CS-E4895 Gaussian Processes
Lecture 10: Latent modelling and unsupervised learning

Markus Heinonen

Aalto University

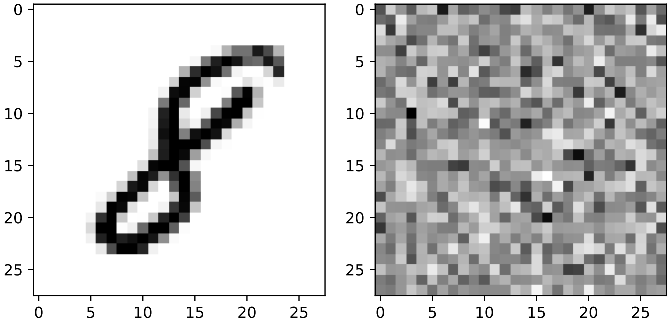
Tuesday 28.3.2023

Agenda for today

- Introduction
  - Why are LVMs useful?
  - Definition of LVMs
- Gaussian process latent variable models
  - Principal Component Analaysis
  - Probabilistic PCA
  - Dual probabilistic PCA
  - GPLVM
- Multi-output models
  - Intrinsic Model of Coregionalisation
  - Semiparametric Latent Factor Model
  - Linear Model of Coregionalisation

Why are LVMs useful?

- Data $\mathbf{x}$ has structure

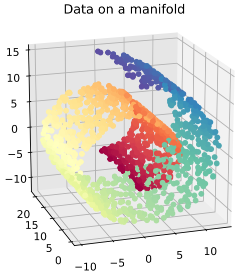


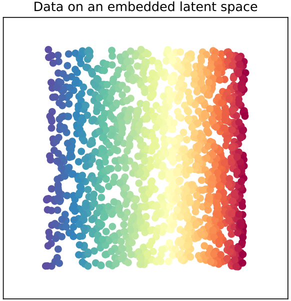
- ... the dimension of this space is large $(D = 784)$
- ... you would never sample this digit randomly

Why are LVMs useful?

- These samples lie on a **very** narrow manifold in $\mathbb{R}^{28\times28}$
- We should only require enough dimensions to describe the digit sufficiently
  - e.g. shape and distortions (rotation, translation, stretching)
- The number of these dimensions is called the *intrinsic dimensionality* and is often significantly smaller than the number of features.
- Its often far easier to perform your inference on this embedded manifold

Moving from $\mathbb{R}^3$ to $\mathbb{R}^2$





Setting:

- Features $\mathbf{X} \in \mathbb{R}^{N \times D}$, latent variables $\mathbf{Z} \in \mathbb{R}^{N \times Q}$, N datapoints
- Often our goal is to find a lower-dimensional latent representation $Q < D$
- Unsupervised learning: no output covariates

Definition of LVMs

Dimensionality reduction: Learning a projection onto a lower dimensional embedding $\mathbf{z}$

Manifold learning: Learning this embedding and a *map* $g : \mathbf{z} \rightarrow \mathbf{x}$

A latent variable model is of the form:

$$\mathbf{x} = g(\mathbf{z}) + \epsilon, \qquad \epsilon \sim p(\epsilon) \tag{1}$$

Our goal is to learn the embedding $\mathbf{z}_n$ of datapoint $\mathbf{x}_n$

Often such models make assumptions of

- Independence across latent samples
- Conditional independence across features given latent samples

Graphical model representation

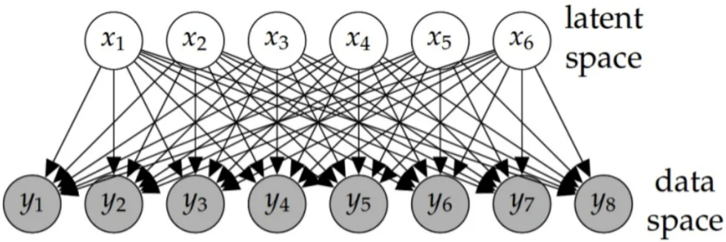
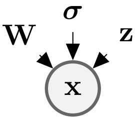


Image from Neil Lawrence, GPSS 2014
(We use $\mathbf{x}$ for data and $\mathbf{z}$ for latents)

The Gaussian process latent variable model (GPLVM)

Principal Component Analysis



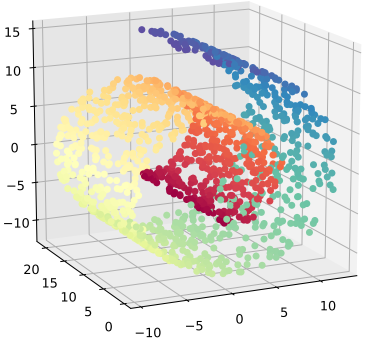
$$\underbrace{\mathbf{x}_n}_{\mathbb{R}^D} = \underbrace{\mathbf{W}}_{\mathbb{R}^{D \times Q}} \underbrace{\mathbf{z}_n}_{\mathbb{R}^Q} + \underbrace{\boldsymbol{\epsilon}_n}_{\mathbb{R}^D}, \quad \boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_D) \tag{2}$$

- Linear projection $\mathbf{W} : \mathbb{R}^Q \mapsto \mathbb{R}^D$
- ... such that the new basis $\mathbf{z}$ is comprised of *principal components*
- ... which span the directions of greatest variance

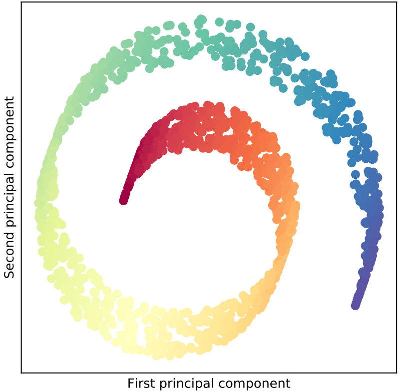
Only works well if data lies on a plane in high dimensional space

No representation of uncertainty

Principal Component Analysis



(a) Swiss roll data



(b) PCA embedding

This is not optimal. This embedding does not capture all of the variance!

Probabilistic PCA

- Likelihood for $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$

$$p(\mathbf{X}|\mathbf{W}, \mathbf{Z}, \sigma) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n|\mathbf{W}\mathbf{z}_n, \sigma^2\mathbf{I}). \tag{3}$$

Probabilistic PCA

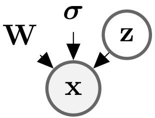
- Likelihood for $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$

$$p(\mathbf{X}|\mathbf{W}, \mathbf{Z}, \sigma) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n | \mathbf{W}\mathbf{z}_n, \sigma^2 \mathbf{I}). \tag{3}$$

- Gaussian prior over embeddings $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_N)$,

$$p(\mathbf{Z}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{z}_n | \mathbf{0}, \mathbf{I}). \tag{4}$$

Probabilistic PCA



- Likelihood for $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$

$$p(\mathbf{X}|\mathbf{W}, \mathbf{Z}, \sigma) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n|\mathbf{W}\mathbf{z}_n, \sigma^2\mathbf{I}). \tag{3}$$

- Gaussian prior over embeddings $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_N)$,

$$p(\mathbf{Z}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{z}_n|\mathbf{0}, \mathbf{I}). \tag{4}$$

- Integrate over latent variables $\mathbf{Z}$ to obtain the marginal likelihood

$$p(\mathbf{X}|\mathbf{W}, \sigma) = \prod_{n=1}^{N} \int p(\mathbf{x}_n|\mathbf{W}, \mathbf{z}_n, \sigma)p(\mathbf{z}_n)d\mathbf{z}_n. \tag{5}$$

Probabilistic PCA

Marginal likelihood

$$p(\mathbf{X}|\mathbf{W},\sigma) = \prod_{n=1}^{N} \int p(\mathbf{x}_n|\mathbf{W},\mathbf{z}_n,\sigma)p(\mathbf{z}_n)d\mathbf{z}_n \tag{6}$$

$$= \prod_{n=1}^{N} \int \mathcal{N}(\mathbf{x}_n|\mathbf{W}\mathbf{z}_n,\sigma^2\mathbf{I})\mathcal{N}(\mathbf{z}_n|\mathbf{0},\mathbf{I})d\mathbf{z}_n. \tag{7}$$

Probabilistic PCA

Marginal likelihood

$$p(\mathbf{X}|\mathbf{W},\sigma) = \prod_{n=1}^{N} \int p(\mathbf{x}_n|\mathbf{W},\mathbf{z}_n,\sigma)p(\mathbf{z}_n)d\mathbf{z}_n \tag{6}$$

$$= \prod_{n=1}^{N} \int \mathcal{N}(\mathbf{x}_n|\mathbf{W}\mathbf{z}_n,\sigma^2\mathbf{I})\mathcal{N}(\mathbf{z}_n|\mathbf{0},\mathbf{I})d\mathbf{z}_n. \tag{7}$$

We can derive

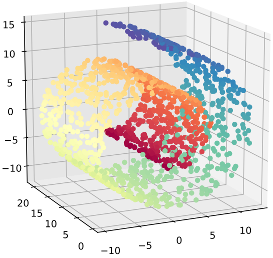
$$p(\mathbf{X}|\mathbf{W},\sigma) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n|\mathbf{0},\underbrace{\mathbf{W}\mathbf{W}^T}_{\mathbb{R}^{D\times D}}+\sigma^2\mathbf{I}) \tag{8}$$

due to

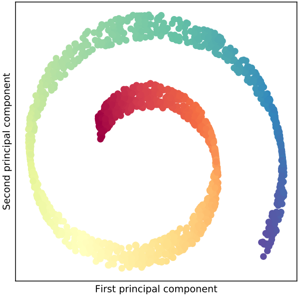
$$\mathrm{cov}[\mathbf{x}] = \mathbb{E}[(\mathbf{W}\mathbf{z}+\epsilon)((\mathbf{W}\mathbf{z}+\epsilon))^T] \tag{9}$$

$$= \mathbb{E}[\mathbf{W}\mathbf{z}\mathbf{z}^T\mathbf{W}^T] + \mathbb{E}[\epsilon\epsilon^T] = \mathbf{W}\mathbf{W}^T + \sigma^2 I. \tag{10}$$

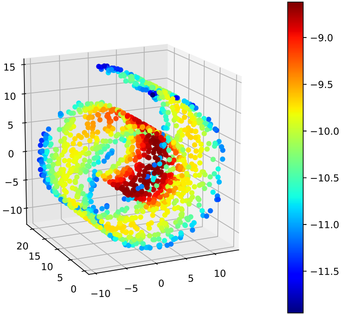
Probabilistic PCA - Swiss roll example



(a) Swiss roll data



(b) PCA embedding



(c) Log-likelihood

Dual PPCA

- Likelihood

$$p(\mathbf{X}|\mathbf{W},\mathbf{Z},\sigma) = \prod_{d=1}^{D} \mathcal{N}(\underbrace{\mathbf{x}_{d\cdot}}_{\mathbb{R}^N} \mid \underbrace{\mathbf{Z}}_{\mathbb{R}^{N\times Q}} \underbrace{\mathbf{w}_{d\cdot}^T}_{\mathbb{R}^{Q\times 1}}, \sigma^2\mathbf{I}) \tag{11}$$

Dual PPCA

- Likelihood

$$p(\mathbf{X}|\mathbf{W},\mathbf{Z},\sigma) = \prod_{d=1}^{D} \mathcal{N}(\underbrace{\mathbf{x}_{d\cdot}}_{\mathbb{R}^N} \mid \underbrace{\mathbf{Z}}_{\mathbb{R}^{N\times Q}} \underbrace{\mathbf{w}_{d\cdot}^T}_{\mathbb{R}^{Q\times 1}}, \sigma^2\mathbf{I}) \tag{11}$$

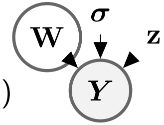
- Gaussian prior over the space of linear transformations

$$p(\mathbf{W}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{w}_{d\cdot}|\mathbf{0},\mathbf{I}) \tag{12}$$

Dual PPCA

- Likelihood

$$p(\mathbf{X}|\mathbf{W},\mathbf{Z},\sigma) = \prod_{d=1}^{D} \mathcal{N}(\underbrace{\mathbf{x}_{d\cdot}}_{\mathbb{R}^N} \mid \underbrace{\mathbf{Z}}_{\mathbb{R}^{N\times Q}} \underbrace{\mathbf{w}_{d\cdot}^T}_{\mathbb{R}^{Q\times 1}}, \sigma^2\mathbf{I}) \tag{11}$$



- Gaussian prior over the space of linear transformations

$$p(\mathbf{W}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{w}_{d\cdot}|\mathbf{0},\mathbf{I}) \tag{12}$$

- ... and integrate over transformation $\mathbf{W}$ to obtain the marginal likelihood

$$p(\mathbf{X}|\mathbf{Z},\sigma) = \prod_{d=1}^{D} \int p(\mathbf{x}_{d\cdot}|\mathbf{W},\mathbf{z},\sigma)p(\mathbf{W})d\mathbf{W} = \prod_{d=1}^{D} \mathcal{N}(\mathbf{x}_{d\cdot}|\mathbf{0}, \underbrace{\mathbf{Z}\mathbf{Z}^T}_{\mathbb{R}^{N\times N}} + \sigma^2\mathbf{I})$$

The kernel

- The dual PPCA marginal likelihood

$$p(\mathbf{X}|\mathbf{Z}, \sigma) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{x}_{d\cdot}|\mathbf{0}, \mathbf{Z}\mathbf{Z}^T + \sigma^2 \mathbf{I}) \tag{13}$$

- The covariance matrix $\mathbf{K} = \mathbf{Z}\mathbf{Z}^T + \sigma^2 \mathbf{I}$ is a linear kernel
- The marginal likelihood for DPPCA is a product of D independent Gaussian processes with a linear kernel
- We can change this kernel and obtain the GPLVM class of models

$$x_d(\mathbf{z}) \sim \mathcal{GP}\big(m(\mathbf{z}), k(\mathbf{z}, \mathbf{z}')\big), \tag{14}$$

where $d = 1, \ldots, D$ iterates over data dimensions

GPLVM summary

- DPPCA is a special case of GPLVM with a linear kernel
- Each dimension x_d of the marginal can be interpreted as an independent GP
- Each dimension x_d is *a priori* assumed independent and identically distributed

GPLVM inference

Analytic solutions

- For PCA, PPCA and DPPCA, an analytic solution exists via solving an eigenvalue problem

[1] where θ are the kernel hyper-parameters and, for example, $\mathbf{K} = \mathbf{z}\mathbf{z}^T + \sigma^2 I$ in the linear kernel case

GPLVM inference

Analytic solutions

- For PCA, PPCA and DPPCA, an analytic solution exists via solving an eigenvalue problem

Maximum marginal likelihood (MLL)

- Once we use a non-linear kernel analytical solutions often become intractable
- Instead we may resort to gradient based optimisation for $(\mathbf{z}, \theta, \sigma)$.[1]

$$\hat{\mathbf{Z}}, \hat{\theta}, \hat{\sigma} = \underset{\mathbf{Z},\theta,\sigma}{\arg\max} \log(\mathbf{X}|\mathbf{Z}, \theta, \sigma) \propto \underset{\mathbf{z},\theta,\sigma}{\arg\max} \left\{ -\frac{D}{2} \log |\mathbf{K_{ZZ}}| - \frac{1}{2} \text{tr}(\mathbf{K}_{\mathbf{ZZ}}^{-1} \mathbf{X}\mathbf{X}^T) \right\}$$

- Initialisation
  - Initialise $\mathbf{z}_n$ for each $\mathbf{x}_n$ randomly
  - Initialise $\mathbf{Z}$ from standard PCA of $\mathbf{X}$

[1] where θ are the kernel hyper-parameters and, for example, $\mathbf{K} = \mathbf{z}\mathbf{z}^T + \sigma^2 I$ in the linear kernel case

GPLVM inference - MAP

Maximum a marginal posterior (MAMP)

- Place a prior on latent variables $\mathbf{Z}$

$$\hat{\mathbf{Z}}, \hat{\theta}, \hat{\sigma} = \arg\max_{\mathbf{Z},\theta,\sigma} \left\{ \log p(\mathbf{X}|\mathbf{Z}, \theta, \sigma) + \log p(\mathbf{Z}) \right\}$$

- We again use gradient based optimisation
- This prior acts to regularise the latent variables

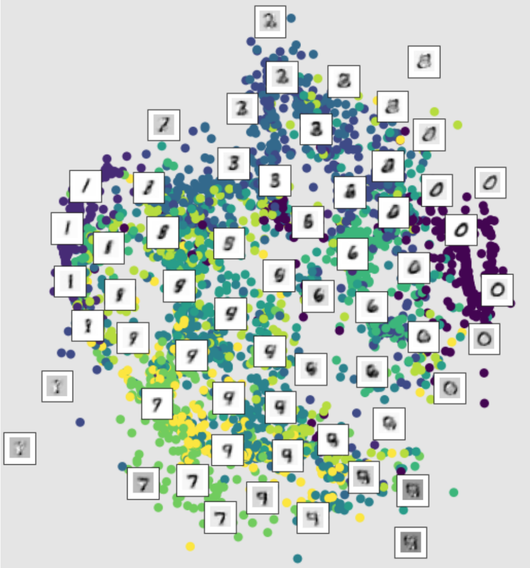
These both optimise over a huge space, but don't do as poorly as you'd expect!

GPLVM - Caveats and practical points

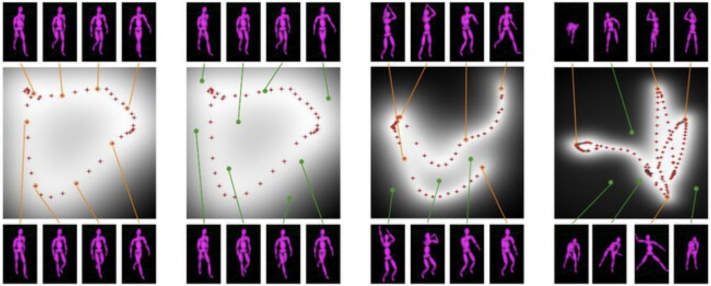
- Optimisation is *very* non-convex.
  - Multiple restarts, initialisation. How do we initialise?
- What is the latent dimensionality?
- Computational cost.

GPLVM for MNIST

- The images $\mathbf{x}_n$ are fixed
- We learn the layout of the embeddings: here two-dimensional
- How to interpolate between images? Use GP regression



GPLVM for motion capture



Bayesian GPLVM

- We may also want to place a prior on $\mathbf{z}$ and integrate it out[2]. The $\mathbf{x}_d$ is the d'th column of $\mathbf{X}$,

$$p(\mathbf{X}|\mathbf{Z}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{x}_d|\mathbf{0}, \mathbf{K}_{\mathbf{Z}\mathbf{Z}} + \sigma^2\mathbf{I})$$

$$p(\mathbf{X}) = \int p(\mathbf{X}|\mathbf{Z})p(\mathbf{Z})d\mathbf{Z} \quad \text{and introduce} \quad p(\mathbf{Z}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{z}_n|\mathbf{0}, \mathbf{I})$$

- This is intractable as $\mathbf{z}$ appears non-linearly in the inverse of the kernel
- Lets try and apply the standard *variational Bayes* approach

[2] We omit θ and σ from the notation for clarity, but they are still part of the model.

Bayesian GPLVM

Introduce a variational distribution

$$p(\mathbf{Z}|\mathbf{X}) \approx q(\mathbf{Z}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{z}_n|\mathbf{m}_n, S_n)$$

And compute the Jensen's lower bound

$$\log p(\mathbf{X}) \geq \underbrace{\sum_{d=1}^{D} \int q(\mathbf{Z}) \log p(\mathbf{x}_d|\mathbf{Z}) d\mathbf{Z}}_{\text{this remains intractable}} - \underbrace{\int q(\mathbf{Z}) \log \frac{q(\mathbf{Z})}{p(\mathbf{Z})} d\mathbf{Z}}_{KL(q||p)}$$

Lets apply the variational sparse methodology of [1] that we learnt in previous lectures

Bayesian GPLVM (revisiting the variational sparse approach)

Lets expand our intractable integral term and augment with inducing variables[3]

$$p(\mathbf{x}_d, \mathbf{f}_d, \mathbf{u}_d | \mathbf{Z}, \mathbf{V}) = p(\mathbf{x}_d | \mathbf{f}_d) p(\mathbf{f}_d | \mathbf{u}_d, \mathbf{Z}, \mathbf{V}) p(\mathbf{u}_d | \mathbf{V})$$

where

$$\begin{aligned} p(\mathbf{x}_d | \mathbf{f}_d) &= \mathcal{N}(\mathbf{x}_d | \mathbf{f}_d, \sigma^2 \mathbf{I}) \\ p(\mathbf{f}_d | \mathbf{u}_d, \mathbf{Z}, \mathbf{V}) &= \mathcal{N}(\mathbf{f}_d | \mathbf{K}_{\mathbf{ZV}} \mathbf{K}_{\mathbf{VV}}^{-1} \mathbf{u}_d, \mathbf{K}_{\mathbf{ZZ}} - \mathbf{K}_{\mathbf{ZV}} \mathbf{K}_{\mathbf{VV}}^{-1} \mathbf{K}_{\mathbf{VZ}}) \\ p(\mathbf{u}_d | \mathbf{V}) &= \mathcal{N}(\mathbf{u}_d | \mathbf{0}, \mathbf{K}_{\mathbf{VV}}). \end{aligned}$$

[3]**Notation reminder:** inducing inputs $\mathbf{V} \in \mathbb{R}^{M \times Q}$, inducing outputs $\mathbf{u}_d \in \mathbb{R}^M$, GP outputs $\mathbf{f}_d \in \mathbb{R}^N$.

Bayesian GPLVM (revisiting the variational sparse approach)

Lets expand our intractable integral term and augment with inducing variables[3]

$$p(\mathbf{x}_d, \mathbf{f}_d, \mathbf{u}_d | \mathbf{Z}, \mathbf{V}) = p(\mathbf{x}_d | \mathbf{f}_d) p(\mathbf{f}_d | \mathbf{u}_d, \mathbf{Z}, \mathbf{V}) p(\mathbf{u}_d | \mathbf{V})$$

where

$$\begin{aligned} p(\mathbf{x}_d | \mathbf{f}_d) &= \mathcal{N}(\mathbf{x}_d | \mathbf{f}_d, \sigma^2 \mathbf{I}) \\ p(\mathbf{f}_d | \mathbf{u}_d, \mathbf{Z}, \mathbf{V}) &= \mathcal{N}(\mathbf{f}_d | \mathbf{K}_{\mathbf{ZV}} \mathbf{K}_{\mathbf{VV}}^{-1} \mathbf{u}_d, \mathbf{K}_{\mathbf{ZZ}} - \mathbf{K}_{\mathbf{ZV}} \mathbf{K}_{\mathbf{VV}}^{-1} \mathbf{K}_{\mathbf{VZ}}) \\ p(\mathbf{u}_d | \mathbf{V}) &= \mathcal{N}(\mathbf{u}_d | \mathbf{0}, \mathbf{K}_{\mathbf{VV}}). \end{aligned}$$

Derive a variational approximation for the posterior

$$p(\mathbf{f}_d, \mathbf{u}_d | \mathbf{X}, \mathbf{Z}, \mathbf{V}) \approx q(\mathbf{f}_n, \mathbf{u}_d) = p(\mathbf{f}_d | \mathbf{u}_d, \mathbf{Z}, \mathbf{V}) q(\mathbf{u}_d)$$

[3] **Notation reminder:** inducing inputs $\mathbf{V} \in \mathbb{R}^{M \times Q}$, inducing outputs $\mathbf{u}_d \in \mathbb{R}^M$, GP outputs $\mathbf{f}_d \in \mathbb{R}^N$.

Bayesian GPLVM (revisiting the variational sparse approach)

Lets expand our intractable integral term and augment with inducing variables[3]

$$p(\mathbf{x}_d, \mathbf{f}_d, \mathbf{u}_d | \mathbf{Z}, \mathbf{V}) = p(\mathbf{x}_d | \mathbf{f}_d) p(\mathbf{f}_d | \mathbf{u}_d, \mathbf{Z}, \mathbf{V}) p(\mathbf{u}_d | \mathbf{V})$$

where

$$\begin{aligned} p(\mathbf{x}_d | \mathbf{f}_d) &= \mathcal{N}(\mathbf{x}_d | \mathbf{f}_d, \sigma^2 \mathbf{I}) \\ p(\mathbf{f}_d | \mathbf{u}_d, \mathbf{Z}, \mathbf{V}) &= \mathcal{N}(\mathbf{f}_d | \mathbf{K}_{\mathbf{ZV}} \mathbf{K}_{\mathbf{VV}}^{-1} \mathbf{u}_d, \mathbf{K}_{\mathbf{ZZ}} - \mathbf{K}_{\mathbf{ZV}} \mathbf{K}_{\mathbf{VV}}^{-1} \mathbf{K}_{\mathbf{VZ}}) \\ p(\mathbf{u}_d | \mathbf{V}) &= \mathcal{N}(\mathbf{u}_d | \mathbf{0}, \mathbf{K}_{\mathbf{VV}}). \end{aligned}$$

Derive a variational approximation for the posterior

$$p(\mathbf{f}_d, \mathbf{u}_d | \mathbf{X}, \mathbf{Z}, \mathbf{V}) \approx q(\mathbf{f}_n, \mathbf{u}_d) = p(\mathbf{f}_d | \mathbf{u}_d, \mathbf{Z}, \mathbf{V}) q(\mathbf{u}_d)$$

And we can use this to derive a new lower bound

$$\int q(\mathbf{Z}) \log p(\mathbf{x}_d | \mathbf{Z}) d\mathbf{Z} \geq \int q(\mathbf{Z}) q(\mathbf{f}_d, \mathbf{u}_d) \log \frac{p(\mathbf{x}_d, \mathbf{f}_d, \mathbf{u}_d | \mathbf{Z}, \mathbf{V})}{q(\mathbf{f}_d, \mathbf{u}_d)} d\mathbf{f}_d d\mathbf{u}_d d\mathbf{Z}$$

[3] **Notation reminder:** inducing inputs $\mathbf{V} \in \mathbb{R}^{M \times Q}$, inducing outputs $\mathbf{u}_d \in \mathbb{R}^M$, GP outputs $\mathbf{f}_d \in \mathbb{R}^N$.

Bayesian GPLVM

And we can use this to derive a new lower bound

$$\int q(\mathbf{Z}) \log p(\mathbf{x}_d|\mathbf{Z}) d\mathbf{Z} \geq \int q(\mathbf{Z}) q(\mathbf{f}_d, \mathbf{u}_d) \log \frac{p(\mathbf{x}_d|\mathbf{f}_d) p(\mathbf{u}_d|\mathbf{V})}{q(\mathbf{u}_d)} d\mathbf{f}_d d\mathbf{u}_d d\mathbf{Z}$$

$$= \int q(\mathbf{Z}) q(\mathbf{f}_d, \mathbf{u}_d) \log p(\mathbf{x}_d|\mathbf{f}_d) d\mathbf{f}_d d\mathbf{u}_d d\mathbf{Z} - \mathsf{KL}(q(\mathbf{u}_d)||p(\mathbf{u}_d|\mathbf{V}))$$

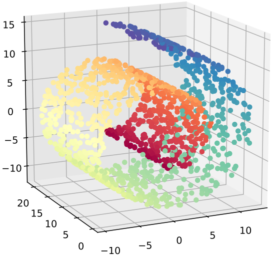
Bayesian GPLVM

And we can use this to derive a new lower bound

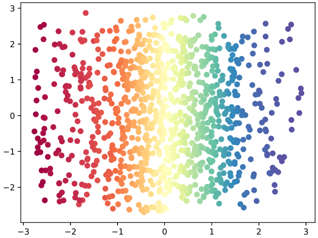
$$\begin{aligned}\int q(\mathbf{Z}) \log p(\mathbf{x}_d|\mathbf{Z}) d\mathbf{Z} &\geq \int q(\mathbf{Z}) q(\mathbf{f}_d, \mathbf{u}_d) \log \frac{p(\mathbf{x}_d|\mathbf{f}_d) p(\mathbf{u}_d|\mathbf{V})}{q(\mathbf{u}_d)} d\mathbf{f}_d d\mathbf{u}_d d\mathbf{Z} \\ &= \int q(\mathbf{Z}) q(\mathbf{f}_d, \mathbf{u}_d) \log p(\mathbf{x}_d|\mathbf{f}_d) d\mathbf{f}_d d\mathbf{u}_d d\mathbf{Z} - \mathsf{KL}(q(\mathbf{u}_d)||p(\mathbf{u}_d|\mathbf{V}))\end{aligned}$$

- We can integrate $q(\mathbf{Z})$ tractably first, and rest of the integral is tractable as well
- For more details see [2] or [3]

Bayesian GPLVM - example



(a) Swiss roll data



(b) BGPLVM embedded mean

The many flavours of GPLVM

- Shared GPLVM - map from a shared latent space to separate observation spaces
- Back constrained GPLVM - preserving locality in the image map
- Dynamic GPLVM (or GP dynamical model) - add a dynamic prior for supervised learning
- 'Deep' GPs - add a GP prior onto $\mathbf{Z}$.

... and many more!

The applications of GPLVM

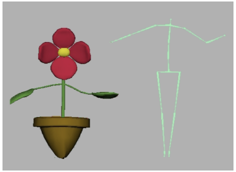


Figure: Shared GPLVM: Disney research (link)

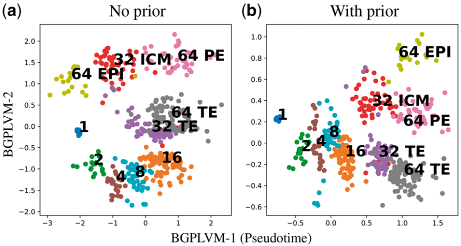


Figure: BGPLVM for single cell data

Further reading

Models

- [4] - PCA $\rightarrow$ PPCA $\rightarrow$ DPPCA $\rightarrow$ GPLVM derivation details
- [3] - Bayesian GPLVM paper
- [2] - Bayesian GPLVM thesis
- [5] - Back constrained GPLVM
- [6] - Shared GPLVM

Applications

- [7] - Disney research Shared GPLVM for animation
- [8] - BGPLVM for single cell data

Multi-output Gaussian processes

Pre-requisites: The Kronecker product

Take two matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & & \vdots \\ a_{m,1} & \dots & a_{m,n} \end{bmatrix} \qquad \mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{1,1}\mathbf{B} & \dots & a_{1,n}\mathbf{B} \\ \vdots & & \vdots \\ a_{m,1}\mathbf{B} & \dots & a_{m,n}\mathbf{B} \end{bmatrix}$$

Pre-requisites: The Kronecker product

Take two matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & & \vdots \\ a_{m,1} & \dots & a_{m,n} \end{bmatrix} \qquad \mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{1,1}\mathbf{B} & \dots & a_{1,n}\mathbf{B} \\ \vdots & & \vdots \\ a_{m,1}\mathbf{B} & \dots & a_{m,n}\mathbf{B} \end{bmatrix}$$

$\boxed{\text{So what is the dimension of } \mathbf{A} \otimes \mathbf{B}\text{?}}$

Pre-requisites: The Kronecker product

Take two matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & & \vdots \\ a_{m,1} & \dots & a_{m,n} \end{bmatrix} \qquad \mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{1,1}\mathbf{B} & \dots & a_{1,n}\mathbf{B} \\ \vdots & & \vdots \\ a_{m,1}\mathbf{B} & \dots & a_{m,n}\mathbf{B} \end{bmatrix}$$

$\boxed{\text{So what is the dimension of } \mathbf{A} \otimes \mathbf{B}\text{?}}$ $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{mp \times nq}$

Pre-requisites: The Kronecker product

Take two matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & & \vdots \\ a_{m,1} & \dots & a_{m,n} \end{bmatrix} \qquad \mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{1,1}\mathbf{B} & \dots & a_{1,n}\mathbf{B} \\ \vdots & & \vdots \\ a_{m,1}\mathbf{B} & \dots & a_{m,n}\mathbf{B} \end{bmatrix}$$

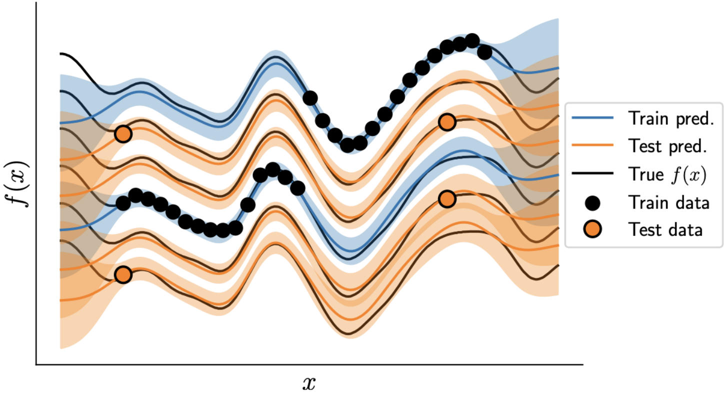
$\boxed{\text{So what is the dimension of } \mathbf{A} \otimes \mathbf{B}?}$ $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{mp \times nq}$

The inversion rule:

$$(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$$

And is invertible *if and only if* both $\mathbf{A}$ and $\mathbf{B}$ are invertible.

The motivation



Saemundson et al, Meta Reinforcement Learning with Latent Variable Gaussian Processes

Motivation: Multiple processes

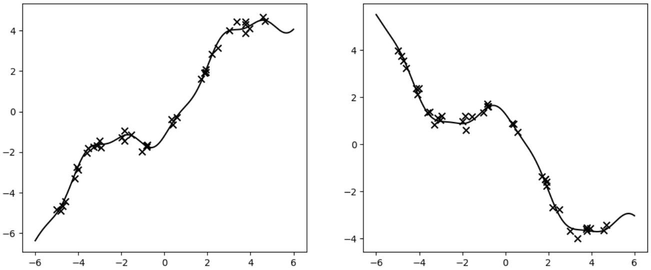


Figure: Two linearly correlated processes

Motivation: Multiple processes

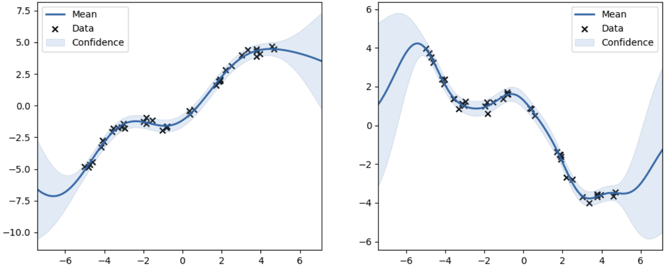


Figure: Two independent Gaussian process fits

$$f_1(\boldsymbol{x}) \sim \mathcal{GP}(0, k_1(\boldsymbol{x}, \boldsymbol{x}'))$$

$$\mathbf{f}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_1)$$

$$f_2(\boldsymbol{x}) \sim \mathcal{GP}(0, k_2(\boldsymbol{x}, \boldsymbol{x}'))$$

$$\mathbf{f}_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_2)$$

Motivation: Multiple processes

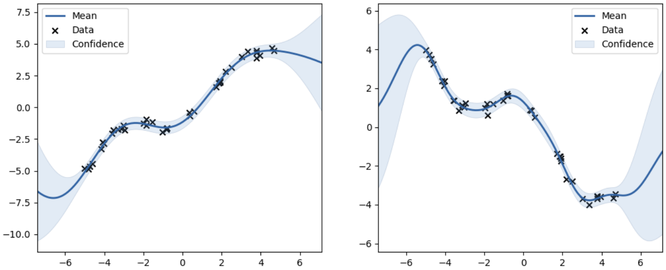


Figure: Two independent Gaussian process fits

$$\begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix}\right)$$

Motivation: Multiple processes

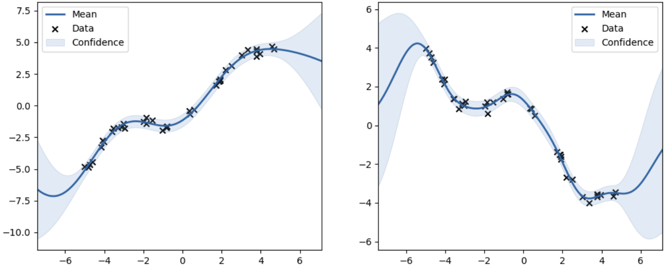


Figure: Two independent Gaussian process fits

$$\begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_1 & ? \\ ? & K_2 \end{bmatrix} \right)$$

Intrinsic Model of Coregionalisation (IMC)

General case

Sample S functions i.i.d. from the shared underlying process $\mathbf{u}^{(s)} \sim \mathcal{GP}(\mathbf{0}, k(\boldsymbol{x}, \boldsymbol{x}'))$.

Intrinsic Model of Coregionalisation (IMC)

General case

Sample S functions i.i.d. from the shared underlying process $\mathbf{u}^{(s)} \sim \mathcal{GP}(\mathbf{0}, k(\boldsymbol{x}, \boldsymbol{x}'))$.

The vector valued process is then defined as a weighted sum,

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{s=1}^{S} \boldsymbol{a}^{(s)} \mathbf{u}^{(s)}(\boldsymbol{x}).$$

where

$$\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_P(\boldsymbol{x})]^T, \quad \boldsymbol{a}^{(s)} = \left[\mathbf{a}_1^{(s)}, \mathbf{a}_2^{(s)}, \ldots, \mathbf{a}_P^{(s)}\right]^T,$$

Intrinsic Model of Coregionalisation (IMC)

General case

Sample S functions i.i.d. from the shared underlying process $\mathbf{u}^{(s)} \sim \mathcal{GP}(\mathbf{0}, k(\boldsymbol{x}, \boldsymbol{x}'))$.

The vector valued process is then defined as a weighted sum,

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{s=1}^{S} \boldsymbol{a}^{(s)} \mathbf{u}^{(s)}(\boldsymbol{x}).$$

where

$$\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_P(\boldsymbol{x})]^T, \quad \boldsymbol{a}^{(s)} = \left[\mathbf{a}_1^{(s)}, \mathbf{a}_2^{(s)}, \ldots, \mathbf{a}_P^{(s)}\right]^T,$$

$$\begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_1 & ? \\ ? & K_2 \end{bmatrix})$$

Intrinsic Model of Coregionalisation (IMC)

General case

$$\boxed{\mathsf{cov}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) = \mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}')^T\right] - \mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x})\right]\mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x}')\right]^T}$$

$$\begin{aligned}
\boldsymbol{f}(\boldsymbol{x}) &= \boldsymbol{a}^{(1)}\mathbf{u}^{(1)}(\boldsymbol{x}) + \boldsymbol{a}^{(2)}\mathbf{u}^{(2)}(\boldsymbol{x}) + \dots \\
\mathsf{cov}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) &= \boldsymbol{a}^{(1)}\boldsymbol{a}^{(1)^T}\mathsf{cov}(u^{(1)}(\boldsymbol{x}), u^{(1)}(\boldsymbol{x}')) + \boldsymbol{a}^{(2)}\boldsymbol{a}^{(2)^T}\mathsf{cov}(u^{(2)}(\boldsymbol{x}), u^{(2)}(\boldsymbol{x}')) + \dots \\
&= \boldsymbol{a}^{(1)}\boldsymbol{a}^{(1)^T}k(\boldsymbol{x}, \boldsymbol{x}') + \boldsymbol{a}^{(2)}\boldsymbol{a}^{(2)^T}k(\boldsymbol{x}, \boldsymbol{x}') + \dots \\
&= \left[\boldsymbol{a}^{(1)}\boldsymbol{a}^{(1)^T} + \boldsymbol{a}^{(2)}\boldsymbol{a}^{(2)^T} + \dots\right]k(\boldsymbol{x}, \boldsymbol{x}')
\end{aligned}$$

Intrinsic Model of Coregionalisation (IMC)

General case

$$\boxed{\mathsf{cov}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) = \mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}')^T\right] - \mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x})\right]\mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x}')\right]^T}$$

$$\begin{aligned}
\boldsymbol{f}(\boldsymbol{x}) &= \boldsymbol{a}^{(1)}\mathbf{u}^{(1)}(\boldsymbol{x}) + \boldsymbol{a}^{(2)}\mathbf{u}^{(2)}(\boldsymbol{x}) + \dots \\
\mathsf{cov}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) &= \boldsymbol{a}^{(1)}\boldsymbol{a}^{(1)^T}\mathsf{cov}(u^{(1)}(\boldsymbol{x}), u^{(1)}(\boldsymbol{x}')) + \boldsymbol{a}^{(2)}\boldsymbol{a}^{(2)^T}\mathsf{cov}(u^{(2)}(\boldsymbol{x}), u^{(2)}(\boldsymbol{x}')) + \dots \\
&= \boldsymbol{a}^{(1)}\boldsymbol{a}^{(1)^T}k(\boldsymbol{x}, \boldsymbol{x}') + \boldsymbol{a}^{(2)}\boldsymbol{a}^{(2)^T}k(\boldsymbol{x}, \boldsymbol{x}') + \dots \\
&= \underbrace{\left[\boldsymbol{a}^{(1)}\boldsymbol{a}^{(1)^T} + \boldsymbol{a}^{(2)}\boldsymbol{a}^{(2)^T} + \dots\right]}_{\hat{\boldsymbol{B}} \in \mathbb{R}^{P \times P}} k(\boldsymbol{x}, \boldsymbol{x}')
\end{aligned}$$

Intrinsic Model of Coregionalisation (IMC)

General case

$$\boxed{\mathrm{cov}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) = \mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}')^T\right] - \mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x})\right]\mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x}')\right]^T}$$

$$\begin{aligned}
\boldsymbol{f}(\boldsymbol{x}) &= \boldsymbol{a}^{(1)}\mathbf{u}^{(1)}(\boldsymbol{x}) + \boldsymbol{a}^{(2)}\mathbf{u}^{(2)}(\boldsymbol{x}) + \dots \\
\mathrm{cov}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) &= \boldsymbol{a}^{(1)}\boldsymbol{a}^{(1)^T}\mathrm{cov}(u^{(1)}(\boldsymbol{x}), u^{(1)}(\boldsymbol{x}')) + \boldsymbol{a}^{(2)}\boldsymbol{a}^{(2)^T}\mathrm{cov}(u^{(2)}(\boldsymbol{x}), u^{(2)}(\boldsymbol{x}')) + \dots \\
&= \boldsymbol{a}^{(1)}\boldsymbol{a}^{(1)^T}k(\boldsymbol{x}, \boldsymbol{x}') + \boldsymbol{a}^{(2)}\boldsymbol{a}^{(2)^T}k(\boldsymbol{x}, \boldsymbol{x}') + \dots \\
&= \underbrace{\left[\boldsymbol{a}^{(1)}\boldsymbol{a}^{(1)^T} + \boldsymbol{a}^{(2)}\boldsymbol{a}^{(2)^T} + \dots\right]}_{\hat{\boldsymbol{B}} \in \mathbb{R}^{P \times P}} k(\boldsymbol{x}, \boldsymbol{x}')
\end{aligned}$$

$$\mathrm{cov}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) = \hat{\boldsymbol{B}}k(\boldsymbol{x}, \boldsymbol{x}')$$

Intrinsic Model of Coregionalisation (IMC)

General case

$$\hat{\boldsymbol{B}} = \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix}, \qquad \text{if } P = 2$$

$$\begin{aligned} \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} &\sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} b_{1,1}\mathbf{K} & b_{1,2}\mathbf{K} \\ b_{2,1}\mathbf{K} & b_{2,2}\mathbf{K} \end{bmatrix} \right) \\ &\sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \hat{\boldsymbol{B}} \otimes \mathbf{K} \right) \end{aligned}$$

Semiparametric latent factor model (SLFM)

General case

Sample Q functions from separate processes $\mathbf{u}_q \sim \mathcal{GP}(\mathbf{0}, k_q(\boldsymbol{x}, \boldsymbol{x}'))$.

Semiparametric latent factor model (SLFM)

General case

Sample Q functions from separate processes $\mathbf{u}_q \sim \mathcal{GP}(\mathbf{0}, k_q(\boldsymbol{x}, \boldsymbol{x}'))$.

The vector valued process is then defined as a weighted sum,

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{q=1}^{Q} \boldsymbol{a}_q \mathbf{u}_q(\boldsymbol{x}).$$

where

$$\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_P(\boldsymbol{x})]^T, \quad \boldsymbol{a}_q = [\mathbf{a}_{q,1}, \mathbf{a}_{q,2}, \ldots, \mathbf{a}_{q,P}]^T,$$

Semiparametric latent factor model (SLFM)

General case

Sample Q functions from separate processes $\mathbf{u}_q \sim \mathcal{GP}(\mathbf{0}, k_q(\boldsymbol{x}, \boldsymbol{x}'))$.

The vector valued process is then defined as a weighted sum,

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{q=1}^{Q} \boldsymbol{a}_q \mathbf{u}_q(\boldsymbol{x}).$$

where

$$\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_P(\boldsymbol{x})]^T, \quad \boldsymbol{a}_q = [\mathbf{a}_{q,1}, \mathbf{a}_{q,2}, \ldots, \mathbf{a}_{q,P}]^T,$$

$$\begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_1 & ? \\ ? & K_2 \end{bmatrix})$$

Semiparametric latent factor model (SLFM)

General case

$$\boxed{\mathrm{cov}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) = \mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}')^T\right] - \mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x})\right]\mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x}')\right]^T}$$

$$\begin{aligned}
\boldsymbol{f}(\boldsymbol{x}) &= \boldsymbol{a}_1\mathbf{u}_1(\boldsymbol{x}) + \boldsymbol{a}_2\mathbf{u}_2(\boldsymbol{x}) + \dots \\
\mathrm{cov}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) &= \boldsymbol{a}_1\boldsymbol{a}_1^T \mathrm{cov}(u_1(\boldsymbol{x}), u_1(\boldsymbol{x}')) + \boldsymbol{a}_2\boldsymbol{a}_2^T \mathrm{cov}(u_2(\boldsymbol{x}), u_2(\boldsymbol{x}')) + \dots \\
&= \underbrace{\boldsymbol{a}_1\boldsymbol{a}_1^T}_{\tilde{\boldsymbol{B}}_1 \in \mathbb{R}^{P\times P}} k_1(\boldsymbol{x}, \boldsymbol{x}') + \underbrace{\boldsymbol{a}_2\boldsymbol{a}_2^T}_{\tilde{\boldsymbol{B}}_2 \in \mathbb{R}^{P\times P}} k_2(\boldsymbol{x}, \boldsymbol{x}') + \dots
\end{aligned}$$

Semiparametric latent factor model (SLFM)

General case

$$\boxed{\mathsf{cov}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) = \mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}')^T\right] - \mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x})\right]\mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x}')\right]^T}$$

$$\begin{aligned}
\boldsymbol{f}(\boldsymbol{x}) &= \boldsymbol{a}_1\mathbf{u}_1(\boldsymbol{x}) + \boldsymbol{a}_2\mathbf{u}_2(\boldsymbol{x}) + \dots \\
\mathsf{cov}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) &= \boldsymbol{a}_1\boldsymbol{a}_1^T\mathsf{cov}(u_1(\boldsymbol{x}), u_1(\boldsymbol{x}')) + \boldsymbol{a}_2\boldsymbol{a}_2^T\mathsf{cov}(u_2(\boldsymbol{x}), u_2(\boldsymbol{x}')) + \dots \\
&= \underbrace{\boldsymbol{a}_1\boldsymbol{a}_1^T}_{\tilde{\boldsymbol{B}}_1 \in \mathbb{R}^{P\times P}} k_1(\boldsymbol{x}, \boldsymbol{x}') + \underbrace{\boldsymbol{a}_2\boldsymbol{a}_2^T}_{\tilde{\boldsymbol{B}}_2 \in \mathbb{R}^{P\times P}} k_2(\boldsymbol{x}, \boldsymbol{x}') + \dots
\end{aligned}$$

$$\mathsf{cov}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) = \tilde{\boldsymbol{B}}_1 k_1(\boldsymbol{x}, \boldsymbol{x}') + \tilde{\boldsymbol{B}}_2 k_2(\boldsymbol{x}, \boldsymbol{x}') + \dots$$

$$\begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \sum_{q=1}^{Q} \tilde{\boldsymbol{B}}_q \otimes \mathbf{K}_q\right), \qquad \text{with } P = 2$$

Linear Model of Coregionalisation (LMC)

General case

Sample S_q functions from Q separate processes $\mathbf{u}_q^{(s)} \sim \mathcal{GP}(\mathbf{0}, k_q(\boldsymbol{x}, \boldsymbol{x}'))$.

Linear Model of Coregionalisation (LMC)

General case

Sample S_q functions from Q separate processes $\mathbf{u}_q^{(s)} \sim \mathcal{GP}(\mathbf{0}, k_q(\boldsymbol{x}, \boldsymbol{x}'))$.

The vector valued process is then defined as a weighted sum,

$$\mathbf{f}(\boldsymbol{x}) = \sum_{q=1}^{Q} \sum_{s=1}^{S} \boldsymbol{a}_q^{(s)} \mathbf{u}_q^{(s)}(\boldsymbol{x}).$$

where

$$\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_P(\boldsymbol{x})]^T, \quad \boldsymbol{a}_q^{(s)} = \left[a_{q,1}^{(s)}, a_{q,2}^{(s)}, \ldots, a_{q,P}^{(s)}\right]^T$$

Linear Model of Coregionalisation (LMC)

General case

$$\boxed{\text{cov}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) = \mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}')^T\right] - \mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x})\right]\mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x}')\right]^T}$$

$$\begin{aligned}\boldsymbol{f}(\boldsymbol{x}) = &\overbrace{\left[\boldsymbol{a}_1^{(1)}\mathbf{u}_1^{(1)}(\boldsymbol{x}) + \boldsymbol{a}_1^{(2)}\mathbf{u}_1^{(2)}(\boldsymbol{x}) + \dots\right]}^{\text{Intrinsic Model of Coregionalization}} \\ &+ \left[\boldsymbol{a}_2^{(1)}\mathbf{u}_2^{(1)}(\boldsymbol{x}) + \boldsymbol{a}_2^{(2)}\mathbf{u}_2^{(2)}(\boldsymbol{x}) + \dots\right] + \dots\end{aligned}$$

$$\begin{aligned}\text{cov}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) = &\overbrace{\left[\boldsymbol{a}_1^{(1)}\boldsymbol{a}_1^{(1)^T} + \boldsymbol{a}_1^{(2)}\boldsymbol{a}_1^{(2)^T}\right]}^{\boldsymbol{B}_1 \in \mathbb{R}^{P\times P}} k_1(\boldsymbol{x}, \boldsymbol{x}') \\ &+ \underbrace{\left[\boldsymbol{a}_2^{(1)}\boldsymbol{a}_2^{(1)^T} + \boldsymbol{a}_2^{(2)}\boldsymbol{a}_2^{(2)^T}\right]}_{\boldsymbol{B}_2 \in \mathbb{R}^{P\times P}} k_2(\boldsymbol{x}, \boldsymbol{x}') + \dots\end{aligned}$$

Linear Model of Coregionalisation (LMC)

General case

$$\boxed{\text{cov}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) = \mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}')^T\right] - \mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x})\right]\mathbb{E}\left[\boldsymbol{f}(\boldsymbol{x}')\right]^T}$$

$$\begin{aligned}\text{cov}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) = & \overbrace{\left[\boldsymbol{a}_1^{(1)}\boldsymbol{a}_1^{(1)^T} + \boldsymbol{a}_1^{(2)}\boldsymbol{a}_1^{(2)^T}\right]}^{\boldsymbol{B}_1 \in \mathbb{R}^{P \times P}} k_1(\boldsymbol{x}, \boldsymbol{x}') \\ & + \underbrace{\left[\boldsymbol{a}_2^{(1)}\boldsymbol{a}_2^{(1)^T} + \boldsymbol{a}_2^{(2)}\boldsymbol{a}_2^{(2)^T}\right]}_{\boldsymbol{B}_2 \in \mathbb{R}^{P \times P}} k_2(\boldsymbol{x}, \boldsymbol{x}') + \dots\end{aligned}$$

$$\text{cov}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x}')) = \boldsymbol{B}_1 k_1(\boldsymbol{x}, \boldsymbol{x}') + \boldsymbol{B}_2 k_2(\boldsymbol{x}, \boldsymbol{x}') + \dots$$

$$\begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \sum_{q=1}^{Q} \boldsymbol{B}_q \otimes \mathbf{K}_q\right), \qquad \text{with } P = 2$$

Linear Model of Coregionalisation (LMC)

Example

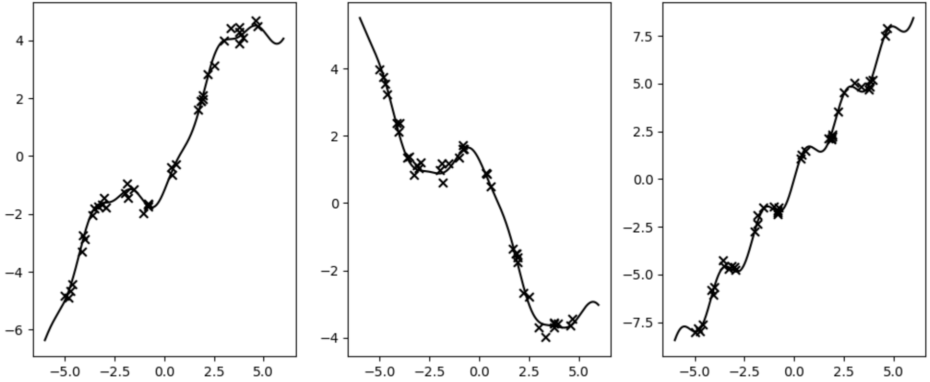


Figure: A third process

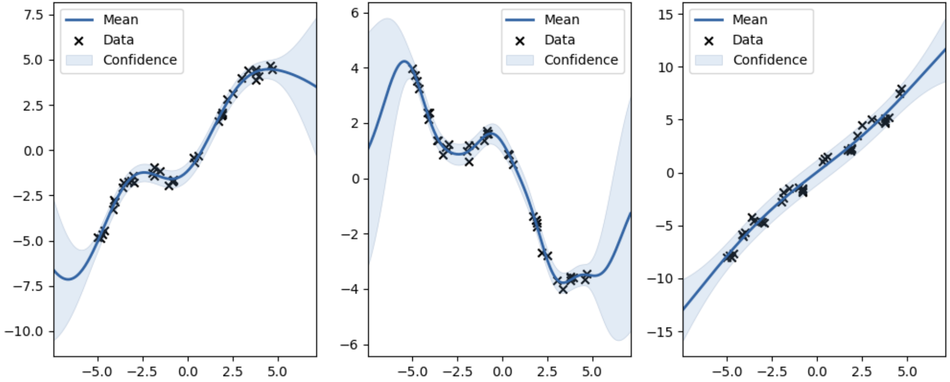


Figure: Independent Gaussian process fits

Linear Model of Coregionalisation (LMC)

Example

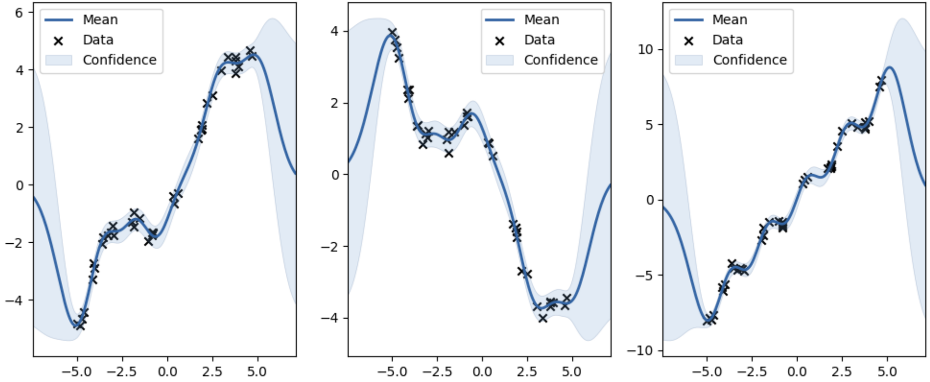
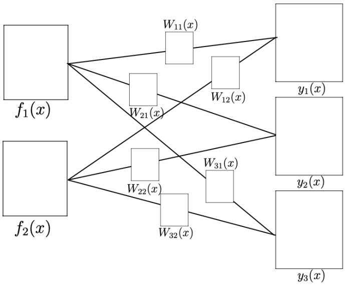


Figure: Linear Model of Coregionalisation fit

Gaussian process regression network (GPRN)

- Assume vector-valued output $\mathbf{y}(\mathbf{x}) \in \mathbb{R}^P$
- The GPRN [Wilson et al, ICML 2012]

$$
\begin{aligned}
\mathbf{y}(\mathbf{x}) &= W(\mathbf{x})[\mathbf{f}(\mathbf{x}) + \epsilon] + \varepsilon \\
f_i(\mathbf{x}) &\sim \mathcal{GP}(0, k_f), \qquad i = 1, \ldots, Q \\
W_{ij}(\mathbf{x}) &\sim \mathcal{GP}(0, k_w), \qquad W(\mathbf{x}) \in \mathbb{R}^{P \times Q}
\end{aligned}
$$



- We learn Q latent GPs that are mixed by PQ mixing GPs
- Variational inference by learning inducing points for both functions, $\mathbf{u}_{ij}^w$ and $\mathbf{u}_i^f$ (or by MCMC)
- Global input space $\mathbf{x} \in \mathbb{R}^D$ for all functions

Spatial interpolation

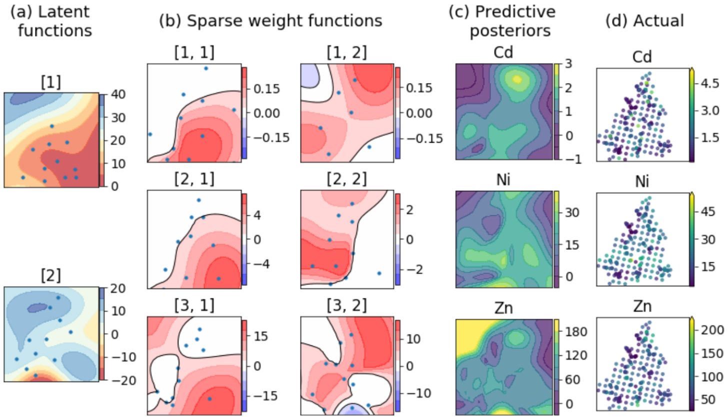


Figure includes a sparse extension, Hegde et al: Variational zero-inflated Gaussian processes with sparse kernels, UAI'2018

Bibliography I

Michalis Titsias.
Variational learning of inducing variables in sparse gaussian processes.
In Artificial Intelligence and Statistics, pages 567–574, 2009.

Andreas Damianou.
Deep Gaussian processes and variational propagation of uncertainty.
PhD thesis, University of Sheffield, 2015.

Michalis Titsias and Neil D Lawrence.
Bayesian gaussian process latent variable model.
In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pages 844–851. JMLR Workshop and Conference Proceedings, 2010.

Neil Lawrence.
Probabilistic non-linear principal component analysis with gaussian process latent variable models.
Journal of machine learning research, 6(Nov):1783–1816, 2005.

Bibliography II

Neil D Lawrence and Joaquin Quiñonero-Candela.
Local distance preservation in the gp-lvm through back constraints.
In Proceedings of the 23rd international conference on Machine learning, pages 513–520, 2006.

Carl Henrik Ek and PHTND Lawrence.
Shared Gaussian process latent variable models.
PhD thesis, Citeseer, 2009.

Katsu Yamane, Yuka Ariki, and Jessica Hodgins.
Animating non-humanoid characters with human motion data.
In Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pages 169–178, 2010.

Sumon Ahmed, Magnus Rattray, and Alexis Boukouvalas.
Grandprix: scaling up the bayesian gplvm for single-cell data.
Bioinformatics, 35(1):47–54, 2019.