then $f_d(\mathbf{x})$ is also a GP.
- We could also include more latent processes $u_1(\mathbf{x}), u_2(\mathbf{x}), \ldots$

$$f_d(\mathbf{x}) = \sum_{q=1}^{Q} \int_{\mathcal{X}} G_{d,q}(\mathbf{x}-\mathbf{z})u_q(\mathbf{z})\mathrm{d}\mathbf{z},$$

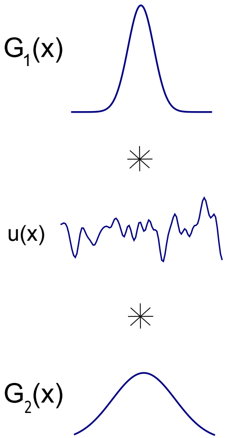
where $\mathrm{cov}[u_q(\mathbf{z}), u_{q'}(\mathbf{z}')] = k_q(\mathbf{z},\mathbf{z}')\delta_{q,q'}$.

Process convolutions

u(x) 

u(x): latent function.

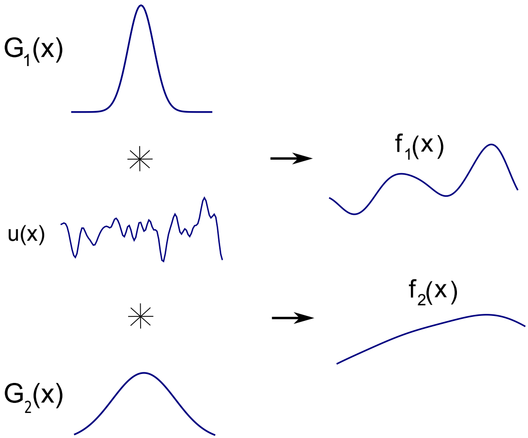
Process convolutions



u(x): latent function.

G(x): smoothing kernel.

Process convolutions



$u(x)$: latent function.

$G(x)$: smoothing kernel.

$f(x)$: output function.

Covariance function

Assume we have D outputs, $\{f_d(\mathbf{x})\}_{d=1}^{D}$. The covariance between $f_d(\mathbf{x})$ and $f_{d'}(\mathbf{x}')$ follows

$$k_{f_d,f_{d'}}(\mathbf{x},\mathbf{x}') = \sum_{q=1}^{Q} \int_{\mathcal{X}} G_{d,q}(\mathbf{x}-\mathbf{z}) \int_{\mathcal{X}} G_{d',q}(\mathbf{x}'-\mathbf{z}')k_q(\mathbf{z},\mathbf{z}')\mathrm{d}\mathbf{z}'\mathrm{d}\mathbf{z}.$$

Latent Force Models

- Suppose we want to model the outputs of different dynamical systems driven by a set of latent functions.

- Each output is given by

$$\mathcal{D}_0^M f_d = \sum_{m=0}^{M} A_{m,d} \frac{\mathrm{d}^m}{\mathrm{d}t^m}[f_d(t)] = \sum_{q=1}^{Q} S_{d,q} u_q(t),$$

where we have introduced an operator $\mathcal{D}_0^M$ that is equivalent to applying the weighted sum of operators $\frac{\mathrm{d}^m}{\mathrm{d}t^m}$.

Green's functions

- The operator $\mathcal{D}_0^M$ is related to a so called *Green's function* $G_d(t,s)$ by

$$\mathcal{D}_0^M[G_d(t,s)] = \delta(t-s),$$

with s fixed.

- The solution for $f_d(t)$ can be written in terms of the Green's function like

$$f_d(t) = \sum_{q=1}^{Q} S_{d,q} f_{d,q}(t, u_q(t)),$$

with

$$f_{d,q}(t, u_q(t)) = \int_{\mathcal{T}} G_d(t,\tau) u_q(\tau) \mathrm{d}\tau.$$

Covariance for the outputs

- We assume that the latent functions $\{u_q(t)\}_{q=1}^{Q}$ are independent.
- We also assume that each $u_q(t)$ follows a Gaussian process prior, this is, $u_q(t) \sim \mathcal{GP}(0, k_q(t, t'))$.
- The covariance $\operatorname{cov}[f_d(t), f_{d'}(t')]$ is the given as

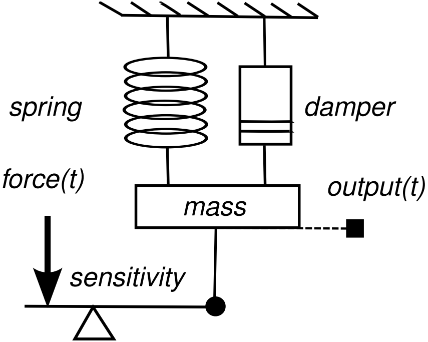
$$\sum_{q=1}^{Q} S_{d,q} S_{d',q} \int_{\mathcal{T}} \int_{\mathcal{T}'} G_d(t-\tau) G_{d'}(t'-\tau') k_q(\tau, \tau') \mathrm{d}\tau' \mathrm{d}\tau.$$

- We define $k_{f_d^q, f_{d'}^q}(t, t') = \operatorname{cov}[f_{d,q}(t, u_q(t)), f_{d',q}(t', u_q(t))]$,

$$k_{f_d^q, f_{d'}^q}(t, t') = \int_{\mathcal{T}} \int_{\mathcal{T}'} G_d(t-\tau) G_{d'}(t'-\tau') k_q(\tau, \tau') \mathrm{d}\tau' \mathrm{d}\tau.$$

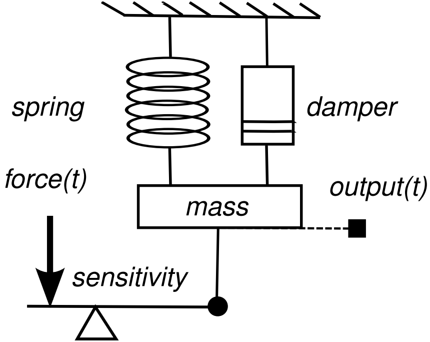
Contents

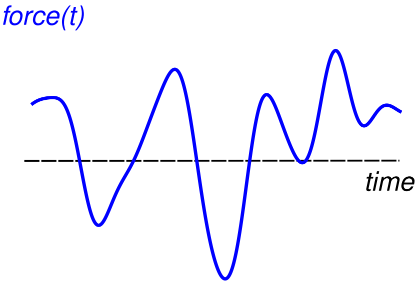
LFM



$force(t) \sim \mathcal{GP}(0, k(t,t'))$

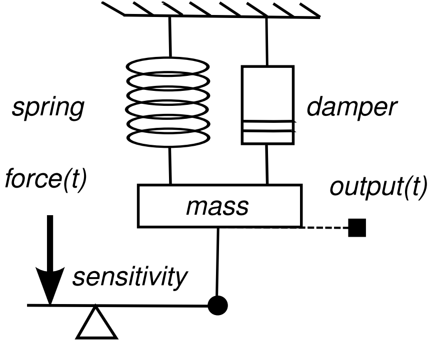
LFM



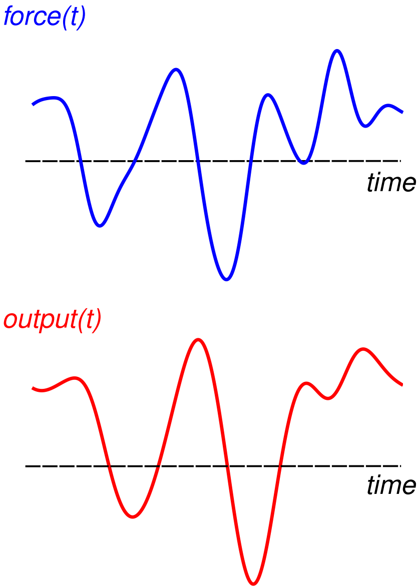


$$force(t) \sim \mathcal{GP}(0, k(t,t'))$$

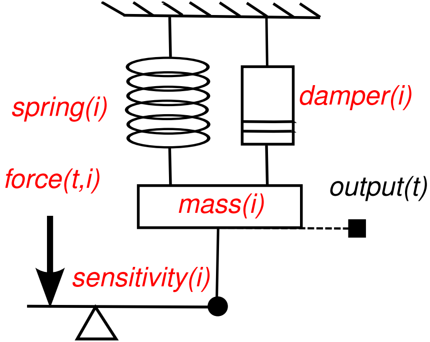
LFM



$$\text{force}(t) \sim \mathcal{GP}(0, k(t,t'))$$

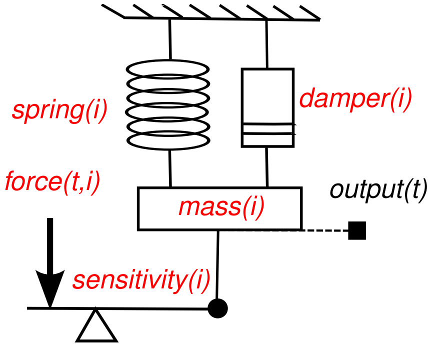


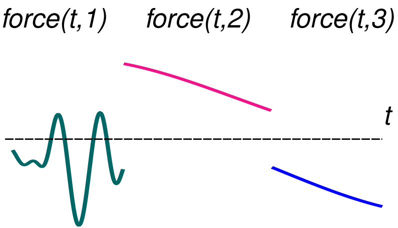
Switched LFM



force(t,i) $\sim \mathcal{GP}$*(0, k(t,t',i))*

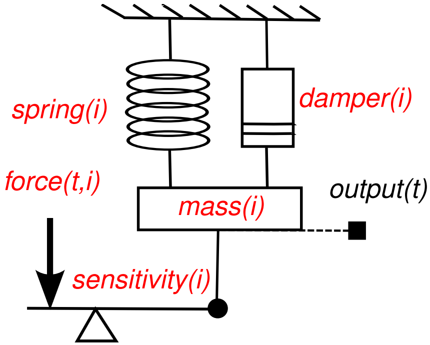
Switched LFM

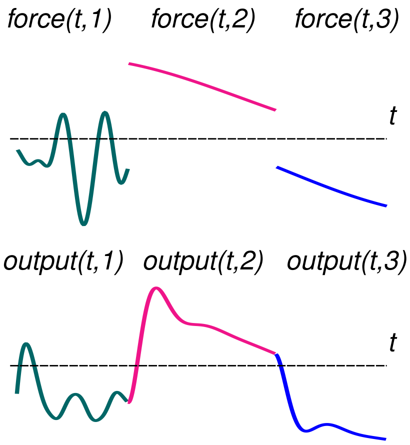




$$\text{force}(t,i) \sim \mathcal{GP}(0, k(t,t',i))$$

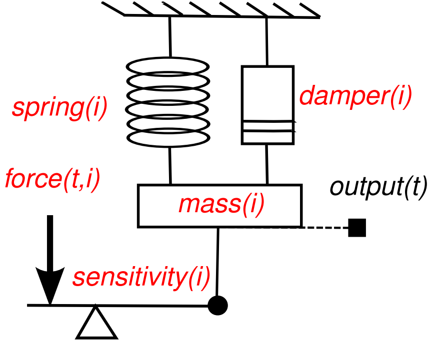
Switched LFM



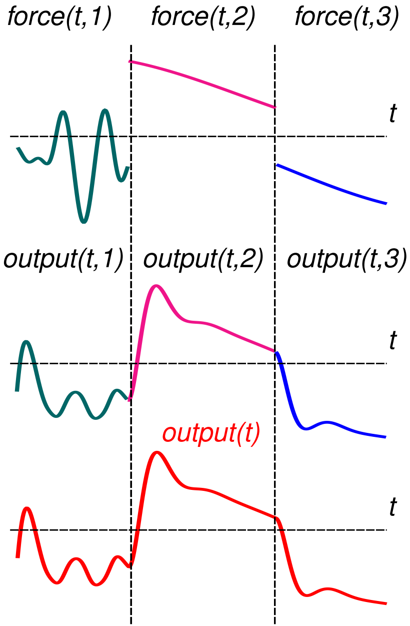


$$\mathit{force}(t,i) \sim \mathcal{GP}(0,\ k(t,t',i))$$

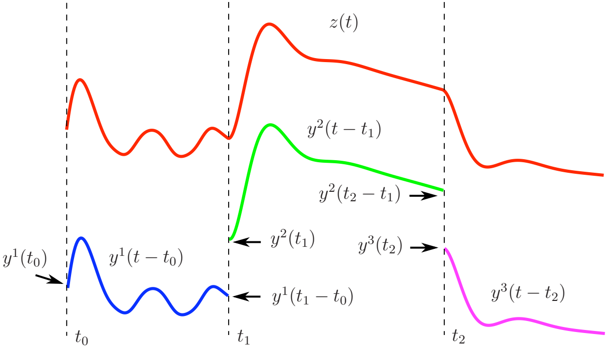
Switched LFM



$$force(t,i) \sim \mathcal{GP}(0, k(t,t',i))$$



Including initial conditions



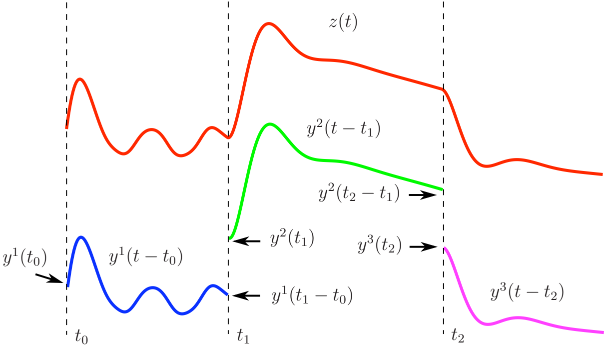
$$y_d(t) = y_d(0)c_d(t) + \dot{y}_d(0)e_d(t) + S_d f_d(t,u),$$

where

$$c_d(t) = e^{-\alpha_d t}\left[\cos(\omega_d t) + \frac{\alpha_d}{\omega_d}\sin(\omega_d t)\right], \quad e_d(t) = \frac{e^{-\alpha_d t}}{\omega_d}\sin(\omega_d t),$$

$$f_d(t,u) = \int_0^t \frac{1}{M_d\omega_d} e^{-\alpha_d(t-\tau)}\sin[(t-\tau)\omega_d]u(\tau)\mathrm{d}\tau.$$

Continuous in the outputs

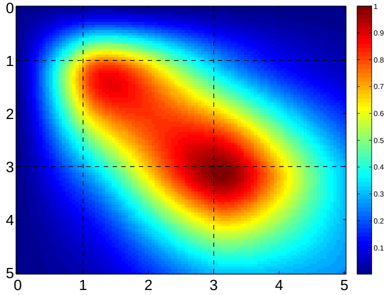


$$z_d(t) = c_d^i(t - t_{i-1}) y_d^i(t_{i-1}) + e_d^i(t - t_{i-1}) \dot{y}_d^i(t_{i-1}) + S_{d,i} f_d^i(t - t_{i-1}, u_{i-1}).$$

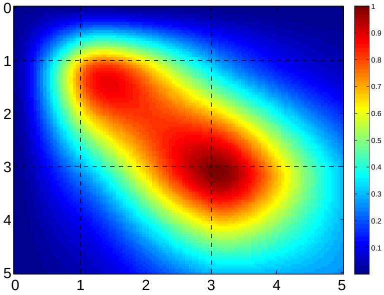
We need to compute

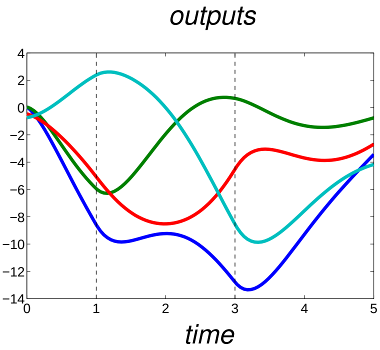
$$\mathrm{cov}[z_d(t), z_{d'}(t')].$$

Covariance and Samples



Covariance and Samples

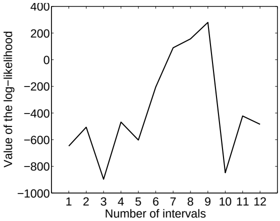




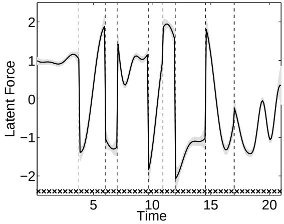
Segmentation of human movement (I)

- The task is to segment discrete movements related to motor primitives.
- Data collection was performed using a Barrett WAM robot as haptic input device, with 7 DOF.

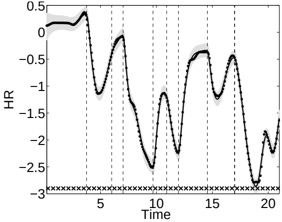
Segmentation of human movement (II)



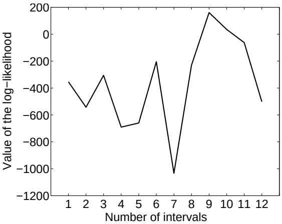
Log-Likelihood Try 1.



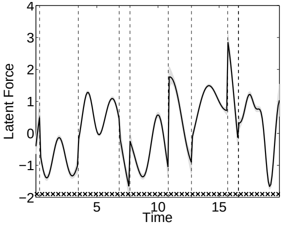
Latent force Try 1.



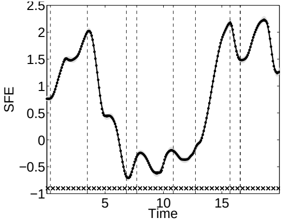
HR Output Try 1.



Log-Likelihood Try 2.



Latent force Try 2.



SFE Output Try 2.

Álvarez et al. NeurIPS (2010).

Contents

Contents

Covariance for LFMs using RFF (I)

Cristian Guarnizo

- The covariance function we need to compute in LFMs is

$$\sum_{q=1}^{Q} S_{d,q} S_{d',q} \int_0^t G_d(t-\tau) \int_0^{t'} G_{d'}(t'-\tau') k_q(\tau,\tau') \mathrm{d}\tau' \mathrm{d}\tau.$$

- We use Bochner theorem to represent $k_q(\tau,\tau')$.

- The cross-covariance function for the LFM, $k_{f_d f_{d'}}(t,t')$ is then

$$\sum_{q=1}^{Q} S_{d,q} S_{d',q} \int_0^t G_d(t-\tau) \int_0^{t'} G_{d'}(t'-\tau') \int p(\lambda) e^{j(\tau-\tau')\lambda} d\lambda \mathrm{d}\tau' \mathrm{d}\tau.$$

Covariance for LFMs using RFF (II)

- The cross-covariance function for the LFM, $k_{f_d f_{d'}}(t, t')$ is then

$$\sum_{q=1}^{Q} S_{d,q} S_{d',q} \int_0^t G_d(t-\tau) \int_0^{t'} G_{d'}(t'-\tau') \int p(\lambda) e^{j(\tau-\tau')\lambda} d\lambda \mathrm{d}\tau' \mathrm{d}\tau.$$

- Organizing the above expression we obtain

$$\sum_{q=1}^{Q} S_{d,q} S_{d',q} \int p(\lambda) v_d(t, \theta_d \lambda) v^*_{d'}(t', \theta_{d'}, \lambda) d\lambda,$$

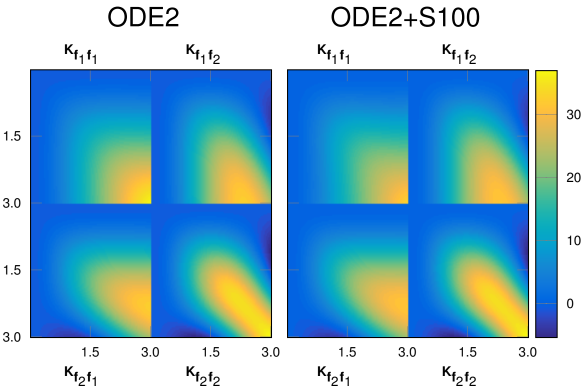
  with

$$v_d(t, \theta_d, \lambda) = \int_0^t G_d(t-\tau) e^{j\lambda\tau} \mathrm{d}\tau,$$

  where θ_d makes reference to the parameters of $G_d(\cdot)$.
- We refer to $v_d(t, \theta_d, \lambda)$ as a *random Fourier response feature*.

Comparison of kernel matrices

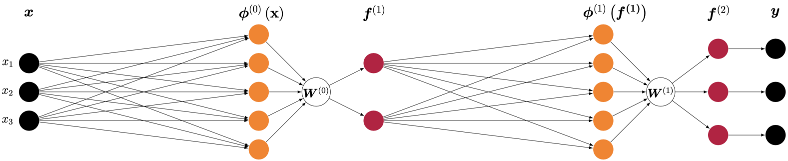
- Responses: $f_1(\cdot)$ is overdamped, $f_2(\cdot)$ is underdamped.
- Input times comprises 100 values in the range from 0s to 3s for each output.
- For $S = 100$, the Frobenius norm between the covariance matrices is 239.1. For $S = 10^5$ samples, the Frobenius norm is 5.8.



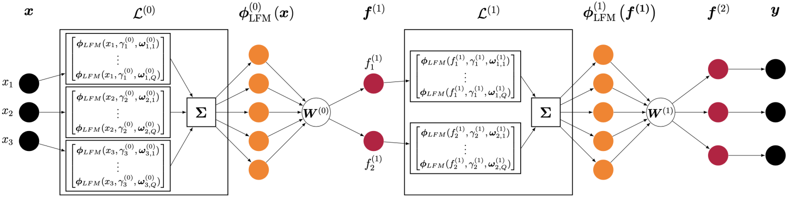
Guarnizo and Álvarez at UAI (2018).

Contents

Deep LFMs

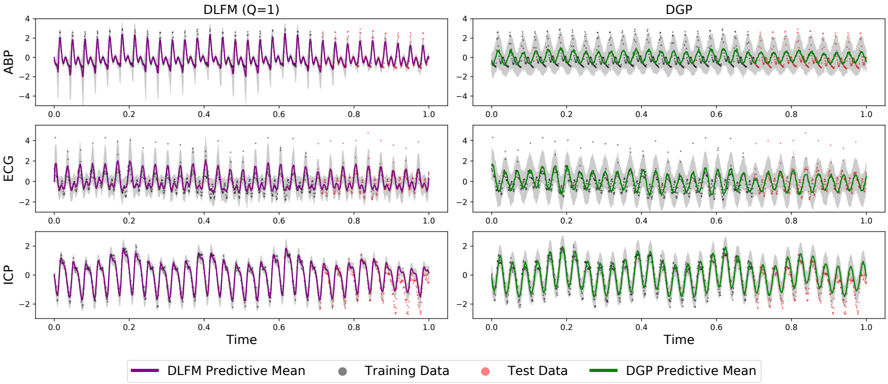


(a) Deep Gaussian process (DGP)

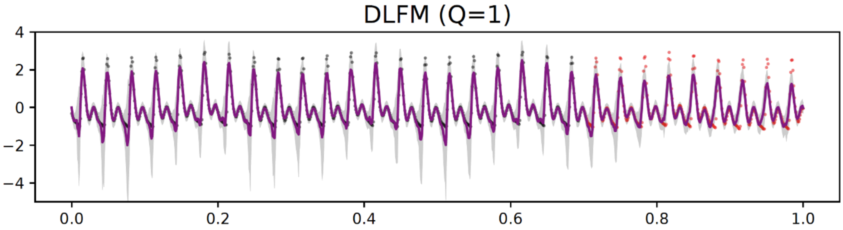


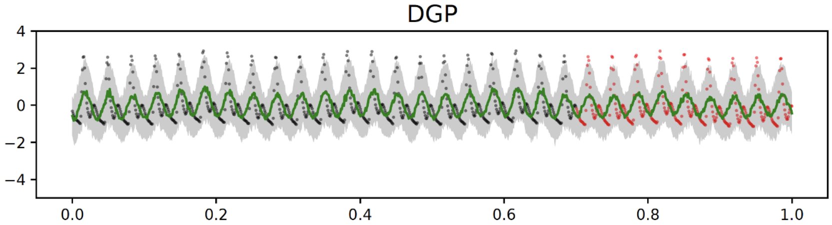
(b) Deep latent force model (DLFM)

Thomas M McDonald



McDonald and Álvarez at NeurIPS (2021).





McDonald and Álvarez at NeurIPS (2021).

Contents

Non-linear process convolutions

- For a given output dimension, d, we approximate the function with a truncated Volterra series as follows

$$f_d^{(C)}(t) = \sum_{c=1}^{C} \int \cdots \int G_d^{(c)}(t-\tau_1, \ldots, t-\tau_c) \prod_{j=1}^{c} u(\tau_j) \mathrm{d}\tau_j,$$

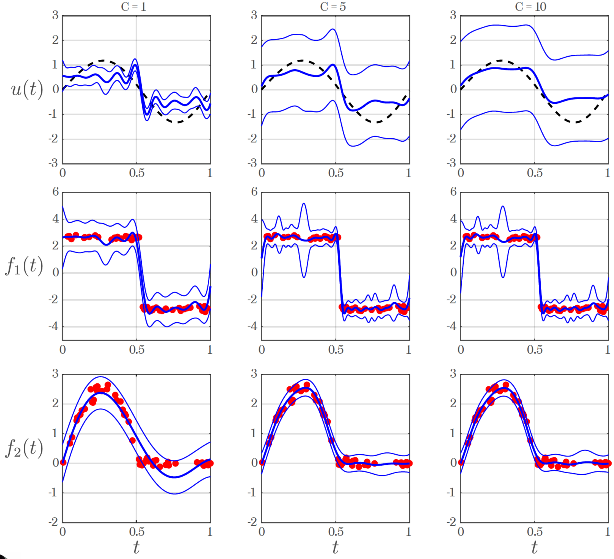
where $G_d^{(c)}$ are c^{th} degree Volterra kernels.

- In contrast to the linear case, the output $f_d^{(C)}$ is no longer a GP.

- We approximate $f_d^{(C)}$ with a GP $\tilde{f}_d^{(C)}(t)$

$$\tilde{f}_d^{(C)}(t) \sim \mathcal{GP}(\mu_d^{(C)}(t), k_{d,d'}^{(C)}(t,t')),$$

where $\mu_d^{(C)}(t) = \mathrm{E}[f_d^{(C)}(t)]$ and $k_{f_d f_{d'}}^{(C)}(t,t') = \mathrm{cov}[f_d^{(C)}(t), f_{d'}^{(C)}(t')]$.



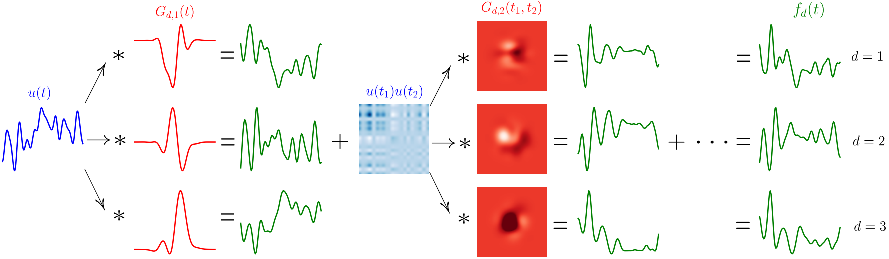
Wil Ward

Cristian Guarnizo

Álvarez et al. at AISTATS (2019).

Contents

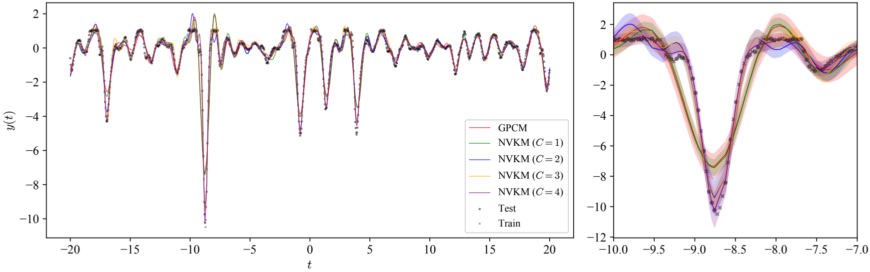
Non-parametric Volterra kernels



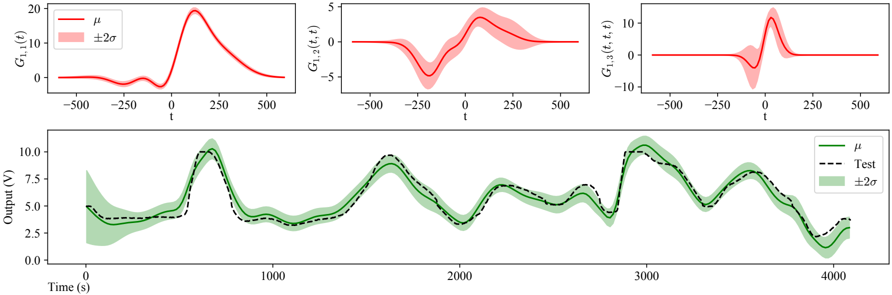
$$\int G_{d,1}(t-\tau)u(\tau)d\tau \quad + \quad \int G_{d,2}(t-\tau_1, t-\tau_2)u(\tau_1)u(\tau_2)d\tau_1 d\tau_2 \quad + \quad \ldots \quad = \quad f_d(t)$$

Magnus Ross

Mike Smith



Ross et al. at NeurIPS (2021).



Ross et al. at NeurIPS (2021).

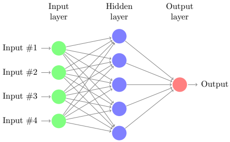
Vision: learning a *continuum* of “physics” from data

White-box modelling

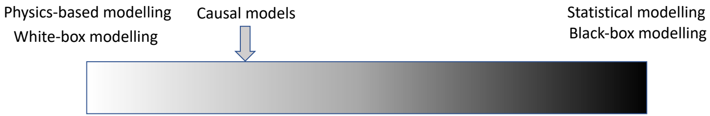
Black-box modelling

$$\frac{\partial f(x,t)}{\partial t} = \gamma \frac{\partial^2 f(x,t)}{\partial x^2} - \lambda f(x,t) + u(x,t)$$

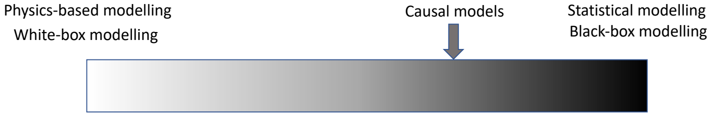
$$\frac{\partial^2 f(x,t)}{\partial t^2} = \beta \left[\frac{\partial^2 f(x,t)}{\partial x^2} + \frac{\partial^2 f(x,t)}{\partial y^2} + \frac{\partial^2 f(x,t)}{\partial z^2} \right] + u(x,t)$$



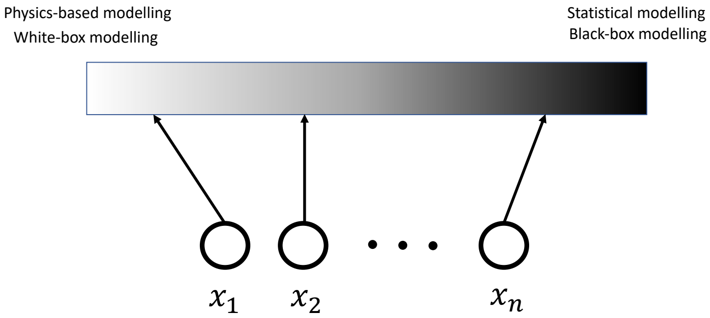
Vision: learning a *continuum* of “physics” from data



Vision: learning a *continuum* of “physics” from data



Vision: learning a *continuum* of "physics" from data



Summary

- LFMs incorporate dynamical systems and PDEs in a kernel function.
- Random Fourier features allow for fast computation for LFMs.
- LFMs with Volterra Series for non-linear systems.
- It is possible to learn non-parametric versions of these models.

Acknowledgements

EPSRC
Engineering and Physical Sciences Research Council

Grants EP/R034303/1 and EP/T00343X/1.