System identification and control with (deep) Gaussian processes

Andreas Damianou

Department of Computer Science, University of Sheffield, UK

MIT, 11 Feb. 2016

Outline

Outline

General area of interest

- Dynamical systems.
- Non-linear models; models with exogenous inputs (NARX)
- Model-based, data-driven approaches for regressive and auto-regressive inference.

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.

Data-driven

Data driven: Learn from data by exploiting patterns through probabilistic modelling.

Pros:

- Complex situations where no ODEs are present etc.
- Prior probabilities can to some degree incorporate side knowledge.
- Principled handling of noise / uncertainty.
- ...

Cons:

- More difficult to incorporate mechanistic knowledge (although there's work attempting to do this).
- Relies on the way the model is optimized (local minima, approximations, computational inefficiencies, numerical problems...)
- ...

Model-based approach

Mechanistic model challenge: create a model which is as simple as possible but also as close to reality as possible.

Probabilistic model challenge: enrich the statistical properties of the model: robustness to outliers, representation learning (e.g. deep, auto-encoders)

Why Model with Gaussian process

- Uncertainty quantification
- learn from few data
- attractive analytical properties
- Bayesian framework: make modelling assumptions explicit.

Cool stuff you can do with GPs #1

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

Outline

Introducing Gaussian Processes:

- A Gaussian distribution depends on a mean and a covariance matrix.
- A Gaussian process depends on a mean and a covariance function.

Infinite model... but we *always* work with finite sets!

Let's start with a multivariate Gaussian:

$$p(\underbrace{f_1, f_2, \cdots, f_s}_{\mathbf{f}_A}, \underbrace{f_{s+1}, f_{s+2}, \cdots, f_N}_{\mathbf{f}_B}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}).$$

with:

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{bmatrix} \text{ and } \mathbf{K} = \begin{bmatrix} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ \mathbf{K}_{BA} & \mathbf{K}_{BB} \end{bmatrix}$$

Marginalisation property:

$$p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}). \quad \text{Then:}$$

$$p(\mathbf{f}_A) = \int_{\mathbf{f}_B} p(\mathbf{f}_A, \mathbf{f}_B) \mathrm{d}\mathbf{f}_B = \mathcal{N}(\boldsymbol{\mu}_A, \mathbf{K}_{AA})$$

Infinite model... but we *always* work with finite sets!

Let's start with a multivariate Gaussian:

$$p(\underbrace{f_1, f_2, \cdots, f_s}_{\mathbf{f}_A}, \underbrace{f_{s+1}, f_{s+2}, \cdots, f_N}_{\mathbf{f}_B}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}).$$

with:

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{bmatrix} \text{ and } \mathbf{K} = \begin{bmatrix} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ \mathbf{K}_{BA} & \mathbf{K}_{BB} \end{bmatrix}$$

Marginalisation property:

$$p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}). \quad \text{Then:}$$

$$p(\mathbf{f}_A) = \int_{\mathbf{f}_B} p(\mathbf{f}_A, \mathbf{f}_B) \mathrm{d}\mathbf{f}_B = \mathcal{N}(\boldsymbol{\mu}_A, \mathbf{K}_{AA})$$

Infinite model... but we *always* work with finite sets!

In the GP context:

$$\boldsymbol{\mu}_\infty = \begin{bmatrix} \boldsymbol{\mu}_{\mathbf{x}} \\ \cdots \\ \cdots \end{bmatrix} \quad \text{and} \quad \mathbf{K}_\infty = \begin{bmatrix} \mathbf{K}_{\mathbf{xx}} & \cdots \\ \cdots & \cdots \end{bmatrix}$$

Posterior is also Gaussian!

$p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$. Then:

$$p(\mathbf{f}_A|\mathbf{f}_B) = \mathcal{N}(\boldsymbol{\mu}_A + \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}(\mathbf{f}_B - \boldsymbol{\mu}_B), \mathbf{K}_{AA} - \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}\mathbf{K}_{BA})$$

In the GP context this can be used for inter/extrapolation:

$$f_*|f_1, \cdots, f_N \sim \mathcal{GP}_{\text{post}}$$

$$p(f_*|f_1, \cdots, f_N) = p(f(x_*)|f(x_1), \cdots, f(x_N))$$

$$\sim \mathcal{N}(\mathbf{K}_*^\top \mathbf{K}^{-1}\mathbf{f}\ ,\ \mathbf{K}_{*,*} - \mathbf{K}_*^\top \mathbf{K}^{-1}\mathbf{K}_*)$$

But where is $\mathbf{K}_{\cdot,\cdot}$ coming from in GPs?

Posterior is also Gaussian!

$p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$. Then:

$$p(\mathbf{f}_A|\mathbf{f}_B) = \mathcal{N}(\boldsymbol{\mu}_A + \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}(\mathbf{f}_B - \boldsymbol{\mu}_B), \mathbf{K}_{AA} - \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}\mathbf{K}_{BA})$$

In the GP context this can be used for inter/extrapolation:

$$\begin{aligned} f_*|f_1, \cdots, f_N &\sim \mathcal{GP}_{\text{post}} \\ p(f_*|f_1, \cdots, f_N) &= p(f(x_*)|f(x_1), \cdots, f(x_N)) \\ &\sim \mathcal{N}(\mathbf{K}_*^\top \mathbf{K}^{-1}\mathbf{f}\ ,\ \mathbf{K}_{*,*} - \mathbf{K}_*^\top \mathbf{K}^{-1}\mathbf{K}_*) \end{aligned}$$

But where is $\mathbf{K}_{..}$ coming from in GPs?

Posterior is also Gaussian!

$p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$. Then:

$$p(\mathbf{f}_A | \mathbf{f}_B) = \mathcal{N}(\boldsymbol{\mu}_A + \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}(\mathbf{f}_B - \boldsymbol{\mu}_B), \mathbf{K}_{AA} - \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}\mathbf{K}_{BA})$$

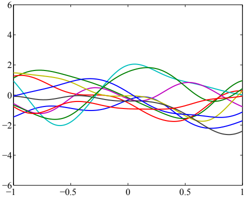
In the GP context this can be used for inter/extrapolation:

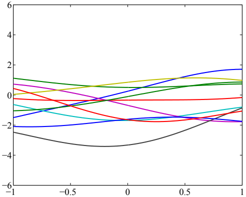
$$\begin{aligned} f_* | f_1, \cdots, f_N &\sim \mathcal{GP}_{\text{post}} \\ p(f_* | f_1, \cdots, f_N) &= p(f(x_*) | f(x_1), \cdots, f(x_N)) \\ &\sim \mathcal{N}(\mathbf{K}_*^\top \mathbf{K}^{-1} \mathbf{f} \,,\ \mathbf{K}_{*,*} - \mathbf{K}_*^\top \mathbf{K}^{-1} \mathbf{K}_*) \end{aligned}$$

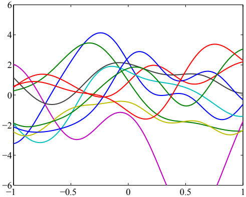
But where is $\mathbf{K}_{\cdot\cdot}$ coming from in GPs?

Covariance samples and hyperparameters

- $k(x, x') = \alpha \exp\left(-\frac{\gamma}{2}(x - x')^\top (x - x')\right)$
- The hyperparameters of the cov. function define the properties (and NOT an explicit form) of the sampled functions





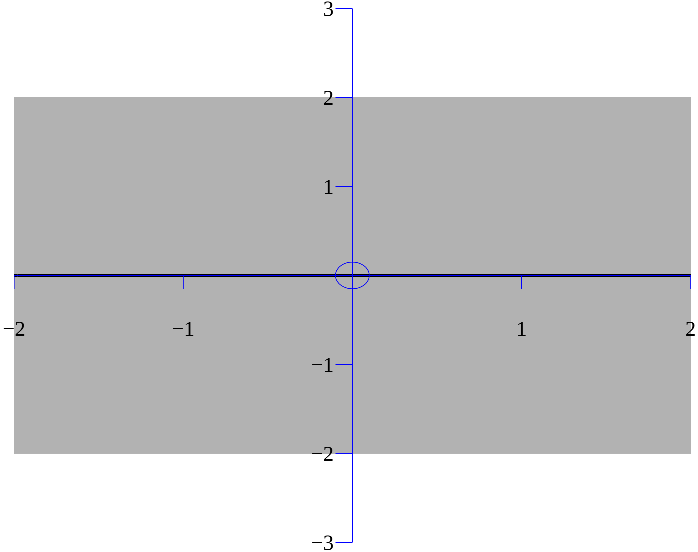


Incorporating Gaussian noise is tractable

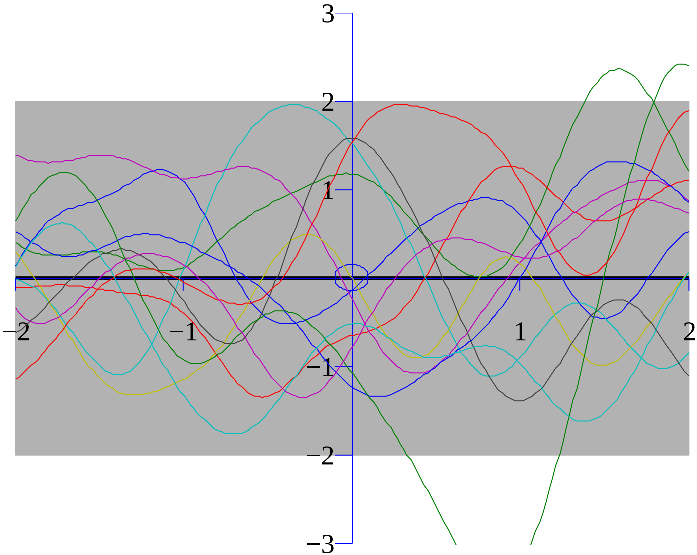
- So far we assumed: $\mathbf{f} = f(\mathbf{X})$
- Assuming that we only observe noisy versions $\mathbf{y}$ of the true outputs $\mathbf{f}$:

$$\mathbf{y} = f(\mathbf{X}) + \epsilon, \;\; \epsilon \sim \mathcal{N}(0, \sigma^2)$$

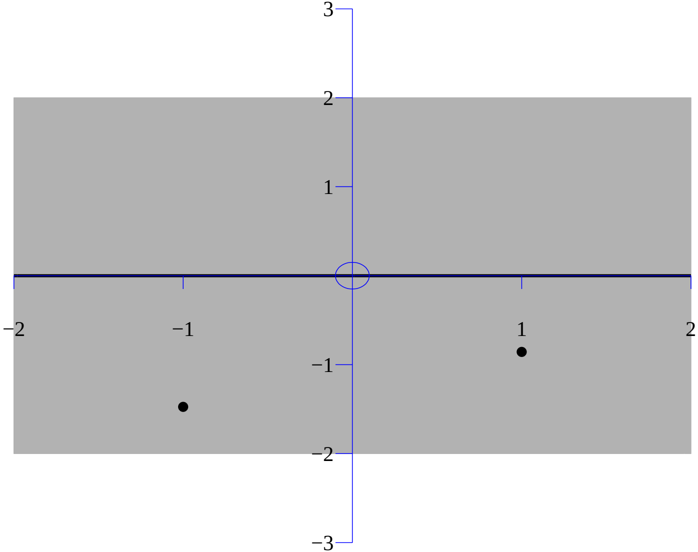
Fitting the data *(shaded area is **uncertainty**)*



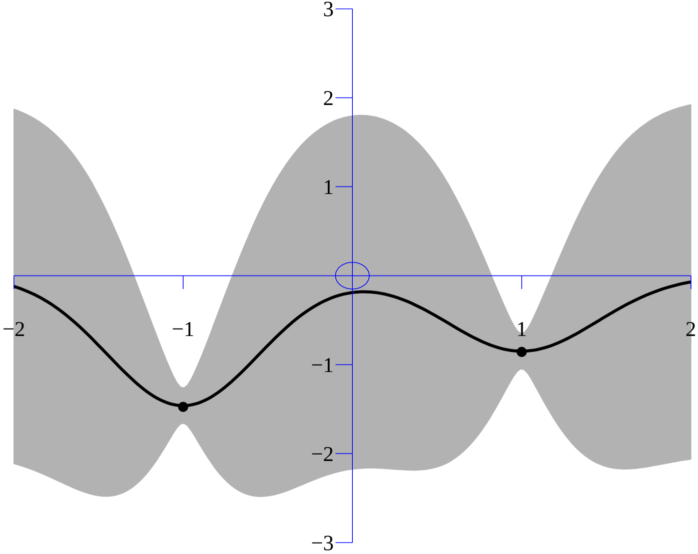
Fitting the data - Prior Samples



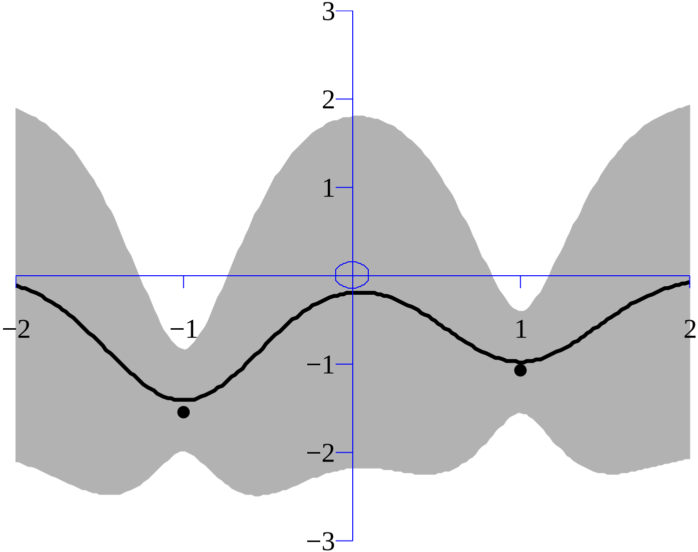
Fitting the data



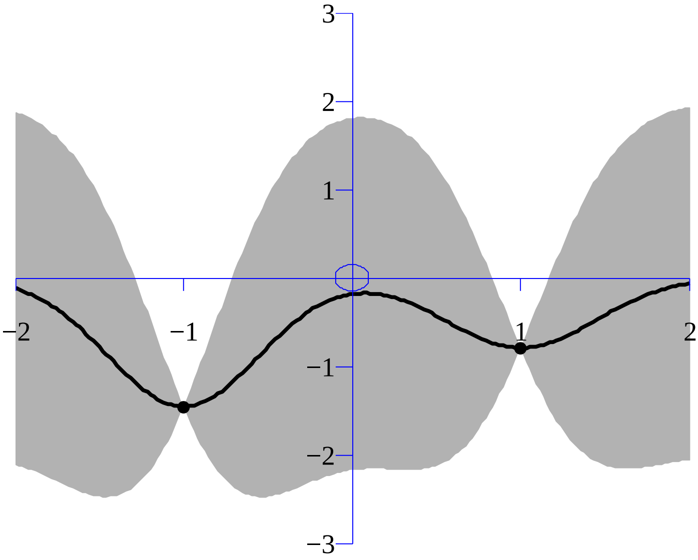
Fitting the data



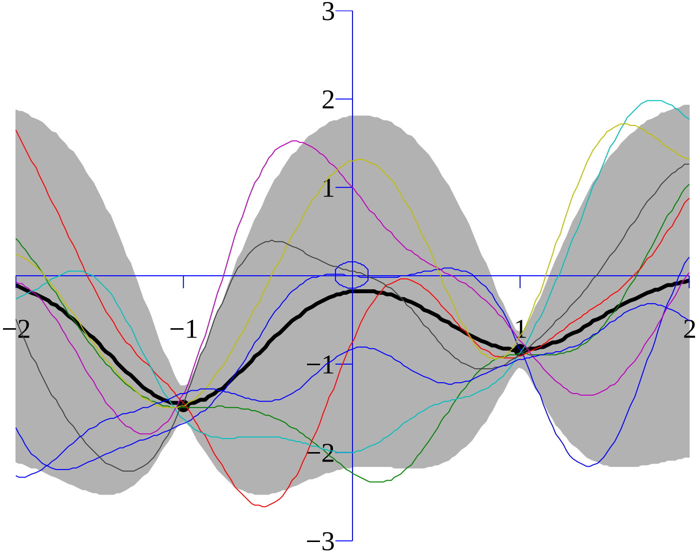
Fitting the data - more noise



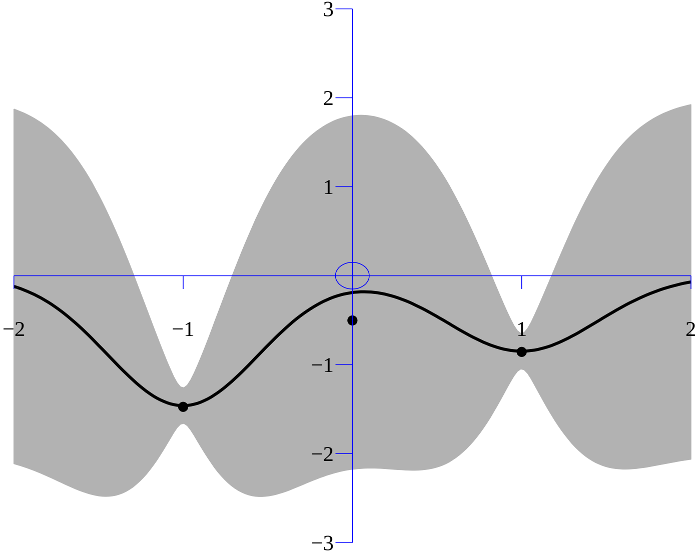
Fitting the data - no noise



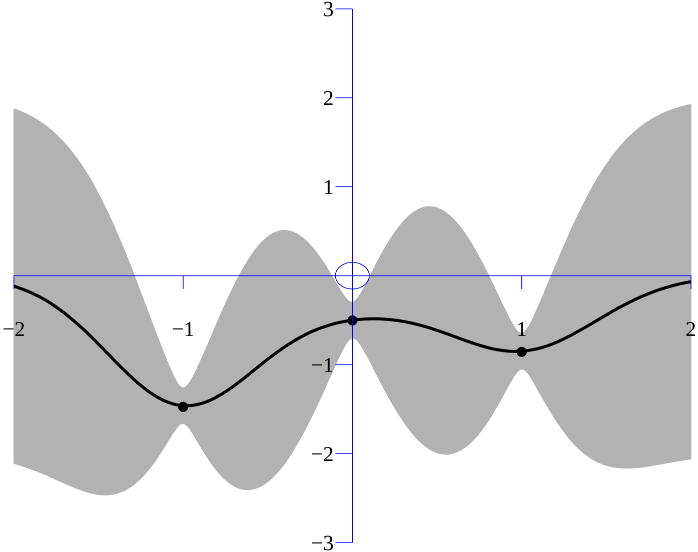
Fitting the data - Posterior samples



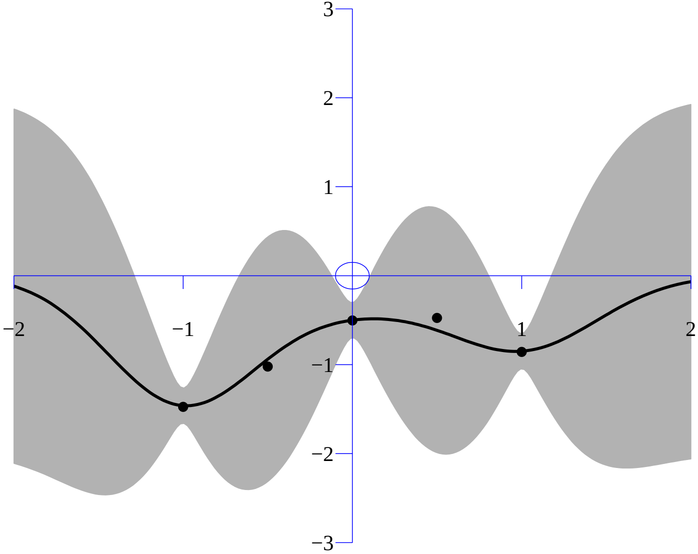
Fitting the data



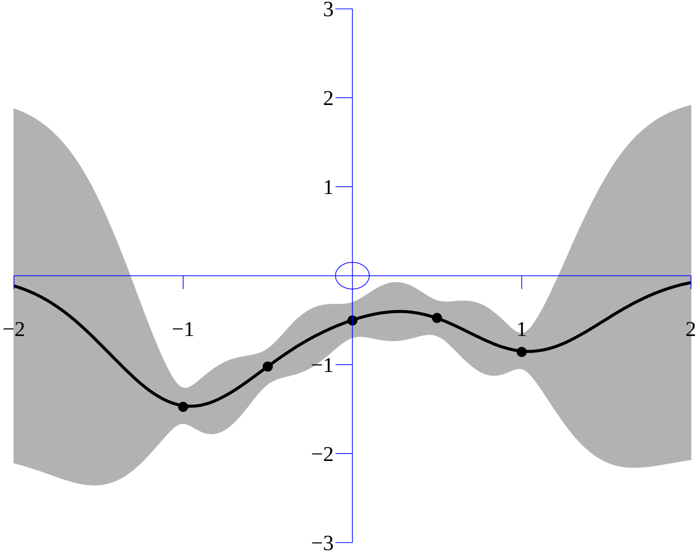
Fitting the data



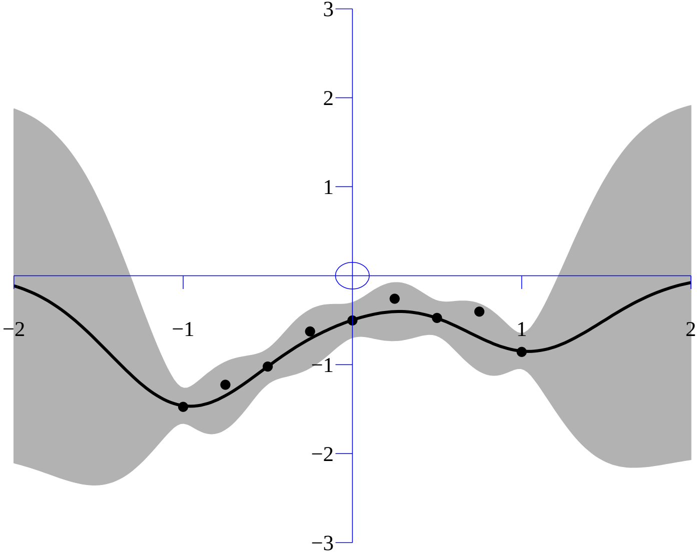
Fitting the data



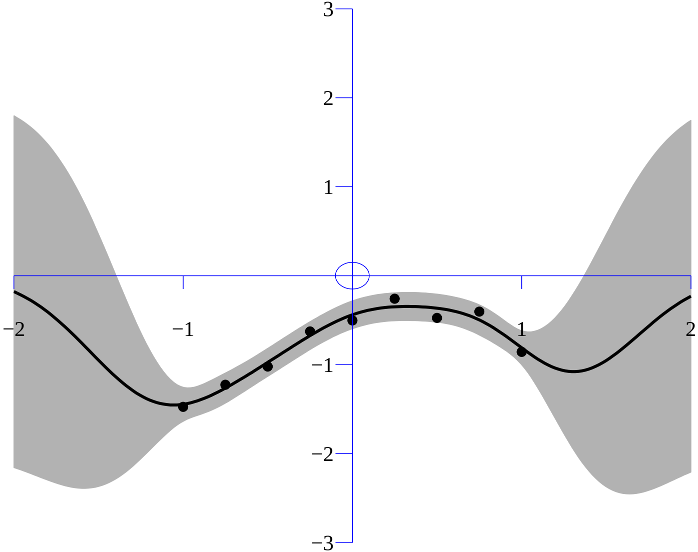
Fitting the data



Fitting the data



Fitting the data



Part 1: 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
- Principled handling of uncertainty

Outline

Part 1: Introduction

Part 2: Gaussian processes
- GPs as infinite dimensional Gaussian distributions

Part 3: In practice
- Autoregressive Dynamics
- Going deeper: Deep Recurrent Gaussian Process
- Regressive dynamics with deep GPs

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

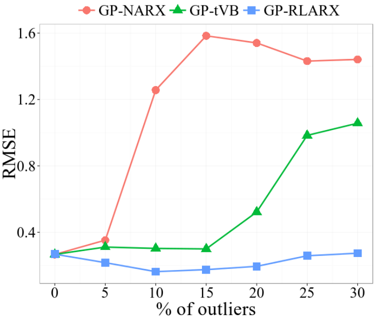
Latent auto-regressive GP model:

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

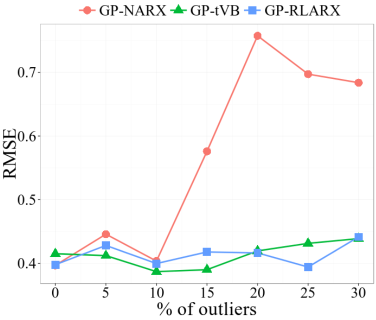
Contribution 3: "Switching-off" outliers by including the above Student-t likelihood for the noise.

Mattos, Damianou, Barreto, Lawrence, 2016

Robust GP autoregressive model: demonstration

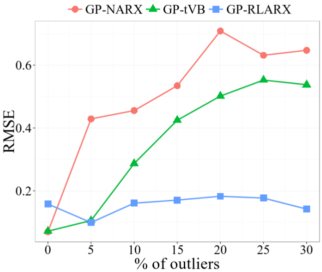


(a) Artificial 1.

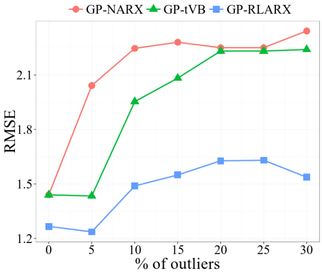


(b) Artificial 2.

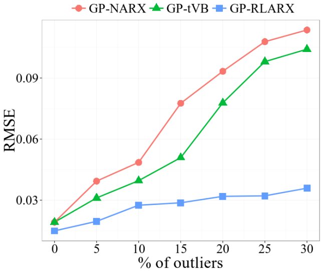
Figure: RMSE values for free simulation on test data with different levels of contamination by outliers.



(c) Artificial 3.



(d) Artificial 4.



(e) Artificial 5.

Going deeper: Deep Recurrent Gaussian Process

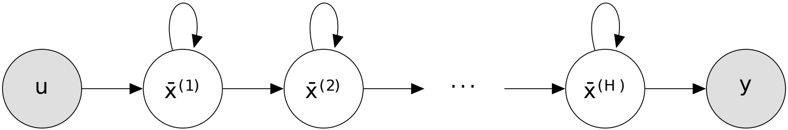


Figure 1: RGP graphical model with H hidden layers.

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

Mattos, Dai, Damianou, Barreto, Lawrence, 2016

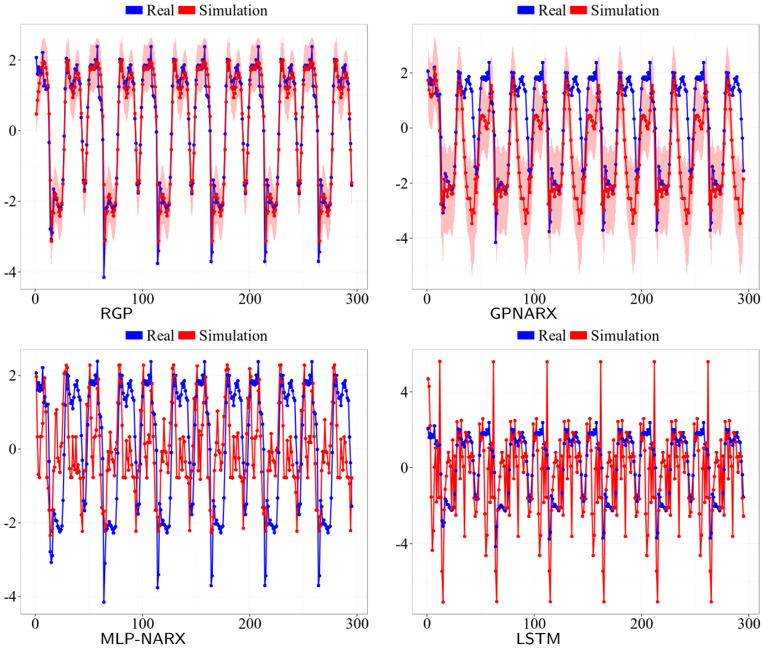
Inference is tricky...

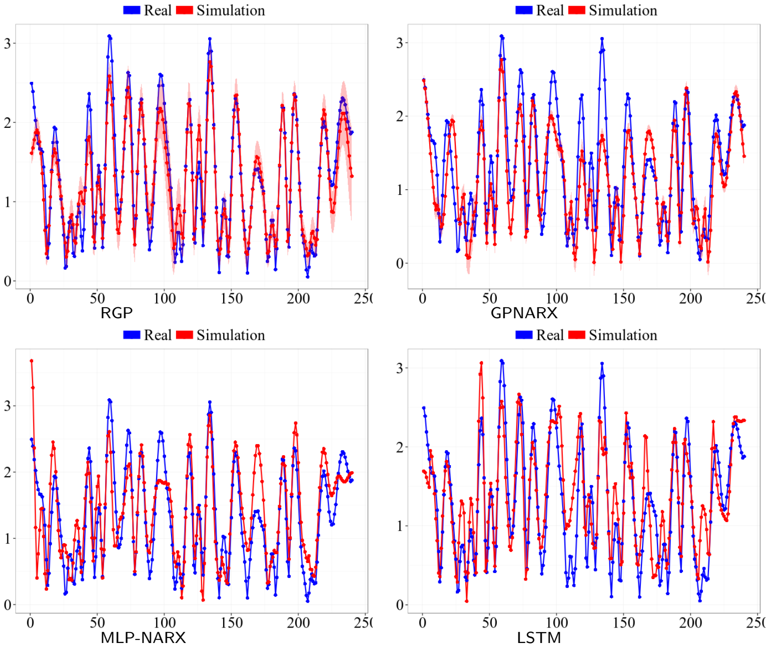
$$
\begin{aligned}
\log p(\mathbf{y}) \geq & -\frac{N-L}{2}\sum_{h=1}^{H+1}\log 2\pi\sigma_h^2 - \frac{1}{2\sigma_{H+1}^2}\Big(\mathbf{y}^\top\mathbf{y} + \Psi_0^{(H+1)} \\
& - \mathsf{Tr}\left(\left(\mathbf{K}_z^{(H+1)}\right)^{-1}\mathbf{\Psi}_2^{(H+1)}\right)\Big) + \frac{1}{2}\left|\mathbf{K}_z^{(H+1)}\right|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

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
0
1
2
3
0
50
100
150
200
250

Avatar control

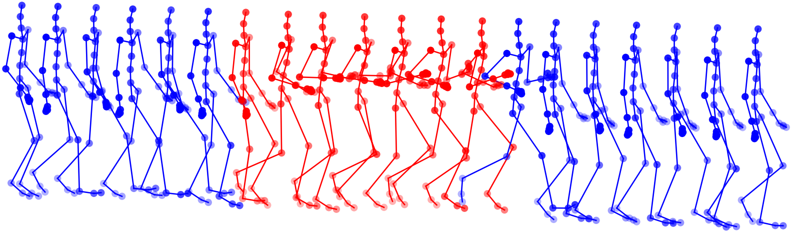


Figure: The generated motion with a step function signal, starting with walking (blue), switching to running (red) and switching back to walking (blue).

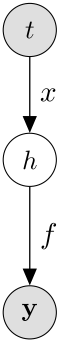
Videos:

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



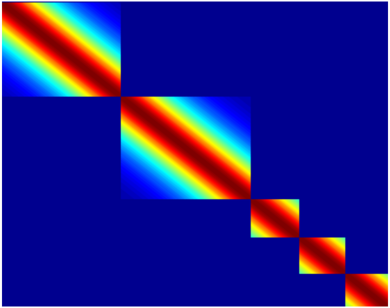
Instead of coupling f's by encoding the Markov property, we couple them by **coupling the f's inputs through another GP with time as input.**

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

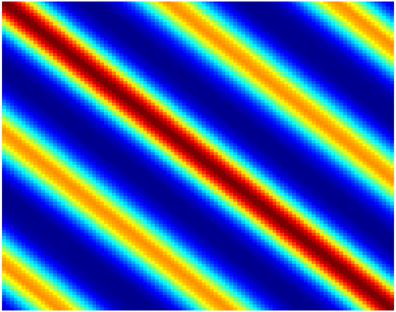
Damianou, Titsias, Lawrence, 2011

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 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 my co-authors: Neil Lawrence, Cesar Lincoln Mattos, Zhenwen Dai, Javier Gonzalez, Guilheme Barreto

Thanks to George Karniadakis, Themis Sapsis, Paris Perdikaris for hosting me