

# Orthogonal Bases for Multi-Output Gaussian Processes

Wessel Bruinsma

University of Cambridge, CBL

5 June 2019

# Contents

- I. Introduction
- II. The Linear Mixing Model
- III. The Orthogonal Linear Mixing Model
- IV. Arbitrary Likelihoods
- V. (Preliminary) Conclusions

# Introduction

- A powerful and popular probabilistic modelling framework for nonlinear functions.
- **Definition:**  $f \sim \mathcal{GP}(m, k)$  if, for all  $(t_1, \dots, t_n) \in \mathcal{T}^n$ ,

$$\begin{bmatrix} f(t_1) \\ \vdots \\ f(t_n) \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} m(t_1) \\ \vdots \\ m(t_n) \end{bmatrix}, \begin{bmatrix} k(t_1, t_1) & \cdots & k(t_1, t_n) \\ \vdots & \ddots & \vdots \\ k(t_n, t_1) & \cdots & k(t_n, t_n) \end{bmatrix} \right).$$

- Inference and learning:  $O(n^3)$  time and  $O(n^2)$  memory.

- Multi-output GPs go long way back (Matheron, 1969).
- Vector-valued** mean function  $m$  and **matrix-valued** kernel  $K$ :

$$m: \mathcal{T} \rightarrow \mathbb{R}^p, \quad K: \mathcal{T}^2 \rightarrow \mathbb{R}^{p \times p}, \quad \begin{array}{c} \leftarrow \text{number of} \\ \text{outputs} \end{array}$$

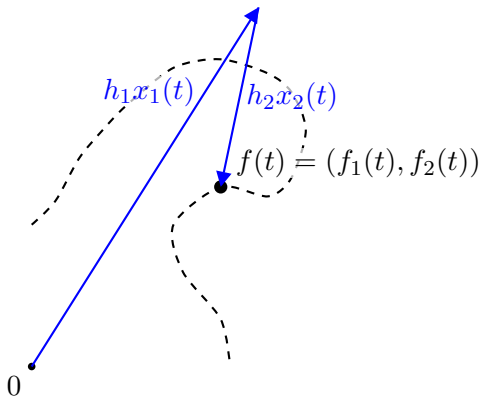
↑  
input space  
(time)

$$m(t) = \begin{bmatrix} \mathbb{E}[f_1(t)] \\ \vdots \\ \mathbb{E}[f_p(t)] \end{bmatrix},$$

$$K(t, t') = \begin{bmatrix} \text{cov}(f_1(t), f_1(t')) & \cdots & \text{cov}(f_1(t), f_p(t')) \\ \vdots & \ddots & \vdots \\ \text{cov}(f_p(t), f_1(t')) & \cdots & \text{cov}(f_p(t), f_p(t')) \end{bmatrix}.$$

- Inference and learning:  $O(n^3 p^3)$  time and  $O(n^2 p^2)$  memory.
  - Often alleviated by **exploiting structure** in  $K$ .

# The Linear Mixing Model



$$f(t) = h_1x_1(t) + h_2x_2(t) = \underbrace{\begin{bmatrix} h_1 & h_2 \end{bmatrix}}_H \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}}_{x(t)}.$$

“mixing matrix”

## Definition (Linear Mixing Model)

$$x \sim \mathcal{GP}(0, K(t, t')), \quad f(t) | H, x = Hx(t), \quad y | f \sim \mathcal{GP}(f(t), \Lambda).$$

“latent processes”

- $f$  is  $p$ -dimensional,  $x$  is  $m$ -dimensional, and  $H$  is  $p \times m$ .
  - Often  $p \gg m$ .
- Equivalently,  $y \sim \mathcal{GP}(0, HK(t, t')H^\top + \Lambda)$ .
- Generalisation of FA to time series setting.
- **Fixed spatial correlation:**  $\mathbb{E}[f(t)f^\top(t)] = HH^\top$  if  $K(t, t) = I$ .
- **Instantaneous mixing:**  $f(t)$  depends on  $x(t')$  only for  $t = t'$ .
- Inference and learning:  $O(n^3 m^3)$  time and  $O(n^2 m^2)$  memory.



## Proposition

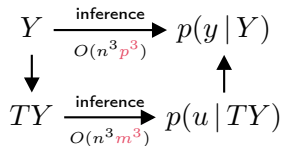
Let  $T$  be the  $(m \times p)$ -matrix  $(H^\top \Lambda^{-1} H)^{-1} H^\top \Lambda^{-1}$ . Then

conditioning  $y \mid f \sim \mathcal{GP}(f(t), \Lambda)$  on data  $Y$ :  $O(n^3 p^3)$

$\iff$

conditioning  $\underbrace{Ty}_u \mid f \sim \mathcal{GP}(\underbrace{Tf(t)}_{x(t)}, T\Lambda T^\top)$  on data  $TY$ :  $O(n^3 m^3)$ .

- $T$ : “ $y$ -space”  $\rightarrow$  “ $x$ -space”.
- $Ty = \arg \min_x \|\Lambda^{\frac{1}{2}}(y - Hx)\|_2$ .

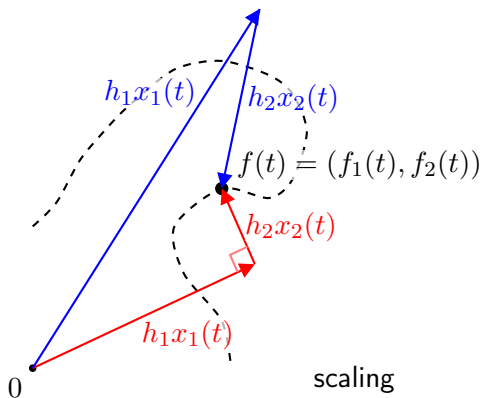


What if  $T\Lambda T^\top$  were **diagonal**? Then inference **decouples** into **independent** problems!

# The Orthogonal Linear Mixing Model

# Fixed Orthogonal Basis with Varying Coefficients

6/16



$$\begin{bmatrix} h_1 & h_2 \end{bmatrix} = H = US^{\frac{1}{2}}.$$

orthogonal

scaling

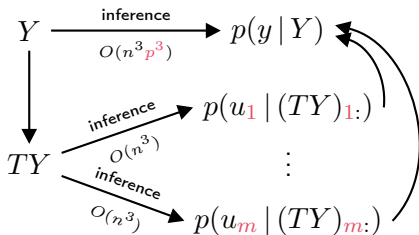
## Definition (Orthogonal Linear Mixing Model)

With  $K(t, t) = I$ ,  $H = US^{\frac{1}{2}}$ , and  $\Lambda = \sigma^2 I + HDH^T$ ,

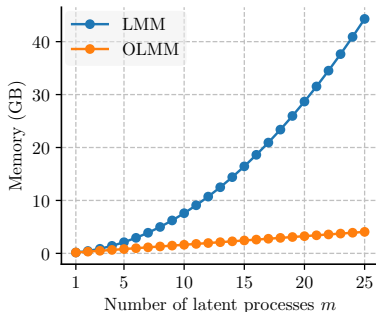
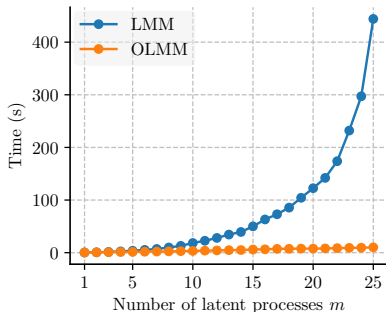
$$x \sim \mathcal{GP}(0, K(t, t')), \quad f(t) | H, x = Hx(t), \quad y | f \sim \mathcal{GP}(f(t), \Lambda).$$

- Generalisation of PPCA (Tipping and Bishop, 1999) to time series setting.
- Like GPFA (Yu et al., 2009), but orthogonality built in.
- **General spatial correlation:**  $\mathbb{E}[f(t)f^T(t)] = USU^T$ .  
⇒ Suggests way to initialise  $U$  and  $S$ .

- Image of noise:  $T\Lambda T^\top = \sigma^2 S^{-1} + D$ . **Diagonal!**



- Inference and learning:  $O(n^3 \textcolor{red}{m})$  time and  $O(n^2 \textcolor{red}{m})$  memory.  
 $\Rightarrow$  **Linear** scaling in the number of degrees of freedom!
- Trivially compatible** with one-dimensional scaling techniques.



## Proposition

The evidence  $\log p(Y)$  is **convex** in  $U$ .

# Arbitrary Likelihoods

- + Computationally efficient
  - + Linear scaling in number of degrees of freedom  $m$
  - + Trivially compatible with one-dimensional scaling techniques
  - + Convex in  $U$
- + Easy to implement
- ± Expressivity
  - ± Restricted to orthogonal bases
  - Linear correlations
- Cannot handle missing data
- Cannot handle inhomogeneous observation noise

**Goal:** arbitrary  $p(y | f)$  whilst retaining computational efficiency.



# Application: Neural Networks with Time-Varying Weights

11/16

**Task:** predict mapping  $z \mapsto \phi(z, t)$  that slowly varies with time.

- ???
- Economics?

**Generative model:**

$$w \sim \text{OLMM}, \quad \phi(z, t) \mid w = \text{NN}_{w(t)}(z).$$

**Inference:** VI with an OLMM as computationally efficient  $q$ .

Prior:

Approximate posterior (Johnson et al., 2016):

$$p(y, f) = \underbrace{p(y | f)}_{\text{arbitrary likelihood}} p(f).$$

$$q(f) = \frac{1}{Z} p(f) \underbrace{p(\hat{y} | f, \hat{\Lambda})}_{\text{likelihood "conjugate" to the OLMM}}.$$

ELBO:

$$\mathcal{L}(\hat{y}, \hat{\Lambda}) = \underbrace{\log Z}_{\text{evidence of pseudo-observations}} + \overbrace{\mathbb{E}_{q(f)} [\log p(y | f) - \log p(\hat{y} | f, \hat{\Lambda})]}^{\text{likelihood correction}}.$$

↑ tractable / Monte Carlo     ↑ tractable

- + Pseudo-evidence  $\log Z$  and approximate posterior  $q(f)$  cheap
- + Easy to implement
- ± Prior  $p(f)$  shared with  $q(f)$
- For  $k$  pseudo-points,  $O(kp)$  parameters

Approximate posterior:

$$\begin{aligned}
 q(f) &= \frac{1}{Z} p(f) p(\hat{y} \mid f, \hat{\Lambda}) \\
 &= \frac{1}{Z} p(f) p(\underbrace{\mathbf{T} \hat{y}}_{\hat{u}} \mid f, \mathbf{T} \hat{\Lambda} \mathbf{T}^\top \underbrace{\mathbf{T} \hat{\Lambda} \mathbf{T}^\top}_{\hat{D}}) = \frac{1}{Z} p(f) p(\hat{u} \mid f, \hat{D}).
 \end{aligned}$$

ELBO:

$$\mathcal{L}(\hat{u}, \hat{D}) = \log Z + \mathbb{E}_{q(f)} [\log p(y \mid f) - \log p(\hat{u} \mid f, \hat{D})].$$

+ For  $k$  pseudo-points,  $O(km)$  parameters

- $q(f) \propto p(f)p(\hat{u} | f, \hat{D})$ : EP-style approximation in VI setting.
- Consider traditional pseudo-point method (Titsias, 2009):

$$\begin{aligned} q(f) &= \int p(f | \hat{f}) q(\hat{f}) d\hat{f} \\ &= \int p(f | \underbrace{\hat{T}\hat{f}\hat{T}}_{\hat{x}}) q(\hat{T}\hat{f}) d\hat{T}\hat{f} = \int p(f | \hat{x}) q(\hat{x}) d\hat{x}. \end{aligned}$$

- Equivalent if  $q(\hat{x}_i) = \mathcal{N}(\hat{u}_i, (K_{x_i}^{-1} + \hat{D}_i^{-1})^{-1})$ .
- When does  $q^*(\hat{x})$  factorise over the latent processes?

$$\prod_{i=1}^m q^*(\hat{x}_i) \stackrel{?}{=} q^*(\hat{x}) = e^{-\mathcal{L}^*} p(\hat{x}) \exp \langle \log p(y | x) \rangle_{p(x | \hat{x})}.$$

① OLMM!

②  $y | x \sim \mathcal{GP}(H\phi(x(t)), \Lambda)$  with  $\phi$  a pointwise nonlinearity.

(Preliminary) Conclusions

- Orthogonal basis **decouples** inference into **independent** problems.
- OLMM can (maybe) function as a **computationally efficient prior / approximate posterior** in larger **spatio-temporal models**.
- Experiments on real-world data!

- Fixing a computational budget, how does OLMM compare?
- How restrictive is the orthogonality assumption?
  - For a given LMM, how close is the closest OLMM?
- When is  $H$  identifiable? Connection to ICA?
- How does learned basis compare to other methods, e.g. PCA?
  - Can pointwise nonlinearity  $\phi$  improve learned basis?
- Can orthogonality alleviate downsides of mean-field VI?

These slides: <https://wessel.page.link/olmm>.

# Appendix



## References

- Johnson, M. J., Duvenaud, D., Wiltchko, A. B., Datta, S. R., & Adams, R. P. (2016). Composing graphical models with neural networks for structured representations and fast inference. *arXiv preprint arXiv:1603.06277*. eprint: <https://arxiv.org/abs/1603.06277>
- Matheron, G. (1969). Le krigeage universel. In *Cahiers du centre de morphologie mathématique de fontainebleau* (Vol. 1). École nationale supérieure des mines de Paris.
- Tipping, M. E., & Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61(3), 611–622.
- Titsias, M. K. (2009). Variational learning of inducing variables in sparse Gaussian processes. *Artificial Intelligence and Statistics*, 12, 567–574.

## References (2)

Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S. I., Shenoy, K. V., & Sahani, M. (2009). Gaussian-Process factor analysis for low-dimensional single-trial analysis of neural population activity. *Advances in Neural Information Processing Systems*, 21, 1881–1888. Retrieved from <http://papers.nips.cc/paper/3494-gaussian-process-factor-analysis-for-low-dimensional-single-trial-analysis-of-neural-population-activity.pdf>