Multioutput Gaussian Processes

Neil D. Lawrence

GPRS
13th February 2014

Outline

Kalman Filter

Outline

Kalman Filter

Simple Markov Chain

- Assume 1-d latent state, a vector over time, $\mathbf{x} = [x_1 \dots x_T]$.
- Markov property,

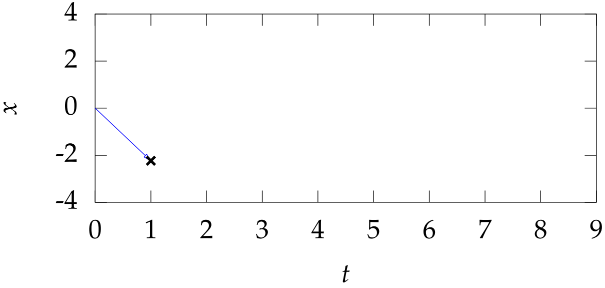
$$\begin{aligned} x_i =& x_{i-1} + \epsilon_i, \\ \epsilon_i \sim& \mathcal{N}(0, \alpha) \\ \implies x_i \sim& \mathcal{N}(x_{i-1}, \alpha) \end{aligned}$$

- Initial state,

$$x_0 \sim \mathcal{N}(0, \alpha_0)$$

- If $x_0 \sim \mathcal{N}(0, \alpha)$ we have a Markov chain for the latent states.
- Markov chain it is specified by an initial distribution (Gaussian) and a transition distribution (Gaussian).

Gauss Markov Chain

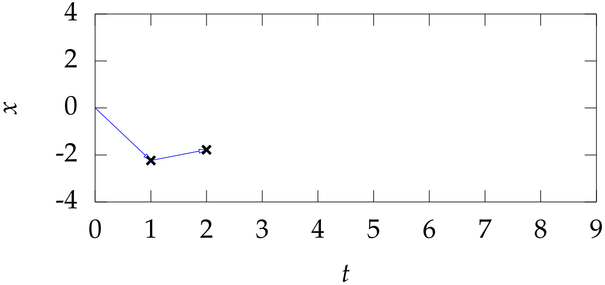


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0, 1)$$

$$x_0 = 0.000, \quad \epsilon_1 = -2.24$$

$$x_1 = 0.000 - 2.24 = -2.24$$

Gauss Markov Chain

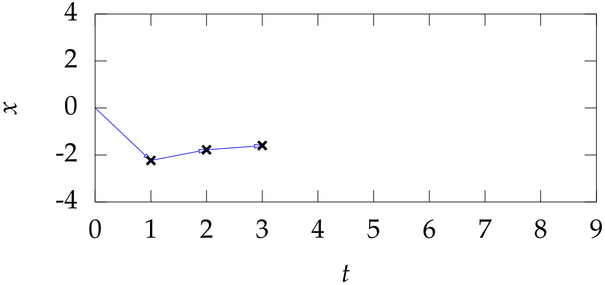


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0, 1)$$

$$x_1 = -2.24, \quad \epsilon_2 = 0.457$$

$$x_2 = -2.24 + 0.457 = -1.78$$

Gauss Markov Chain

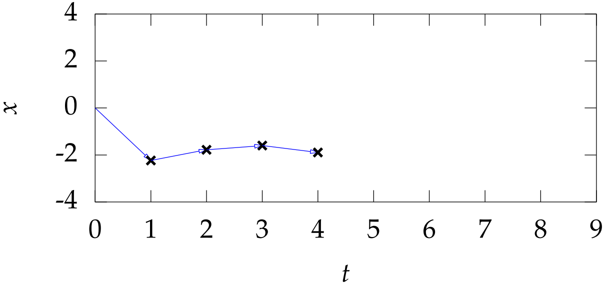


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0, 1)$$

$$x_2 = -1.78, \quad \epsilon_3 = 0.178$$

$$x_3 = -1.78 + 0.178 = -1.6$$

Gauss Markov Chain

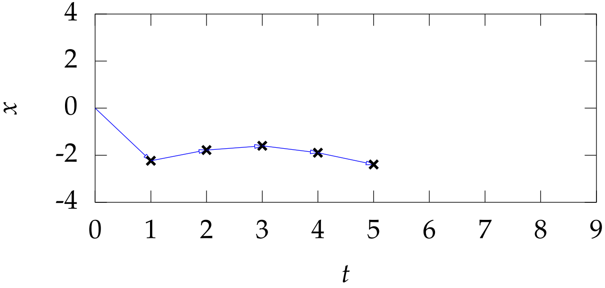


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0, 1)$$

$$x_3 = -1.6, \quad \epsilon_4 = -0.292$$

$$x_4 = -1.6 - 0.292 = -1.89$$

Gauss Markov Chain

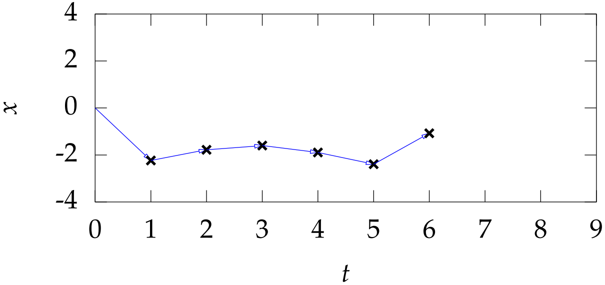


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0, 1)$$

$$x_4 = -1.89, \quad \epsilon_5 = -0.501$$

$$x_5 = -1.89 - 0.501 = -2.39$$

Gauss Markov Chain

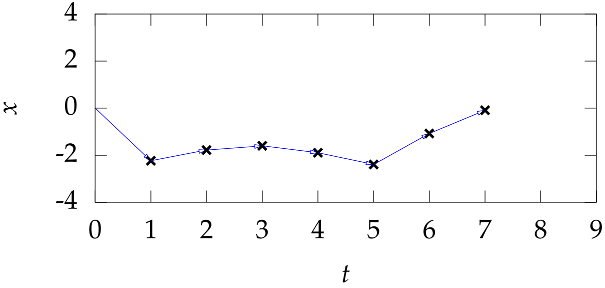


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0, 1)$$

$$x_5 = -2.39, \quad \epsilon_6 = 1.32$$

$$x_6 = -2.39 + 1.32 = -1.08$$

Gauss Markov Chain

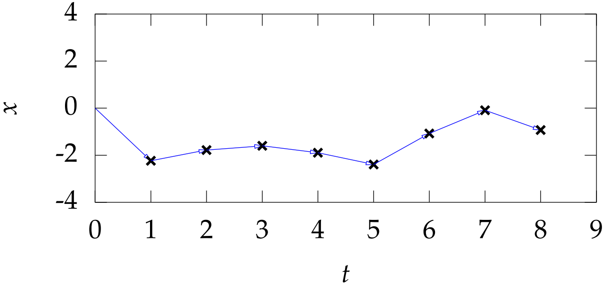


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0, 1)$$

$$x_6 = -1.08, \quad \epsilon_7 = 0.989$$

$$x_7 = -1.08 + 0.989 = -0.0881$$

Gauss Markov Chain

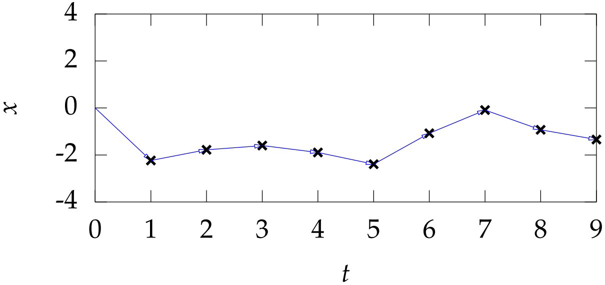


$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0, 1)$$

$$x_7 = -0.0881, \quad \epsilon_8 = -0.842$$

$$x_8 = -0.0881 - 0.842 = -0.93$$

Gauss Markov Chain



$$x_0 = 0, \quad \epsilon_i \sim \mathcal{N}(0, 1)$$

$$x_8 = -0.93, \quad \epsilon_9 = -0.41$$

$$x_9 = -0.93 - 0.410 = -1.34$$

Multivariate Gaussian Properties: Reminder

If

$$\mathbf{z} \sim \mathcal{N}\left(\boldsymbol{\mu}, \mathbf{C}\right)$$

and

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \mathbf{b}$$

then

$$\mathbf{x} \sim \mathcal{N}\left(\mathbf{W}\boldsymbol{\mu} + \mathbf{b}, \mathbf{W}\mathbf{C}\mathbf{W}^\top\right)$$

Multivariate Gaussian Properties: Reminder

Simplified: If

$$\mathbf{z} \sim \mathcal{N}\left(0, \sigma^2\mathbf{I}\right)$$

and

$$\mathbf{x} = \mathbf{W}\mathbf{z}$$

then

$$\mathbf{x} \sim \mathcal{N}\left(0, \sigma^2\mathbf{W}\mathbf{W}^\top\right)$$

Matrix Representation of Latent Variables

$$\begin{bmatrix} \boxed{x_1} \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{bmatrix}$$

$$x_1 = \epsilon_1$$

Matrix Representation of Latent Variables

$$\begin{bmatrix} x_1 \\ \boxed{x_2} \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{bmatrix}$$

$$x_2 = \epsilon_1 + \epsilon_2$$

Matrix Representation of Latent Variables

$$\begin{bmatrix} x_1 \\ x_2 \\ \boxed{x_3} \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{bmatrix}$$

$$x_3 = \epsilon_1 + \epsilon_2 + \epsilon_3$$

Matrix Representation of Latent Variables

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \boxed{x_4} \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{bmatrix}$$

$$x_4 = \epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4$$

Matrix Representation of Latent Variables

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \boxed{x_5} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ \boxed{1} & \boxed{1} & \boxed{1} & \boxed{1} & \boxed{1} \end{bmatrix} \times \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{bmatrix}$$

$$x_5 = \epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4 + \epsilon_5$$

Matrix Representation of Latent Variables

$$\mathbf{x} \quad = \quad \mathbf{L}_1 \quad \times \quad \epsilon$$

Multivariate Process

- Since $\mathbf{x}$ is linearly related to $\boldsymbol{\epsilon}$ we know $\mathbf{x}$ is a Gaussian process.
- Trick: we only need to compute the mean and covariance of $\mathbf{x}$ to determine that Gaussian.

Latent Process Mean

$$\mathbf{x} = \mathbf{L}_1\boldsymbol{\epsilon}$$

Latent Process Mean

$$\langle \mathbf{x} \rangle = \langle \mathbf{L}_1 \epsilon \rangle$$

Latent Process Mean

$$\langle \mathbf{x} \rangle = \mathbf{L}_1 \langle \boldsymbol{\epsilon} \rangle$$

Latent Process Mean

$$\langle \mathbf{x} \rangle = \mathbf{L}_1 \langle \boldsymbol{\epsilon} \rangle$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \alpha \mathbf{I}\right)$$

Latent Process Mean

$$\langle \mathbf{x} \rangle = \mathbf{L}_1 \mathbf{0}$$

Latent Process Mean

$$\langle \mathbf{x} \rangle = \mathbf{0}$$

Latent Process Covariance

$$\mathbf{x}\mathbf{x}^\top = \mathbf{L_1}\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\mathbf{L_1}^\top$$

$$\mathbf{x}^\top = \boldsymbol{\epsilon}^\top\mathbf{L}^\top$$

Latent Process Covariance

$$\langle \mathbf{x}\mathbf{x}^\top \rangle = \left\langle \mathbf{L_1} \boldsymbol{\epsilon} \boldsymbol{\epsilon}^\top \mathbf{L_1}^\top \right\rangle$$

Latent Process Covariance

$$\left\langle \mathbf{x}\mathbf{x}^\top \right\rangle = \mathbf{L_1} \left\langle \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top \right\rangle \mathbf{L_1}^\top$$

Latent Process Covariance

$$\langle \mathbf{x}\mathbf{x}^\top \rangle = \mathbf{L_1} \langle \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top \rangle \mathbf{L_1}^\top$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \alpha \mathbf{I})$$

Latent Process Covariance

$$\langle \mathbf{x}\mathbf{x}^\top \rangle = \alpha \mathbf{L}_\mathbf{1} \mathbf{L}_\mathbf{1}^\top$$

Latent Process

$$\mathbf{x} = \mathbf{L}_1 \boldsymbol{\epsilon}$$

Latent Process

$$\mathbf{x} = \mathbf{L}_1 \boldsymbol{\epsilon}$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \alpha \mathbf{I})$$

Latent Process

$$\mathbf{x} = \mathbf{L}_1 \boldsymbol{\epsilon}$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \alpha \mathbf{I}\right)$$

$$\Longrightarrow$$

Latent Process

$$\mathbf{x} = \mathbf{L_1}\boldsymbol{\epsilon}$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \alpha\mathbf{I})$$

$$\Longrightarrow$$

$$\mathbf{x} \sim \mathcal{N}\left(\mathbf{0}, \alpha\mathbf{L_1}\mathbf{L_1}^\top\right)$$

Covariance for Latent Process II

- Make the variance dependent on time interval.
- Assume variance grows *linearly* with time.
- Justification: sum of two Gaussian distributed random variables is distributed as Gaussian with sum of variances.
- If variable's movement is additive over time (as described) variance scales linearly with time.

Covariance for Latent Process II

- Given
$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \alpha \mathbf{I}\right) \Longrightarrow \boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \alpha \mathbf{L_1} \mathbf{L_1}^\top\right).$$
Then
$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \Delta t \alpha \mathbf{I}\right) \Longrightarrow \boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \Delta t \alpha \mathbf{L_1} \mathbf{L_1}^\top\right).$$
where Δt is the time interval between observations.

Covariance for Latent Process II

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(0, \alpha \Delta t \mathbf{I}\right), \quad \mathbf{x} \sim \mathcal{N}\left(0, \alpha \Delta t \mathbf{L_1} \mathbf{L_1}^\top\right)$$

Covariance for Latent Process II

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(0, \alpha \Delta t \mathbf{I}\right), \quad \mathbf{x} \sim \mathcal{N}\left(0, \alpha \Delta t \mathbf{L_1} \mathbf{L_1}^\top\right)$$

$$\mathbf{K} = \alpha \Delta t \mathbf{L_1} \mathbf{L_1}^\top$$

Covariance for Latent Process II

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(0, \alpha\Delta t\mathbf{I}\right), \quad \mathbf{x} \sim \mathcal{N}\left(0, \alpha\Delta t\mathbf{L_1}\mathbf{L_1}^\top\right)$$

$$\mathbf{K} = \alpha\Delta t\mathbf{L_1}\mathbf{L_1}^\top$$

$$k_{i,j} = \alpha\Delta t\mathbf{l}_{:,i}^\top\mathbf{l}_{:,j}$$

where $\mathbf{l}_{:,k}$ is a vector from the kth row of $\mathbf{L_1}$: the first k elements are one, the next $T - k$ are zero.

Covariance for Latent Process II

$$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \alpha\Delta t\mathbf{I}), \quad \mathbf{x} \sim \mathcal{N}\left(0, \alpha\Delta t\mathbf{L_1}\mathbf{L_1}^\top\right)$$

$$\mathbf{K} = \alpha\Delta t\mathbf{L_1}\mathbf{L_1}^\top$$

$$k_{i,j} = \alpha\Delta t\mathbf{l}_{:,i}^\top\mathbf{l}_{:,j}$$

where $\mathbf{l}_{:,k}$ is a vector from the kth row of $\mathbf{L_1}$: the first k elements are one, the next $T-k$ are zero.

$$k_{i,j} = \alpha\Delta t \min(i, j)$$

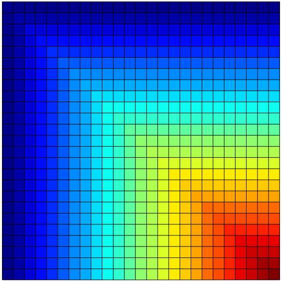
define $\Delta t i = t_i$ so

$$k_{i,j} = \alpha \min(t_i, t_j) = k(t_i, t_j)$$

Markov Process

$$k(t, t') = \alpha \min(t, t')$$

- Covariance matrix is built using the *inputs* to the function t.



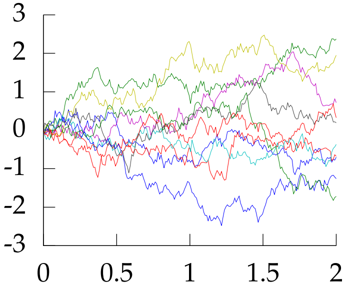
Covariance Functions

Where did this covariance matrix come from?

Markov Process

$$k(t, t') = \alpha \min(t, t')$$

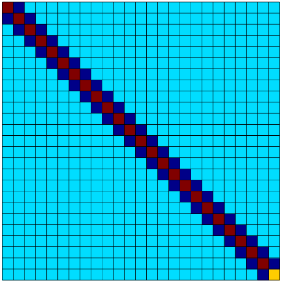
- Covariance matrix is built using the *inputs* to the function t.



Markov Process

Visualization of inverse covariance (precision).

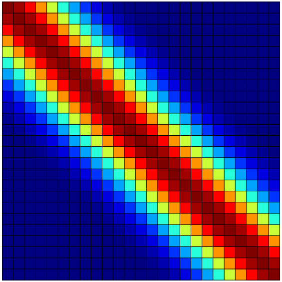
- Precision matrix is sparse: only neighbours in matrix are non-zero.
- This reflects *conditional* independencies in data.
- In this case *Markov* structure.



Exponentiated Quadratic Kernel Function (RBF, Squared Exponential, Gaussian)

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\ell^2}\right)$$

- Covariance matrix is built using the *inputs* to the function $\mathbf{x}$.
- For the example above it was based on Euclidean distance.
- The covariance function is also know as a kernel.



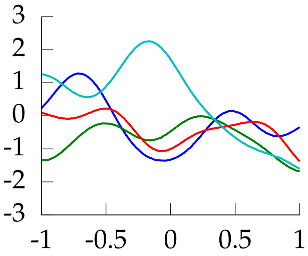
Covariance Functions

Where did this covariance matrix come from?

Exponentiated Quadratic Kernel Function (RBF, Squared Exponential, Gaussian)

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\ell^2}\right)$$

- Covariance matrix is built using the *inputs* to the function $\mathbf{x}$.
- For the example above it was based on Euclidean distance.
- The covariance function is also know as a kernel.



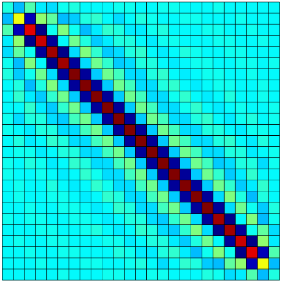
Covariance Functions

Where did this covariance matrix come from?

Exponentiated Quadratic

Visualization of inverse covariance (precision).

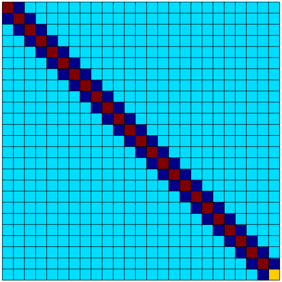
- Precision matrix is not sparse.
- Each point is dependent on all the others.
- In this case non-Markovian.



Markov Process

Visualization of inverse covariance (precision).

- Precision matrix is sparse: only neighbours in matrix are non-zero.
- This reflects *conditional* independencies in data.
- In this case *Markov* structure.



Simple Kalman Filter I

- We have state vector $\mathbf{X} = \left[\mathbf{x}_1 \ldots \mathbf{x}_q\right] \in \mathbb{R}^{T \times q}$ and if each state evolves independently we have

$$p(\mathbf{X}) = \prod_{i=1}^{q} p(\mathbf{x}_{:,i})$$
$$p(\mathbf{x}_{:,i}) = \mathcal{N}\left(\mathbf{x}_{:,i}|\mathbf{0}, \mathbf{K}\right).$$

- We want to obtain outputs through:

$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:}$$

Stacking and Kronecker Products I

- Represent with a 'stacked' system:

$$p(\mathbf{x}) = \mathcal{N}\left(\mathbf{x}|\mathbf{0}, \mathbf{I} \otimes \mathbf{K}\right)$$

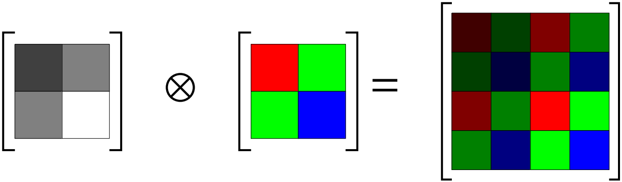
  where the stacking is placing each column of $\mathbf{X}$ one on top of another as

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{:,1} \\ \mathbf{x}_{:,2} \\ \vdots \\ \mathbf{x}_{:,q} \end{bmatrix}$$

Kronecker Product

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \otimes \mathbf{K} = \begin{bmatrix} a\mathbf{K} & b\mathbf{K} \\ c\mathbf{K} & d\mathbf{K} \end{bmatrix}$$

Kronecker Product



Stacking and Kronecker Products I

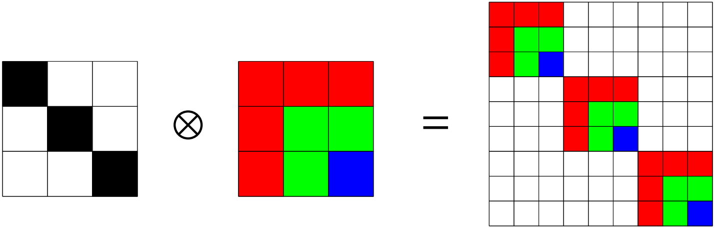
- Represent with a 'stacked' system:

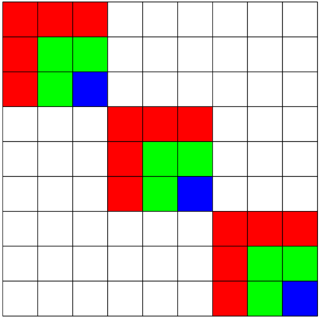
$$p(\mathbf{x}) = \mathcal{N}\left(\mathbf{x}|\mathbf{0}, \mathbf{I} \otimes \mathbf{K}\right)$$

  where the stacking is placing each column of $\mathbf{X}$ one on top of another as

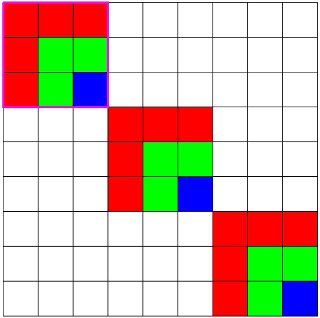
$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{:,1} \\ \mathbf{x}_{:,2} \\ \vdots \\ \mathbf{x}_{:,q} \end{bmatrix}$$

Column Stacking

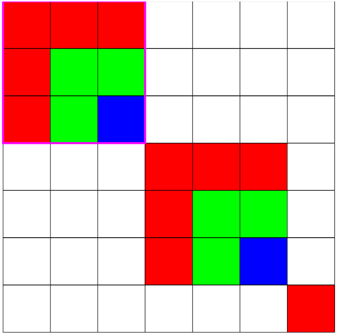




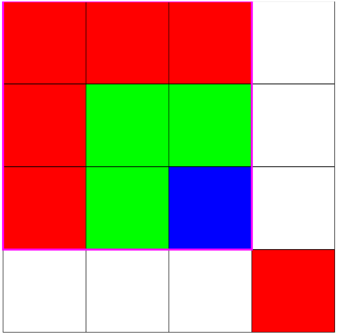
For this stacking the marginal distribution over *time* is given by the block diagonals.



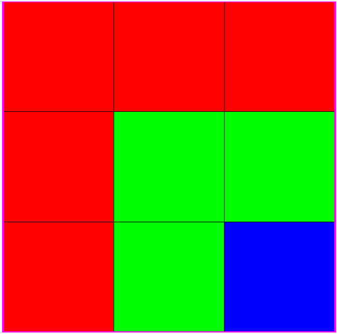
For this stacking the marginal distribution over *time* is given by the block diagonals.



For this stacking the marginal distribution over *time* is given by the block diagonals.



For this stacking the marginal distribution over *time* is given by the block diagonals.



For this stacking the marginal distribution over *time* is given by the block diagonals.

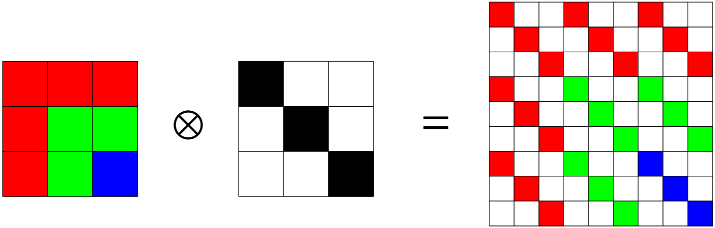
Two Ways of Stacking

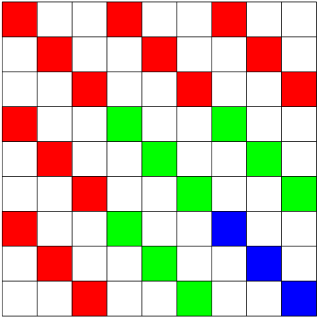
Can also stack each row of $\mathbf{X}$ to form column vector:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{1,:} \\ \mathbf{x}_{2,:} \\ \vdots \\ \mathbf{x}_{T,:} \end{bmatrix}$$

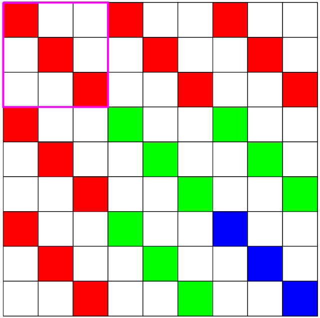
$$p(\mathbf{x}) = \mathcal{N}\left(\mathbf{x}|\mathbf{0}, \mathbf{K} \otimes \mathbf{I}\right)$$

Row Stacking

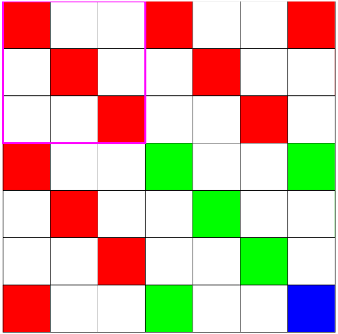




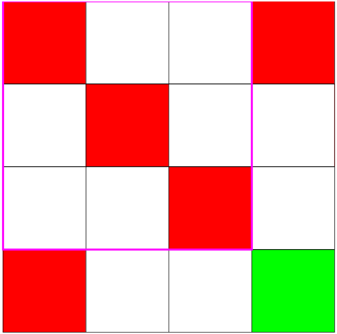
For this stacking the marginal distribution over the latent *dimensions* is given by the block diagonals.



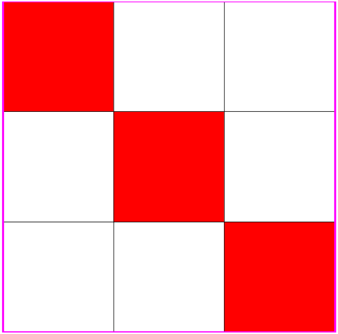
For this stacking the marginal distribution over the latent *dimensions* is given by the block diagonals.



For this stacking the marginal distribution over the latent *dimensions* is given by the block diagonals.



For this stacking the marginal distribution over the latent *dimensions* is given by the block diagonals.



For this stacking the marginal distribution over the latent *dimensions* is given by the block diagonals.

Observed Process

The observations are related to the latent points by a linear mapping matrix,

$$\mathbf{y}_{i,:} = \mathbf{W}\mathbf{x}_{i,:} + \boldsymbol{\epsilon}_{i,:}$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(0, \sigma^2\mathbf{I}\right)$$

Mapping from Latent Process to Observed

$$\begin{bmatrix} \mathbf{W} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{W} \end{bmatrix} \times \begin{bmatrix} \mathbf{x}_{1,:} \\ \mathbf{x}_{2,:} \\ \mathbf{x}_{3,:} \end{bmatrix} = \begin{bmatrix} \mathbf{W}\mathbf{x}_{1,:} \\ \mathbf{W}\mathbf{x}_{2,:} \\ \mathbf{W}\mathbf{x}_{3,:} \end{bmatrix}$$

Output Covariance

This leads to a covariance of the form

$$(\mathbf{I} \otimes \mathbf{W})(\mathbf{K} \otimes \mathbf{I})(\mathbf{I} \otimes \mathbf{W}^\top) + \mathbf{I}\sigma^2$$

Using $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}$ This leads to

$$\mathbf{K} \otimes \mathbf{W}\mathbf{W}^\top + \mathbf{I}\sigma^2$$

or

$$\mathbf{y} \sim \mathcal{N}\left(0, \mathbf{W}\mathbf{W}^\top \otimes \mathbf{K} + \mathbf{I}\sigma^2\right)$$

Kernels for Vector Valued Outputs: A Review

Foundations and Trends® in
Machine Learning
Vol. 4, No. 3 (2011) 195–266

DOI: 10.1561/2200000036

Kernels for Vector-Valued Functions: A Review

By Mauricio A. Álvarez,
Lorenzo Rosasco and Neil D. Lawrence

Kronecker Structure GPs

- This Kronecker structure leads to several published models.
$$(\mathbf{K}(\mathbf{x}, \mathbf{x}'))_{d,d'} = k(\mathbf{x}, \mathbf{x}')k_T(d, d'),$$
where k has $\mathbf{x}$ and k_T has n as inputs.
- Can think of multiple output covariance functions as covariances with augmented input.
- Alongside $\mathbf{x}$ we also input the d associated with the *output* of interest.

Separable Covariance Functions

- Taking $\mathbf{B} = \mathbf{W}\mathbf{W}^\top$ we have a matrix expression across outputs.
$$\mathbf{K}(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}')\mathbf{B},$$
where $\mathbf{B}$ is a $p \times p$ symmetric and positive semi-definite matrix.
- $\mathbf{B}$ is called the *coregionalization* matrix.
- We call this class of covariance functions *separable* due to their product structure.

Sum of Separable Covariance Functions

- In the same spirit a more general class of kernels is given by

$$\mathbf{K}(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{q} k_j(\mathbf{x}, \mathbf{x}')\mathbf{B}_j.$$

- This can also be written as

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \sum_{j=1}^{q} \mathbf{B}_j \otimes k_j(\mathbf{X}, \mathbf{X}),$$

- This is like several Kalman filter-type models added together, but each one with a different set of latent functions.
- We call this class of kernels sum of separable kernels (SoS kernels).

Geostatistics

- Use of GPs in Geostatistics is called kriging.
- These multi-output GPs pioneered in geostatistics: prediction over vector-valued output data is known as *cokriging*.
- The model in geostatistics is known as the *linear model of coregionalization* (LMC, Journel and Huijbregts (1978); Goovaerts (1997)).
- Most machine learning multitask models can be placed in the context of the LMC model.

Weighted sum of Latent Functions

- In the linear model of coregionalization (LMC) outputs are expressed as linear combinations of independent random functions.
- In the LMC, each component f_d is expressed as a linear sum

$$f_d(\mathbf{x}) = \sum_{j=1}^{q} w_{d,j} u_j(\mathbf{x}).$$

  where the latent functions are independent and have covariance functions $k_j(\mathbf{x}, \mathbf{x}')$.
- The processes $\{f_j(\mathbf{x})\}_{j=1}^{q}$ are independent for $q \neq j'$.

Kalman Filter Special Case

- The Kalman filter is an example of the LMC where $u_i(\mathbf{x}) \rightarrow x_i(t)$.
- I.e. we've moved form time input to a more general input space.
- In matrix notation:
  1. Kalman filter
  $$\mathbf{F} = \mathbf{W}\mathbf{X}$$
  2. LMC
  $$\mathbf{F} = \mathbf{W}\mathbf{U}$$

  where the rows of these matrices $\mathbf{F}$, $\mathbf{X}$, $\mathbf{U}$ each contain q samples from their corresponding functions at a different time (Kalman filter) or spatial location (LMC).

Intrinsic Coregionalization Model

- If one covariance used for latent functions (like in Kalman filter).
- This is called the intrinsic coregionalization model (ICM, Goovaerts (1997)).
- The kernel matrix corresponding to a dataset $\mathbf{X}$ takes the form

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{B} \otimes k(\mathbf{X}, \mathbf{X}).$$

Autokrigeability

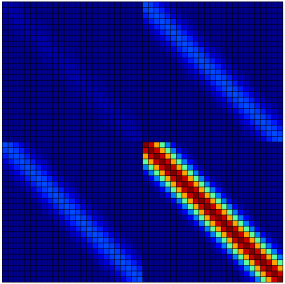
- If outputs are noise-free, maximum likelihood is equivalent to independent fits of $\mathbf{B}$ and $k(\mathbf{x}, \mathbf{x}')$ (Helterbrand and Cressie, 1994).
- In geostatistics this is known as autokrigeability (Wackernagel, 2003).
- In multitask learning its the cancellation of intertask transfer (Bonilla et al., 2008).

Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \mathbf{w}\mathbf{w}^\top \otimes k(\mathbf{X},\mathbf{X}).$$

$$\mathbf{w} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 5 \\ 5 & 25 \end{bmatrix}$$

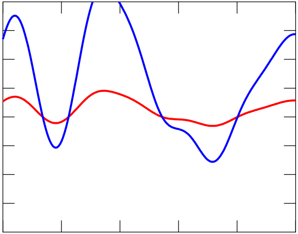


Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{w}\mathbf{w}^\top \otimes k(\mathbf{X}, \mathbf{X}).$$

$$\mathbf{w} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 5 \\ 5 & 25 \end{bmatrix}$$

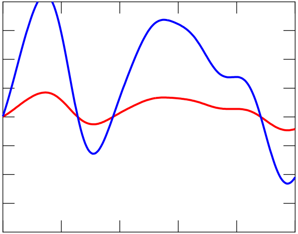


Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{w}\mathbf{w}^\top \otimes k(\mathbf{X}, \mathbf{X}).$$

$$\mathbf{w} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 5 \\ 5 & 25 \end{bmatrix}$$

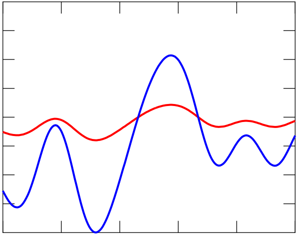


Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \mathbf{w}\mathbf{w}^{\top} \otimes k(\mathbf{X},\mathbf{X}).$$

$$\mathbf{w} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 5 \\ 5 & 25 \end{bmatrix}$$

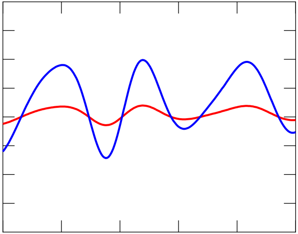


Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{w}\mathbf{w}^\top \otimes k(\mathbf{X}, \mathbf{X}).$$

$$\mathbf{w} = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

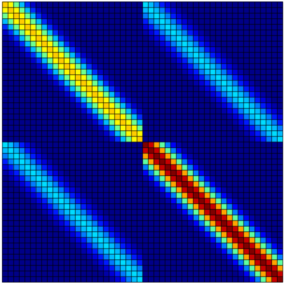
$$\mathbf{B} = \begin{bmatrix} 1 & 5 \\ 5 & 25 \end{bmatrix}$$



Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \mathbf{B} \otimes k(\mathbf{X},\mathbf{X}).$$

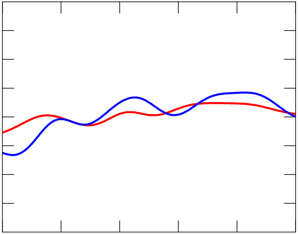
$$\mathbf{B} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.5 \end{bmatrix}$$



Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{B} \otimes k(\mathbf{X}, \mathbf{X}).$$

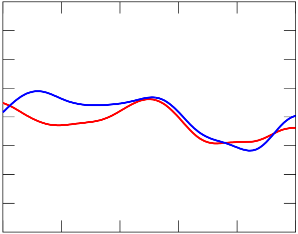
$$\mathbf{B} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.5 \end{bmatrix}$$



Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{B} \otimes k(\mathbf{X}, \mathbf{X}).$$

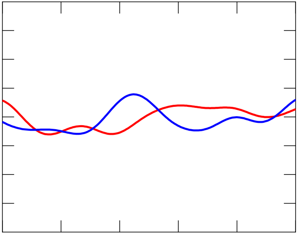
$$\mathbf{B} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.5 \end{bmatrix}$$



Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{B} \otimes k(\mathbf{X}, \mathbf{X}).$$

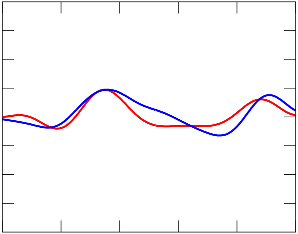
$$\mathbf{B} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.5 \end{bmatrix}$$



Intrinsic Coregionalization Model

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{B} \otimes k(\mathbf{X}, \mathbf{X}).$$

$$\mathbf{B} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.5 \end{bmatrix}$$



LMC Samples

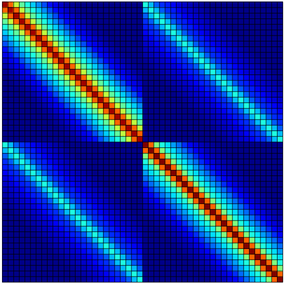
$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \mathbf{B}_1 \otimes k_1(\mathbf{X},\mathbf{X}) + \mathbf{B}_2 \otimes k_2(\mathbf{X},\mathbf{X})$$

$$\mathbf{B}_1 = \begin{bmatrix} 1.4 & 0.5 \\ 0.5 & 1.2 \end{bmatrix}$$

$$\ell_1 = 1$$

$$\mathbf{B}_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.3 \end{bmatrix}$$

$$\ell_2 = 0.2$$



LMC Samples

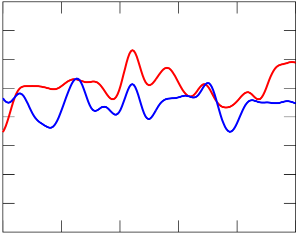
$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \mathbf{B}_1 \otimes k_1(\mathbf{X},\mathbf{X}) + \mathbf{B}_2 \otimes k_2(\mathbf{X},\mathbf{X})$$

$$\mathbf{B}_1 = \begin{bmatrix} 1.4 & 0.5 \\ 0.5 & 1.2 \end{bmatrix}$$

$$\ell_1 = 1$$

$$\mathbf{B}_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.3 \end{bmatrix}$$

$$\ell_2 = 0.2$$



LMC Samples

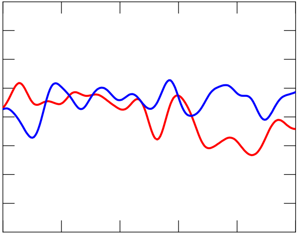
$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \mathbf{B}_1 \otimes k_1(\mathbf{X},\mathbf{X}) + \mathbf{B}_2 \otimes k_2(\mathbf{X},\mathbf{X})$$

$$\mathbf{B}_1 = \begin{bmatrix} 1.4 & 0.5 \\ 0.5 & 1.2 \end{bmatrix}$$

$$\ell_1 = 1$$

$$\mathbf{B}_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.3 \end{bmatrix}$$

$$\ell_2 = 0.2$$



LMC Samples

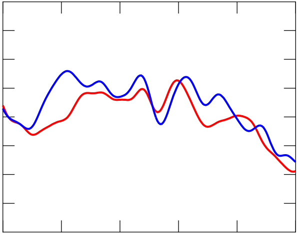
$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{B}_1 \otimes k_1(\mathbf{X}, \mathbf{X}) + \mathbf{B}_2 \otimes k_2(\mathbf{X}, \mathbf{X})$$

$$\mathbf{B}_1 = \begin{bmatrix} 1.4 & 0.5 \\ 0.5 & 1.2 \end{bmatrix}$$

$$\ell_1 = 1$$

$$\mathbf{B}_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.3 \end{bmatrix}$$

$$\ell_2 = 0.2$$



LMC Samples

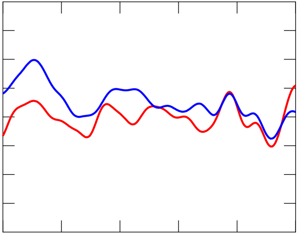
$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{B}_1 \otimes k_1(\mathbf{X}, \mathbf{X}) + \mathbf{B}_2 \otimes k_2(\mathbf{X}, \mathbf{X})$$

$$\mathbf{B}_1 = \begin{bmatrix} 1.4 & 0.5 \\ 0.5 & 1.2 \end{bmatrix}$$

$$\ell_1 = 1$$

$$\mathbf{B}_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1.3 \end{bmatrix}$$

$$\ell_2 = 0.2$$



LMC in Machine Learning and Statistics

- Used in machine learning for GPs for multivariate regression and in statistics for computer emulation of expensive multivariate computer codes.
- Imposes the correlation of the outputs explicitly through the set of coregionalization matrices.
- Setting $\mathbf{B} = \mathbf{I}_p$ assumes outputs are conditionally independent given the parameters $\boldsymbol{\theta}$. (Minka and Picard, 1997; Lawrence and Platt, 2004; Yu et al., 2005).
- More recent approaches for multiple output modeling are different versions of the linear model of coregionalization.

Semiparametric Latent Factor Model

- Coregionalization matrices are rank 1 Teh et al. (2005). rewrite equation (**??**) as

$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \sum_{j=1}^{q} \mathbf{w}_{:,j}\mathbf{w}_{:,j}^{\top} \otimes k_j(\mathbf{X},\mathbf{X}).$$

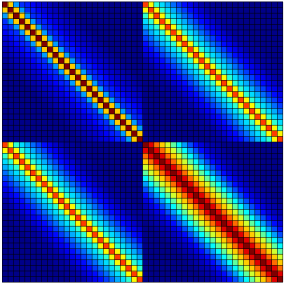
- Like the Kalman filter, but each latent function has a *different* covariance.
- Authors suggest using an exponentiated quadratic characteristic length-scale for each input dimension.

Semiparametric Latent Factor Model Samples

$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \mathbf{w}_{:,1}\mathbf{w}_{:,1}^{\top} \otimes k_1(\mathbf{X},\mathbf{X}) + \mathbf{w}_{:,2}\mathbf{w}_{:,2}^{\top} \otimes k_2(\mathbf{X},\mathbf{X})$$

$$\mathbf{w}_1 = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

$$\mathbf{w}_2 = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

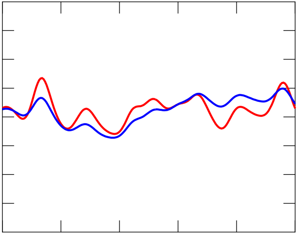


Semiparametric Latent Factor Model Samples

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{w}_{:,1}\mathbf{w}_{:,1}^{\top} \otimes k_1(\mathbf{X}, \mathbf{X}) + \mathbf{w}_{:,2}\mathbf{w}_{:,2}^{\top} \otimes k_2(\mathbf{X}, \mathbf{X})$$

$$\mathbf{w}_1 = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

$$\mathbf{w}_2 = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

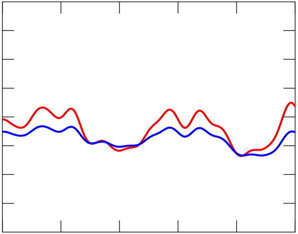


Semiparametric Latent Factor Model Samples

$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \mathbf{w}_{:,1}\mathbf{w}_{:,1}^{\top} \otimes k_1(\mathbf{X},\mathbf{X}) + \mathbf{w}_{:,2}\mathbf{w}_{:,2}^{\top} \otimes k_2(\mathbf{X},\mathbf{X})$$

$$\mathbf{w}_1 = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

$$\mathbf{w}_2 = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

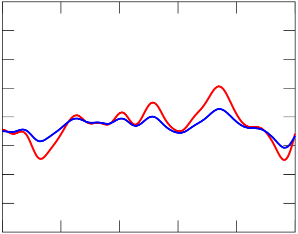


Semiparametric Latent Factor Model Samples

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{w}_{:,1}\mathbf{w}_{:,1}^{\top} \otimes k_1(\mathbf{X}, \mathbf{X}) + \mathbf{w}_{:,2}\mathbf{w}_{:,2}^{\top} \otimes k_2(\mathbf{X}, \mathbf{X})$$

$$\mathbf{w}_1 = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

$$\mathbf{w}_2 = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

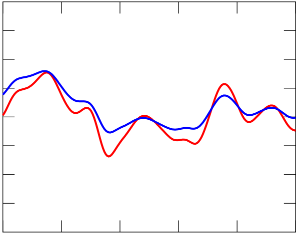


Semiparametric Latent Factor Model Samples

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \mathbf{w}_{:,1}\mathbf{w}_{:,1}^\top \otimes k_1(\mathbf{X}, \mathbf{X}) + \mathbf{w}_{:,2}\mathbf{w}_{:,2}^\top \otimes k_2(\mathbf{X}, \mathbf{X})$$

$$\mathbf{w}_1 = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

$$\mathbf{w}_2 = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$



Gaussian processes for Multi-task, Multi-output and Multi-class

- Bonilla et al. (2008) suggest ICM for multitask learning.
- Use a PPCA form for $\mathbf{B}$: similar to our Kalman filter example.
- Refer to the autokrigeability effect as the cancellation of inter-task transfer.
- Also discuss the similarities between the multi-task GP and the ICM, and its relationship to the SLFM and the LMC.

Multitask Classification

- Mostly restricted to the case where the outputs are conditionally independent given the hyperparameters ϕ (Minka and Picard, 1997; Williams and Barber, 1998; Lawrence and Platt, 2004; Seeger and Jordan, 2004; Yu et al., 2005; Rasmussen and Williams, 2006).
- Intrinsic coregionalization model has been used in the multiclass scenario. Skolidis and Sanguinetti (2011) use the intrinsic coregionalization model for classification, by introducing a probit noise model as the likelihood.
- Posterior distribution is no longer analytically tractable: approximate inference is required.

Computer Emulation

- A statistical model used as a surrogate for a computationally expensive computer model.
- Higdon et al. (2008) use the linear model of coregionalization to model images representing the evolution of the implosion of steel cylinders.
- In Conti and O'Hagan (2009) use the ICM to model a vegetation model: called the Sheffield Dynamic Global Vegetation Model (Woodward et al., 1998).

References I

E. V. Bonilla, K. M. Chai, and C. K. I. Williams. Multi-task Gaussian process prediction. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20, Cambridge, MA, 2008. MIT Press.

S. Conti and A. O'Hagan. Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140(3):640–651, 2009. [DOI].

P. Goovaerts. *Geostatistics For Natural Resources Evaluation*. Oxford University Press, 1997. [Google Books] .

J. D. Helterbrand and N. A. C. Cressie. Universal cokriging under intrinsic coregionalization. *Mathematical Geology*, 26(2):205–226, 1994.

D. M. Higdon, J. Gattiker, B. Williams, and M. Rightley. Computer model calibration using high dimensional output. *Journal of the American Statistical Association*, 103(482):570–583, 2008.

A. G. Journel and C. J. Huijbregts. *Mining Geostatistics*. Academic Press, London, 1978. [Google Books] .

N. D. Lawrence and J. C. Platt. Learning to learn with the informative vector machine. In R. Greiner and D. Schuurmans, editors, *Proceedings of the International Conference in Machine Learning*, volume 21, pages 512–519. Omnipress, 2004. [PDF].

T. P. Minka and R. W. Picard. Learning how to learn is learning with point sets. Available on-line., 1997. [URL]. Revised 1999, available at `http://www.stat.cmu.edu/~{}minka/`.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006. [Google Books] .

M. Seeger and M. I. Jordan. Sparse Gaussian Process Classification With Multiple Classes. Technical Report 661, Department of Statistics, University of California at Berkeley,

G. Skolidis and G. Sanguinetti. Bayesian multitask classification with Gaussian process priors. *IEEE Transactions on Neural Networks*, 22(12):2011 – 2021, 2011.

Y. W. Teh, M. Seeger, and M. I. Jordan. Semiparametric latent factor models. In R. G. Cowell and Z. Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 333–340, Barbados, 6-8 January 2005. Society for Artificial Intelligence and Statistics.

H. Wackernagel. *Multivariate Geostatistics: An Introduction With Applications*. Springer-Verlag, 3rd edition, 2003. [Google Books] .

References II

C. K. Williams and D. Barber. Bayesian Classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998.

I. Woodward, M. R. Lomas, and R. A. Betts. Vegetation-climate feedbacks in a greenhouse world. *Philosophical Transactions: Biological Sciences*, 353(1365):29–39, 1998.

K. Yu, V. Tresp, and A. Schwaighofer. Learning Gaussian processes from multiple tasks. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*, pages 1012–1019, 2005.