Fully Bayesian Deep Gaussian Processes for Uncertainty Quantification

N. Zabaras[1] S. Atkinson[1]

Center for Informatics and Computational Science
Department of Aerospace and Mechanical Engineering
University of Notre Dame

SIAM CSE17, March 1, 2017

Motivation

- State-of-the-art simulation codes have enormous computational costs (millions of CPU-hrs/run).
- Need for computationally-efficient surrogate models that can approximately evaluate the system's response surface accurately and quickly.
- Stochastic collocation / gPC suffer from the curse of dimensionality.
- Limited training data? Gaussian processes are an attractive tool...

Challenges

- Capturing nonstationary behavior/discontinuities in the response surface
- Problems where the stochastic dimensionality is high and the underlying intrinsic dimensionality and its probability distribution are unknown.

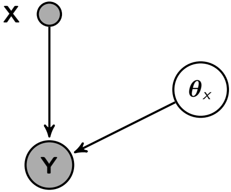
Problem Statement

- Consider some physical system with an input space with spatial, temporal, and stochastic components: $\mathcal{X} = \mathcal{X}_s \times \mathcal{X}_t \times \mathcal{X}_\Xi \subset \mathbb{R}^{d_t d_s d_\Xi}$
- We would like to construct a surrogate model to approximately infer the response surface $\hat{y}(\boldsymbol{x}_s, t, \boldsymbol{\Xi}) \in \mathbb{R}^{p_o}$ for the purpose of computing statistical quantities of interest $\mathcal{Q} = \int f(\hat{y})p(\boldsymbol{\Xi})d\boldsymbol{\Xi}$.
- We will use a supervised deep Gaussian process model as our surrogate model to approximate $\hat{y}$.
- We will also explore using unsupervised Gaussian process latent variable models to reduce the dimensionality of the stochastic input and learn a reduced (latent) stochastic space $\mathcal{X}_\xi$ that captures the underlying correlations in our data from $\mathcal{X}_\Xi$.

Outline

Definition of a GP

- A Gaussian Process is a distribution over functions that defines a mapping $\mathcal{GP} : \mathcal{X} \subset \mathbb{R}^{p_i} \mapsto \mathcal{Y} \subset \mathbb{R}^{p_o}$.
- For a finite number of input points $\{\boldsymbol{x}_{i,:} \in \mathcal{X}\}_{i=1}^{n}$, the marginal distribution of the corresponding outputs $\{\boldsymbol{y}_{i,:} \in \mathcal{Y}\}_{i=1}^{n}$ is a multivariate Gaussian distribution.



Bayesian inference with GPs

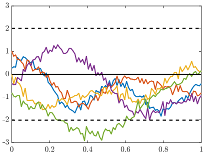
Prior:

$$\begin{aligned}
p(\boldsymbol{y}|\mathbf{X},\boldsymbol{\theta}_x) &= \mathcal{N}\left(\boldsymbol{y}|\mathbf{0},\mathbf{K}_{xx}\right) \\
\left(\mathbf{K}_{xx}\right)_{ij} &= k(\boldsymbol{x}_{i,:},\boldsymbol{x}_{j,:};\boldsymbol{\theta}_x) \\
k(\boldsymbol{x}_{i,:},\boldsymbol{x}_{j,:};\boldsymbol{\theta}_x) &= \sigma_{SE}^2 \exp\left(\sum_{k=1}^{p_i} w_k^2 (x_{ik}-x_{jk})^2\right)
\end{aligned}$$

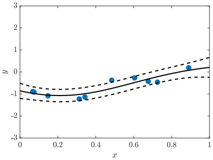
Given some observations $\mathbf{X}$ and $\boldsymbol{y}$, the posterior is

$$\begin{aligned}
p(\boldsymbol{y}^*|\mathbf{X}^*,\mathbf{X},\boldsymbol{y},\boldsymbol{\theta}_x) &= \mathcal{N}\left(\boldsymbol{y}^*|\mathbf{K}_{*x}\mathbf{K}_{xx}^{-1}\boldsymbol{y},\mathbf{K}_{**}-\mathbf{K}_{*x}\mathbf{K}_{xx}^{-1}\mathbf{K}_{x*}\right) \\
\left(\mathbf{K}_{*x}\right)_{ij} &= k(x_i^*,x_j;\boldsymbol{\theta}_x) \\
\left(\mathbf{K}_{**}\right)_{ij} &= k(x_i^*,x_j^*;\boldsymbol{\theta}_x) \\
\mathbf{K}_{x*} &= \mathbf{K}_{*x}^{\intercal}
\end{aligned}$$

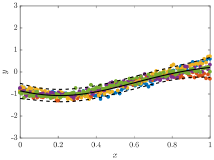
(See: C. M. Bishop, 2006)



(a) Prior samples



(b) Introduce observations



(c) Samples from posterior

GPs can be used as generative models.

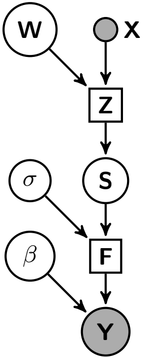
The standardized covariance function

- Squared exponential kernel function with all hyperparameters fixed to unity:

$$k_s(\mathbf{z}_{i,:}, \mathbf{z}_{j,:}) = \exp\left(-\frac{1}{2}\sum_{k=1}^{d_i}(z_{ik} - z_{jk})^2\right).$$

- Hyperparameters enter into model through deterministic transformations before and after the GP:

$$\begin{aligned}\mathbf{Z} &= \mathbf{HW},\\ \mathbf{F} &= \sigma\mathbf{S}.\end{aligned}$$



(See: M. K. Titsias and N. D. Lawrence, 2013)

Unsupervised Gaussian Process Latent Variable Model (GP-LVM)

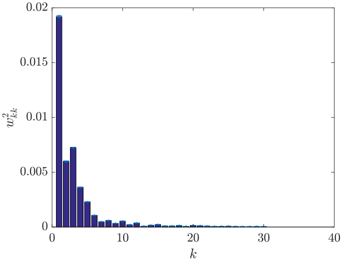
(M. K. Titsias and N. D. Lawrence, 2010)

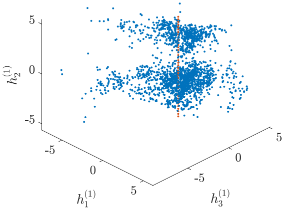
- Unlabeled data enters as model outputs.
- Inputs are not specified. Instead, they are latent variables and we must learn them.
  - Define a prior distribution over every training data.
  - Variational posterior distribution is a factorized Gaussian; variational parameters are subject to optimization.
  - Optimize the model log likelihood where the latent variables are marginalized out.
- Training yields a reduced-order model of the training data as well as nonlinear projection and uplifting mappings between the observed data space and the learned latent space.

Example: pictures of faces

- High-dimensional data
- Bayesian training automatically selects the number of latent dimensions (automatic relevance determination).

Data obtained from:
github.com/SheffieldML/vargplvm





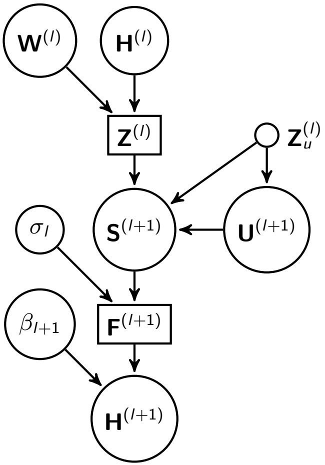
We may select points in latent space and uplift them to produce new data (YouTube)

Outline

- Create a model comprised of L layers.
- Each layer is a sparse Gaussian process.
- The output of layer l is the input of layer $l+1$. (Output of layer L is the observed data.)
- Allows for increased representative capacity in unsupervised models.
- Can be used for supervised problems to learn more complex mapping functions than what can be encoded in a single GP, including nonstationary behavior in the data being modeled.
- (See: A. C. Damianou and N. D. Lawrence, 2013)

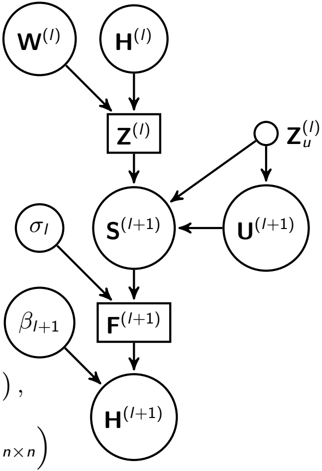
Outline

- Sparse GP with auxiliary (inducing) input-output pairs given by the matrices $\mathbf{Z}_u^{(l)}$ and $\mathbf{U}^{(l+1)}$.
- Hidden layers are connected through the latent variables only.
- All other model parameters decouple across layers.



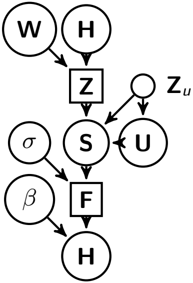
Priors:

$$
\begin{aligned}
p\left(\mathbf{H}^{(l)}\right) &= \prod_{i=1}^{n}\prod_{j=1}^{q_l}\mathcal{N}\left(h_{ij}^{(l)}|0,1\right),\\
p\left(\mathbf{W}^{(l)}\right) &= \prod_{i=1}^{q_l}\prod_{j=1}^{q_l}\mathcal{N}\left(w_{ij}^{(l)}|0,\delta_{ij}\right),\\
p(\mathbf{S}^{(l+1)}|\mathbf{Z}^{(l)}) &= \prod_{j=1}^{q_{l+1}}\mathcal{N}\left(\boldsymbol{s}_j^{(l+1)}|\mathbf{0},\mathbf{K}_{ss}^{(l)}\right),\\
p(\mathbf{U}^{(l+1)}|\mathbf{Z}_u^{(l)}) &= \prod_{j=1}^{q_{l+1}}\mathcal{N}\left(\boldsymbol{u}_j^{(l+1)}|\mathbf{0},\mathbf{K}_{uu}^{(l)}\right)\\
p\left(\sigma_l\right) &= \mathrm{Gamma}\left(\sigma_l|a_{\sigma_l},b_{\sigma_l}\right),\\
p\left(\beta_{l+1}\right) &= \mathrm{Gamma}\left(\beta_{l+1}|a_{\beta_{l+1}},b_{\beta_{l+1}}\right),\\
p(\mathbf{H}^{(l+1)}|\mathbf{F}^{(l+1)},\beta_{l+1}) &= \prod_{j=1}^{q_{l+1}}\mathcal{N}\left(\boldsymbol{h}_j^{(l+1)}|\boldsymbol{f}_j^{(l+1)},\beta_{l+1}^{-1}\mathbf{I}_{n\times n}\right)
\end{aligned}
$$



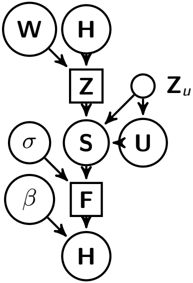
Conditional GP prior:

$$
\begin{aligned}
p(\mathbf{F}^{(l+1)}|\mathbf{U}^{(l+1)}, \mathbf{Z}_u^{(l)}, \sigma_l, \mathbf{Z}^{(l)}) &= \prod_{j=1}^{q_{l+1}} \mathcal{N}\left(\boldsymbol{f}_j^{(l+1)} | \sigma_l \boldsymbol{\eta}_j^{(l)}, \sigma_l^2 \tilde{\mathbf{K}}^{(l)}\right) \\
\boldsymbol{\eta}_j^{(l)} &= \mathbf{K}_{su}^{(l)} (\mathbf{K}_{uu}^{(l)})^{-1} \boldsymbol{u}_j^{(l+1)}, \\
\tilde{\mathbf{K}}^{(l)} &= \mathbf{K}_{ss}^{(l)} - \mathbf{K}_{su}^{(l)} (\mathbf{K}_{uu}^{(l)})^{-1} \mathbf{K}_{us}^{(l)}, \\
\left(\mathbf{K}_{su}^{(l)}\right)_{ij} &= k_s(\boldsymbol{z}_{i,:}^{(l)}, (\mathbf{Z}_u^{(l)})_{j,:}),
\end{aligned}
$$



Variational posteriors:

$$\tilde{p}\left(\mathbf{H}^{(l)}\right) = \prod_{i=1}^{n}\prod_{j=1}^{q_l}\mathcal{N}\left(h_{ij}^{(l)}|\left(\mu_h^{(l)}\right)_{ij},\left(s_h^{(l)}\right)_{ij}\right),$$
$$\tilde{p}\left(\mathbf{W}^{(l)}\right) = \prod_{i=1}^{q_l}\prod_{j=1}^{q_l}\mathcal{N}\left(w_{ij}^{(l)}|\delta_{ij}\left(\mu_w^{(l)}\right)_{ij},\delta_{ij}\left(s_w^{(l)}\right)_{ij}\right),$$
$$\tilde{p}\left(\sigma_l\right) = \mathrm{Gamma}\left(\sigma_l|a_{\sigma_l}^*,b_{\sigma_l}^*\right),$$
$$\tilde{p}\left(\beta_{l+1}\right) = \mathrm{Gamma}\left(\beta_{l+1}|a_{\beta_{l+1}}^*,b_{\beta_{l+1}}^*\right),$$



The variational parameters in these expressions will be subject to numerical optimization.

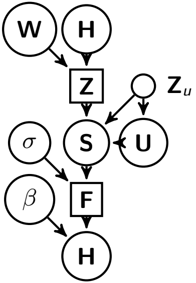
Variational posteriors (cont'd):

$$\tilde{p}\left(\mathbf{U}^{(l+1)}\right) = \prod_{j=1}^{q_{l+1}} \mathcal{N}\left(\boldsymbol{u}_j^{(l+1)} | \left(\boldsymbol{\mu}_{(u)}^{(l+1)}\right)_j, \boldsymbol{\Sigma}_u^{(l+1)}\right),$$

$$\left(\boldsymbol{\mu}_{(u)}^{(l+1)}\right)_j = \bar{\sigma}_l \mathbf{K}_{uu}^{(l)} \left(\mathbf{K}_{\psi}^{(l)}\right)^{-1} \phi_j^{(l)},$$

$$\boldsymbol{\Sigma}_u^{(l+1)} = \bar{\beta}_{l+1}^{-1} \mathbf{K}_{uu}^{(l)} \left(\mathbf{K}_{\psi}^{(l)}\right)^{-1} \mathbf{K}_{uu}^{(l)}.$$

This expression is optimal, given the other parameters.



Approximating the model likelihood

Start by writing the joint distribution for a single layer. For a single layer l:

$$\begin{aligned}\mathcal{P}_l &\equiv p(\mathbf{H}^{(l+1)}, \beta_{l+1}, \mathbf{F}^{(l+1)}, \mathbf{U}^{(l+1)}, \sigma_l, \mathbf{W}^{(l)} | \mathbf{Z}_u^{(l)}, \mathbf{H}^{(l)}) \\ &= p\left(\mathbf{H}^{(l+1)} | \beta_{l+1}, \mathbf{F}^{(l+1)}\right) p\left(\beta_{l+1}\right) p\left(\mathbf{F}^{(l+1)} | \mathbf{U}^{(l+1)}, \mathbf{Z}_u^{(l)}, \sigma_l, \mathbf{W}^{(l)}, \mathbf{H}^{(l)}\right) \\ &\quad p\left(\mathbf{U}^{(l+1)} | \mathbf{Z}_u^{(l)}\right) p\left(\sigma_l\right) p\left(\mathbf{W}^{(l)}\right).\end{aligned}$$

For the whole model:

$$\mathcal{P} = p\left(\mathbf{Y}, \left\{\mathbf{H}^{(l)}, \beta_{l+1}, \mathbf{F}^{(l+1)}, \sigma_l, \mathbf{U}^{(l+1)}, \mathbf{W}^{(l)}\right\}_{l=1}^{L} \middle| \left\{\mathbf{Z}_u^{(l)}\right\}_{l=1}^{L}\right) = p\left(\mathbf{H}^{(1)}\right) \prod_{l=1}^{L} \mathcal{P}_l,$$

where $\mathbf{H}^{(l+1)} \equiv \mathbf{Y}$, and $\mathbf{H}^{(1)} \equiv \mathbf{X}$ (for supervised applications).

The marginal log likelihood is

$$\log p(\mathbf{Y}| \left\{\mathbf{Z}_u^{(l)}\right\}_{l=1}^{L}) = \log \int \tilde{\mathcal{P}} \frac{\mathcal{P}}{\tilde{\mathcal{P}}} d\mathbf{\Omega}$$

where we introduce the shorthand

$$d\mathbf{\Omega} = d\left\{\mathbf{H}^{(l)}\right\}_{l=1}^{L} d\left\{\mathbf{U}^{(l+1)}\right\}_{l=1}^{L} d\left\{\mathbf{W}^{(l)}\right\}_{l=1}^{L} d\left\{\sigma_l\right\}_{l=1}^{L} d\left\{\beta_{l+1}\right\}_{l=1}^{L},$$

for brevity.
Applying Jensen's inequality, we can write the lower bound

$$\log p(\mathbf{Y}| \left\{\mathbf{Z}_u^{(l)}\right\}_{l=1}^{L}) \geq \mathfrak{F} = \int \tilde{\mathcal{P}} \log \frac{\mathcal{P}}{\tilde{\mathcal{P}}} d\mathbf{\Omega}.$$

After some manipulation and substituting in the optimal $\tilde{p}\left(\mathbf{U}^{(l+1)}\right)$, we get the following "partially-collapsed" lower bound:

$$
\begin{aligned}
\mathfrak{F} &= \mathfrak{F}\left(\left\{\mathbf{H}^{(l)}, \mathbf{W}^{(l)}, \sigma_l, \beta_{l+1}\right\}_{l=1}^{L}\right) \\
&= \sum_{l=1}^{L} \mathfrak{F}^{(l)}\left(\tilde{p}\left(\mathbf{H}^{(l+1)}\right), \tilde{p}\left(\mathbf{H}^{(l)}\right), \tilde{p}\left(\mathbf{W}^{(l)}\right), \tilde{p}\left(\sigma_l\right), \tilde{p}\left(\beta_{l+1}\right)\right) \\
&+ \sum_{l=2}^{L} \mathcal{H}\left[\tilde{p}\left(\mathbf{H}^{(l)}\right)\right] - \sum_{\alpha=\left\{\mathbf{H}^{(1)}, \left\{\sigma_l, \beta_{l+1}, \mathbf{W}^{(l)}\right\}_{l=1}^{L}\right\}} \mathrm{KL}\left(\tilde{p}(\alpha) \,\|\, p(\alpha)\right)
\end{aligned}
$$

where the first summand is given by

$$
\begin{aligned}
\mathfrak{F}^{(l)} &= \frac{q_{l+1}}{2}\left(n\left(\mathbb{E}\left[\log \beta_{l+1}\right] - \log(2\pi)\right) + \log\left|\mathbf{K}_{uu}^{(l)}\right| - \log\left|\bar{\beta}_{l+1}\mathbf{K}_{\psi}^{(l)}\right|\right) \\
&- \frac{\bar{\beta}_{l+1}}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{q_{l+1}}\left(\mu_h^{(l+1)}\right)_{ij}^{2} + \left(s_h^{(l)}\right)_{ij} - \bar{\sigma}_l^2 \mathrm{Tr}\left(\left(\mathbf{\Phi}^{(l)}\right)^{\mathsf{T}}\left(\mathbf{K}_{\psi}^{(l)}\right)^{-1}\mathbf{\Phi}^{(l)}\right)\right. \\
&\left. + q_{l+1}\mathbb{E}\left[\sigma_l^2\right]\left(\psi_0^{(l)} - \mathrm{Tr}\left(\left(\mathbf{K}_{uu}^{(l)}\right)^{-1}\mathbf{\Psi}_2^{(l)}\right)\right)\right).
\end{aligned}
$$

Model statistics:

$$
\begin{aligned}
\mathbf{\Psi}_2^{(l)} &= \sum_{i=1}^{n} \left(\hat{\mathbf{\Psi}}_2^{(l)} \right)_i \in \mathbb{R}^{m_l \times m_l} \\
\mathbf{\Phi}^{(l)} &= \sum_{i=1}^{n} \hat{\mathbf{\Phi}}_i^{(l)} \in \mathbb{R}^{m_l \times q_{l+1}}
\end{aligned}
$$

Can be distributed w.r.t. data points.

To recap:

- We have an analytical “semi-collapsed” lower bound to the model’s marginal log likelihood in terms of variational parameters.
- All model hyperparameters are given variational posterior distributions instead of being treated as point estimates which can be used to characterize our model uncertainty.

Outline

Obtaining an Initial Guess

- Rescale data to have zero mean and unit standard deviation.
- For each layer, perform PCA on the layer output means $\boldsymbol{\mu}_h^{(l+1)}$ to obtain the layer's input latent variables.
- One may optionally perform a "mini optimization" on each GP-LVM layer before proceeding to the layer above it.

Optimization

- The model's variational parameters are optimized by conjugate gradient.
- Computation of the gradient is parallelized over data points.
- Optimization terminates when either the objective function (negative log likelihood) stops decreasing, the optimization state stops evolving, or a predetermined number of iterations is exceeded.

Predictions with the model

The predictive posterior distribution for each individual layer, given some input, is derived using the inducing input-output pairs instead of the training data and is Gaussian.

$$
\begin{aligned}
p\left(\mathbf{F}^{(l+1)*} | \mathbf{H}^{(l)*}, \mathbf{Z}_u^{(l)}, \mathbf{U}^{(l+1)}, \mathbf{W}^{(l)}, \sigma_l\right) &= \prod_{j=1}^{q_{l+1}} p\left(\boldsymbol{f}_j^{(l+1)*} | \mathbf{H}^{(l)*}, \mathbf{Z}_u^{(l)}, \boldsymbol{u}_j^{(l+1)}, \mathbf{W}^{(l)}, \sigma_l\right) \\
&= \prod_{j=1}^{q_{l+1}} \mathcal{N}\left(\boldsymbol{f}_j^{(l+1)*} | \left(\boldsymbol{\mu}_f^{(l)*}\right)_j, \boldsymbol{\Sigma}_f^{(l)*}\right),
\end{aligned}
$$

where

$$
\begin{aligned}
\left(\boldsymbol{\mu}_f^{(l)*}\right)_j &= \sigma_l \mathbf{K}_{*u}^{(l)} \left(\mathbf{K}_{uu}^{(l)}\right)^{-1} \boldsymbol{u}_j^{(l+1)}, \\
\boldsymbol{\Sigma}_f^{(l)*} &= \sigma_l^2 \left(\mathbf{K}_{**}^{(l)} - \mathbf{K}_{*u}^{(l)} \left(\mathbf{K}_{uu}^{(l)}\right)^{-1} \mathbf{K}_{u*}^{(l)}\right), \\
\left(\mathbf{K}_{**}^{(l)}\right) &= k_s\left(\boldsymbol{h}_{i,:}^{(l)*} \mathbf{W}^{(l)}, \boldsymbol{h}_{j,:}^{(l)*} \mathbf{W}^{(l)}\right), \\
\left(\mathbf{K}_{*u}^{(l)}\right) &= k_s\left(\boldsymbol{h}_{i,:}^{(l)*} \mathbf{W}^{(l)}, \left(\mathbf{Z}_u^{(l)}\right)_{j,:}\right).
\end{aligned}
$$

We can marginalize out the inducing outputs $\mathbf{U}^{(l+1)}$:

$$p\left(\boldsymbol{f}_j^{(l+1)*}|\mathbf{H}^{(l)*}, \mathbf{Z}_u^{(l)}, \mathbf{W}^{(l),\sigma_l}\right) \quad = \quad \mathcal{N}\left(\boldsymbol{f}_j^{(l+1)*}|\left(\boldsymbol{\mu}_f^{(l)*}\right)_j, \boldsymbol{\Sigma}_f^{(l)*}\right),$$

where

$$\begin{aligned}
\left(\boldsymbol{\mu}_f^{(l)*}\right)_j &= \sigma_l \bar{\sigma}_l \mathbf{K}_{*u}^{(l)} \left(\mathbf{K}_\psi^{(l)}\right)^{-1} \phi_j^{(l)}, \\
\boldsymbol{\Sigma}_f^{(l)*} &= \sigma_l^2 \left(\mathbf{K}_{**}^{(l)} - \mathbf{K}_{*u}^{(l)} \left(\left(\mathbf{K}_{uu}^{(l)}\right)^{-1} - \left(\bar{\beta}_{l+1}\mathbf{K}_\psi^{(l)}\right)^{-1}\right)\right) \mathbf{K}_{u*}^{(l)}.
\end{aligned}$$

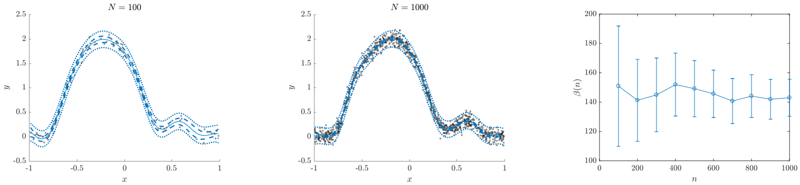
Furthermore, we have

$$p\left(\boldsymbol{h}_j^{(l+1)}|\boldsymbol{f}_j^{(l+1)}, \beta_{l+1}\right) = \mathcal{N}\left(\boldsymbol{h}_j^{(l+1)}|\boldsymbol{f}_j^{(l+1)}, \beta_{l+1}^{-1}\mathbf{I}_{n^*\times n^*}\right).$$

It is straightforward to marginalize out the test points' latent outputs $\boldsymbol{f}_j^{(l+1)}$ to obtain:

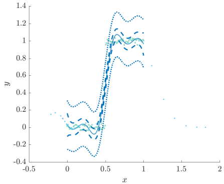
$$p\left(\boldsymbol{h}_j^{(l+1)}|\mathbf{H}^{(l)*}, \mathbf{Z}_u^{(l)}, \mathbf{W}^{(l)}, \sigma_l\right) = \mathcal{N}\left(\boldsymbol{f}_j^{(l+1)*}|\left(\boldsymbol{\mu}_f^{(l)*}\right)_j, \boldsymbol{\Sigma}_f^{(l)*} + \beta_{l+1}^{-1}\mathbf{I}_{n^*\times n^*}\right),$$

1D Regression

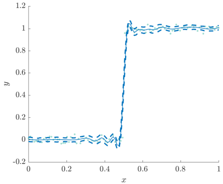


Bayesian training provides confidence intervals on model hyperparameters (quantifies epistemic uncertainty).

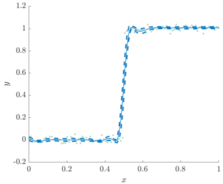
Step Function



(o) $L = 0$



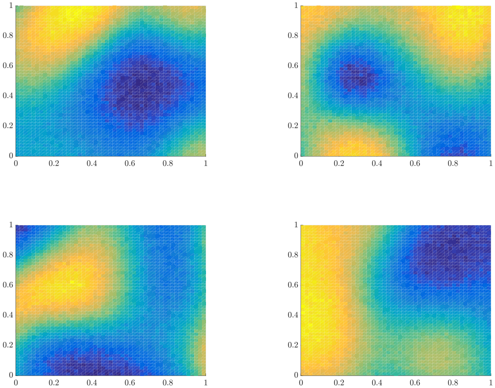
(p) $L = 1$



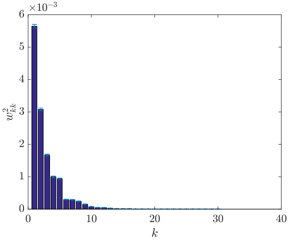
(q) $L = 2$

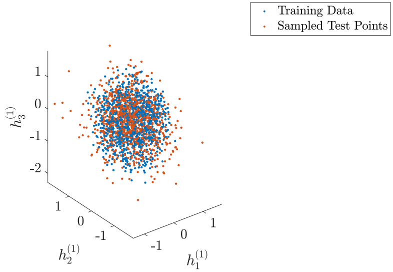
Introducing hidden layers increases model capacity and allows us to better model the nonstationary behavior.

Example: random fields

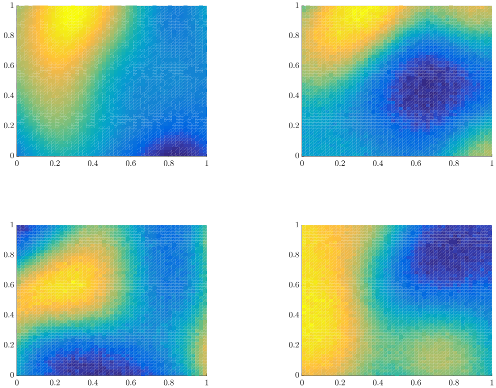


Example: random fields





Example: random fields



KO-2

The Kraichnan-Orszag three-mode problem is a dynamical problem described by:

$$
\begin{aligned}
\frac{dy_1}{dt} &= y_1 y_3, \\
\frac{dy_2}{dt} &= -y_2 y_3, \\
\frac{dy_3}{dt} &= y_2^2 - y_1^2.
\end{aligned}
$$

We consider the initial conditions

$$
\begin{aligned}
y_1(0) &= 1, \\
y_2(0) &= 0.1\xi_1, \\
y_3(0) &= \xi_2.
\end{aligned}
$$

We assume the stochastic variables are uniformly distributed in $[-1, 1]^2$. The response surface has a bifurcation in y_2 at $\xi_1 = 0$.

Predictions Near the Bifurcation

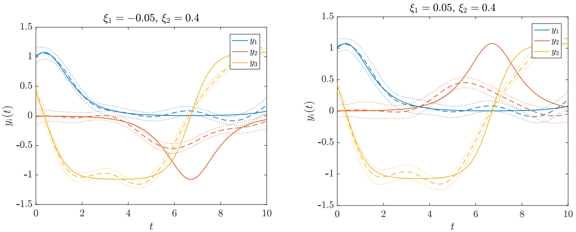


Figure: $L = 0$ (Sparse GP)

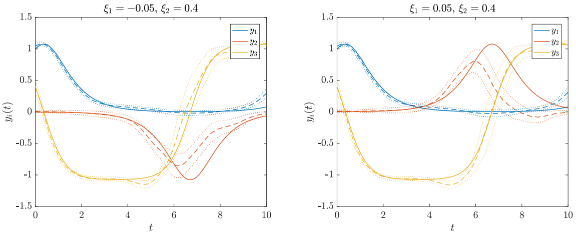
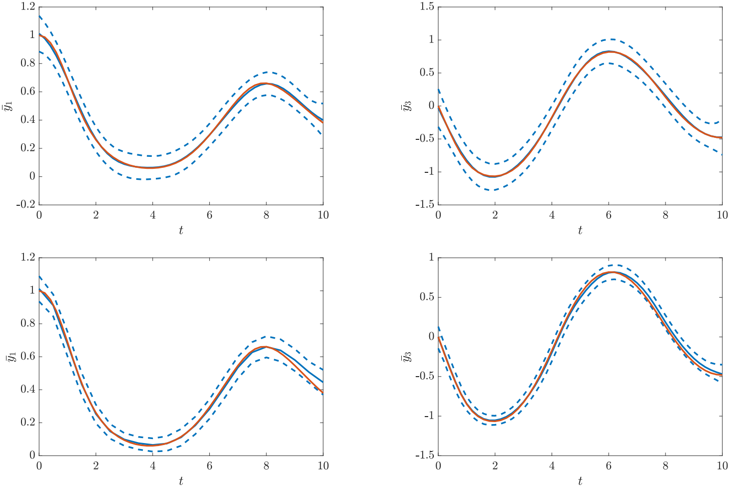
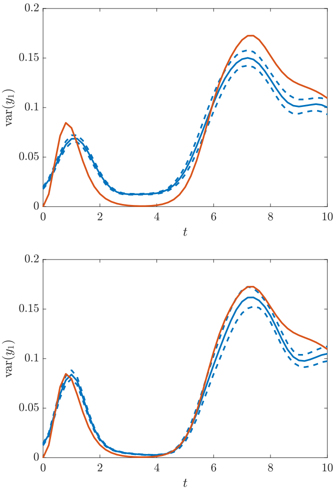


Figure: $L = 2$


1.2
1
0.8
0.6
0.4
0.2
0
-0.2
$\bar{y}_1$
0 2 4 6 8 10
t
1.5
1
0.5
0
-0.5
-1
-1.5
$\bar{y}_3$
0 2 4 6 8 10
t
1.2
1
0.8
0.6
0.4
0.2
0
$\bar{y}_1$
0 2 4 6 8 10
t
1
0.5
0
-0.5
-1
-1.5
$\bar{y}_3$
0 2 4 6 8 10
t



0.2
0.15
0.1
0.05
0
var(y_1)
0 2 4 6 8 10
t
0.8
0.6
0.4
0.2
0
var(y_3)
0 2 4 6 8 10
t
0.2
0.15
0.1
0.05
0
var(y_1)
0 2 4 6 8 10
t
0.8
0.6
0.4
0.2
0
var(y_3)
0 2 4 6 8 10
t

Summary

- Bayesian deep Gaussian processes are promising tools for performing dimensionality reduction of high-dimensional data and learning response surfaces with discontinuities and other nonstationary features.
- The fully Bayesian training paradigm allows for us to quantify our model uncertainty and provide confidence intervals for its predictions.

Immediate Tasks

- Working on combined approach in which an unsupervised model is used to perform dimensionality reduction on high-dimensional input and a supervised model learns the response surface from from the reduced stochastic space.
- Further investigations into model refinement including the role of the number of inducing points and augmenting the training data with a dense set of design points.

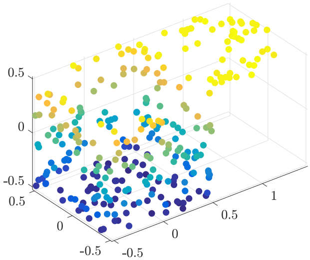
Further Goals

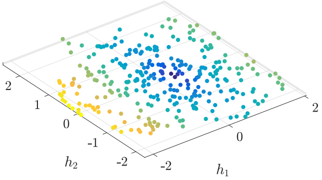
- Exploiting the hierarchical structure of DGPs in applications to multiscale problems.
- Speeding up projections from (output) data space to latent space using a Bayesian extension of the back constraints framework of [cite]

Thank you

This work was supported from the Computer Science and Mathematics Division of ORNL under the DARPA EQUiPS program. N.Z. thanks the Technische Universität München, Institute for Advanced Study for support through a Hans Fisher Senior Fellowship, funded by the German Excellence Initiative and the European Union Seventh Framework Programme under Grant Agreement No. 291763.

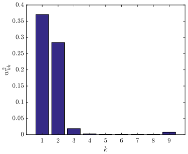
Example: “open box” manifold

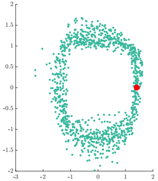


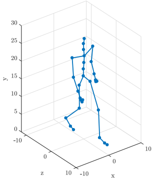


Example: motion capture video sequences

- 1023 frames, 59 dimensions
- Trained using a GP-LVM with 9-dimensional latent space; ARD identifies two principal dimensions.
- Uplifting points from the torus generates a new walk.

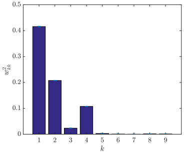






YouTube

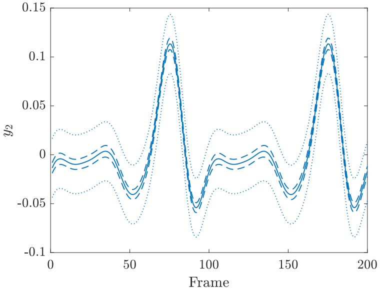
The data used in this project was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217.



- The model is capable of finding structure in limited data sets and segmenting latent space

Figures/Mocap/1Walk1Run/YT-eps-conv

Motion capture (cont'd)



- Model provides confidence intervals for the latent outputs as well as the noise of the observed data.
- Bayesian training provides uncertainty estimates for the model hyperparameters ("statistics of statistics").

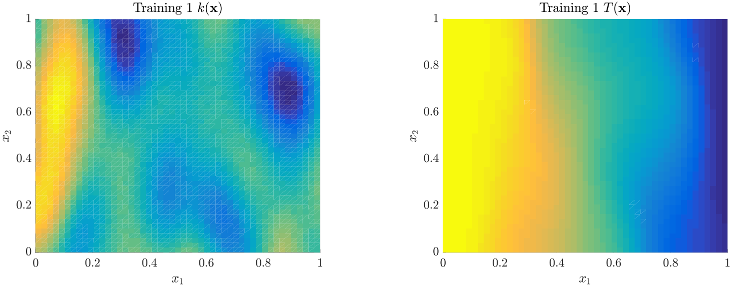
Example: Elliptic PDE

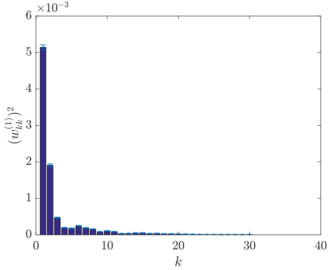
We consider an Elliptic PDE given by:

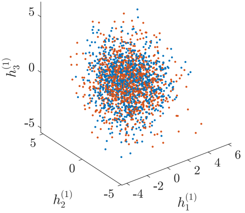
$$
\begin{aligned}
\nabla \cdot (k(\boldsymbol{x}) \nabla T(\boldsymbol{x})) &= 0, \\
\boldsymbol{x} \in \mathcal{X}_s &= [0,1]^2 \subset \mathbb{R}^2, \\
T(x_1 = 0, 1) &= 1 - x_1, \\
\left.\frac{dT}{dx_2}\right|_{x_2=0,1} &= 0, \\
\log k(\boldsymbol{x}) &\sim \mathcal{GP}\left(\mathbf{0}, c(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta})\right).
\end{aligned}
$$

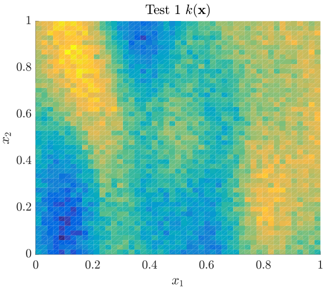
Example: Elliptic PDE

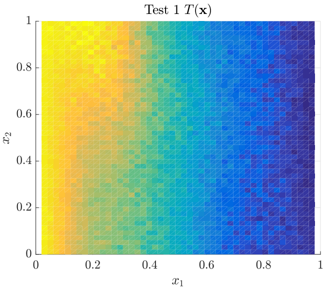
- Concatenate random property field measurements $\log k(\boldsymbol{x})$ and $T(\boldsymbol{x})$ for a given simulation run into a single datum. (Number of training data = number of simulations)
- Use unsupervised model to learn a common latent space for the "input" and "output" dimensions in the data.


Training 1 $k(\mathbf{x})$
Training 1 $T(\mathbf{x})$
x_1
x_2
0
0.2
0.4
0.6
0.8
1


$\times 10^{-3}$
6
5
4
3
2
1
0
$(w_{kk}^{(1)})^2$
0
10
20
30
40
k


5
0
-5
$h_3^{(1)}$
5
0
-5
$h_2^{(1)}$
-4
-2
0
2
4
6
$h_1^{(1)}$


Test 1 $k(\mathbf{x})$
x_2
x_1
0
0.2
0.4
0.6
0.8
1


Test 1 $T(\mathbf{x})$
x_2
x_1
0
0.2
0.4
0.6
0.8
1