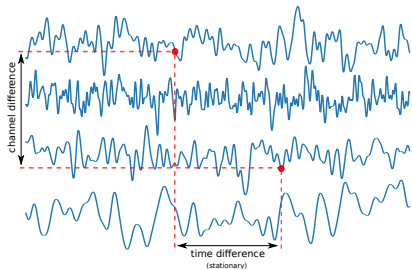
MultiOutput Gaussian Processes

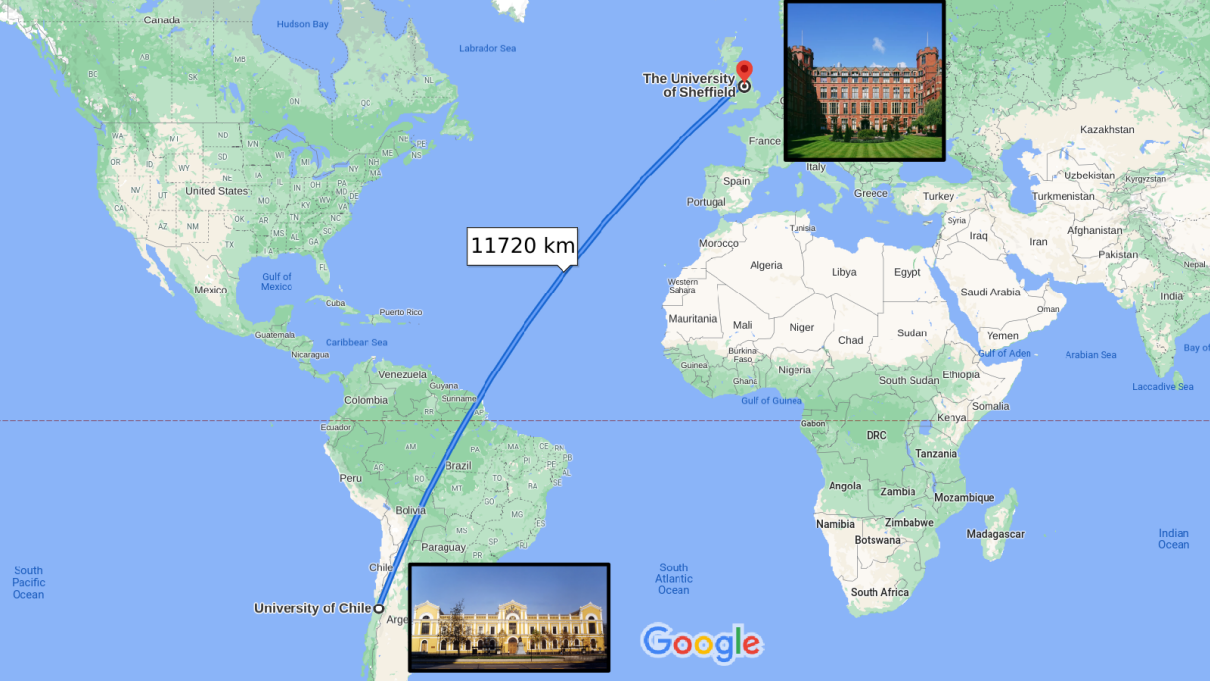
GPSS 2021

Felipe Tobar

Initiative for Data & Artificial Intelligence
Universidad de Chile

14 September 2021




11720 km
The University of Sheffield
University of Chile
Canada
Hudson Bay
Labrador Sea
United States
Mexico
Gulf of Mexico
Cuba
Puerto Rico
Guatemala
Nicaragua
Caribbean Sea
Venezuela
Guyana
Suriname
Colombia
Ecuador
Peru
Brazil
Bolivia
Paraguay
Chile
Argentina
South Pacific Ocean
South Atlantic Ocean
France
Spain
Portugal
Italy
Greece
Turkey
Kazakhstan
Uzbekistan
Kyrgyzstan
Turkmenistan
Afghanistan
Pakistan
Nepal
India
Iran
Iraq
Syria
Morocco
Tunisia
Algeria
Libya
Egypt
Western Sahara
Mauritania
Mali
Niger
Chad
Sudan
Saudi Arabia
Oman
Yemen
Gulf of Aden
Arabian Sea
Laccadive Sea
Bay of
Guinea
Burkina Faso
Nigeria
Ghana
Gulf of Guinea
Ethiopia
South Sudan
Somalia
Kenya
Gabon
DRC
Tanzania
Angola
Zambia
Mozambique
Namibia
Zimbabwe
Botswana
Madagascar
South Africa
Indian Ocean
Google

Today: Bahía Inglesa

University of Chile

11720 km

The University of Sheffield

This talk

What: A one-hour crash course on MOGPs

How:

- Motivation: why do we need jointly processing time series?
- What is an MOGP?
- Building the covariance function
- Examples
- Demo
- Discussion

A personal note: How did I end up working with GPs?

2010: BSC / MSc (Electrical Enginnering, Control Systems, **Particle Filters**)

2014: PhD (Signal Processing, Adaptive Filters, **Kernel Metods**)

2015: Postdoc (Machine Learning, **GPs**)

2016-2020: Research Fellow (Maths, **GPs & Spectrum estimation**, apps)

2021: Lecturer (Data & AI, GPs, **Optimal Transport**, applications)

This talk

What: A one-hour crash course on MOGPs

How:

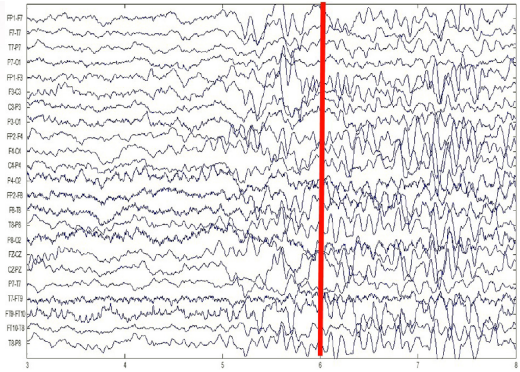
- Motivation: why do we need jointly processing time series?
- What is an MOGP?
- Building the covariance function
- Examples
- Demo
- Discussion

A personal note: How did I end up working with GPs?

2010: BSC / MSc (Electrical Enginnering, Control Systems, **Particle Filters**)
2014: PhD (Signal Processing, Adaptive Filters, **Kernel Metods**)
2015: Postdoc (Machine Learning, **GPs**)
2016-2020: Research Fellow (Maths, **GPs & Spectrum estimation**, apps)
2021: Lecturer (Data & AI, GPs, **Optimal Transport**, applications)

The need for multivariate processing

Shared sources of uncertainty



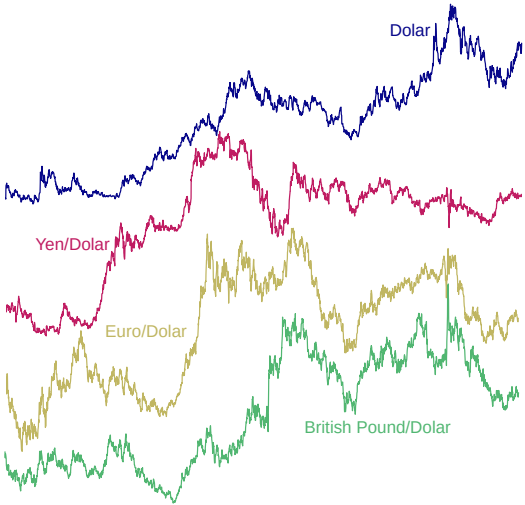
Left: EMOTIV EPOC+, https://lucid.me/blog/wed-love-try-emotiv-epoc/

Right: Xun, G., Jia, X. & Zhang, A. Detecting epileptic seizures with electroencephalogram via a context-learning model. BMC Med Inform Decis Mak 16, 70 (2016)

The need for multivariate processing

Shared sources of uncertainty

Left: The Charging Bull. Copyright Andrew Henkelman (photo) & Arturo Di Modica (sculpture) https://upload.wikimedia.org/wikipedia/en/c/c9/Charging_Bull_statue.jpg



Right: Dolar and three change rates. Own work, data form Banco Central de Chile

The need for multivariate processing

Relationship across measurements

Top: Caterpillar 797 (20 cylinder, 4 turbo, 4000 hp, 400-ton payload). Mina Radomiro Tomic & Radio Cooperativa

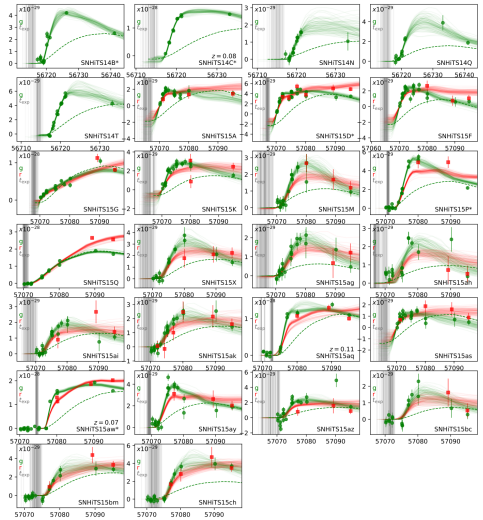
Bottom: From project with PSInet-CMM-Codelco

The need for multivariate processing

Only some channels are observed at some locations (incomplete measurements)

Left: 2018zd Super nova (colour composition form Observatorio Las Cumbres and Hubble Telescope, white point) and host galaxy NGC 2146 (left).

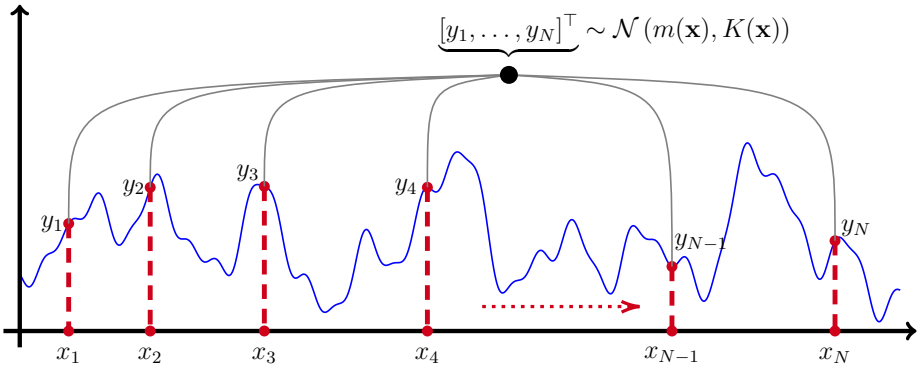
Right: Light curve fluxes at different bands. From Förster. et al. The delay of shock breakout due to circumstellar material evident in most type II supernovae. Nature Astronomy 2, 808–818 (2018).



Gaussian processes

a 1-slide reminder

Definition: A GP is a stochastic process such that any finite collection of values follows a multivariate normal distribution.

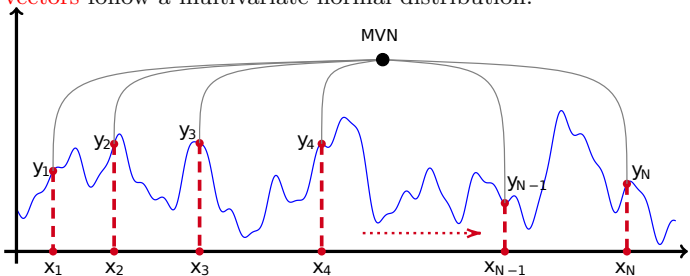


Notation: $f \sim \mathcal{GP}(\mu, K) \iff f(\mathbf{x}) \sim \mathcal{N}(m(\mathbf{x}), K(\mathbf{x}, \mathbf{x})), \forall \mathbf{x} \in \mathcal{X}^n, n \in \mathbb{N}$

MOGPs

Attempt of a definition

Definition: An MOGP is a vector-valued stochastic process such that any finite collection of ~~values~~ vectors follow a multivariate normal distribution.



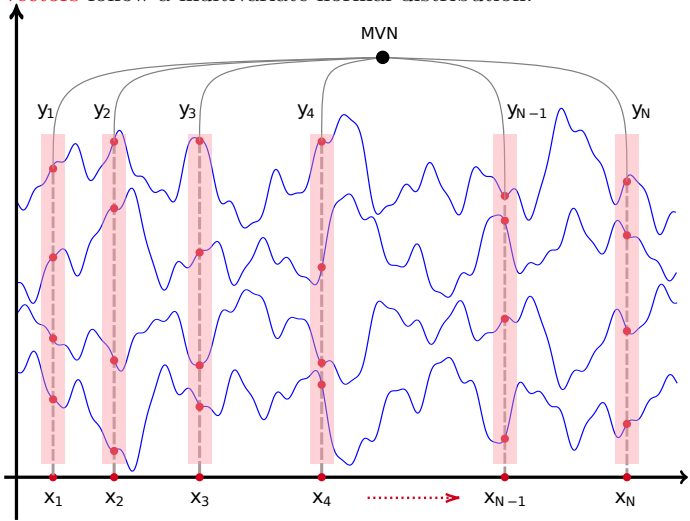
However, due to the marginalisation property of the MVN, vectors need not be chosen in full. Therefore:

Definition: An MOGP is a vector-valued stochastic process such that any finite collection of values, chosen from any channel at any time, follow a multivariate normal distribution.

MOGPs

Attempt of a definition

Definition: An MOGP is a vector-valued stochastic process such that any finite collection of ~~values~~ vectors follow a multivariate normal distribution.



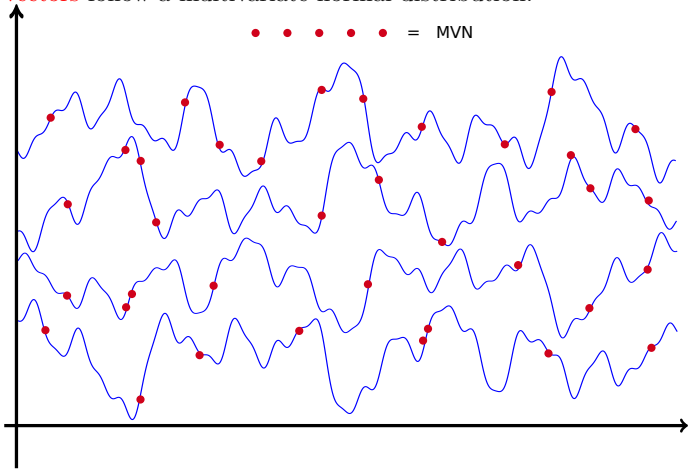
However, due to the marginalisation property of the MVN, vectors need not be chosen in full. Therefore:

Definition: An MOGP is a vector-valued stochastic process such that any finite collection of values, chosen from any channel at any time, follow a multivariate normal distribution.

MOGPs

Attempt of a definition

Definition: An MOGP is a vector-valued stochastic process such that any finite collection of ~~values~~ vectors follow a multivariate normal distribution.



However, due to the marginalisation property of the MVN, vectors need not be chosen in full. Therefore:

Definition: An MOGP is a vector-valued stochastic process such that any finite collection of values, chosen from any channel at any time, follow a multivariate normal distribution.

How do we specify the covariance of an MOGP?

Let us introduce some notation:

- Input space: $\mathcal{X}$ (usually $\mathbb{R}$ but in general $\mathbb{R}^n$)
- GP: $f = [f_1, f_2, \ldots, f_m]^\top \sim \mathcal{MOGP}(\mu, K)$
- Number of channels $= m$

An MOGP is specified by its covariance function which is a four-way array:

$$K : \{1, 2, \ldots, n\}^2 \times \mathcal{X}^2 \to \mathbb{R}$$
$$(i, j, x, x') \mapsto K_{ij}(x, x'),$$

meaning that for a pair of values $f_i(x)$ and $f_j(x')$, we have $\mathbb{V}(f_i(x), f_j(x')) = K_{ij}(x, x')$.

Definition

A two-input matrix-valued function $\mathcal{K}(x, x') : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{m \times m}$ defined element-wise by $[\mathcal{K}(x, x')]_{ij} = k_{ij}(x, x')$ is a multivariate covariance function (kernel) if it is:

- Symmetric, i.e., $\mathcal{K}(x, x') = \mathcal{K}(x', x)^\top, \forall x, x' \in \mathcal{X}$, and
- Positive definite, i.e., $\forall N \in \mathbb{N}, \{c_p\}_{p=1}^N \subseteq \mathbb{R}^m, \{x_p\}_{p=1}^N \subseteq \mathcal{X}$, we have:

$$\sum_{i,j=1}^{m} \sum_{p,q=1}^{N} c_{pi} c_{qj} k_{ij}(x_p, x_q) \geq 0.$$

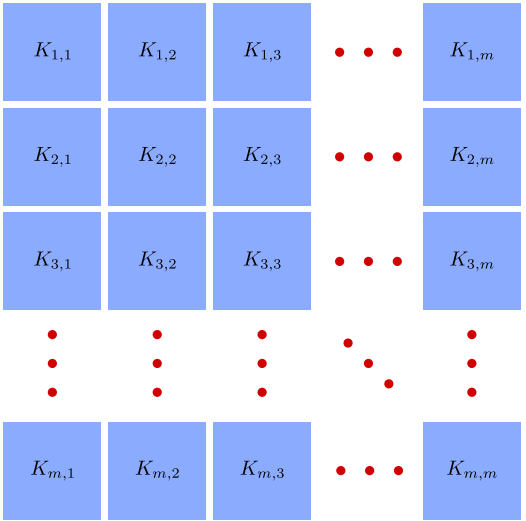
Multioutput Covariance

A few observations

Visualisation: The 4-way array can be shown by *slicing*, that is, we slice the covariance over its countable dimensions (channels) and show each slice.

Parametrisation: This is the tricky part. In the sliced representation the covariance is not continuous, since it has channel jumps. That makes parametrisation quite difficult.

Stacked representation: This covariance corresponds to the concatenation of all datapoints stacked, where one loses channel information. That is useful for coding and also for understanding the definition of positive semidefiniteness via index reshaping.



The Assumption of Stationarity

From 4D to 3D

Recall that a covariance is stationary iff

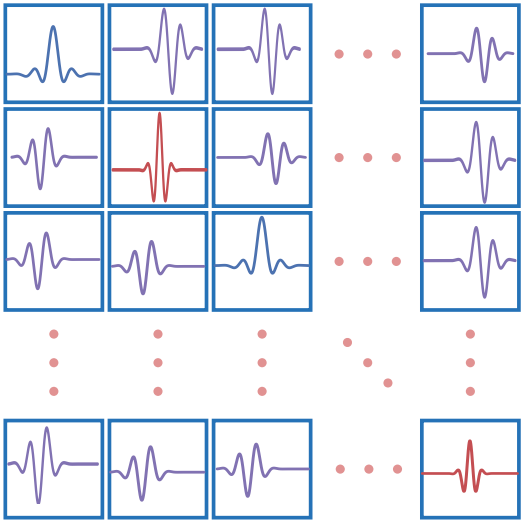
$$K(x, x') = K(x - x').$$

This condition, for MOGPs, turns into

$$K_{i,j}(x, x') = K_{i,j}(x - x'), \forall i, j = 1, \ldots, m.$$

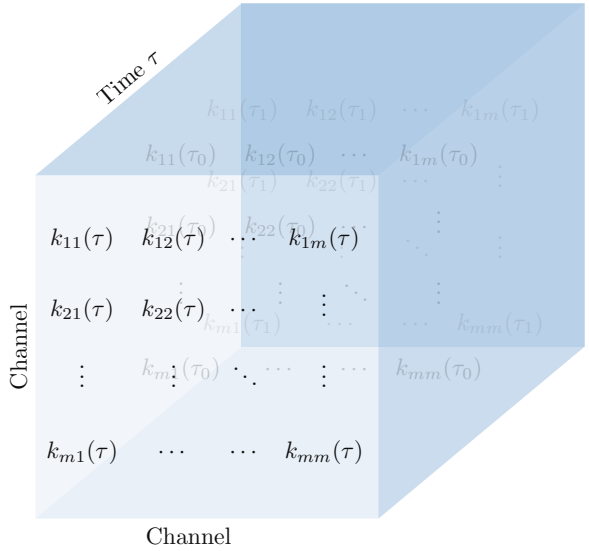
We've got one fewer dimension!

(This might sound like it isn't much, but it'll allow for the design of vector-valued covariance kernels via direct parametrisation)

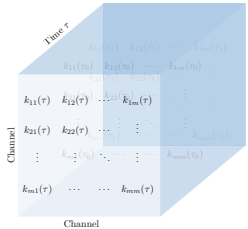


Under the stationarity assumption, we work with this neat 3D representation

(also assume 1-dimensional time)



How to design the kernel?

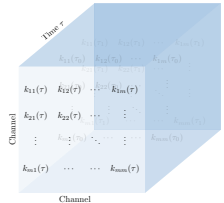


Challenge: Due to the positive semidefiniteness condition, designing MOGP kernels via direct parametrisation is tough if one aim to move from the trivial choices.

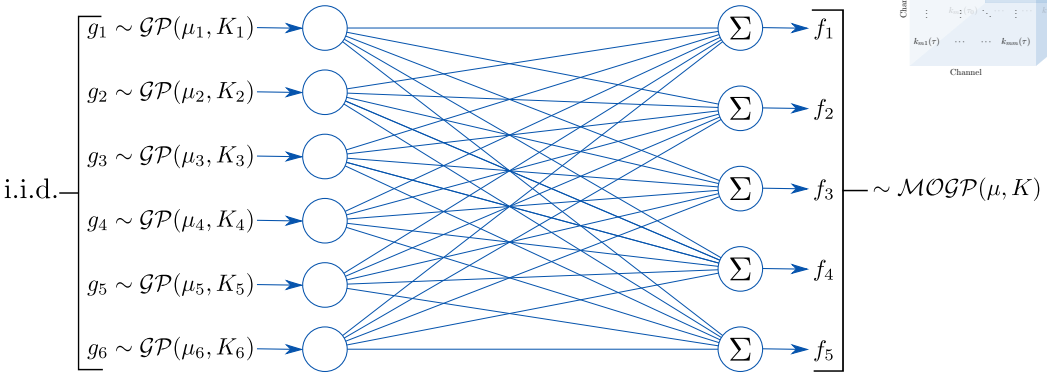
- We want kernels to be as expressive as possible
- Hopefully all structure is discovered (we want to be agnostic)
- Kernel parametrisation are constrained in a very (parameter-wise) unintuitive way

Let us start with something simple

MOGP design, first chapter: Induced kernels

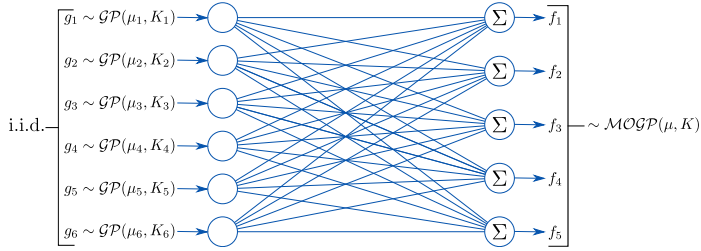


Bypass covariance parametrisation by mixing independent GPs.



where $[\theta]_{ij}$ denotes the weight between input node i and output node j.

Induced kernels: general form



Denote:

- θ the weights
- $g = [g_1, g_2, \ldots, g_n]$
- $f = [f_1, f_2, \ldots, f_m]$

We have that $f = \theta^\top g$ (or $f_j = \theta_j^\top g$) and thus

$$\begin{aligned}
\mathbb{E}(f) &= \theta^\top \mathbb{E}(f) = \theta^\top \mu \\
\mathbb{V}(f(x), f(x')) &= \mathbb{V}(\theta^\top g(x), \theta^\top g(x')) \\
&= \theta^\top \mathbb{V}(g(x), g(x'))\theta \\
&= \sum_{q=1}^{Q} \theta_q \theta_q^\top K_q(x - x')
\end{aligned}$$

Particular cases of MOGP kernels induced by linear mixing

- The Linear Model of Corregionalisation[1]
  Assumption: some of the Q latent GPs, though independent, share the same covariance. Grouping GPs with common covariance, the kernel is

$$\mathbb{V}(f(x), f(x')) = \sum_{q=1}^{Q} \underbrace{\sum_{r=1}^{R_q} \theta_q^{(r)} \theta_q^{(r)\top}}_{C_q} K_q(x - x'), \tag{0.1}$$

  where $\theta_q^{(r)}$ is the vector of weights associated to the r-th member of the q-th group of inputs. The matrices $C_q, q \in \{1, 2, \ldots, m\}$, are called the *corregionalisation matrices*
- The Intrinsic Model of Corregionalisation IMC[2]
  Assumption: All kernels are equal ($Q = 1$ above)
- The Semiparametric Latent Factor Model [3]
  Assumption: The weights θ are free

[1] A. G. Journel and C. J. Huijbregts. Mining Geostatistics. Academic Press, London, 1978.
[2] P. Goovaerts. Geostatistics For Natural Resources Evaluation. Oxford University Press, USA, 1997.
[3] Y. W. Teh, M. Seeger, and M. I. Jordan. Semiparametric latent factor models. AISTATS, pp. 333–340, 2005.

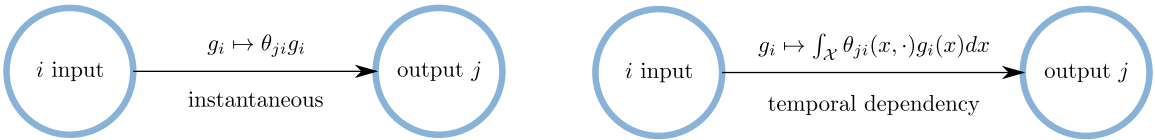
A drawback of the above construction

solved via moving averages

The mixing model above is *instantaneous* meaning that the output at x :

- $K(x, x')$ only depends on $K_q(x, x')$ $(\forall q)$
- all temporal structure in f is given by that of g and the mixture only introduces channel correlations.
- kernels are separable, i.e., $K_{i,j}(x, x') = B_{i,j}K(x, x')$
- $f(x)$ is conditionally independent of its future and its past given $g(x)$

Fix: Let us consider a more general linear operator

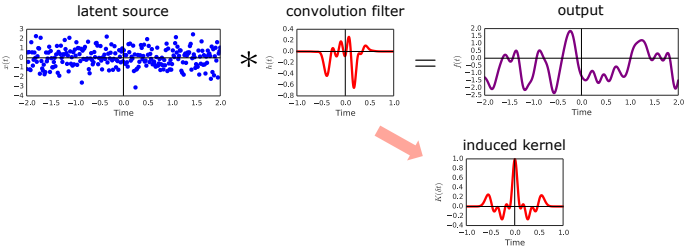


$$K(f_i(x), f_j(x')) = \sum_q \int \int \theta_{iq}(x - z)\theta_{jq}(x' - z')K_q(z, z')dzdz'$$

Warning 1: For the convolution to converge, the kernel has to be square integrable (one particular case are kernels with compact support). Warning 2: Causal vs non-Causal models.

Models based on the convolution construction

Though the convolution might sound a bit hard to grasp or implement, there are certain families of windows and driving noises that have explicit forms and/or interpretations.



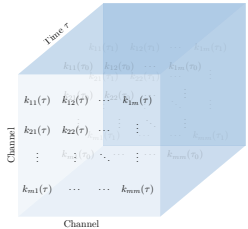
For instance:

- the driving process is white noise[4]
- if $\theta_{iq}(x - z)$ and $K_q(z, z')$ are SE[5], then $K(f_i(x), f_j(x'))$ is SE as well.
- θ is a draw from a GP: scalar case[6] and extensions[7]

[4] P. Boyle and M. Frean. Dependent Gaussian processes. NeurIPS 2005.
[5] M. A. Álvarez and N. D. Lawrence. Sparse convolved Gaussian processes for multi-output regression. NeurIPS 2009.
[6] F. Tobar, T. Bui, and R. Turner. Learning stationary time series using GPs with nonparametric kernels. NeurIPS 2015.
[7] W. Bruinsma, The Generalised Gaussian Process Convolution Model, 2006 (Masters thesis)

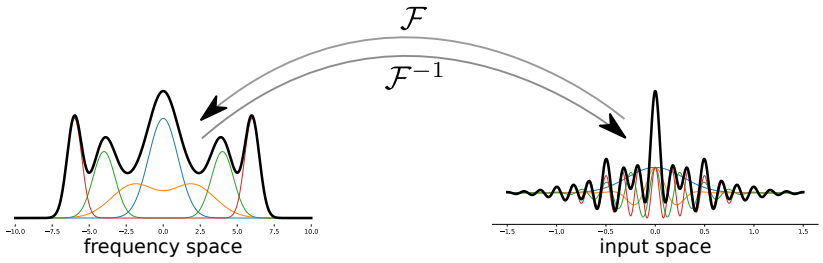
MOGP design, second chapter: spectral parametrisation

Idea: Use an unconstrained representation of MO kernel and then transform it - a sort of *reparametrisation trick*.



This can be implemented by the *spectral mixture* kernel, which parametrises the Fourier power spectral density of the kernel by a Gaussian mixture. Then, the kernel is given by the anti-Fourier transform of such mixture,

For the scalar case, this looks like[8]



$$S(\xi) = \sum_{q=1}^{Q} \sigma_q^2 \frac{1}{\sqrt{2\pi l_q^2}} \exp\left(-\frac{1}{2l_q^2}(\xi \pm \mu_q)^2\right) \qquad K(x) = \sum_{q=1}^{Q} \sigma_q^2 \exp\left(-2\pi^2 l_q^2 x^2\right) \cos\left(2\pi \mu_q x\right)$$

[8] A. Wilson and R. Adams, "Gaussian process kernels for pattern discovery and extrapolation," ICML, 2013.

The Multioutput Spectral Mixture Kernel (1)

Theorem (Cramér's theorem simplified)

A family $\{k_{ij}(\tau)\}_{i,j=1}^{m}$ *of integrable functions are the covariance functions of a weakly-stationary multivariate stochastic process if and only if they (i) admit the representation*

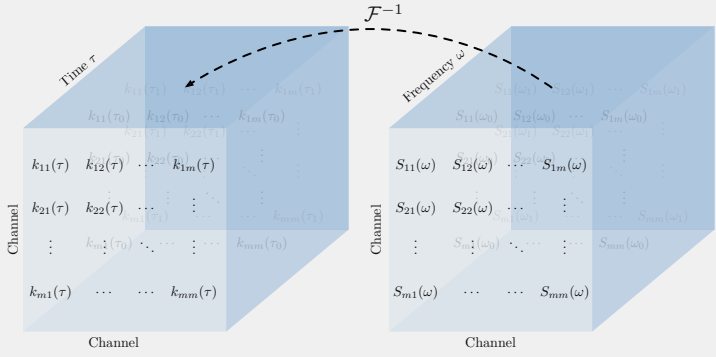
$$k_{ij}(\tau) = \int_{\mathbb{R}^n} e^{\iota \omega^\top \tau} S_{ij}(\omega) \mathrm{d}\omega \qquad \forall i, j \in \{1, \ldots, m\}$$

where each S_{ij} *is an integrable complex-valued function* $S_{ij} : \mathbb{R}^n \to \mathbb{C}$ *known as the spectral density associated to the function* $k_{ij}(\tau)$*, and (ii) fulfil the positive definiteness condition*

$$\sum_{i,j=1}^{m} \overline{z_i} z_j S_{ij}(\omega) \geq 0 \qquad \forall \{z_1, \ldots, z_m\} \subset \mathbb{C},\ \omega \in \mathbb{R}^n$$

The Multioutput Spectral Mixture Kernel (2)

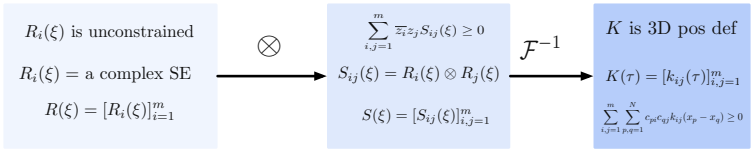
Theorem (Cramér's theorem illustrated)



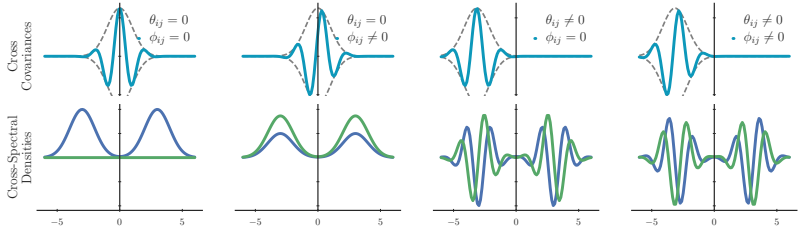
The kernel needs "3D" positive definiteness while the power spectrum only needs 2D positive definiteness (since the condition along the third dimension is guaranteed by the anti-Fourier transform)

The Multioutput Spectral Mixture Kernel (3)

MOSM bypasses the task of building a 3D pos def function by a set 2D PD matrices (via Cholesky):



With this, the MOSM kernel is $k_{ij}(\tau) = \alpha_{ij} \exp\left(-\frac{1}{2}(\tau + \theta_{ij})^\top \Sigma_{ij} (\tau + \theta_{ij})\right) \cos\left((\tau + \theta_{ij})^\top \mu_{ij} + \phi_{ij}\right)$ and looks like this:

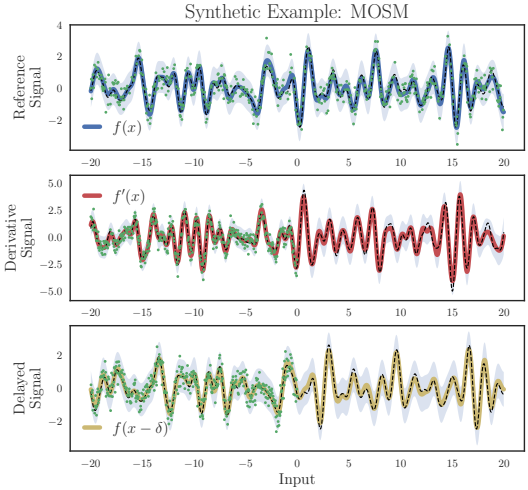


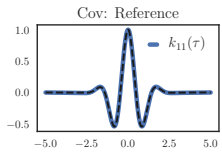
NB: MOSM's key feature is the consideration of relative delays among channels.

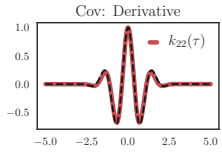
NB: MOSM is equivalent to other models via a restriction of its hypers, e.g., the cross-spectral mixture.

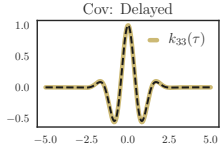
An illustration: learning derivatives and delayed signals

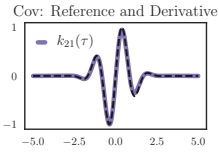
Observations in green, ground truth in dashed lines, MOSM prediction in colours

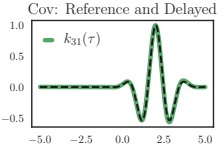






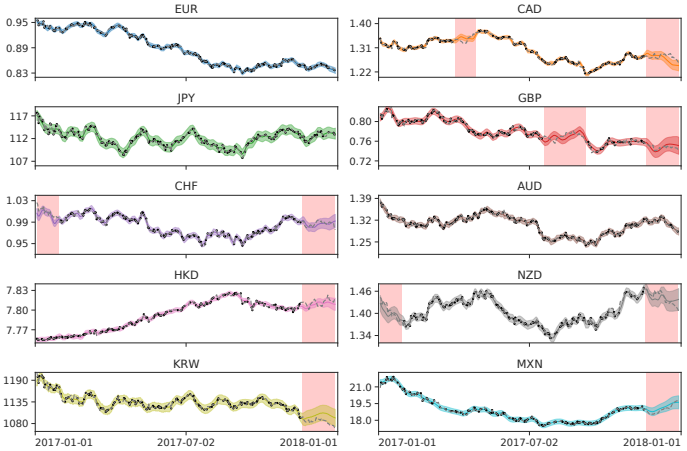


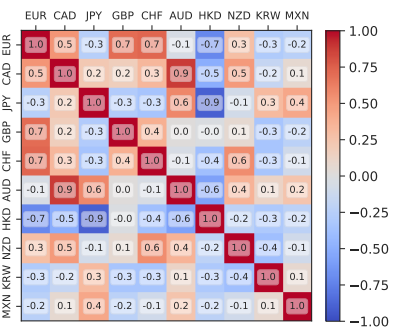




MOGP in action (1): GP-MOSM for finance

Aim: Imputation, filtering, anomaly detection
How: Fit GP-MOSM, inspect hypers
Next: Incorporate non-Gaussian likelihood (stylised facts).





T. de Wolff, A. Cuevas & F. Tobar, Gaussian process imputation of multiple financial series. ICASSP 2017

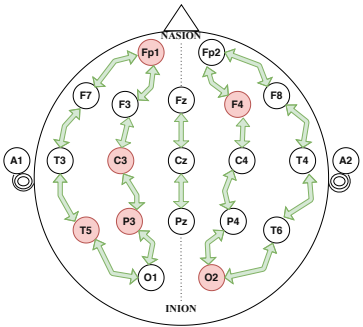
G. Rios & F. Tobar, Compositionally-warped Gaussian processes. Elsevier Neural Networks, 2019.

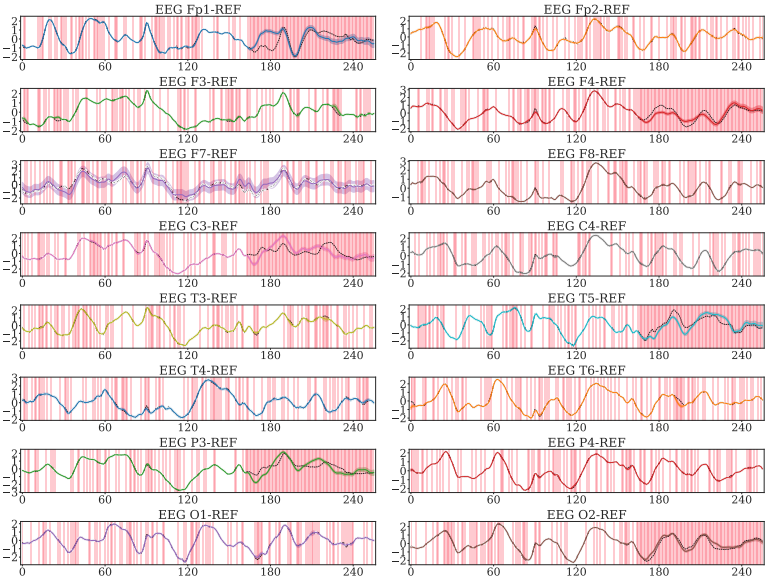
MOGP in action (2): GP-MOSM for electroencephalography

Aim: Detection of neonatal epileptic fits, critical for brain development

How: Adjust GP-MOSM, inspect hypers

Next: Automatic selection of quasistationary windows, change-point detection, nonstationarity





F. Tobar, Modelling Neonatal EEG, Google Latin American Research Awards, 2021

V. Caro, J Ho, S. Witting & F. Tobar, Modelling neonatal EEG using MOGPs. In preparation 2021

MOGP in action (∞): MOGPs for all

Neurocomputing
Available online 17 November 2020
In Press, Journal Pre-proof

Original software publication

MOGPTK: The Multi-Output Gaussian Process Toolkit

Taco Wolff de, Alejandro Cuevas, Felipe Tobar

Show more

Share Cite

README.md

Multi-Output Gaussian Process Toolkit

Paper - **API Documentation** - **Tutorials & Examples**

The Multi-Output Gaussian Process Toolkit is a Python toolkit for training and interpreting Gaussian process models with multiple data channels. It builds upon PyTorch to provide an easy way to train multi-output models effectively on CPUs and GPUs. The main authors are Taco de Wolff, Alejandro Cuevas, and Felipe Tobar as part of the Center for Mathematical Modelling at the University of Chile.

mogptk / examples /

tdewolff ... yesterday

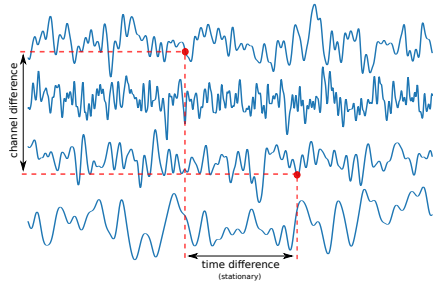
..	
data	2 days ago
00_Quick_Start.ipynb	3 days ago
01_Data_Loading.ipynb	3 days ago
02_Data_Preparation.ipynb	3 days ago
03_Parameter_Initialization.ipynb	3 days ago
04_Model_Training.ipynb	3 days ago
05_Error_Metrics.ipynb	3 days ago
06_Custom_Kernels_and_Mean_Functio…	yesterday
example_airline_passengers.ipynb	3 days ago
example_bramblemet.ipynb	3 days ago
example_currency_exchange.ipynb	2 days ago
example_gold_oil_NASDAQ_USD.ipynb	6 days ago
example_human_activity_recognition.ip…	3 days ago
example_mauna_loa.ipynb	3 days ago

github.com/GAMES-UChile/mogptk

Demo of MOGPTK

Notebooks available at the project's repository

Closing Remarks



Summary

- MOGPs are like GPs, but vector valued :)
- main challenge is covariance design
- covariance can be induced or parametrised
- different models are equivalent modulo parametric constraints

We didn't cover:

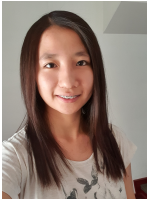
- non-Gaussian likelihood [1]
- sparsity [1]
- nonstationarity [2]
- negative transfer of knowledge [3]

[1] F. Leibfried, V. Dutordoir, ST John, N. Durrande, A tutorial on sparse GPs and VI, arXiv, 2020.
[2] S. Remes, M. Heinonen, S. Kaski, Non-stationary spectral kernels, NeurIPS 2017
[3] R. Kontar, G. Raskutti, S. Zhou, Minimizing negative transfer of knowledge in multivariate Gaussian processes: A scalable and regularized approach, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.

team

Taco de Wolff
Data Scientist
Inria Chile

Alejandro Cuevas
Data Scientist
NoiseGrasp



Jou-Hui Ho
MSc, Electrical Eng.
U de Chile

Víctor Caro
MSc, Data Science
U de Chile

Matías Altamirano
MSc, Applied Maths
U de Chile

many thanks!