Latent Variables, Differential Equations and Gaussian Processes

Neil Lawrence
School of Computer Science
University of Manchester

12th November 2007

Outline

1. Latent Variable Models
   - Data Representation Example

2. Dynamics and Latent Variable Models
   - Markov Assumptions

3. Differential Equations
   - Linear Differential Equation and GP
   - Transcription Factor Concentrations
   - Non-linear Response Model

4. Acknowledgements

Online Resources

- Source code and slides are available online
- This talk available from home page (see talks link on side).
- Scripts available in the 'gpsim' toolbox
    - `http://www.cs.man.ac.uk/~neill/gpsim/`.
- MATLAB commands used for examples given in `typewriter font`.

High Dimensional Data

- High dimensional data: curse of dimensionality. Does it exist?
- Only if data is *inherently* high dimensional.
- In practice most data ‘lives’ on a lower dimensional space.
- Latent variable models allow us to capture the structure of such data.

Stick Man

- Example: GP-LVM [Lawrence, 2004, 2005]
    - Non-linear dimensional reduction using Gaussian processes.
- Powerful uncertainly handling of GPs leads to suprising properties.
    - Non-linear models can be used where there are fewer data points than dimensions *without overfitting*.
    - Example: Modelling a stick man in 102 dimensions with 55 data points

Stick Man II

`demStick1`

Figure: The latent space for the stick man motion capture data.

Stick Man II

demStick1

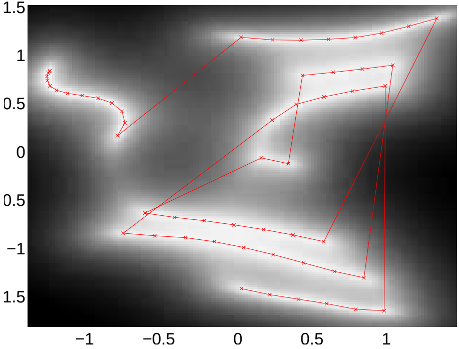


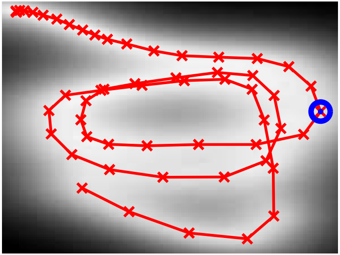
Figure: The latent space for the stick man motion capture data.

MAP Solutions for Dynamics Models

- Data often has a temporal ordering.
- Markov-based dynamics are often used.
- For the GP-LVM
  - Marginalising such dynamics is intractable.
  - *But*: MAP solutions are trivial to implement.
- Many choices: Kalman filter, Markov chains *etc*..
- Wang et al. [2006] suggest using an autoregressive Gaussian Process.
  - This has been applied in the context of tracking models [Urtasun et al., 2006].

Auto Regressive Gaussian Process Dynamics

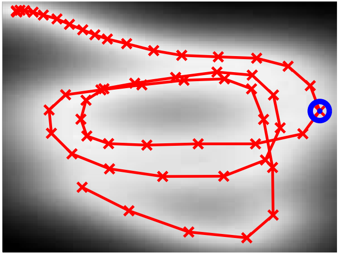
- Autoregressive Gaussian process mapping in latent space between time points.



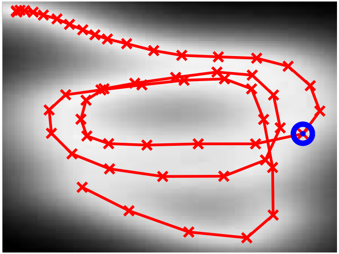
t

Auto Regressive Gaussian Process Dynamics

- Autoregressive Gaussian process mapping in latent space between time points.



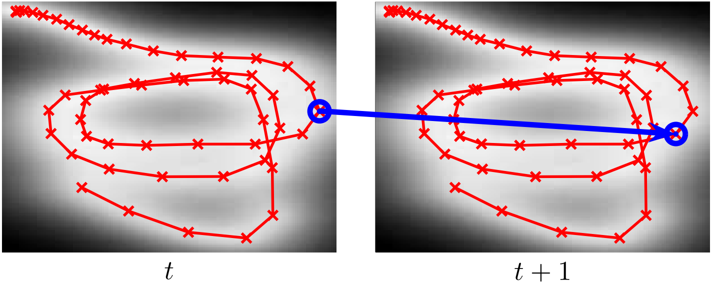
t



$t+1$

Auto Regressive Gaussian Process Dynamics

- Autoregressive Gaussian process mapping in latent space between time points.



Motion Capture Results

`demStick1` and `demStick2`

Figure: The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*right*) based on an RBF kernel.

Motion Capture Results

demStick1 and demStick2

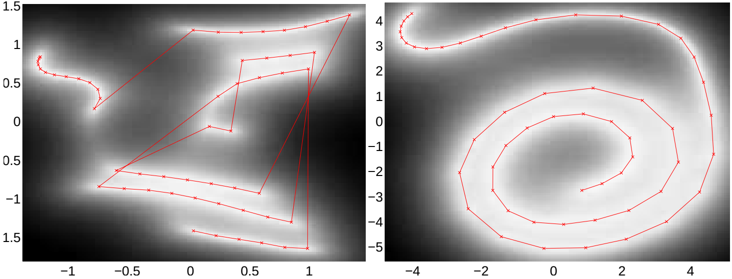
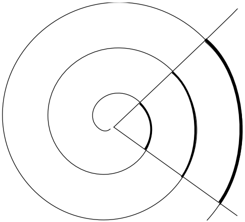


Figure: The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*right*) based on an RBF kernel.

Regressive Dynamics

Inner Groove Distortion

- Autoregressive unimodal dynamics, $p\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right)$.
- Forces spiral visualisation.
- Poorer model due to inner groove distortion.



Regressive Dynamics

- Instead of auto-regressive dynamics, consider regressive dynamics.
- Take **t** as an input, for the prior distribution over latent space, $p(\mathbf{F}|\mathbf{t})$.
- Use a Gaussian process prior for $p(\mathbf{F}|\mathbf{t})$.
  (also allows us to consider variable sample rate data).

Motion Capture Results

`demStick1`, `demStick2` and `demStick5`

Figure: The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*middle*) and with regressive dynamics (*right*) based on an RBF kernel.

Motion Capture Results

`demStick1`, `demStick2` and `demStick5`

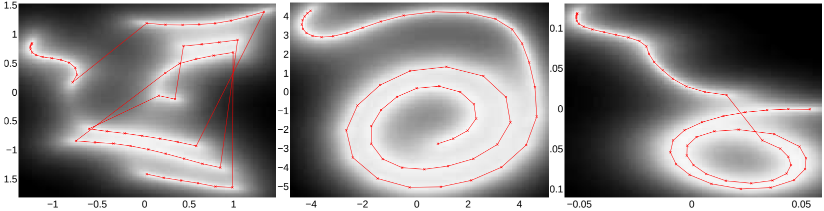


Figure: The latent space for the motion capture data without dynamics (*left*), with auto-regressive dynamics (*middle*) and with regressive dynamics (*right*) based on an RBF kernel.

Dynamic Latent Variable Model

- Defined a GP priors as function of time over the latent space.

$$f_i(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_t)$$

- GP-LVM defines a non-linear relationship the latent space and the data, **Y**.

$$y_j(t) = g_j\left(\{f_i(t)\}_{i=1}^{q}\right) + \eta. \quad \eta \sim \mathcal{N}\left(\imath, \sigma^{\in}\right)$$

- In contrast Probabilistic PCA [Tipping and Bishop, 1999, Roweis, 1998] defines a linear relationship.

$$y_j(t) = \sum_{i=1}^{q} S_{ji} f_i(t) + \eta. \quad \eta \sim \mathcal{N}\left(\imath, \sigma^{\in}\right)$$

Differential Equation Dynamics

- Alternative extension. Instead of 'nonlinearising', introduce dynamics explicitly.

$$m_j \frac{\mathrm{d}y_j^2\left(t\right)}{\mathrm{d}t} + C_j \frac{\mathrm{d}y_j\left(t\right)}{\mathrm{d}t} + D_j y_j\left(t\right) = \sum_{i=1}^{q} S_{ji} f_i\left(t\right)$$

$$y_j(t) = \frac{1}{m_j \omega_j} \sum_{i=1}^{q} S_{ji} \exp(-\alpha_j t) \int_0^t f_i(u) \exp(\alpha_j u) \sin(\omega_j (t-u)) \mathrm{d}u$$

where

$$\alpha_k = \frac{C_k}{2m_k}, \quad \omega_k^2 = \frac{D_k}{m_k} - \alpha_k^2,$$

this can be written

$$y_j(t) = \sum_{i=1}^{q} L_{ij}[f_i](t)$$

- If we model $f\left(t\right)$ as a GP then as (2) only involves linear operations $x_j\left(t\right)$ is also a GP.

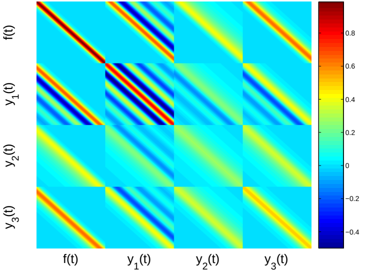
Covariance for Transcription Model

- RBF Kernel function for $f\left(t\right)$

$$y_j(t) = \frac{1}{m_j\omega_j}\sum_{i=1}^{q} S_{ji}\exp(-\alpha_j t)\int_0^t f_i(u)\exp(\alpha_j u)\sin(\omega_j(t-u))\mathrm{d}u$$

- Joint distribution for $y_1\left(t\right)$, $y_2\left(t\right)$, $y_3\left(t\right)$ and $f\left(t\right)$. Damping ratios:

ζ_1	ζ_2	ζ_3
0.125	2	1



Joint Sampling of $x(t)$ and $f(t)$

- `demLfmSample`

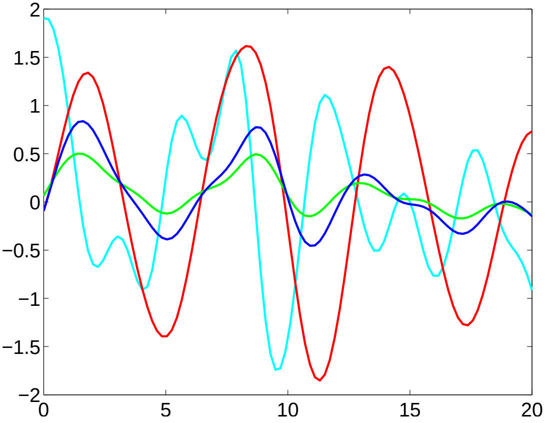


Figure: Joint samples from the ODE covariance, *cyan*: $f(t)$, *red*: $y_1(t)$(underdamped) and *green*: $y_2(t)$ (overdamped) and *blue*: $y_3(t)$ (critically damped).

Joint Sampling of $x(t)$ and $f(t)$

- `demLfmSample`

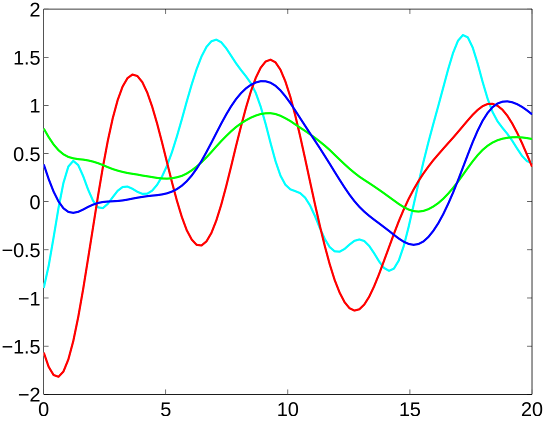


Figure: Joint samples from the ODE covariance, *cyan*: $f(t)$, *red*: $y_1(t)$(underdamped) and *green*: $y_2(t)$ (overdamped) and *blue*: $y_3(t)$ (critically damped).

Joint Sampling of $x(t)$ and $f(t)$

- `demLfmSample`

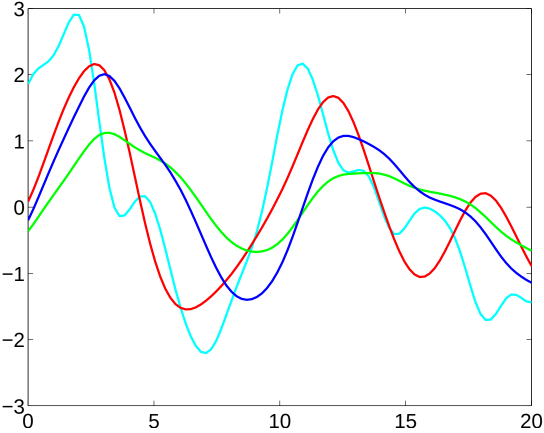


Figure: Joint samples from the ODE covariance, *cyan*: $f(t)$, *red*: $y_1(t)$(underdamped) and *green*: $y_2(t)$ (overdamped) and *blue*: $y_3(t)$ (critically damped).

Joint Sampling of $x(t)$ and $f(t)$

- `demLfmSample`

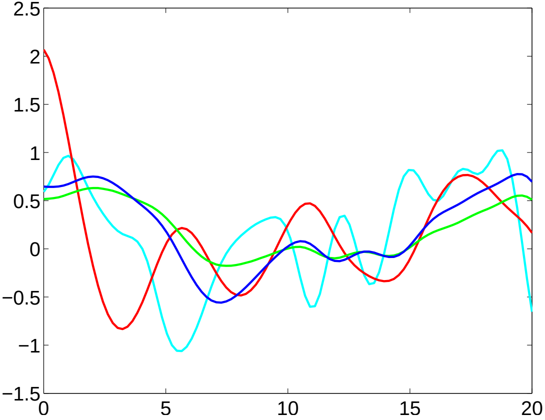


Figure: Joint samples from the ODE covariance, *cyan*: $f(t)$, *red*: $y_1(t)$(underdamped) and *green*: $y_2(t)$ (overdamped) and *blue*: $y_3(t)$ (critically damped).

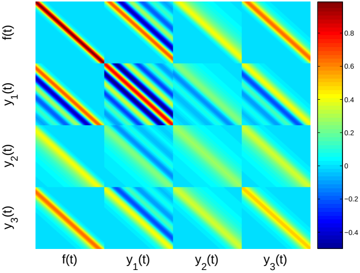
Covariance for ODE

- RBF Kernel function for $f(t)$

$$y_j(t) = \frac{1}{m_j\omega_j}\sum_{i=1}^{q} S_{ji}\exp(-\alpha_j t)\int_0^t f_i(u)\exp(\alpha_j u)\sin(\omega_j(t-u))\mathrm{d}u$$

- Joint distribution for $y_1(t)$, $y_2(t)$, $y_3(t)$ and $f(t)$.
- Damping ratios:

ζ_1	ζ_2	ζ_3
0.125	2	1



Linear Differential Equation Model

- We will now focus on a protein concentration example.
  - Gaussian processes (GPs) are probabilistic models for functions.
  - GPs provide a framework for performing inference about these functions in the presence of uncertainty.
- Data consists of T measurements of mRNA expression level for N different genes.
- We relate gene expression, $x_j(t)$, to TFC, $f(t)$, by

$$\frac{dx_j(t)}{dt} = B_j + S_j f(t) - D_j x_j(t). \tag{1}$$

B_j basal transcription rate of gene j,
S_j is sensitivity of gene j
D_j is the decay rate of the mRNA.

- Dependence of mRNA transcription rate on TF is linear.

Linear Response Solution

- Solve for TFC
  - The equation given in (1) can be solved to recover

$$x_j\left(t\right) = \frac{B_j}{D_j} + S_j \exp\left(-D_j t\right) \int_0^t f\left(u\right) \exp\left(D_j u\right) du. \tag{2}$$

Computation of Joint Covariance

- Covariance Function Computation
- We rewrite equation (2) as

$$x_j(t) = \frac{B_j}{D_j} + L_j[f](t)$$

where

$$L_j[f](t) = S_j \exp(-D_j t) \int_0^t f(u) \exp(D_j u)\, du \tag{3}$$

is a linear operator.

Induced Covariance

- Gene's Covariance
- The new covariance function is then given by

$$\mathrm{cov}\left(L_j\left[f\right](t), L_k\left[f\right](t')\right) = L_j \otimes L_k\left[k_{ff}\right]\left(t, t'\right).$$

  more explicitly

$$k_{x_j x_k}\left(t, t'\right) = S_j S_k \exp\left(-D_j t - D_k t'\right) \int_0^t \exp\left(D_j u\right) \times \int_0^{t'} \exp\left(D_k u'\right) k_{ff}\left(u, u'\right) du' du.$$

- With RBF covariance these integrals are tractable.

Covariance Result

- Covariance Result

$$k_{x_j x_k}\left(t,t'\right)=S_j S_k\frac{\sqrt{\pi}}{2}\left[h_{kj}\left(t',t\right)+h_{jk}\left(t,t'\right)\right]$$

where

$$\begin{aligned}h_{kj}\left(t',t\right)=&\frac{\exp\left(\gamma_k\right)^2}{D_j+D_k}\\&\times\left\{\exp\left[-D_k\left(t'-t\right)\right]\left[\operatorname{erf}\left(\frac{t'-t}{l}-\gamma_k\right)+\operatorname{erf}\left(\frac{t}{l}+\gamma_k\right)\right]\right.\\&\left.-\exp\left[-\left(D_k t'+D_j\right)\right]\left[\operatorname{erf}\left(\frac{t'}{l}-\gamma_k\right)+\operatorname{erf}\left(\gamma_k\right)\right]\right\}.\end{aligned}$$

Here $\gamma_k=\frac{D_k l}{2}$.

Cross Covariance

- Correlation of $y_j(t)$ and $f(t')$
  - Need the "cross-covariance" terms between $y_j(t)$ and $f(t')$, which is obtained as

$$k_{y_j f}(t, t') = S_j \exp(-D_j t) \int_0^t \exp(D_j u) k_{ff}(u, t') du. \tag{4}$$

  - For RBF we have

$$k_{y_j f}(t', t) = \frac{\sqrt{\pi} l S_j e^{2\gamma_j}}{2} \exp\left[-D_j (t' - t)\right] \left[\mathrm{erf}\left(\frac{t' - t}{l} - \gamma_j\right) + \mathrm{erf}\left(\frac{t}{l} + \gamma_j\right)\right].$$

Posterior for f

- Prediction for TFC
  - Standard Gaussian process regression techniques [see *e.g.* Rasmussen and Williams, 2006] yield

$$\langle f \rangle_{\mathrm{post}} = K_{f\mathbf{y}} K_{yy}^{-1} \mathbf{y}$$
$$K_{ff}^{\mathrm{post}} = K_{ff} - K_{f\mathbf{y}} K_{yy}^{-1} K_{yf}$$

  - Model parameters B_j, D_j and S_j estimated by type II maximum likelihood,

$$\log p\left(\mathbf{y}\right) = N\left(\mathbf{y}|\mathbf{0}, K_{\mathbf{yy}}\right)$$

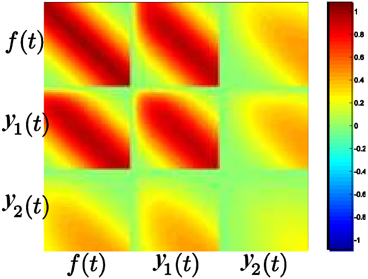
Covariance for Transcription Model

- RBF Kernel function for $f(t)$

$$y_i(t) = \frac{B_i}{D_i} + S_i \exp(-D_i t) \int_0^t f(u) \exp(D_i u)\, du.$$

- Joint distribution for $y_1(t)$, $y_2(t)$ and $f(t)$.
  - Here:

D_1	S_1	D_2	S_2
5	5	0.5	0.5



Joint Sampling of $y(t)$ and $f(t)$

- `gpsimTest`

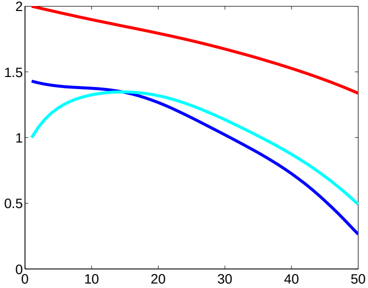


Figure: *Left*: joint samples from the transcription covariance, *blue*: $f(t)$, *cyan*: $y_1(t)$ and *red*: $y_2(t)$. *Right*: numerical solution for $f(t)$ of the differential equation from $y_1(t)$ and $y_2(t)$ (blue and cyan). True $f(t)$ included for comparison.

Joint Sampling of $y(t)$ and $f(t)$

- `gpsimTest`

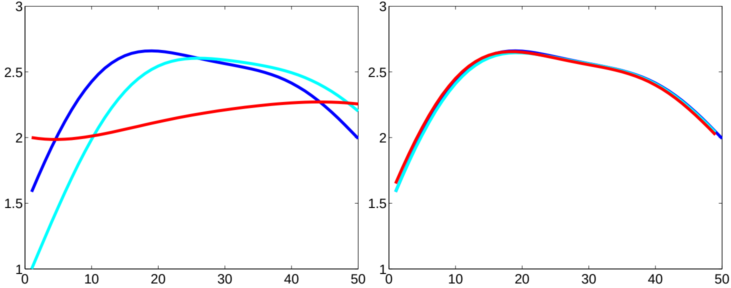


Figure: *Left*: joint samples from the transcription covariance, *blue*: $f(t)$, *cyan*: $y_1(t)$ and *red*: $y_2(t)$. *Right*: numerical solution for $f(t)$ of the differential equation from $y_1(t)$ and $y_2(t)$ (blue and cyan). True $f(t)$ included for comparison.

Joint Sampling of $y(t)$ and $f(t)$

- `gpsimTest`

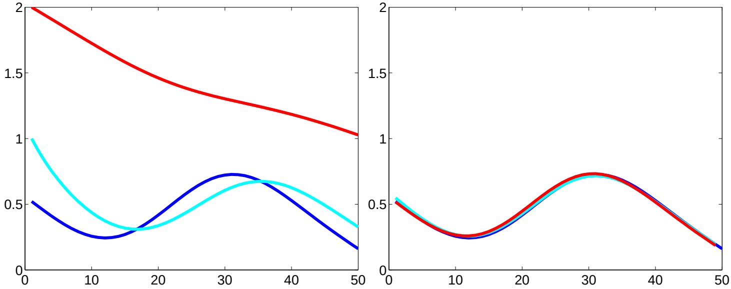


Figure: *Left*: joint samples from the transcription covariance, *blue*: $f(t)$, *cyan*: $y_1(t)$ and *red*: $y_2(t)$. *Right*: numerical solution for $f(t)$ of the differential equation from $y_1(t)$ and $y_2(t)$ (blue and cyan). True $f(t)$ included for comparison.

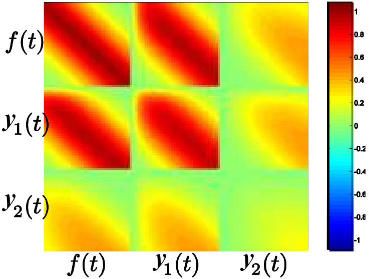
Covariance for Transcription Model

- RBF Kernel function for $f\left(t\right)$

$$y_i\left(t\right) = \frac{B_i}{D_i} + S_i \exp\left(-D_i t\right) \int_0^t f\left(u\right) \exp\left(D_i u\right) du.$$

- Joint distribution for $y_1\left(t\right)$, $y_2\left(t\right)$ and $f\left(t\right)$.
  - Here:

D_1	S_1	D_2	S_2
5	5	0.5	0.5



Overview

- Application Background
  - Understanding of cellular processes is improving through microarrays, chromatin immunoprecipitation *etc.*.
  - Quantitative description of regulatory mechanisms requires:
    - transcription factor (TF) concentrations,
    - gene-specific constants such as the baseline expression, mRNA decay rate and sensitivity to TF concentrations (TFCs).

Example Application

- Transcription Factor Inference
- Model the dynamics of gene transcription.
- Infer transcription factor concentration (TFC).
- Infer parameters of the transcription model (decay rates, sensitivities *etc.*)
- Infer these quantities using a set of known target genes.

Methodology

- Treat TFC as a *latent function* in a differential equation model.
- Assume a Gaussian process (GP) prior distribution for the latent function.
- Derive GP covariance jointly for genes and transcription factor.
- Maximise likelihood with respect to parameters (mostly physically meaningful).
    - These quantities are hard to *measure directly*.
- They can be *inferred* using a systems biology model and Gaussian processes (GPs).

Gaussian Processes

GP Advantages

- GPs allow for inference of continuous profies, accounting naturally for temporal structure.
  - GPs avoid cumbersome interpolation to estimate mRNA production rates.
- GPs deal consistently with the uncertainty inherent in the measurements.
- GPs outstrip MCMC for computational efficiency.
- *Note*: GPs have previously been proposed for solving differential equations [Graepel, 2003] and in dynamical systems [Murray-Smith and Pearlmutter].

Noise Corruption

Estimate Underlying Noise

- Allow the mRNA abundance of each gene at each time point to be corrupted by noise, for observations at t_i for $i = 1, \ldots, T$,

$$y_j(t_i) = x_j(t_i) + \epsilon_j(t_i) \tag{5}$$

with $\epsilon_j(t_i) \sim \mathcal{N}\left(0, \sigma^2_{ji}\right)$.

  - Estimate noise level using probe-level processing techniques of Affymetrix microarrays (*e.g.* mmgMOS, [Liu et al., 2005]).
  - The covariance of the noisy process is then $K_{\mathbf{yy}} = \Sigma + K_{\mathbf{xx}}$, with $\Sigma = \mathrm{diag}\left(\sigma^2_{11}, \ldots, \sigma^2_{1T}, \ldots, \sigma^2_{N1}, \ldots, \sigma^2_{NT}\right)$.

Artificial Data

- Results from an artificial data set.
- We used a 'known TFC' and derived six 'mRNA profiles'.
  - Known TFC composed of three Gaussian basis functions.
  - mRNA profiles derived analytically.
- Fourteen subsamples were taken and corrupted by noise.
- This 'data' was then used to infer a distribution over plausible TFCs.

Artificial Data Results

`demToyProblem1`

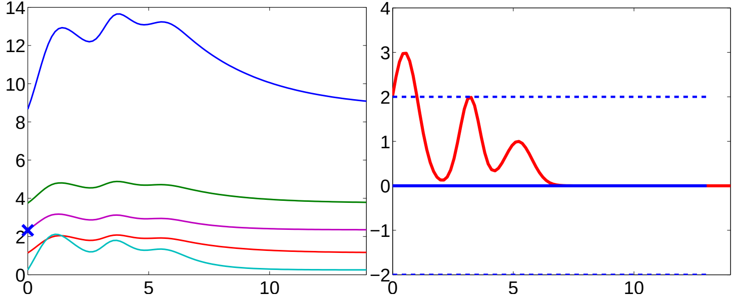


Figure: *Left:* The TFC, $f(t)$, which drives the system. *Middle:* Five gene mRNA concentration profiles each obtained by using different parameter sets $\{B_i, S_i, D_i\}_{i=1}^{5}$ (lines) along with noise corrupted 'data' . *Right:* The inferred TFC (with error bars).

Artificial Data Results

`demToyProblem1`

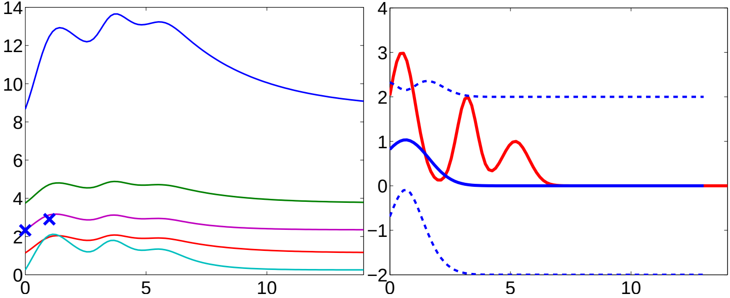


Figure: *Left:* The TFC, $f(t)$, which drives the system. *Middle:* Five gene mRNA concentration profiles each obtained by using different parameter sets $\{B_i, S_i, D_i\}_{i=1}^{5}$ (lines) along with noise corrupted 'data' . *Right:* The inferred TFC (with error bars).

Artificial Data Results

`demToyProblem1`

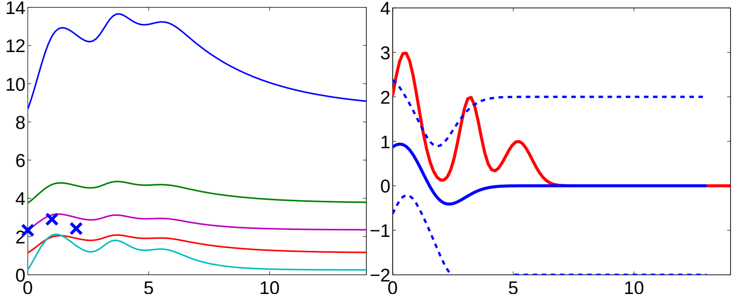


Figure: *Left:* The TFC, $f(t)$, which drives the system. *Middle:* Five gene mRNA concentration profiles each obtained by using different parameter sets $\{B_i, S_i, D_i\}_{i=1}^{5}$ (lines) along with noise corrupted 'data' . *Right:* The inferred TFC (with error bars).

Artificial Data Results

`demToyProblem1`

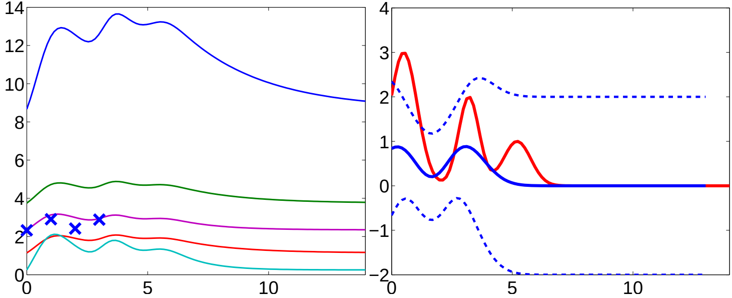


Figure: *Left:* The TFC, $f(t)$, which drives the system. *Middle:* Five gene mRNA concentration profiles each obtained by using different parameter sets $\{B_i, S_i, D_i\}_{i=1}^{5}$ (lines) along with noise corrupted 'data' . *Right:* The inferred TFC (with error bars).

Artificial Data Results

`demToyProblem1`

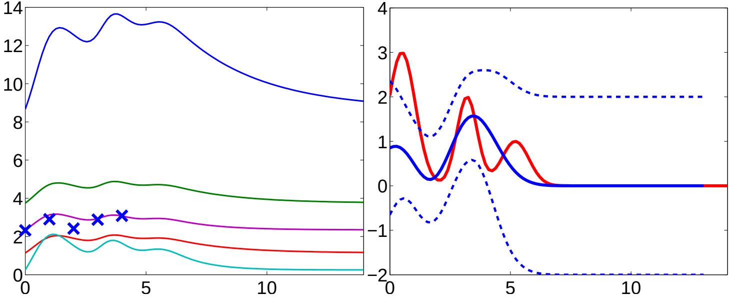


Figure: *Left:* The TFC, $f(t)$, which drives the system. *Middle:* Five gene mRNA concentration profiles each obtained by using different parameter sets $\{B_i, S_i, D_i\}_{i=1}^{5}$ (lines) along with noise corrupted 'data' . *Right:* The inferred TFC (with error bars).

Artificial Data Results

`demToyProblem1`

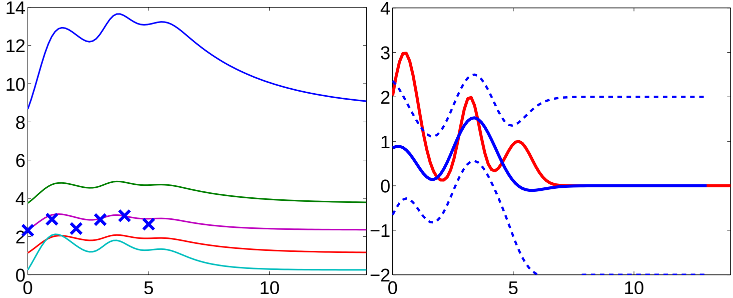


Figure: *Left:* The TFC, $f(t)$, which drives the system. *Middle:* Five gene mRNA concentration profiles each obtained by using different parameter sets $\{B_i, S_i, D_i\}_{i=1}^{5}$ (lines) along with noise corrupted 'data' . *Right:* The inferred TFC (with error bars).

Artificial Data Results

demToyProblem1

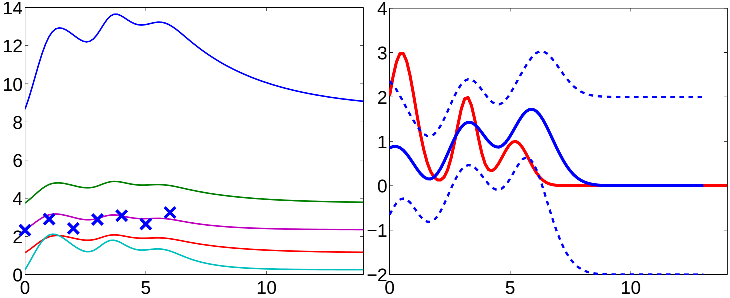


Figure: *Left:* The TFC, $f(t)$, which drives the system. *Middle:* Five gene mRNA concentration profiles each obtained by using different parameter sets $\{B_i, S_i, D_i\}_{i=1}^{5}$ (lines) along with noise corrupted 'data' . *Right:* The inferred TFC (with error bars).

Artificial Data Results

`demToyProblem1`

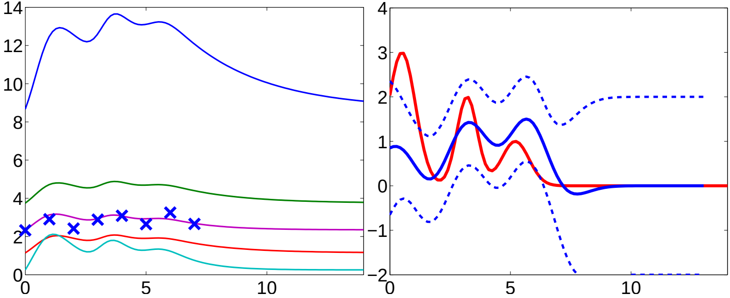


Figure: *Left:* The TFC, $f(t)$, which drives the system. *Middle:* Five gene mRNA concentration profiles each obtained by using different parameter sets $\{B_i, S_i, D_i\}_{i=1}^{5}$ (lines) along with noise corrupted 'data' . *Right:* The inferred TFC (with error bars).

Artificial Data Results

`demToyProblem1`

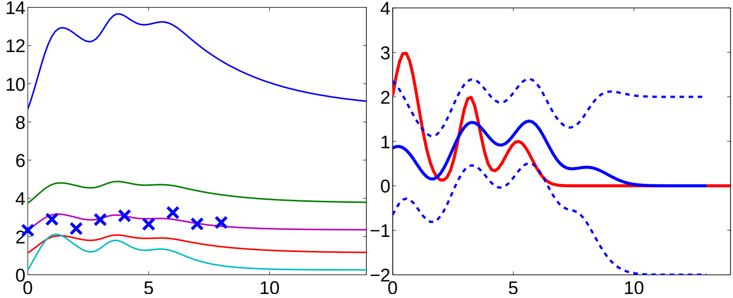


Figure: *Left:* The TFC, $f(t)$, which drives the system. *Middle:* Five gene mRNA concentration profiles each obtained by using different parameter sets $\{B_i, S_i, D_i\}_{i=1}^{5}$ (lines) along with noise corrupted 'data' . *Right:* The inferred TFC (with error bars).

Artificial Data Results

`demToyProblem1`

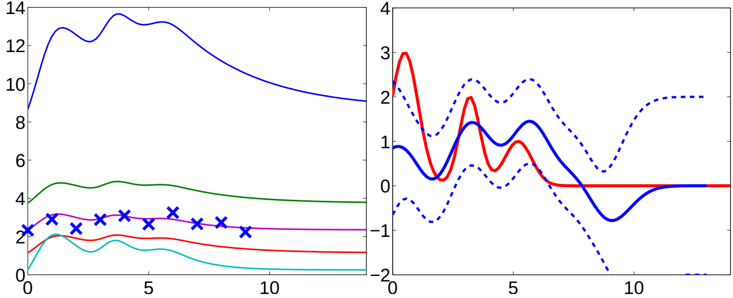


Figure: *Left:* The TFC, $f(t)$, which drives the system. *Middle:* Five gene mRNA concentration profiles each obtained by using different parameter sets $\{B_i, S_i, D_i\}_{i=1}^{5}$ (lines) along with noise corrupted 'data' . *Right:* The inferred TFC (with error bars).

Artificial Data Results

`demToyProblem1`

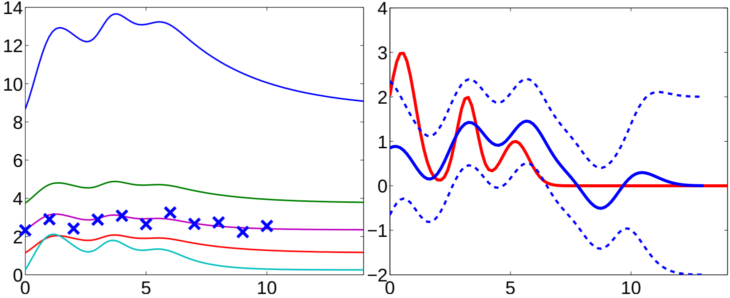


Figure: *Left:* The TFC, $f(t)$, which drives the system. *Middle:* Five gene mRNA concentration profiles each obtained by using different parameter sets $\{B_i, S_i, D_i\}_{i=1}^{5}$ (lines) along with noise corrupted 'data' . *Right:* The inferred TFC (with error bars).

Artificial Data Results

`demToyProblem1`

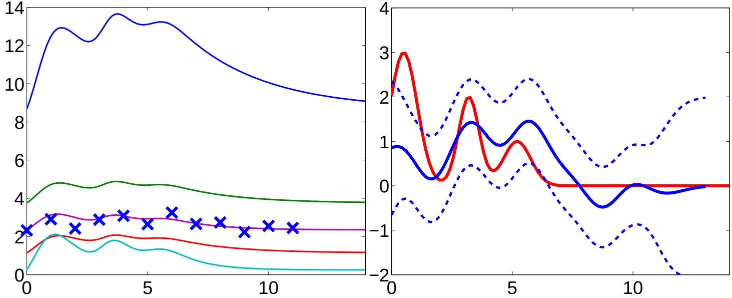


Figure: *Left:* The TFC, $f(t)$, which drives the system. *Middle:* Five gene mRNA concentration profiles each obtained by using different parameter sets $\{B_i, S_i, D_i\}_{i=1}^{5}$ (lines) along with noise corrupted 'data' . *Right:* The inferred TFC (with error bars).

Artificial Data Results

`demToyProblem1`

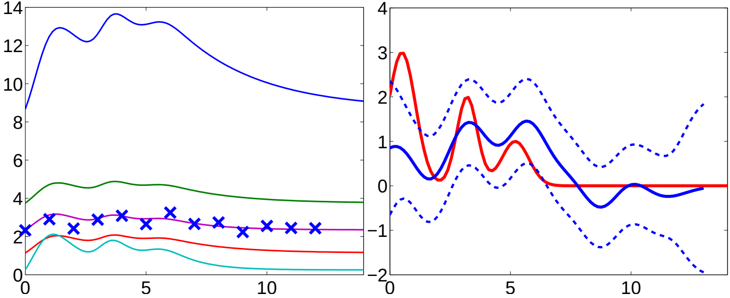


Figure: *Left:* The TFC, $f(t)$, which drives the system. *Middle:* Five gene mRNA concentration profiles each obtained by using different parameter sets $\{B_i, S_i, D_i\}_{i=1}^{5}$ (lines) along with noise corrupted 'data' . *Right:* The inferred TFC (with error bars).

Artificial Data Results

`demToyProblem1`

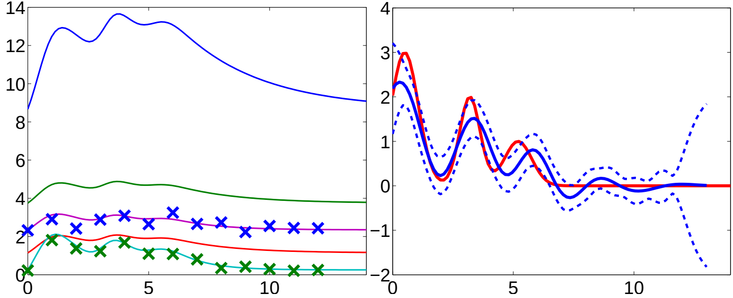


Figure: *Left:* The TFC, $f(t)$, which drives the system. *Middle:* Five gene mRNA concentration profiles each obtained by using different parameter sets $\{B_i, S_i, D_i\}_{i=1}^{5}$ (lines) along with noise corrupted 'data' . *Right:* The inferred TFC (with error bars).

Artificial Data Results

`demToyProblem1`

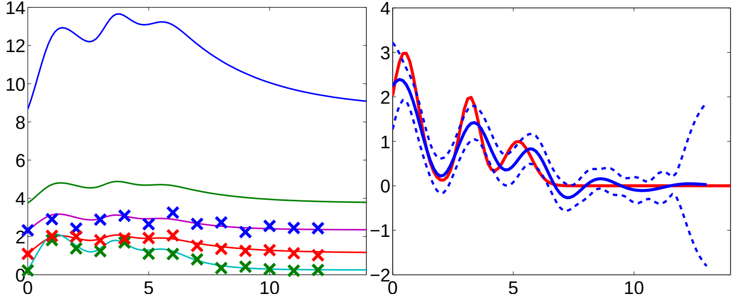


Figure: *Left:* The TFC, $f(t)$, which drives the system. *Middle:* Five gene mRNA concentration profiles each obtained by using different parameter sets $\{B_i, S_i, D_i\}_{i=1}^{5}$ (lines) along with noise corrupted 'data' . *Right:* The inferred TFC (with error bars).

Artificial Data Results

`demToyProblem1`

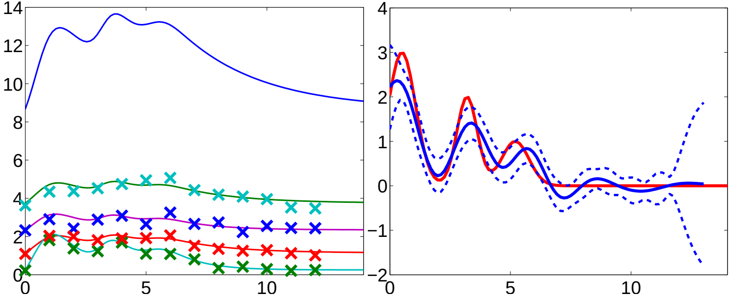


Figure: *Left:* The TFC, $f(t)$, which drives the system. *Middle:* Five gene mRNA concentration profiles each obtained by using different parameter sets $\{B_i, S_i, D_i\}_{i=1}^{5}$ (lines) along with noise corrupted 'data' . *Right:* The inferred TFC (with error bars).

Artificial Data Results

`demToyProblem1`

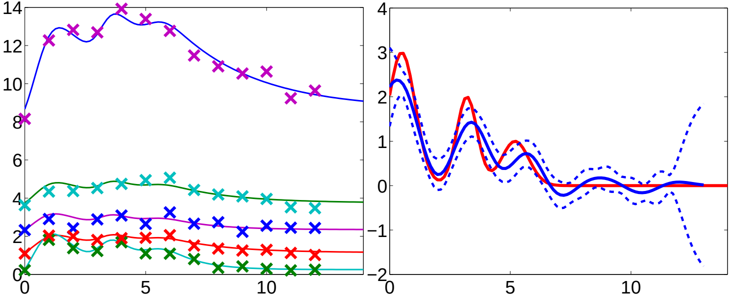


Figure: *Left:* The TFC, $f(t)$, which drives the system. *Middle:* Five gene mRNA concentration profiles each obtained by using different parameter sets $\{B_i, S_i, D_i\}_{i=1}^{5}$ (lines) along with noise corrupted 'data' . *Right:* The inferred TFC (with error bars).

Mesoderm Development

- Development of the mesoderm in *Drosophila*.

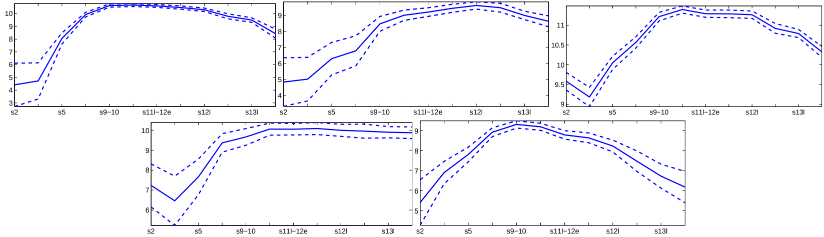


Figure: mRNA expression levels for target genes of tinman. (a) pannier, (b) hibris, (c) CG12744, (d) CG10516 (e) CG31368 .

Results — Drosophila

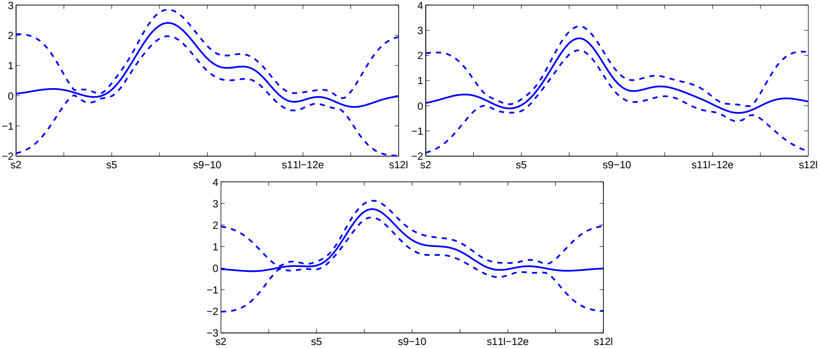


Figure: Inferred Transcription Factor Activities for tinman.

Results

- Recently published biological data set studied using linear response model by Barenco et al. [2006].
- Study focused on the tumour suppressor protein p53.
- mRNA abundance measured for five targets: *DDB2*, *p21*, *SESN1/hPA26*, *BIK* and *TNFRSF10b*.
- Quadratic interpolation for the mRNA production rates to obtain gradients.
- They used MCMC sampling to obtain estimates of the model parameters B_j, S_j, D_j and $f(t)$.

Linear response analysis

- We analysed data using the linear response model
- Raw data was processed using the mmgMOS model of Liu et al. [2005] which provides variance as well as expression level.
- We present posterior distribution over TFCs.
- Results of inference on the values of the hyperparameters B_j, S_j and D_j.
- Samples from the posterior distribution were obtained using Hybrid Monte Carlo (see *e.g.* Neal, 1996).

Linear Response Results

`demBarenco1`

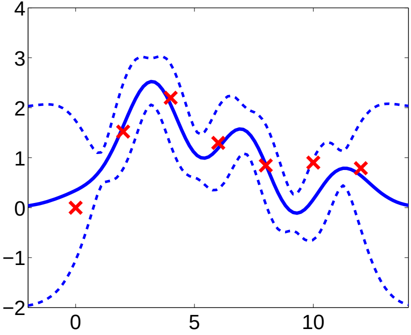


Figure: Predicted protein concentration for p53. Solid line is mean, dashed lines 95% credibility intervals. The prediction of [Barenco et al., 2006] was pointwise and is shown as crosses.

Results — Transcription Rates

- Estimation of Equation Parameters `demBarenco1`

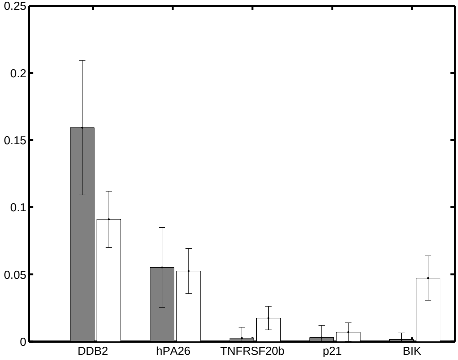


Figure: Basal transcription rates. Our results (black) compared with Barenco et al. [2006] (white).

Results — Transcription Rates

- Estimation of Equation Parameters `demBarenco1`

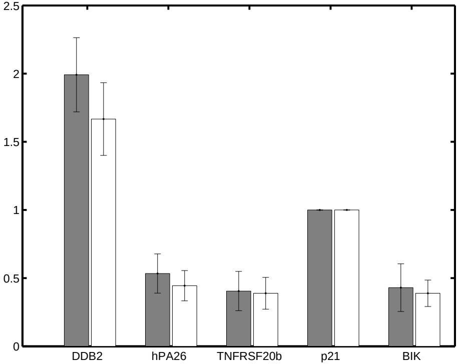


Figure: Sensitivities. Our results (black) compared with Barenco et al. [2006] (white).

Results — Transcription Rates

- Estimation of Equation Parameters `demBarenco1`

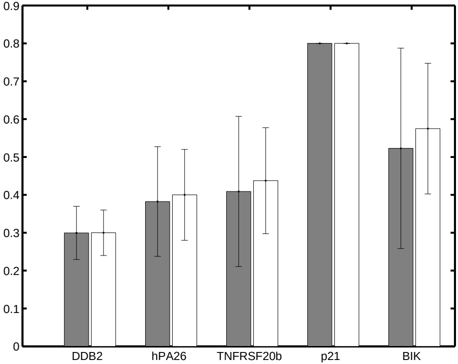


Figure: Decays. Our results (black) compared with Barenco et al. [2006] (white).

Linear Response Discussion

- Note oscillatory behaviour, possible artifact of RBF covariance Rasmussen and Williams [see page 123 in 2006].
- Results are in good accordance with the results obtained by Barenco et al..
- Differences in estimates of the basal transcription rates probably due to:
  - different methods used for probe-level processing of the microarray data.
  - Our failure to constrain $f(0) = 0$.

Non-linear Response Model

- All the quantities in equation (1) are positive, but direct samples from a GP will not be.
- Linear models don't account for saturation.
- *Solution*: model response using a positive nonlinear function.

Non Linear Response Formalism

- Introduce a non-linearity $g(\cdot)$ parameterised by $\boldsymbol{\theta}_j$

$$
\begin{aligned}
\frac{dy_j}{dt} &= B_j + g(f(t), \boldsymbol{\theta}_j) - D_j x_j \\
y_j(t) &= \frac{B_j}{D_j} + \exp\left(-D_j t\right) \int_0^t \mathrm{d}u\, g(f(u), \boldsymbol{\theta}_j) \exp\left(D_j u\right) \ .
\end{aligned}
$$

- The induced distribution of $y_j(t)$ is no longer a GP.
- Derive the functional gradient and learn a MAP solution for $f(t)$.
- Also compute Hessian so we can approximate the marginal likelihood.

Implementation

- Implementation requires a discretised time.
- Compute the gradient and Hessian on a grid.
- Integrate them by approximate Riemann quadrature.
- We choose a uniform grid $\{t_p\}_{p=1}^M$ so that $\Delta = t_p - t_{p-1}$ is constant.
- The vector $\mathbf{f} = \{f_p\}_{p=1}^M$ is the function f at the grid points.

$$I\left(t\right) = \int_0^t f\left(u\right) \exp\left(D_j u\right) du$$

$$I\left(t\right) \approx \sum_{p=1}^{M} f\left(t_p\right) \exp\left(D_j t_p\right) \Delta$$

Log Likelihood

- Given noise-corrupted data $y_j(t_i)$ the log-likelihood is

$$\log p(Y|f,\theta_j) = -\frac{1}{2}\sum_{i=1}^{T}\sum_{j=1}^{N}\left[\frac{\left(x_j(t_i) - y_j(t_i)\right)^2}{\sigma_{ji}^2} - \log\left(\sigma_{ji}^2\right)\right] - \frac{NT}{2}\log(2\pi)$$

- The functional derivative of the log-likelihood wrt f is

$$\frac{\delta \log p(Y|f)}{\delta f(t)} = -\sum_{i=1}^{T}\Theta(t_i - t)\sum_{j=1}^{N}\frac{(x_j(t_i) - y_j(t_i))}{\sigma_{ji}^2}g'(f(t))\mathrm{e}^{-D_j(t_i - t)}$$

$\Theta(x)$ — Heaviside step function.

Log Likelihood

- Functional Hessian
- Given noise-corrupted data $y_j\left(t_i\right)$ the log-likelihood is

$$\log p(Y|f,\theta_j) = -\frac{1}{2}\sum_{i=1}^{T}\sum_{j=1}^{N}\left[\frac{\left(x_j(t_i) - y_j\left(t_i\right)\right)^2}{\sigma_{ji}^2} - \log\left(\sigma_{ji}^2\right)\right] - \frac{NT}{2}\log(2\pi)$$

- The negative Hessian of the log-likelihood wrt f is

$$\begin{aligned} w(t,t') = &\sum_{i=1}^{T}\Theta(t_i - t)\delta\left(t - t'\right)\sum_{j=1}^{N}\frac{\left(x_j(t_i) - y_j\left(t_i\right)\right)}{\sigma_{ji}^2}g''(f(t))\mathrm{e}^{-D_j(t_i - t)} \\ &+ \sum_{i=1}^{T}\Theta(t_i - t)\Theta\left(t_i - t'\right)\sum_{j=1}^{N}\sigma_{ji}^{-2}g'\left(f(t)\right)g'\left(f(t')\right)\mathrm{e}^{-D_j(2t_i - t - t')} \end{aligned}$$

$$g'(f) = \partial g/\partial f \text{ and } g''(f) = \partial^2 g/\partial f^2.$$

Implementation II

- Combine with Prior
- Combine these with prior to compute gradient and Hessian of log posterior $\Psi(\mathbf{f}) = \log p(Y|\mathbf{f}) + \log p(\mathbf{f})$ [see Rasmussen and Williams, 2006, chapter 3]

$$\begin{aligned}\frac{\partial \Psi(\mathbf{f})}{\partial \mathbf{f}} &= \frac{\partial \log p(Y|\mathbf{f})}{\partial \mathbf{f}} - K^{-1}\mathbf{f} \\ \frac{\partial^2 \Psi(\mathbf{f})}{\partial \mathbf{f}^2} &= -(W + K^{-1})\end{aligned} \qquad (6)$$

K prior covariance evaluated at the grid points.

- Use to find a MAP solution via, $\hat{\mathbf{f}}$, using Newton's algorithm.
- The Laplace approximation is then

$$\log p(Y) \simeq \log p(Y|\hat{\mathbf{f}}) - \tfrac{1}{2}\hat{\mathbf{f}}^T K^{-1}\hat{\mathbf{f}} - \tfrac{1}{2}\log|I + KW|. \qquad (7)$$

Example: linear response

- Linear response with *non*-RBF kernels
- Start by taking $g(\cdot)$ to be linear.
- Provides 'sanity check' and *i.e.* non-stochastic) covariance function.
- Avoids double numerical integral that would normally be required.

Response Results

- `demBarencoMap1, demBarencoMap2`

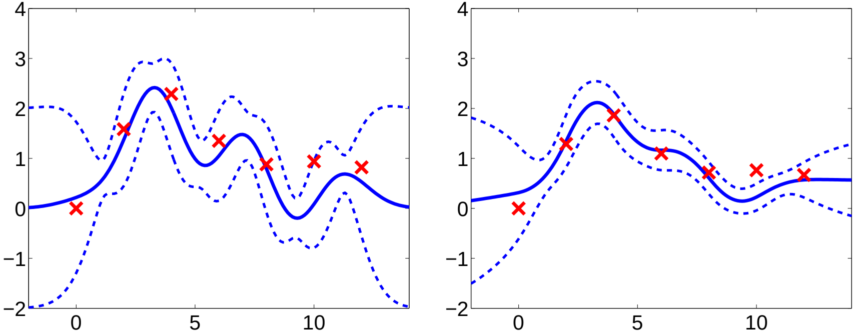


Figure: *Left:* RBF prior on f (log likelihood -101.4); *Right:* MLP prior on f (log likelihood -105.6). Solid line is mean prediction, dashed lines are 95% credibility intervals.

Non-linear response analysis

- Non-linear responses
- Exponential response model (constrains protein concentrations positive).
- $\log(1+\exp(f))$ response model.
- $\frac{3}{1+\exp(-f)}$
- Inferred MAP solutions for the latent function f are plotted below.

$\exp(\cdot)$ Response Results

- `demBarencoMap3`, `demBarencoMap4`

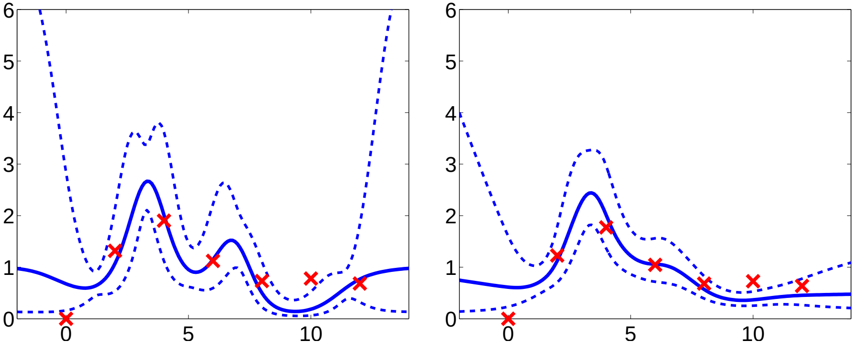


Figure: Exponential response: *Left:* squared exponential prior covariance on f (log likelihood -100.6); *Right:* MLP prior covariance on f (log likelihood -106.4). Solid line is mean prediction, dashed lines show 95% credibility intervals.

$\log(1 + \exp(f))$ Response Results

- `demBarencoMap5`, `demBarencoMap6`

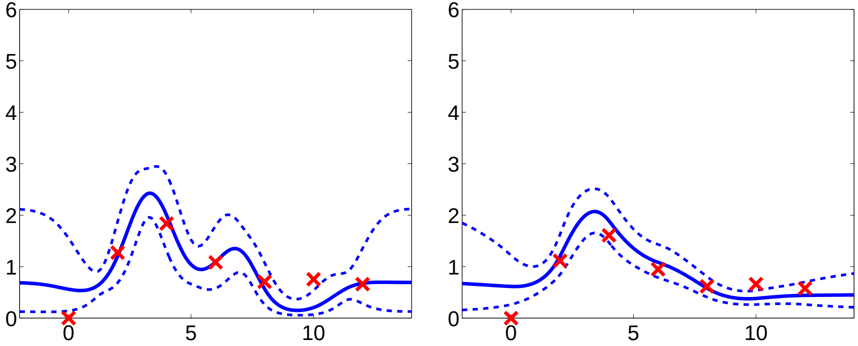


Figure: *Left:* squared exponential prior covariance on f (log likelihood -100.9); *Right:* shows MLP prior covariance on f (log likelihood -110.0).

$\frac{3}{1+\exp(-f)}$ Response Results

- `demBarencoMap7`, `demBarencoMap8`

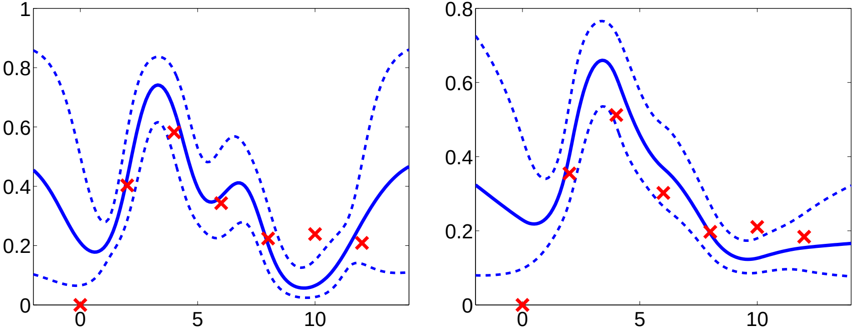


Figure: P*Left:* squared exponential prior covariance on f (log likelihood -104.1); *Right:* an MLP prior covariance on f (log likelihood -111.2). Solid line is mean prediction, dashed lines show 95% credibility intervals.

Discussion

- Promising applications for dynamics in latent variable modelling.
- Exampled showed how GPs can be used in modelling dynamics of a simple regulatory network motif.
- We are applying similar models to motion capture data (second order ODEs).
- there is no need to restrict the inference to the observed time points, the temporal continuity of the inferred functions is accounted for naturally.
- GPs allow us to handle uncertainty in a natural way.
- Code on-line `http://www.cs.man.ac.uk/~neill/gpsim/`.

Acknowledgements

- Data and Support
  - Martino Barenco and Mike Hubank for data and inspiration.
  - Funding
    - ⋆ PUMA Project http://www.cs.man.ac.uk/~neill/projects/puma. BBSRC Grant No BBS/B/0076X "Improved processing of microarray data with probabilistic models"
    - ⋆ TIGRA Project http://www.cs.man.ac.uk/~neill/projects/tigra/. EPSRC Grant No EP/F005873/1 "Gaussian Process Models for Systems Identification with Applications in Systems Biology"
- Collaborators
  - Magnus Rattray (co-I on both grants above)
  - David Luengo (visitor in Manchester, worked on 2nd order model)
  - Pei Gao, Michalis Titsias and Guido Sanguinetti (post-docs on above grants)

References I

M. Barenco, D. Tomescu, D. Brewer, R. Callard, J. Stark, and M. Hubank. Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome Biology*, 7(3):R25, 2006.

T. Graepel. Solving noisy linear operator equations by Gaussian processes: Application to ordinary and partial differential equations. In T. Fawcett and N. Mishra, editors, *Proceedings of the International Conference in Machine Learning*, volume 20, pages 234–241. AAAI Press, 2003. ISBN 1-57735-189-4.

N. D. Lawrence. Gaussian process models for visualisation of high dimensional data. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, pages 329–336, Cambridge, MA, 2004. MIT Press.

N. D. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 11 2005.

X. Liu, M. Milo, N. D. Lawrence, and M. Rattray. A tractable probabilistic model for Affymetrix probe-level analysis across multiple chips. *Bioinformatics*, 21(18):3637–3644, 2005. doi: 10.1093/bioinformatics/bti583.

R. Murray-Smith and B. A. Pearlmutter. Transformations of Gaussian process priors.

R. M. Neal. *Bayesian Learning for Neural Networks*. Springer, 1996. Lecture Notes in Statistics 118.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006. ISBN 026218253X.

S. T. Roweis. EM algorithms for PCA and SPCA. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10, pages 626–632, Cambridge, MA, 1998. MIT Press.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, B*, 6(3): 611–622, 1999.

R. Urtasun, D. J. Fleet, and P. Fua. 3D people tracking with Gaussian process dynamical models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 238–245, New York, U.S.A., 17–22 Jun. 2006. IEEE Computer Society Press.

J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, Cambridge, MA, 2006. MIT Press.