Latent variable and deep modeling with Gaussian processes; application to system identification

Andreas Damianou

Department of Computer Science, University of Sheffield, UK

Brown University, 17 Feb. 2016

Outline

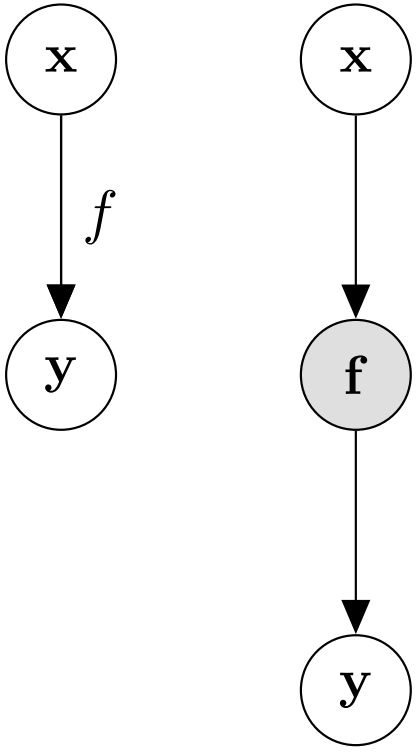
Outline

Part 1: GP Take-home messages

- Gaussian processes as infinite dimensional Gaussian distributions
- $\Rightarrow$ can be used as priors over functions
- Non-parametric: training data act as parameters
- Uncertainty Quantification
- Learning from scarce data

Notation and graphical models

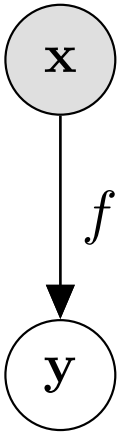
Graphical models



- White nodes: Observed variables
- Shaded nodes: Unobserved, or *latent* variables.
- Convention: Sometimes the latent function will be placed next to the arrow; sometimes I'll explicitly include the collection of instantiations $\mathbf{f}$

Outline

Unsupervised learning: GP-LVM



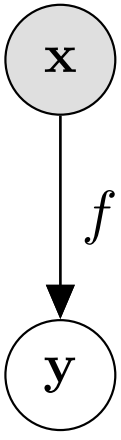
- If $\mathbf{X}$ is unobserved, treat it as a parameter and optimize over it.
- GP-LVM is interpreted as non-linear, non-parametric probabilistic PCA *(Lawrence, JMLR 2015)*.
- Objective (likelihood function) is similar to GPs:

$$p(\mathbf{y}|\mathbf{x}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{x})d\mathbf{f} = \mathcal{N}(\mathbf{y}|\mathbf{0}, K_{ff} + \sigma^2\mathbf{I})$$

but now x's are optimized too.

Neil Lawrence, JMLR, 2005.

Unsupervised learning: GP-LVM



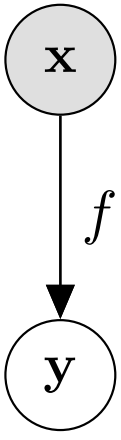
- If $\mathbf{X}$ is unobserved, treat it as a parameter and optimize over it.
- GP-LVM is interpreted as non-linear, non-parametric probabilistic PCA *(Lawrence, JMLR 2015)*.
- Objective (likelihood function) is similar to GPs:

$$p(\mathbf{y}|\mathbf{x}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{x})d\mathbf{f} = \mathcal{N}(\mathbf{y}|\mathbf{0}, K_{ff} + \sigma^2\mathbf{I})$$

but now x's are optimized too.

Neil Lawrence, JMLR, 2005.

Unsupervised learning: GP-LVM



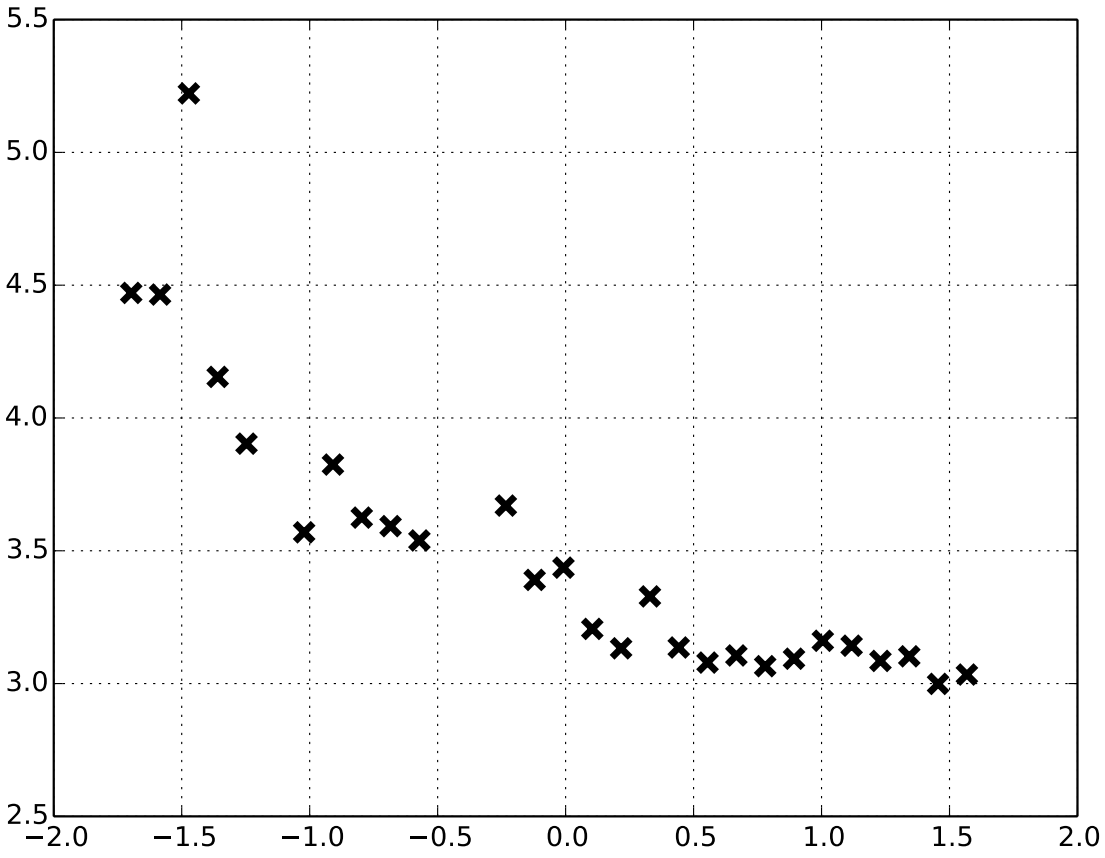
- If $\mathbf{X}$ is unobserved, treat it as a parameter and optimize over it.
- GP-LVM is interpreted as non-linear, non-parametric probabilistic PCA *(Lawrence, JMLR 2015)*.
- Objective (likelihood function) is similar to GPs:

$$p(\mathbf{y}|\mathbf{x}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{x})d\mathbf{f} = \mathcal{N}(\mathbf{y}|\mathbf{0}, K_{ff} + \sigma^2\mathbf{I})$$

but now x's are optimized too.

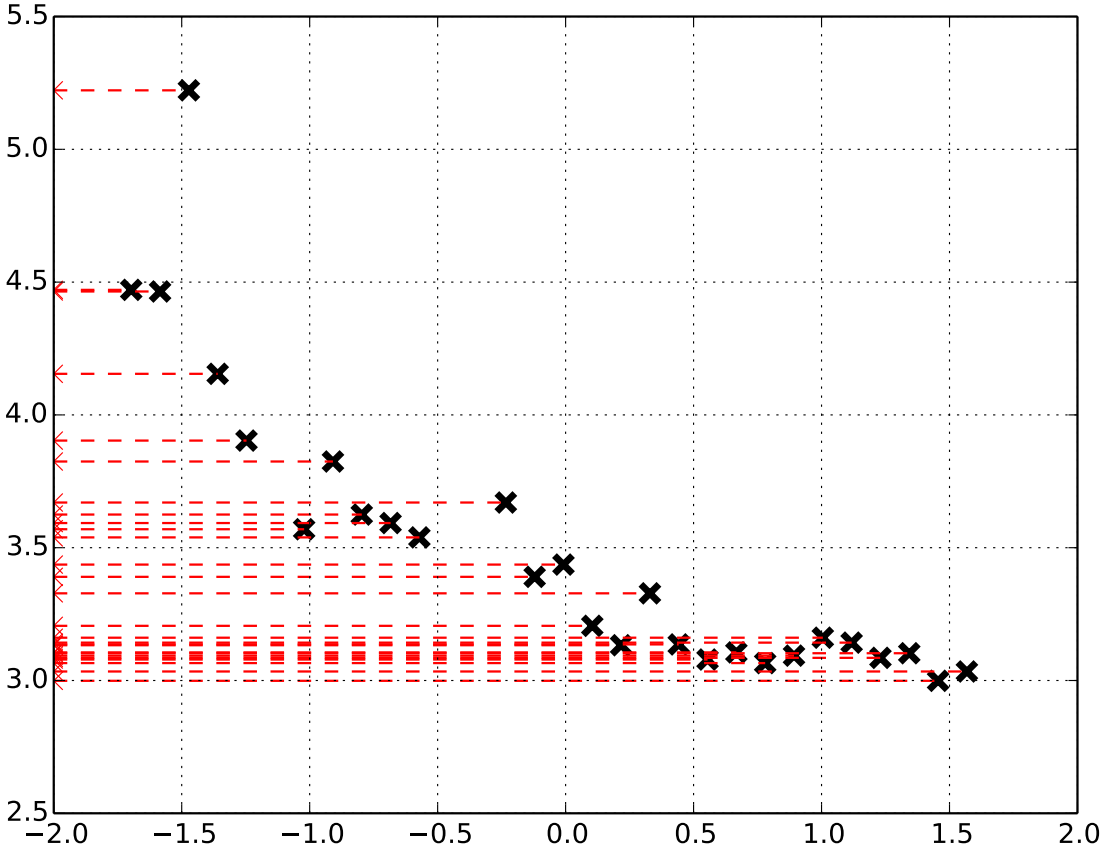
Neil Lawrence, JMLR, 2005.

Fitting the GP-LVM



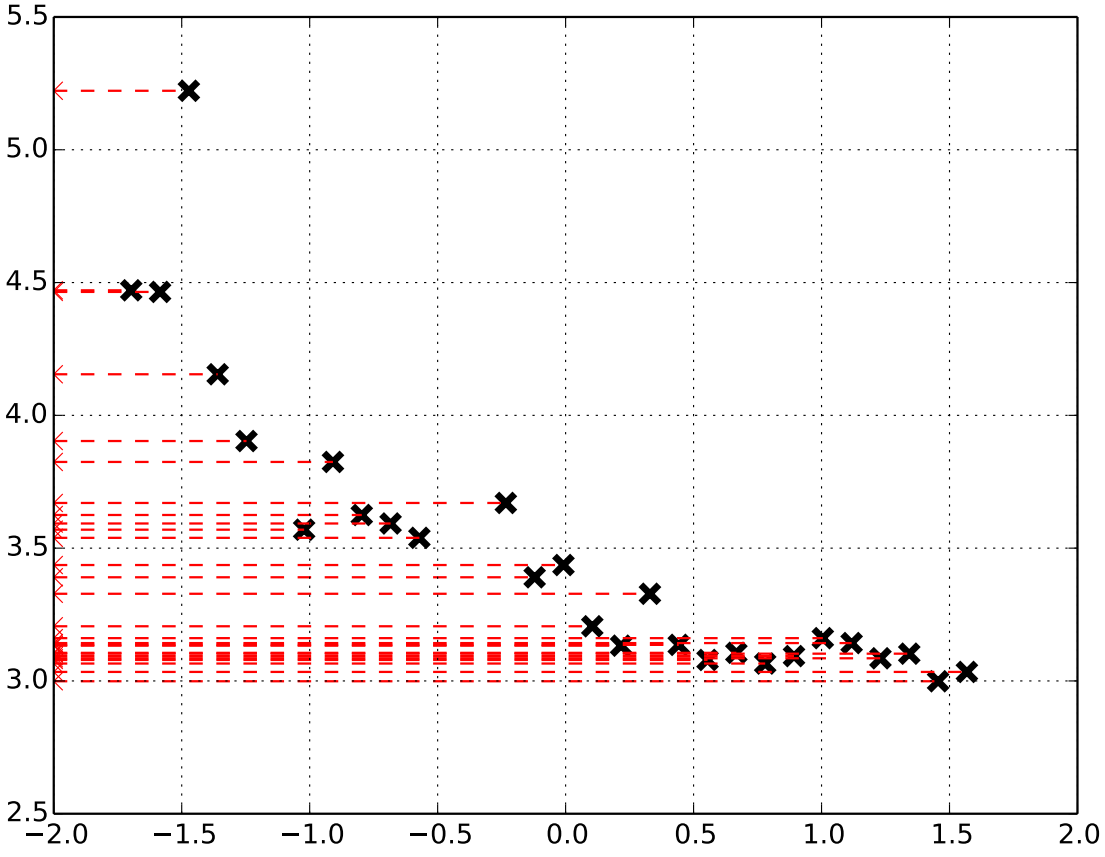
Fitting the GP-LVM

Figure credits: C. H. Ek



Fitting the GP-LVM

Figure credits: C. H. Ek



- Additional difficulty: x's are also missing!
- Improvement: Invoke the Bayesian methodology to find x's too.

Bayesian GP-LVM

Additionally integrate out the latent space.

$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{x})p(\mathbf{x})\mathbf{df}\mathbf{dx}$$

where

$$p(\mathbf{x}) \sim \mathcal{N}(0, \mathbf{I})$$

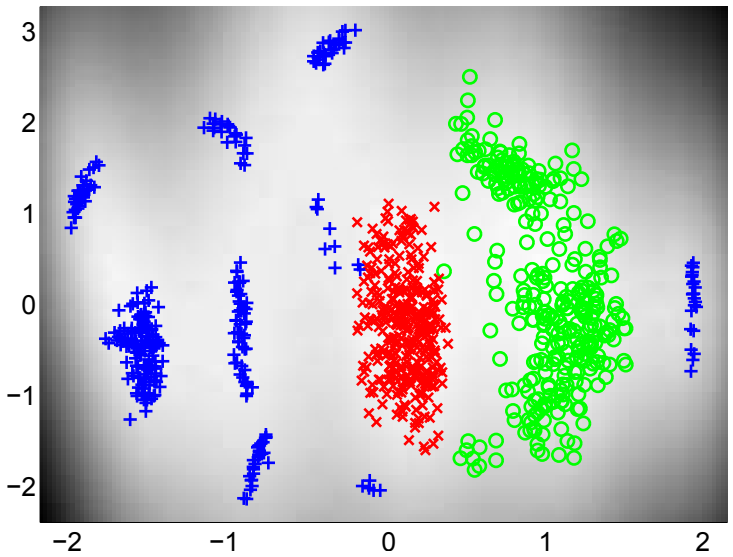
Titsias and Lawrence 2010, AISTATS; Damianou et al. 2015, JMLR

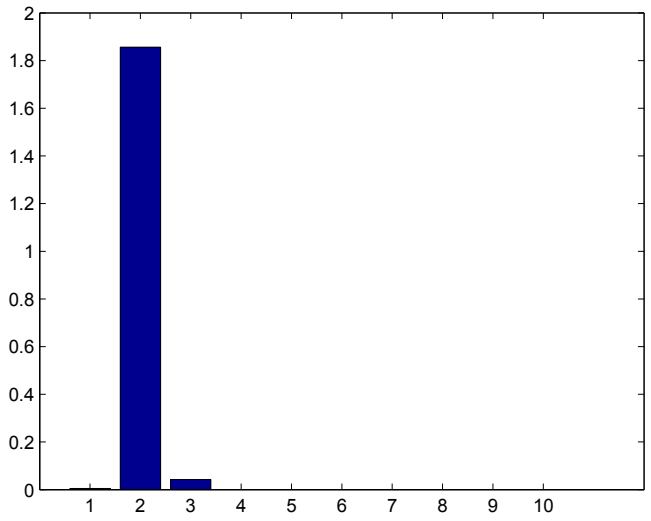
Automatic dimensionality detection

- In general, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ with $\mathbf{x}^{(n)} \in \Re^Q$.
- Automatic dimenionality detection by employing automatic relevance determination (ARD) priors for the mapping f.
- $f \sim \mathcal{GP}(\mathbf{0}, k_f)$ with:

$$k_f\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \sigma^2 \exp\left(-\frac{1}{2}\sum_{q=1}^{Q} w^{(q)}\left(x^{(i,q)} - x^{(j,q)}\right)^2\right)$$

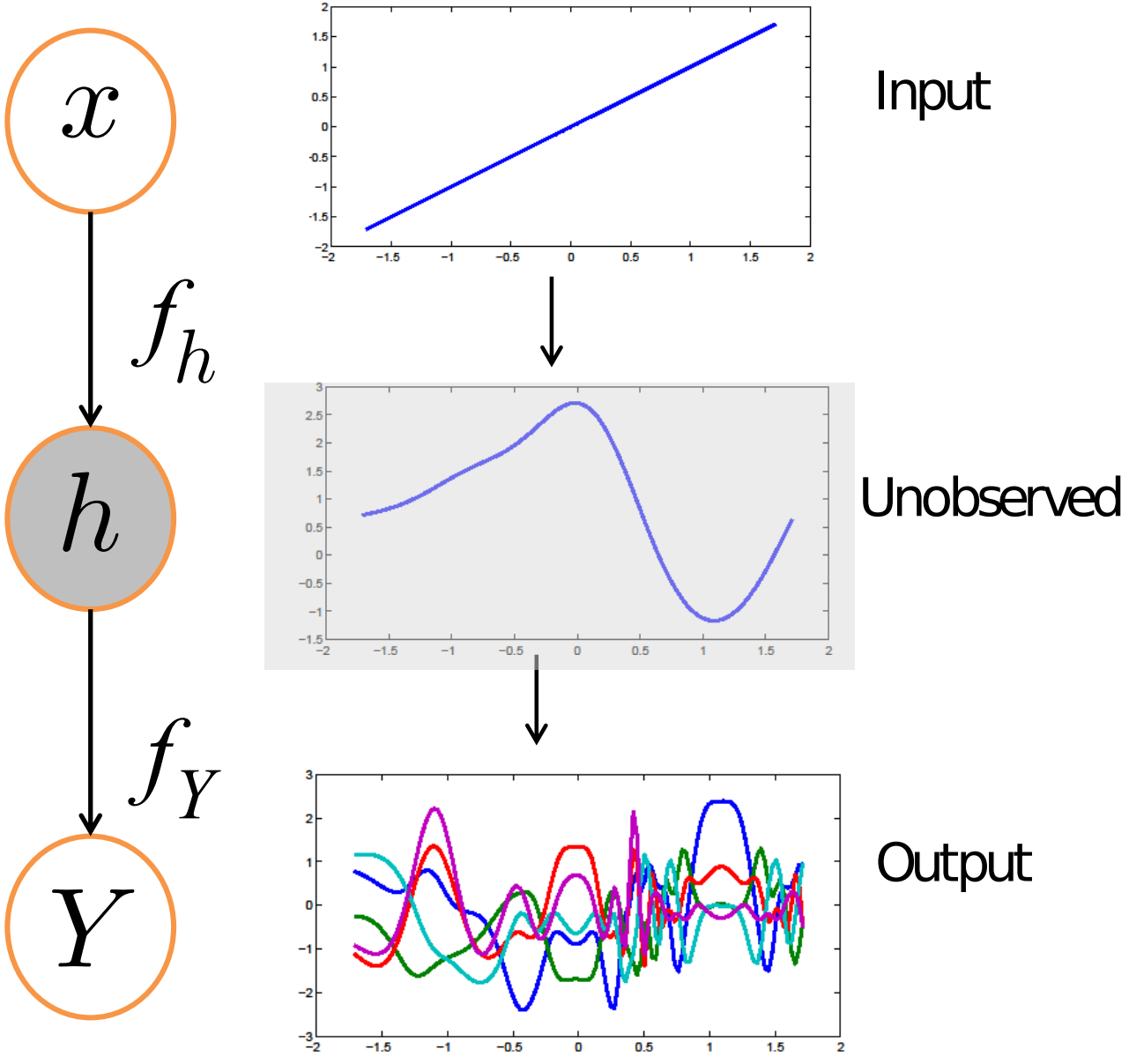
- Example:





Outline

Sampling from a deep GP



Model

$$\mathbf{x} = \text{given}$$
$$\mathbf{f}_h = f_h(x) = \mathcal{GP}(\mathbf{0}, k_h(x, x))$$
$$\mathbf{h} = f_h(x) + \epsilon_h$$
$$\mathbf{f}_y = f_y(h) = \mathcal{GP}(\mathbf{0}, k_f(h, h))$$
$$\mathbf{y} = f_y(h) + \epsilon_y$$

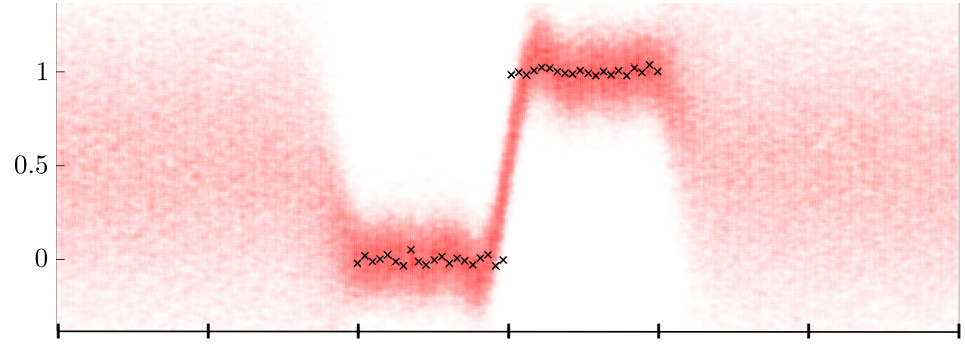
So: $\mathbf{y} = f_y(f_h(\mathbf{x}) + \epsilon_h) + \epsilon_y$

Objective:

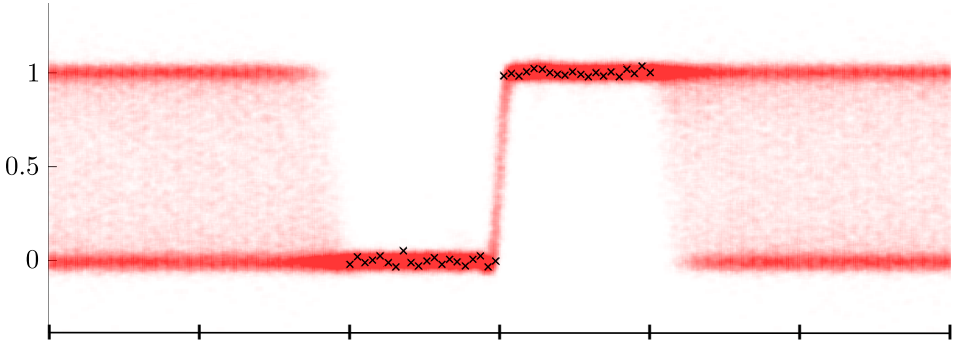
$$p(\mathbf{y}|\mathbf{x}) = \int p(\mathbf{y}|\mathbf{f}_y)p(\mathbf{f}_y|\mathbf{h})p(\mathbf{h}|\mathbf{f}_h)p(\mathbf{f}_h|\mathbf{x})\mathrm{d}\mathbf{f}_y\mathrm{d}\mathbf{f}_h\mathrm{d}\mathbf{h}$$

Damianou and Lawrence, AISTATS 2013; Damianou, PhD Thesis, 2015

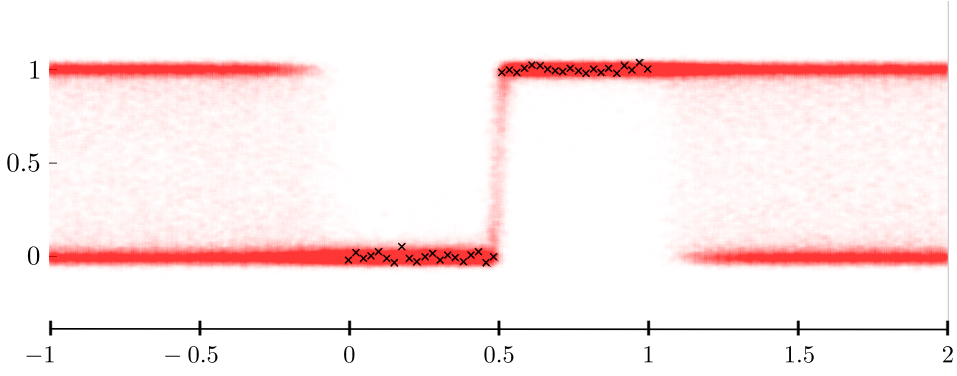
Deep GP: Step function (credits for idea to J. Hensman, R. Calandra, M. Deisenroth)



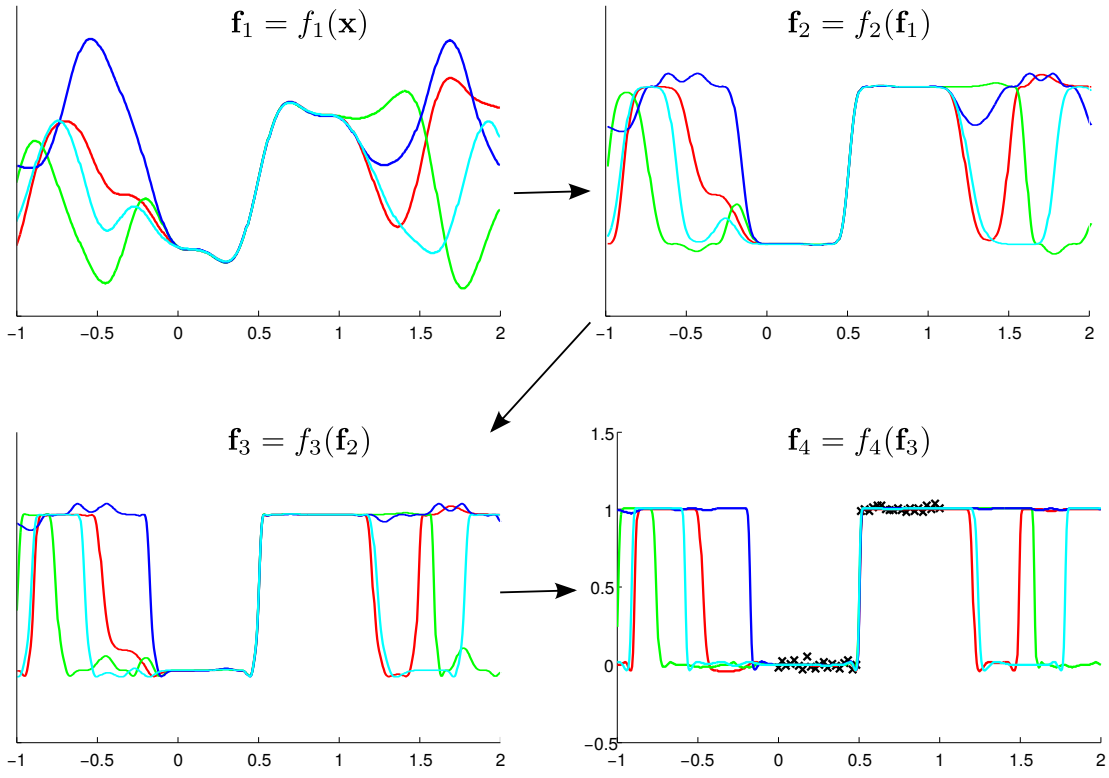
(standard GP)



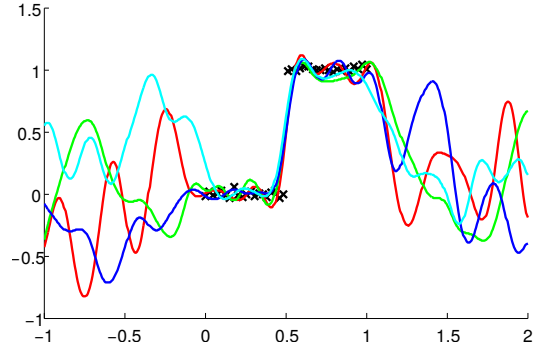
(deep GP - 1 hidden layer)



(deep GP - 3 hidden layers)



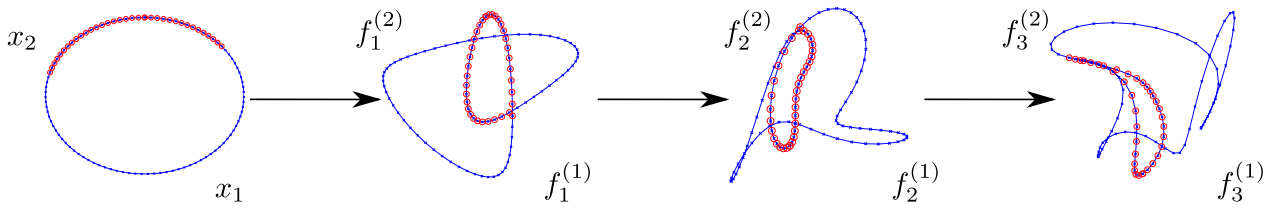
Deep GP with three hidden plus one warping layer



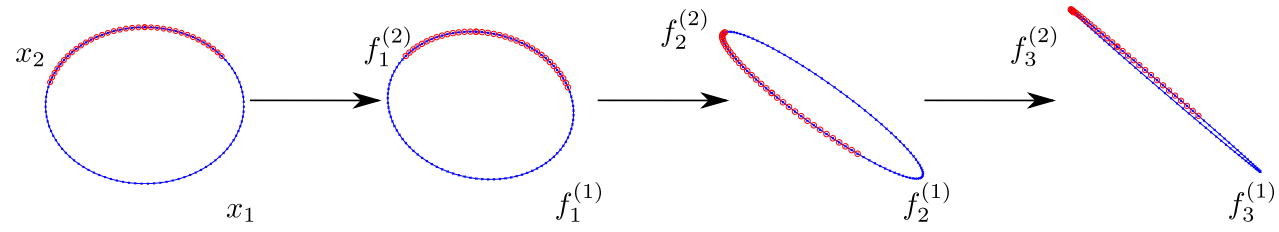
Standard GP

Non-linear feature learning

Stacked GP (nonlinear)



Stacked PCA (linear)



- Successive warping creates "knots" which act as features.
- Features discovered in one layer remain in the next ones (ie knots are not un-tied)

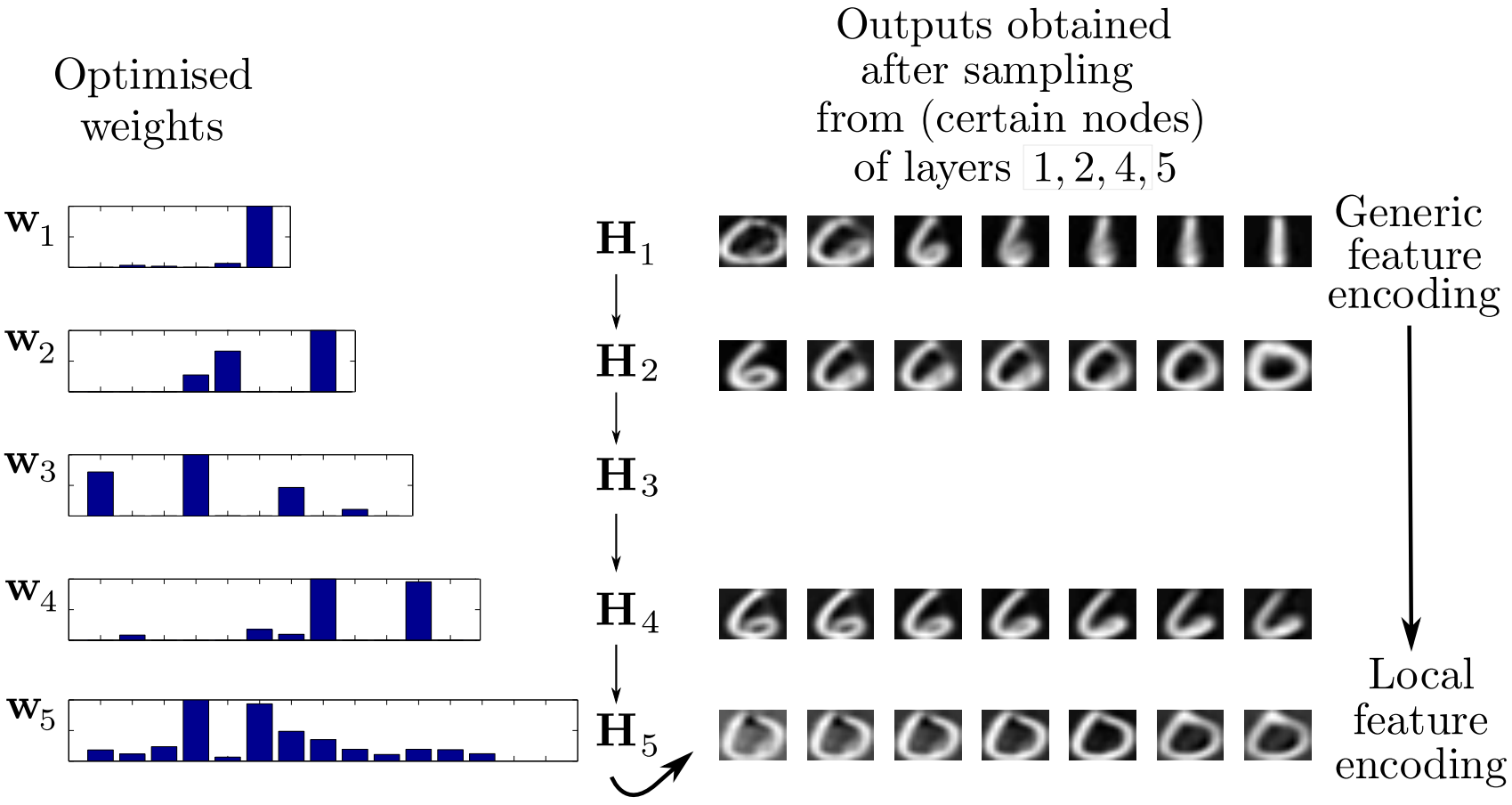
$$f_1(f_2(f_3(x)))$$

- With linear warpings (stacked PCA) we can't achieve this effect:

$$W_1(W_2(W_3x))) = W'x$$

Damianou, PhD Thesis, 2015

Deep GP: MNIST example



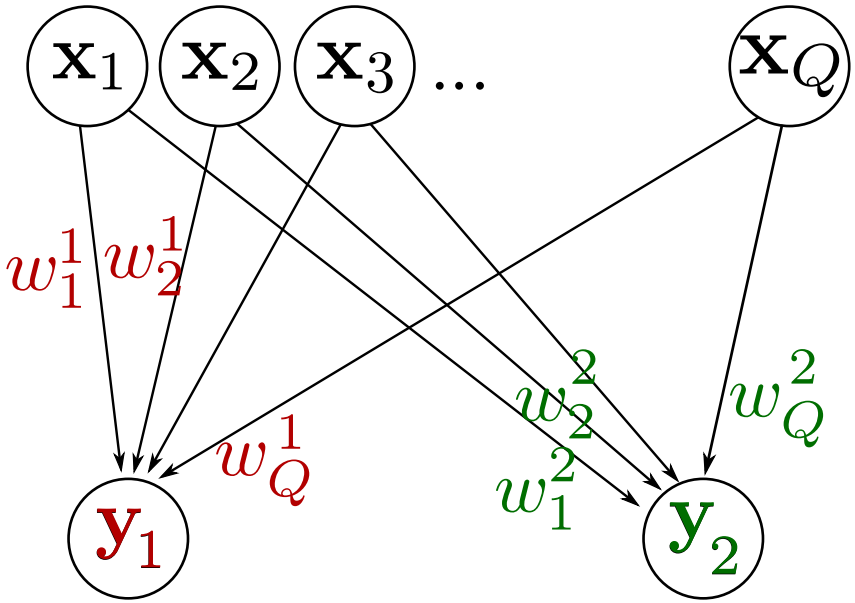
▸ https://youtu.be/E8-vxt8wxBU (Live sampling video)

Outline

Multi-view: Manifold Relevance Determination (MRD)

Multi-view data arise from multiple information sources. These sources naturally contain some overlapping, or *shared* signal (since they describe the same "phenomenon"), but also have some *private* signal.

MRD: Model such data using overlapping sets of latent variables



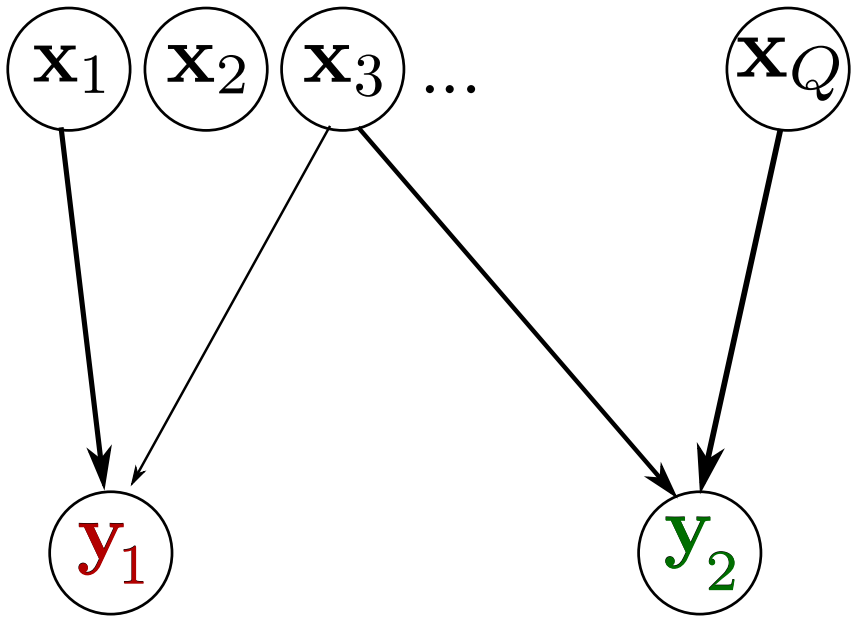
Demo: ▸ https://youtu.be/rIPX3CIOhKY *Faces*, ▸ https://youtu.be/xOXnJAAVgLE *mocap-silhouette*

Damianou et al., ICML 2012; Damianou, PhD Thesis, 2015

Multi-view: Manifold Relevance Determination (MRD)

Multi-view data arise from multiple information sources. These sources naturally contain some overlapping, or *shared* signal (since they describe the same "phenomenon"), but also have some *private* signal.

MRD: Model such data using overlapping sets of latent variables



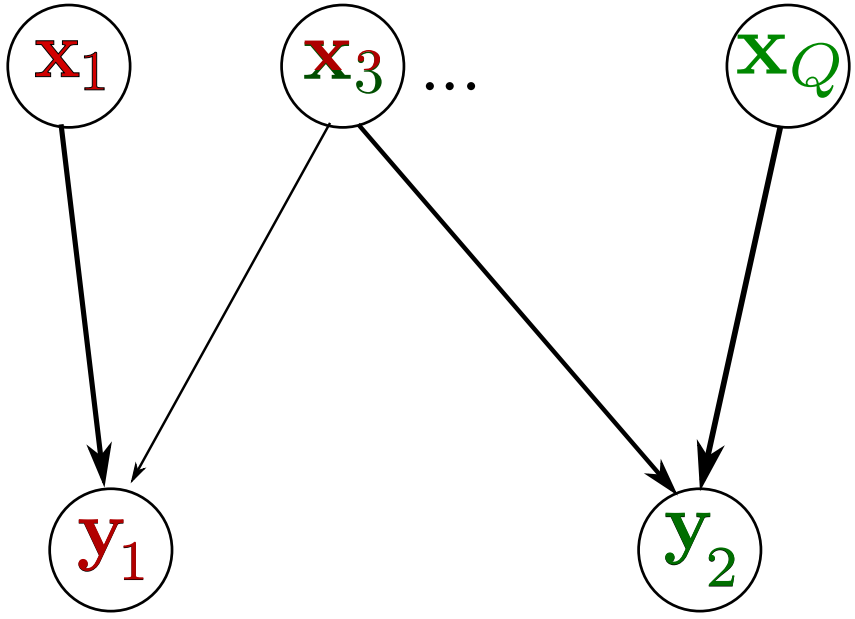
Demo: ▸ https://youtu.be/rIPX3CIOhKY *Faces*, ▸ https://youtu.be/xOXnJAAVgLE *mocap-silhouette*

Damianou et al., ICML 2012; Damianou, PhD Thesis, 2015

Multi-view: Manifold Relevance Determination (MRD)

Multi-view data arise from multiple information sources. These sources naturally contain some overlapping, or *shared* signal (since they describe the same "phenomenon"), but also have some *private* signal.

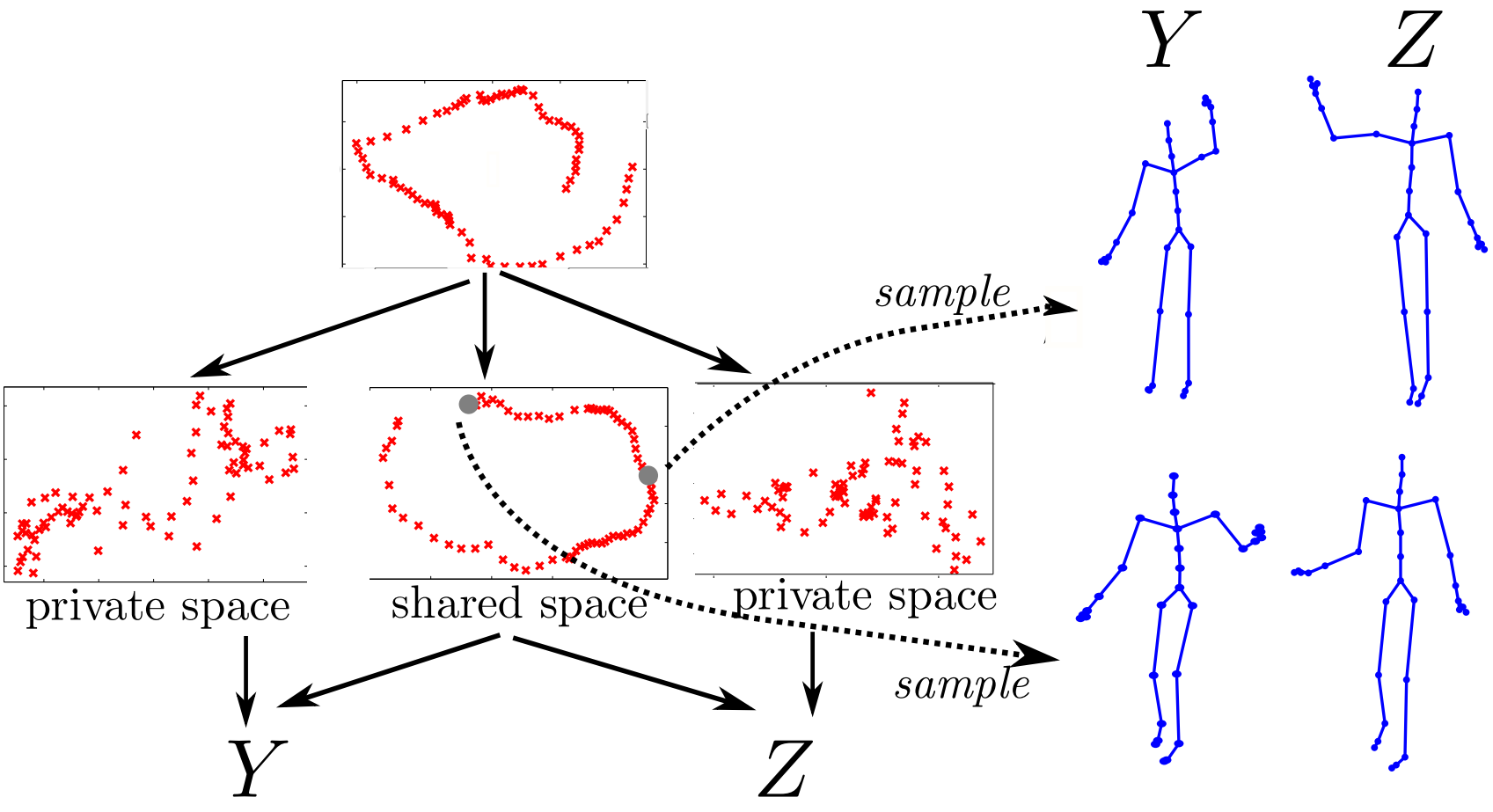
MRD: Model such data using overlapping sets of latent variables



Demo: ▸ https://youtu.be/rIPX3CIOhKY *Faces*, ▸ https://youtu.be/xOXnJAAVgLE *mocap-silhouette*

Damianou et al., ICML 2012; Damianou, PhD Thesis, 2015

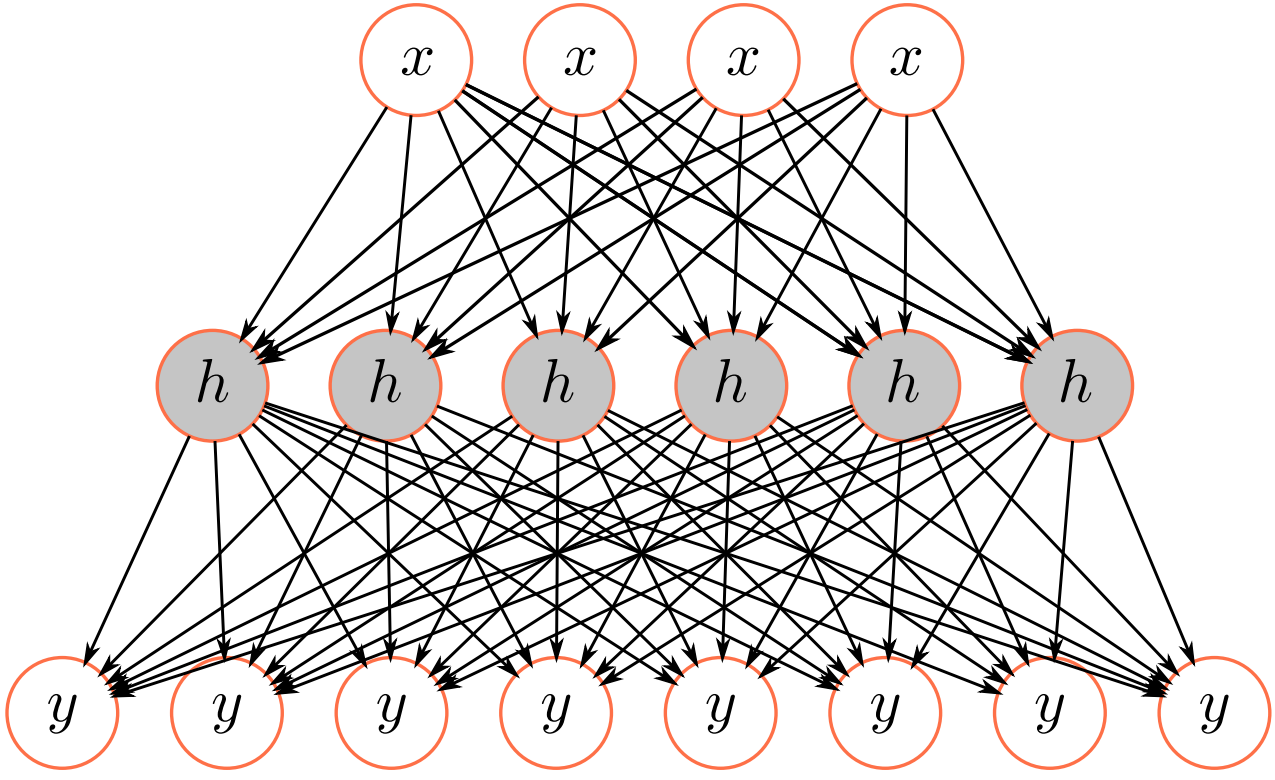
Deep GPs: Another multi-view example



Automatic structure discovery summary: ARD and MRD

Tools:

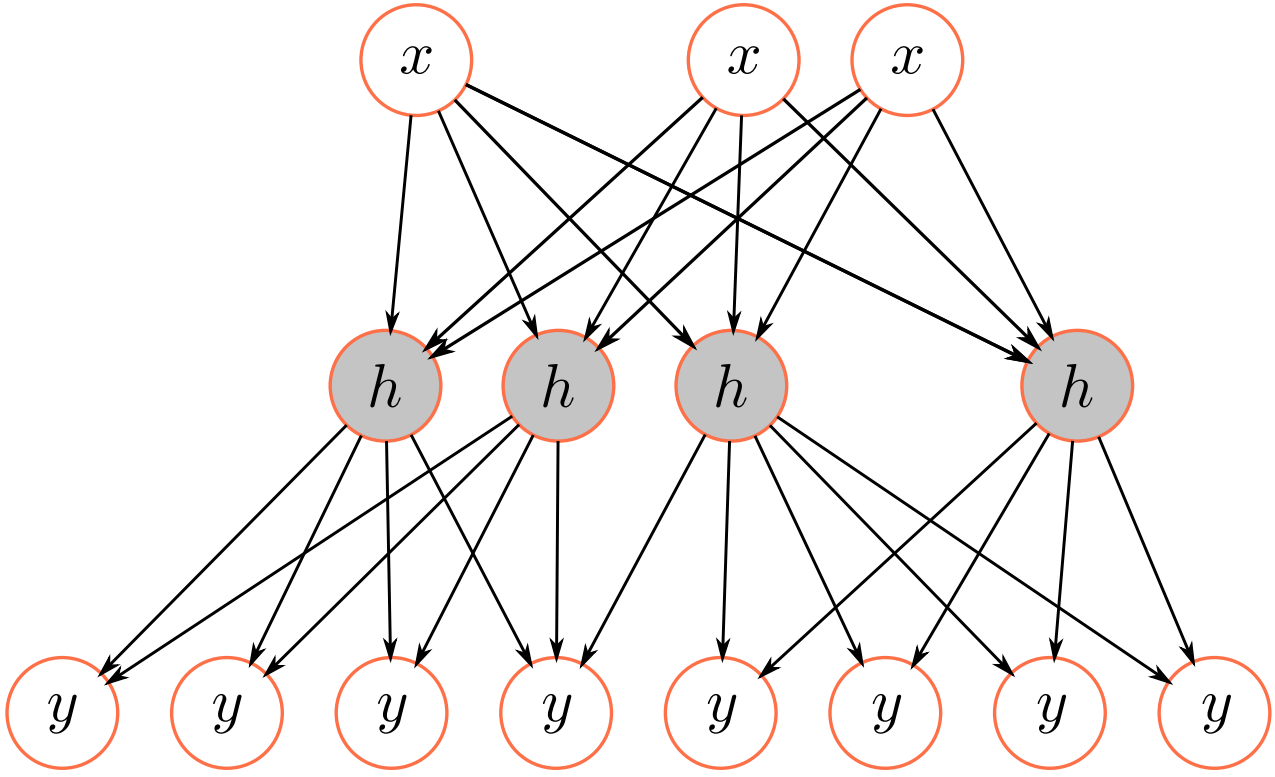
- ARD: Eliminate uncessary nodes/connections
- MRD: Conditional independencies
- Approximating evidence: Number of layers (?)



Automatic structure discovery summary: ARD and MRD

Tools:

- ARD: Eliminate uncessary nodes/connections
- MRD: Conditional independencies
- Approximating evidence: Number of layers (?)



More deepGP representation learning examples...

- https://youtu.be/s4zATH1TjG8 iCub interaction
- https://youtu.be/uSQ0vxLcfVU Faces - autoencoder

Outline

Dynamical systems examples

Through a model, we wish to learn to:

- perform free simulation by learning patterns coming from the latent generation process (a mechanistic system we do not know)
- perform inter/extrapolation in time-series data which are **very** high-dimensional (e.g. video)
- detect outliers in data coming from a dynamical system
- optimize policies for control based on a model of the data.

Policy learning

Model-based policy learning

- https://www.youtube.com/watch?v=XiigTGKZfks *(Cart-pole)*

- http://mlg.eng.cam.ac.uk/?portfolio=andrew-mchutchon *(Unicycle)*

Work by: Marc Deisenroth, Andrew McHutchon, Carl Rasmussen

NARX model

A **standard NARX model** considers an input vector $\mathbf{x}_i \in \mathbb{R}^D$ comprised of L_y past observed outputs $y_i \in \mathbb{R}$ and L_u past exogenous inputs $u_i \in \mathbb{R}$:

$$\mathbf{x}_i = [y_{i-1}, \cdots, y_{i-L_y}, u_{i-1}, \cdots, u_{i-L_u}]^\top,$$
$$y_i = f(\mathbf{x}_i) + \epsilon_i^{(y)}, \qquad \epsilon_i^{(y)} \sim \mathcal{N}(\epsilon_i^{(y)}|0, \sigma_y^2),$$

Latent auto-regressive GP model:

$$x_i = f(x_{i-1}, \cdots, x_{i-L_x} u_{i-1}, \cdots, u_{i-L_u}) + \epsilon_i^{(x)},$$
$$y_i = x_i + \epsilon_i^{(y)},$$

Contribution 1: Simultaneous auto-regressive and representation learning.

Contribution 2: Latents avoid the feedback of possibly corrupted observations into the dynamics.

Mattos, Damianou, Barreto, Lawrence, 2016

NARX model

A **standard NARX model** considers an input vector $\mathbf{x}_i \in \mathbb{R}^D$ comprised of L_y past observed outputs $y_i \in \mathbb{R}$ and L_u past exogenous inputs $u_i \in \mathbb{R}$:

$$\mathbf{x}_i = [y_{i-1}, \cdots, y_{i-L_y}, u_{i-1}, \cdots, u_{i-L_u}]^\top,$$
$$y_i = f(\mathbf{x}_i) + \epsilon_i^{(y)}, \qquad \epsilon_i^{(y)} \sim \mathcal{N}(\epsilon_i^{(y)}|0, \sigma_y^2),$$

Latent auto-regressive GP model:

$$x_i = f(x_{i-1}, \cdots, x_{i-L_x} u_{i-1}, \cdots, u_{i-L_u}) + \epsilon_i^{(x)},$$
$$y_i = x_i + \epsilon_i^{(y)},$$

Contribution 1: Simultaneous auto-regressive and representation learning.

Contribution 2: Latents avoid the feedback of possibly corrupted observations into the dynamics.

Mattos, Damianou, Barreto, Lawrence, 2016

Robustness to outliers

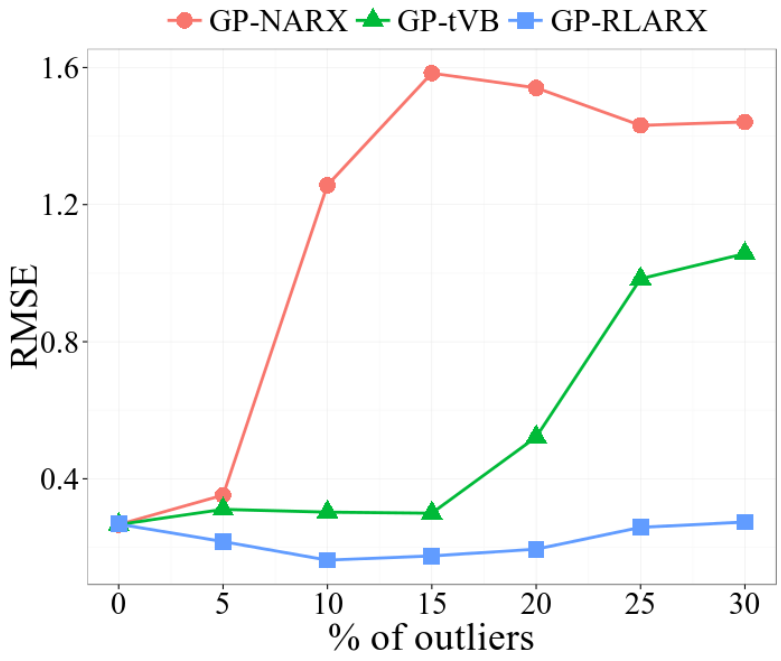
Latent auto-regressive GP model:

$$
\begin{aligned}
x_i &= f(x_{i-1}, \cdots, x_{i-L_x} u_{i-1}, \cdots, u_{i-L_u}) + \epsilon_i^{(x)}, \\
y_i &= x_i + \epsilon_i^{(y)}, \\
\epsilon_i^{(x)} &\sim \mathcal{N}(\epsilon_i^{(x)} | 0, \sigma_x^2), \\
\epsilon_i^{(y)} &\sim \mathcal{N}(\epsilon_i^{(y)} | 0, \tau_i^{-1}), \quad \tau_i \sim \Gamma(\tau_i | \alpha, \beta),
\end{aligned}
$$

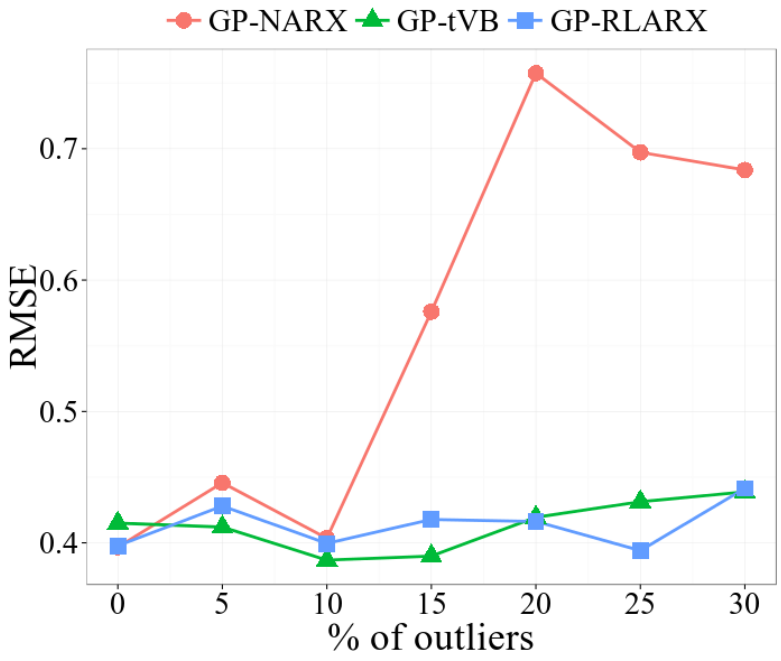
Contribution 3: "Switching-off" outliers by including the above Student-t likelihood for the noise.

Mattos, Damianou, Barreto, Lawrence, DYCOPS 2016

Robust GP autoregressive model: demonstration

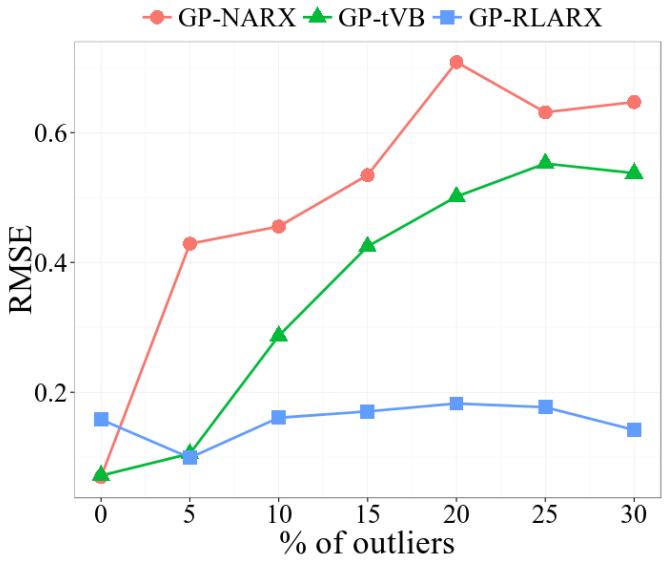


(a) Artificial 1.

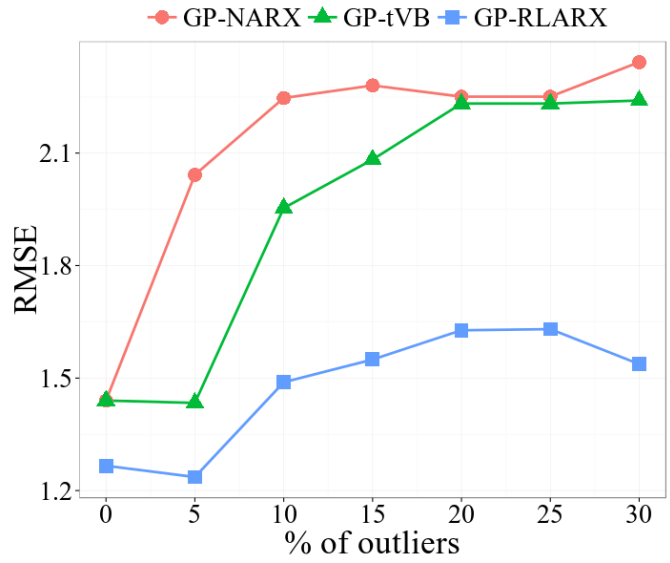


(b) Artificial 2.

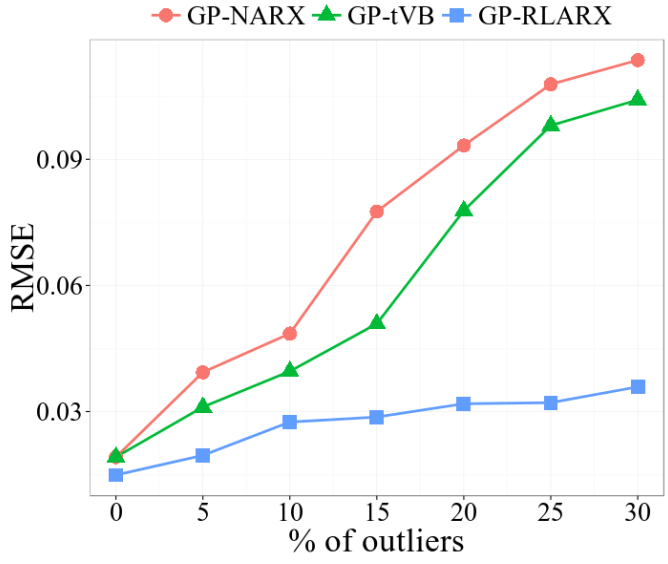
Figure: RMSE values for free simulation on test data with different levels of contamination by outliers.



(c) Artificial 3.



(d) Artificial 4.



(e) Artificial 5.

Going deeper: Deep Recurrent Gaussian Process

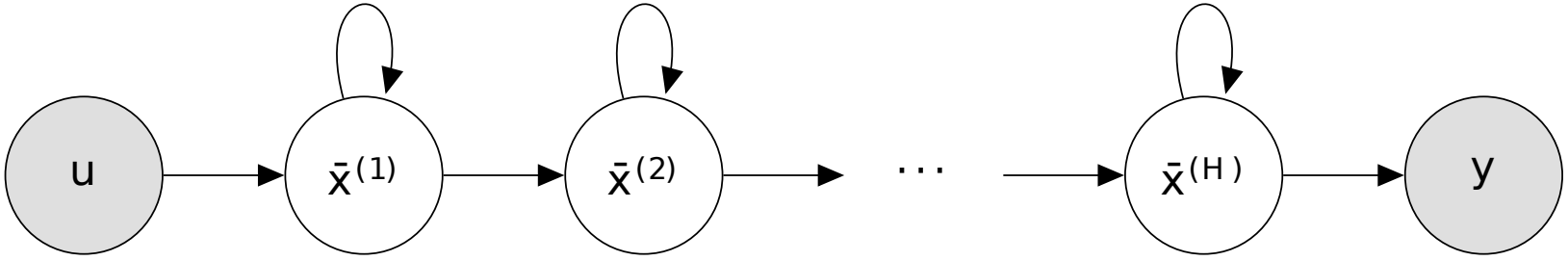
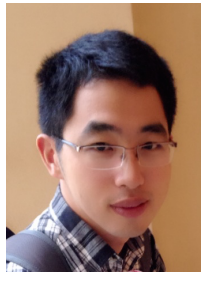


Figure 1: RGP graphical model with H hidden layers.

$\tilde{x}$ is the lagged latent function values augmented with the lagged exogenous inputs.

Mattos, Dai, Damianou, Barreto, Lawrence, ICLR 2016



Inference is tricky...

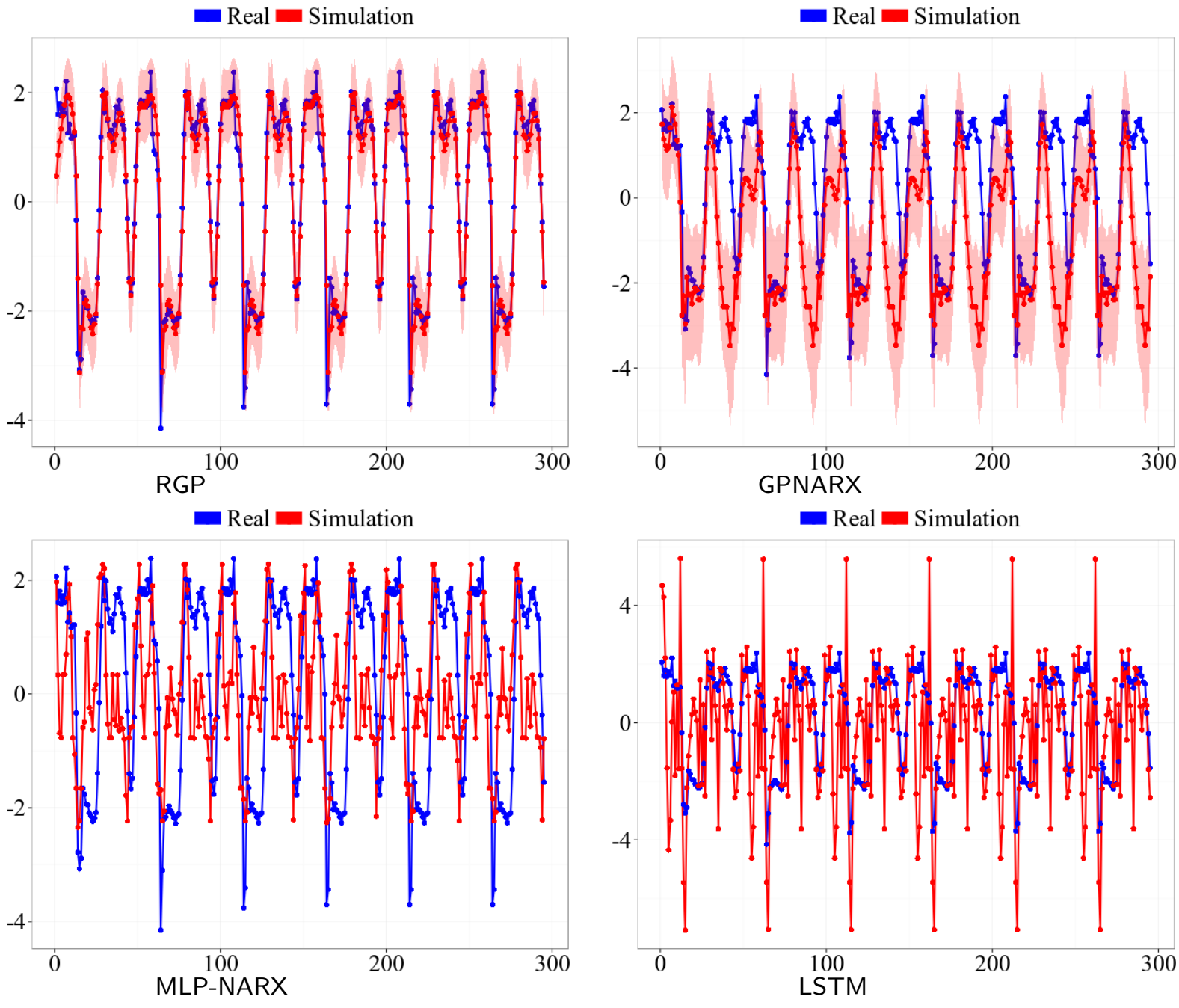
$$
\begin{aligned}
\log p(\mathbf{y}) \geq & -\frac{N-L}{2}\sum_{h=1}^{H+1}\log 2\pi\sigma_h^2 - \frac{1}{2\sigma_{H+1}^2}\Big(\mathbf{y}^\top\mathbf{y} + \Psi_0^{(H+1)} \\
& - \mathsf{Tr}\left(\left(\mathbf{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{(H+1)}\right)\Big) + \frac{1}{2}\left|\mathbf{K}_z^{(H+1)}\right| - \frac{1}{2}\left|\mathbf{K}_z^{(H+1)} + \frac{1}{\sigma_{H+1}^2}\boldsymbol{\Psi}_2^{(H+1)}\right| \\
& + \frac{1}{2(\sigma_{H+1}^2)^2}\mathbf{y}^\top\boldsymbol{\Psi}_1^{(H+1)}\left(\mathbf{K}_z^{(H+1)} + \frac{1}{\sigma_{H+1}^2}\boldsymbol{\Psi}_2^{(H+1)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(H+1)}\right)^\top\mathbf{y} \\
& + \sum_{h=1}^{H}\left\{-\frac{1}{2\sigma_h^2}\left(\sum_{i=L+1}^{N}\lambda_i^{(h)} + \left(\mu^{(h)}\right)^\top\mu^{(h)} + \Psi_0^{(h)} - \mathsf{Tr}\left(\left(\mathbf{K}_z^{(h)}\right)^{-1}\boldsymbol{\Psi}_2^{(h)}\right)\right)\right. \\
& + \frac{1}{2}\left|\mathbf{K}_z^{(h)}\right| - \frac{1}{2}\left|\mathbf{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right| \\
& + \frac{1}{2(\sigma_h^2)^2}\left(\mu^{(h)}\right)^\top\boldsymbol{\Psi}_1^{(h)}\left(\mathbf{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(h)}\right)^\top\mu^{(h)} \\
& \left. - \sum_{i=L+1}^{N}\int_{x_i^{(h)}} q\left(x_i^{(h)}\right)\log q\left(x_i^{(h)}\right) + \sum_{i=1}^{L}\int_{x_i^{(h)}} q\left(x_i^{(h)}\right)\log p\left(x_i^{(h)}\right)\right\}.
\end{aligned}
$$

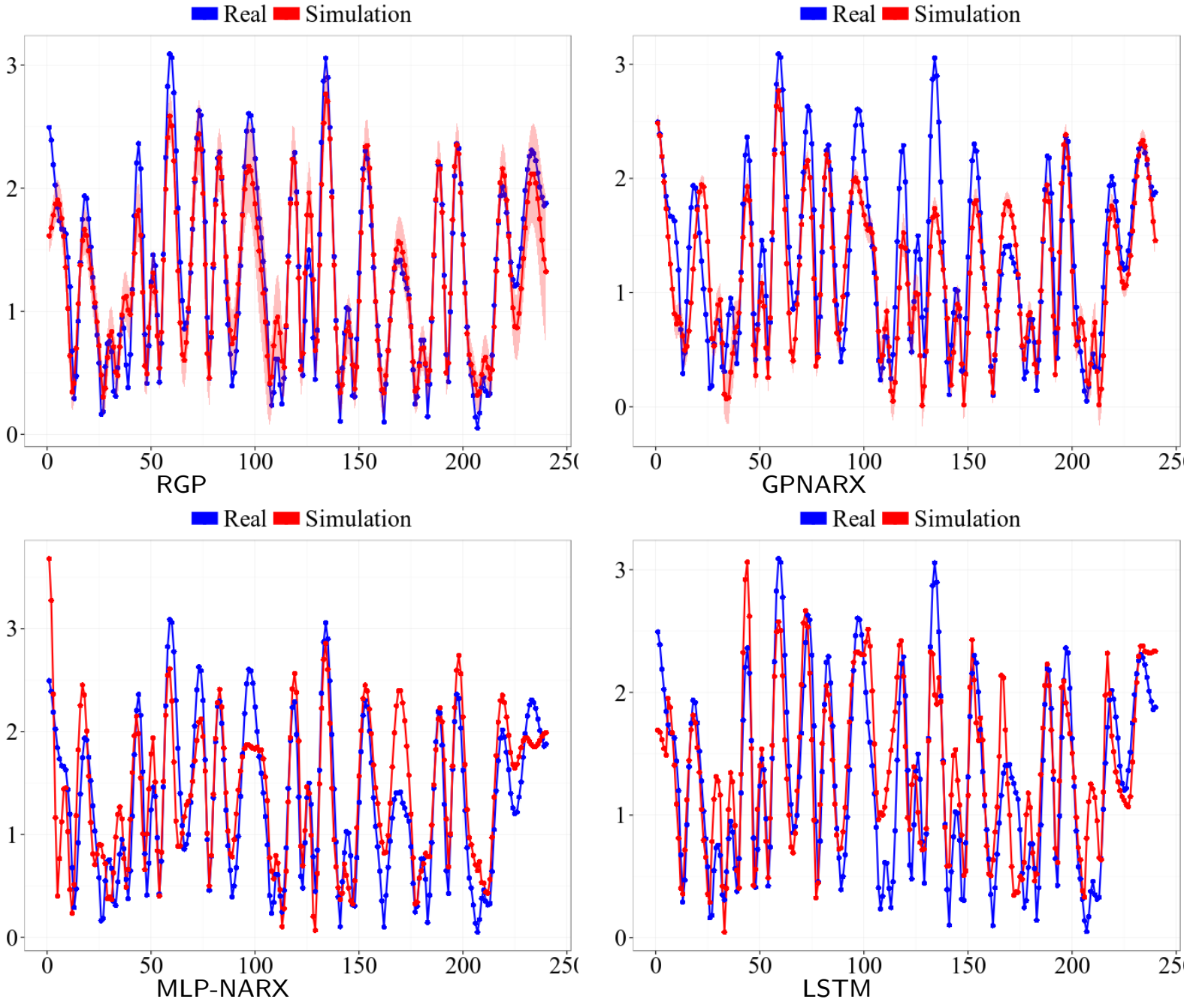
Results

Results in nonlinear systems identification:

1. artificial dataset

2. "drive" dataset: by a system with two electric motors that drive a pulley using a flexible belt.
   - input: the sum of voltages applied to the motors
   - output: speed of the belt.


Real
Simulation
2
0
-2
-4
0
100
200
300
RGP
Real
Simulation
2
0
-2
-4
0
100
200
300
GPNARX
Real
Simulation
2
0
-2
-4
0
100
200
300
MLP-NARX
Real
Simulation
4
0
-4
0
100
200
300
LSTM


Real
Simulation
RGP
GPNARX
MLP-NARX
LSTM

Avatar control

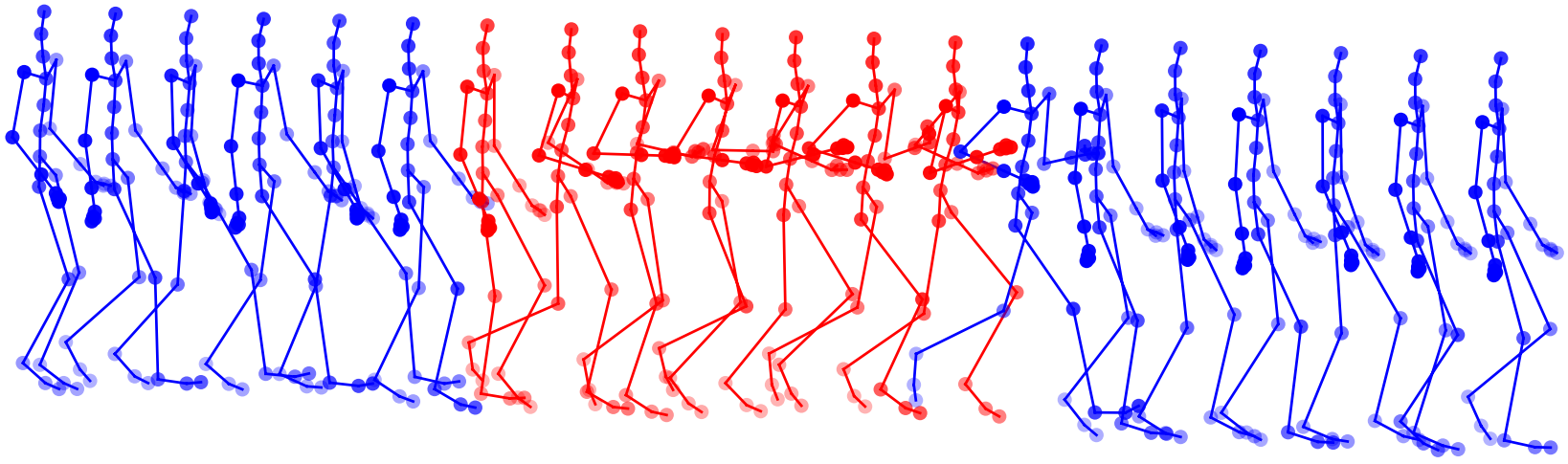
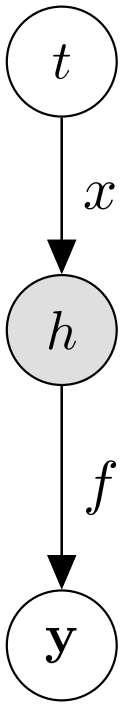


Figure: The generated motion with a step function signal, starting with walking (blue), switching to running (red) and switching back to walking (blue).

Videos:

- https://youtu.be/FR-oeGxV6yY *Switching between learned speeds*
- https://youtu.be/AT0HMtoPgjc *Interpolating (un)seen speed*
- https://youtu.be/FuF-uZ83VMw Constant unseen speed

Regressive dynamics with deep GPs



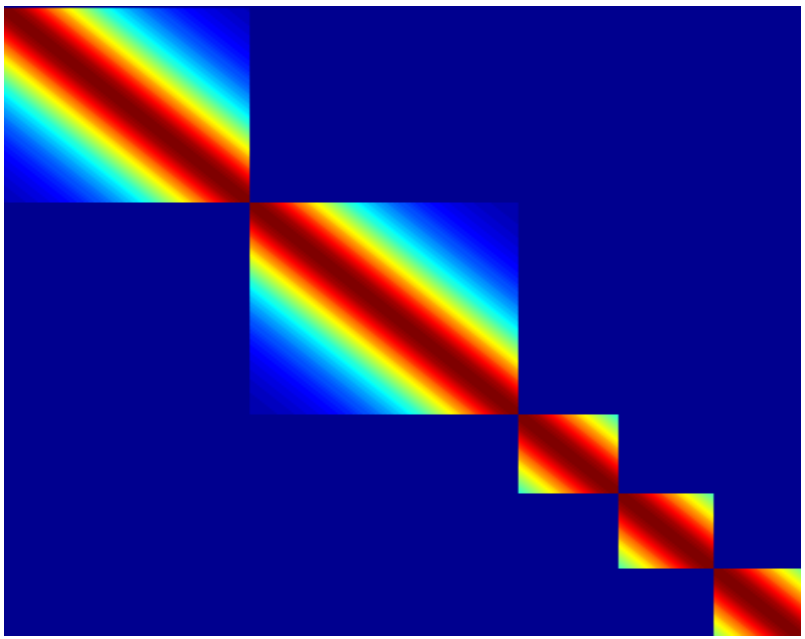
Instead of coupling f's by encoding the Markov property, we couple them by **coupling the f's inputs through another GP with time as input.**

$$y = f(x) + \epsilon$$
$$x \sim \mathcal{GP}(0, k_x(t, t))$$
$$f \sim \mathcal{GP}(0, k_f(x, x))$$

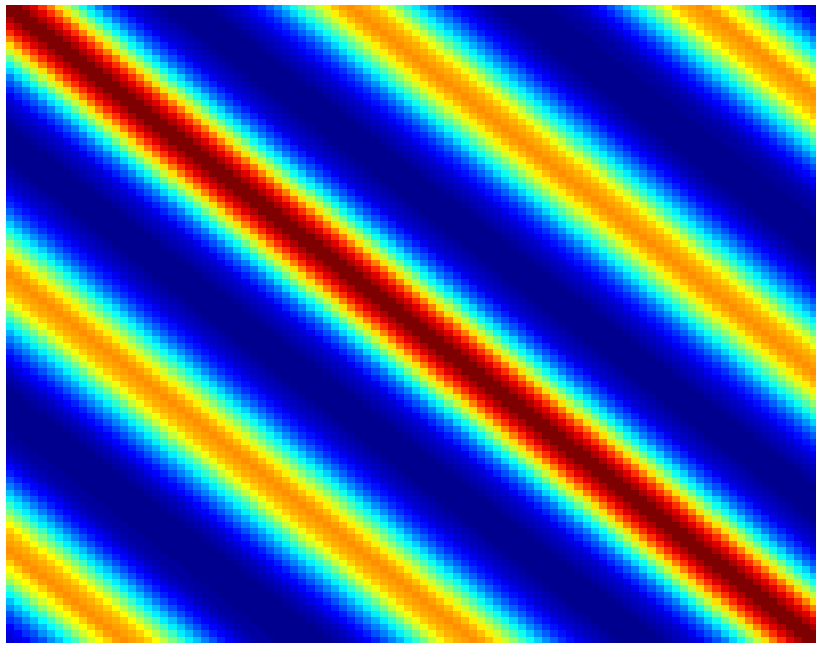
Damianou et al., NIPS 2011; Damianou, Titsias and Lawrence, JMLR, 2015

Dynamics

- Dynamics are encoded in the covariance matrix $\mathbf{K} = k(\mathbf{t}, \mathbf{t})$.
- We can consider special forms for $\mathbf{K}$.



Model individual sequences



Model periodic data

- ▸ https://www.youtube.com/watch?v=i9TEoYxaBxQ *(missa)*
- ▸ https://www.youtube.com/watch?v=mUY1XHPnoCU *(dog)*
- ▸ https://www.youtube.com/watch?v=fHDWloJtgk8 (mocap)

Summary

- Data-driven, model-based approach to real and control problems
- Gaussian processes: Uncertainty quantification / propagation gives an advantage
- Deep Gaussian processes: Representation learning + dynamics learning
- Future work: Deep Gaussian processes + mechanistic information; consider "real" applications.

Thanks!

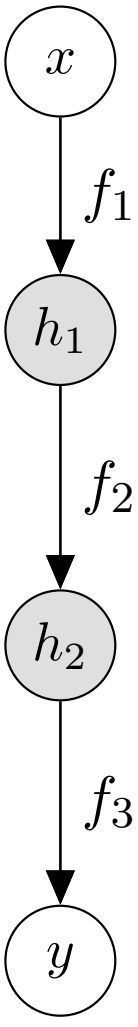
Thanks

Thanks to: Neil Lawrence, Michalis Titsias, Carl Henrik Ek, James Hensman, Cesar Lincoln Mattos, Zhenwen Dai, Javier Gonzalez, Tony Prescott, Uriel Martinez-Hernandez, Luke Boorman

Thanks to George Karniadakis, Paris Perdikaris for hosting me

BACKUP SLIDES 1: Deep GP Optimization - variational inference

MAP optimisation?



- Joint Distr. $= p(y|h_2)p(h_2|h_1)p(h_1|x)$
- MAP optimization is extremely problematic because:
  - Dimensionality of hs has to be decided a priori
  - Prone to overfitting, if h are treated as parameters
  - Deep structures are not supported by the model's objective but have to be forced [Lawrence & Moore '07]
- We want:
  - To use the *marginal likelihood* as the objective: marg. lik. $= \int_{h_2,h_1} p(y|h_2)p(h_2|h_1)p(h_1|x)$
  - Further regularization tools.

Marginal likelihood is intractable

Let's try to marginalize out the top layer only:

$$
\begin{aligned}
p(\mathbf{h}_2) &= \int p(\mathbf{h}_2|\mathbf{h}_1)p(\mathbf{h}_1)\mathrm{d}\mathbf{h}_1 \\
&= \int\int p(\mathbf{h}_2|\mathbf{f}_2)p(\mathbf{f}_2|\mathbf{h}_1)p(\mathbf{h}_1)\mathrm{d}\mathbf{f}_2\mathbf{h}_1 \\
&= \int p(\mathbf{h}_2|\mathbf{f}_2)\Big[\underbrace{\int p(\mathbf{f}_2|\mathbf{h}_1)p(\mathbf{h}_1)\mathrm{d}\mathbf{h}_1}_{\text{Intractable!}}\Big]\mathrm{d}\mathbf{f}_2
\end{aligned}
$$

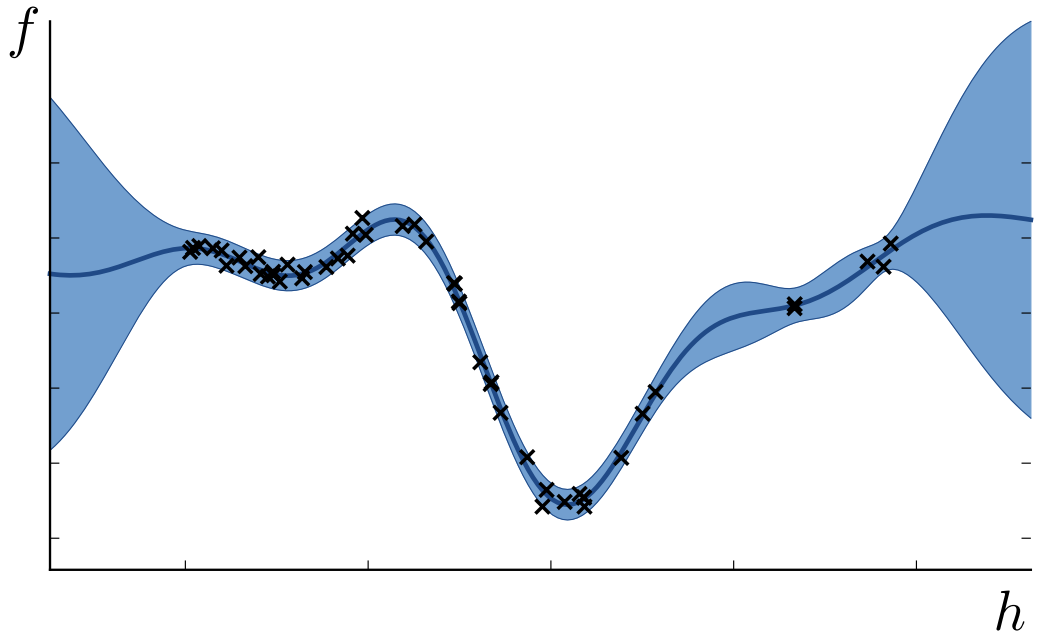
Intractability: $\mathbf{h}_1$ appears non-linearly in $p(\mathbf{f}_2|\mathbf{h}_1)$, inside $\mathbf{K}^{-1}$ (and also the determinant term), where $\mathbf{K} = k(\mathbf{h}_1, \mathbf{h}_1)$.

Solution: Variational Inference

- Similar issues arise for 1-layer models. Solution was given by Titsias and Lawrence, 2010. A small modification to that solution does the trick in deep GPs too.
- Extend Titsias' method for *variational learning of inducing variables in Sparse GPs*.
- Analytic variational bound $\mathcal{F} \leq p(y|x)$
- *Approximately* marginalise out h
- Hence obtain the approximate posterior $q(h)$

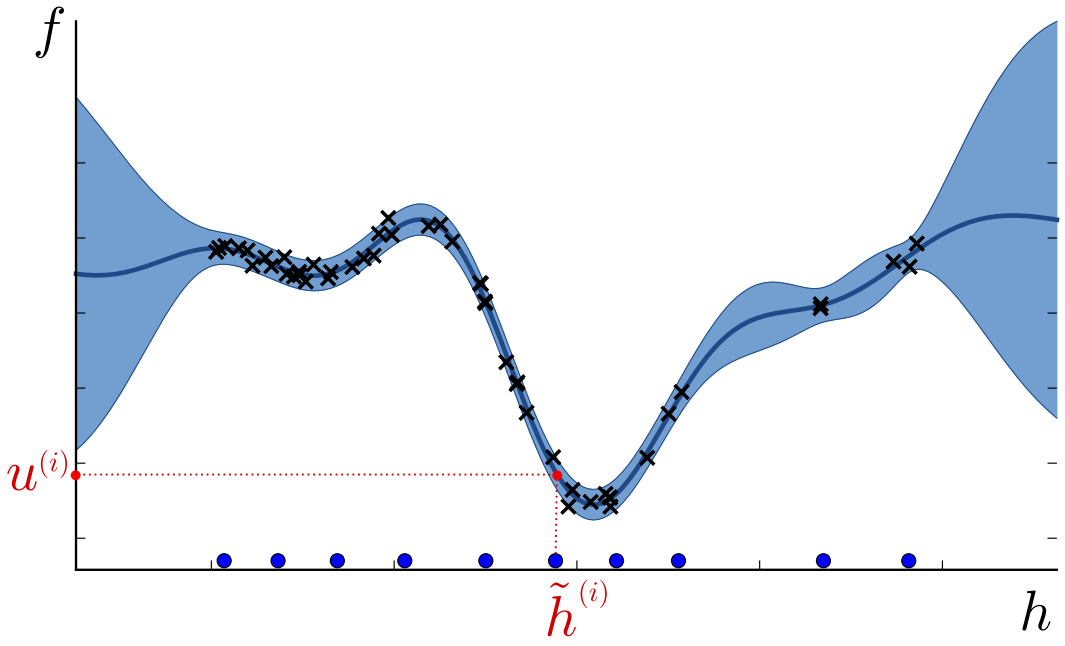
Inducing points: sparseness, tractability and Big Data

$h^{(1)}$	$\mathbf{f}^{(1)}$
$h^{(2)}$	$\mathbf{f}^{(2)}$
$\dots$	$\dots$
$h^{(30)}$	$\mathbf{f}^{(30)}$
$h^{(31)}$	$\mathbf{f}^{(31)}$
$\dots$	$\dots$
$h^{(N)}$	$\mathbf{f}^{(N)}$



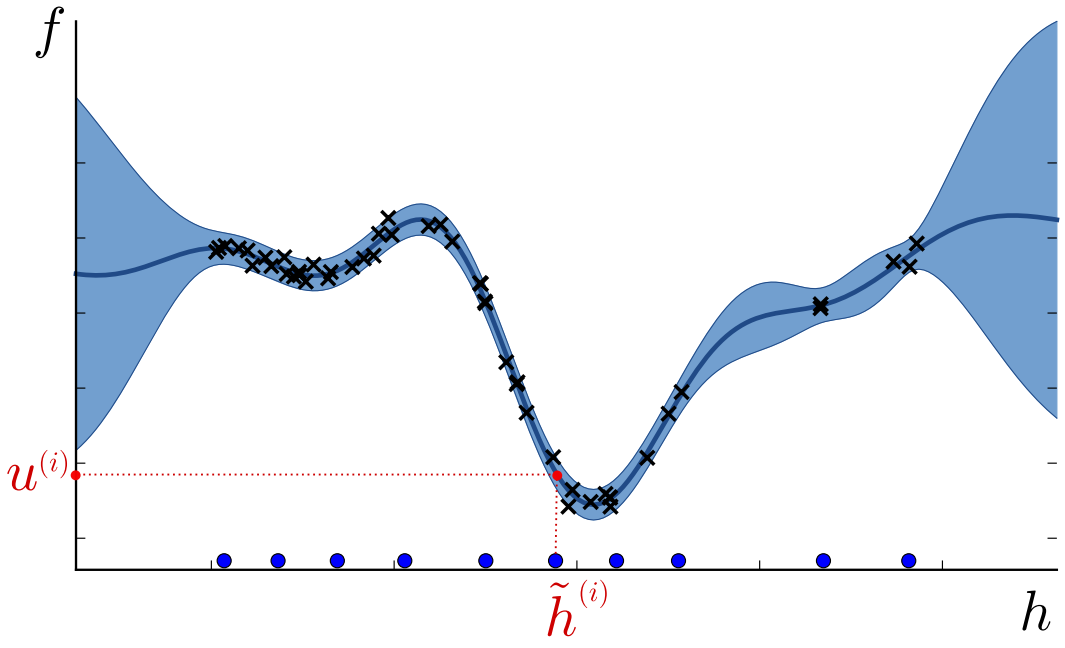
Inducing points: sparseness, tractability and Big Data

$h^{(1)}$	$\mathbf{f}^{(1)}$
$h^{(2)}$	$\mathbf{f}^{(2)}$
$\dots$	$\dots$
$h^{(30)}$	$\mathbf{f}^{(30)}$
$\tilde{h}^{(i)}$	$u^{(i)}$
$h^{(31)}$	$\mathbf{f}^{(31)}$
$\dots$	$\dots$
$h^{(N)}$	$\mathbf{f}^{(N)}$



Inducing points: sparseness, tractability and Big Data

$h^{(1)}$	$\mathbf{f}^{(1)}$
$h^{(2)}$	$\mathbf{f}^{(2)}$
$\dots$	$\dots$
$h^{(30)}$	$\mathbf{f}^{(30)}$
$\tilde{h}^{(i)}$	$u^{(i)}$
$h^{(31)}$	$\mathbf{f}^{(31)}$
$\dots$	$\dots$
$h^{(N)}$	$\mathbf{f}^{(N)}$



- Inducing points originally introduced for faster **(sparse) GPs**
- But this also induces **tractability** in our models, due to the conditional independencies assumed
- Viewing them as **global variables**
  $\Rightarrow$ extension to **Big Data** [Hensman et al., UAI 2013]

Bayesian regularization

$$\mathcal{F} = \text{Data Fit} \qquad \underbrace{-\mathsf{KL}\left(q(\mathbf{h}_1) \,\|\, p(\mathbf{h}_1)\right)}_{\text{Regularisation}} + \sum_{l=2}^{L} \underbrace{\mathcal{H}\left(q(\mathbf{h}_l)\right)}_{\text{Regularisation}}$$