Latent Force Models with Gaussian Processes

Neil D. Lawrence

Bayesian Research Kitchen, Wordsworth Hotel, Grasmere

6th September 2008
1. Introduction

2. Covariance Functions

3. Convolutions and Computational Complexity

4. Non-linear Response Models

5. Cascaded Differential Equations

6. Discussion and Future Work

7. Acknowledgements
Linear relationship between the data, $X \in \mathbb{R}^{N \times d}$, and a reduced dimensional representation, $F \in \mathbb{R}^{N \times q}$, where $q \ll d$.

$$X = FW + \epsilon,$$

$$\epsilon \sim \mathcal{N}(0, \Sigma)$$

Integrate out $X$, optimize with respect to $W$.

For temporal data and a particular Gaussian prior in the latent space: Kalman filter/smoother

More generally consider a Gaussian process (GP) prior,

$$p(F|t) = \prod_{i=1}^{q} \mathcal{N}(f_{i,|0}, K_{f_{i,|0},f_{i,|0}}).$$
Given the covariance functions for $\{f_i(t)\}$ the implied covariance functions for $\{x_i(t)\}$ — semi-parametric latent factor model (Teh et al., 2005).

Kalman filter/smooother approach has been preferred

- linear computational complexity in $N$.
- Advances in sparse approximations have made the general GP framework practical. (Snelson and Ghahramani, 2006; Quiñonero Candela and Rasmussen, 2005, also Titsias tomorrow).
These models rely on the latent variables to provide the dynamic information.

We now introduce a further dynamical system with a mechanistic inspiration.

Physical Interpretation:

- the latent functions, $f_i(t)$ are $q$ forces.
- We observe the displacement of $d$ springs to the forces.
- Interpret system as the force balance equation, $XD = FS\epsilon$.
- Forces act, e.g. through levers — a matrix of sensitivities, $S \in \mathbb{R}^{q \times d}$.
- Diagonal matrix of spring constants, $D \in \mathbb{R}^{d \times d}$.
- Original System: $W = SD^{-1}$. 
Add a damper and give the system mass.

\[ FS = \ddot{X}M + \dot{X}C + XD + \epsilon. \]

Now have a second order mechanical system.

It will exhibit inertia and resonance.

There are many systems that can also be represented by differential equations.

- When being forced by latent function(s), \( \{f_i(t)\}_{i=1}^q \), we call this a latent force model.
For Gaussian process we can compute the covariance matrices for the output displacements.

For one displace the model is

\[
    m_k \ddot{x}_k(t) + c_k \dot{x}_k(t) + d_k x_k(t) = b_k + \sum_{i=0}^{M} s_{ik} f_i(t),
\]

(1)

where, \( m_k \) is the \( k \)th diagonal element from \( \mathbf{M} \) and similarly for \( c_k \) and \( d_k \). \( s_{ik} \) is the \( i, k \)th element of \( \mathbf{S} \).

Model the latent forces as \( q \) independent, GPs with RBF covariances

\[
    k_{f_if_l}(t, t') = \exp \left( -\frac{(t - t')^2}{\sigma_i^2} \right) \delta_{il}.
\]
Covariance for ODE Model

- RBF Kernel function for \( f(t) \)

\[
x_j(t) = \frac{1}{m_j \omega_j} \sum_{i=1}^{q} S_{ji} \exp(-\alpha_j t) \int_{0}^{t} f_i(u) \exp(\alpha_j u) \sin(\omega_j(t-u)) \, du
\]

- Joint distribution for \( x_1(t), x_2(t), x_3(t) \) and \( f(t) \).

Damping ratios:

<table>
<thead>
<tr>
<th>( \zeta_1 )</th>
<th>( \zeta_2 )</th>
<th>( \zeta_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.125</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Joint Sampling of \( x(t) \) and \( f(t) \)

Figure: Joint samples from the ODE covariance, cyan: \( f(t) \), red: \( x_1(t) \) (underdamped) and green: \( x_2(t) \) (overdamped) and blue: \( x_3(t) \) (critically damped).
Joint Sampling of $x(t)$ and $f(t)$

Figure: Joint samples from the ODE covariance, cyan: $f(t)$, red: $x_1(t)$ (underdamped) and green: $x_2(t)$ (overdamped) and blue: $x_3(t)$ (critically damped).
Joint Sampling of $x(t)$ and $f(t)$

Figure: Joint samples from the ODE covariance, cyan: $f(t)$, red: $x_1(t)$ (underdamped) and green: $x_2(t)$ (overdamped) and blue: $x_3(t)$ (critically damped).

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

Figure: Joint samples from the ODE covariance, cyan: $f(t)$, red: $x_1(t)$ (underdamped) and green: $x_2(t)$ (overdamped) and blue: $x_3(t)$ (critically damped).
Covariance for ODE

- RBF Kernel function for \( f(t) \)

\[
x_j(t) = \frac{1}{m_j \omega_j} \sum_{i=1}^{q} S_{ji} \exp(-\alpha_j t) \int_{0}^{t} f_i(u) \exp(\alpha_j u) \sin(\omega_j(t-u)) du
\]

- Joint distribution for \( x_1(t), x_2(t), x_3(t) \) and \( f(t) \).
- Damping ratios:

<table>
<thead>
<tr>
<th>( \zeta_1 )</th>
<th>( \zeta_2 )</th>
<th>( \zeta_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.125</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Example: Transcriptional Regulation

- **First Order Differential Equation**

\[
\frac{dx_j(t)}{dt} = B_j + S_j f(t) - D_j x_j(t)
\]

- Can be used as a model of gene transcription: Barenco et al., 2006.
- \(x_j(t)\) – concentration of gene \(j\)’s mRNA
- \(f(t)\) – concentration of active transcription factor
- Model parameters: baseline \(B_j\), sensitivity \(S_j\) and decay \(D_j\)
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when \(f(t)\) is not observed?
Example: Transcriptional Regulation

- First Order Differential Equation

\[ \frac{dx_j(t)}{dt} = B_j + S_j f(t) - D_j x_j(t) \]

- Can be used as a model of gene transcription: Barenco et al., 2006.

- \( x_j(t) \) – concentration of gene j’s mRNA
- \( f(t) \) – concentration of active transcription factor
- Model parameters: baseline \( B_j \), sensitivity \( S_j \) and decay \( D_j \)
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when \( f(t) \) is not observed?
First Order Differential Equation

\[ \frac{dx_j(t)}{dt} = B_j + S_j f(t) - D_j x_j(t) \]

Can be used as a model of gene transcription: Barenco et al., 2006.

- \( x_j(t) \) – concentration of gene \( j \)'s mRNA
- \( f(t) \) – concentration of active transcription factor
- Model parameters: baseline \( B_j \), sensitivity \( S_j \) and decay \( D_j \)
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when \( f(t) \) is not observed?
First Order Differential Equation

\[
\frac{dx_j(t)}{dt} = B_j + S_j f(t) - D_j x_j(t)
\]

- Can be used as a model of gene transcription: Barenco et al., 2006.
- \(x_j(t)\) – concentration of gene \(j\)’s mRNA
- \(f(t)\) – concentration of active transcription factor
- Model parameters: baseline \(B_j\), sensitivity \(S_j\) and decay \(D_j\)
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when \(f(t)\) is not observed?
Example: Transcriptional Regulation

- **First Order Differential Equation**

\[
\frac{dx_j(t)}{dt} = B_j + S_j f(t) - D_j x_j(t)
\]

- Can be used as a model of gene transcription: Barenco et al., 2006.
- \(x_j(t)\) – concentration of gene \(j\)’s mRNA
- \(f(t)\) – concentration of active transcription factor
- Model parameters: baseline \(B_j\), sensitivity \(S_j\) and decay \(D_j\)
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when \(f(t)\) is not observed?
Example: Transcriptional Regulation

- First Order Differential Equation

\[
\frac{dx_j(t)}{dt} = B_j + S_j f(t) - D_j x_j(t)
\]

- Can be used as a model of gene transcription: Barenco et al., 2006.
- \(x_j(t)\) – concentration of gene \(j\)’s mRNA
- \(f(t)\) – concentration of active transcription factor
- Model parameters: baseline \(B_j\), sensitivity \(S_j\) and decay \(D_j\)
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when \(f(t)\) is not observed?
Example: Transcriptional Regulation

- First Order Differential Equation

\[
\frac{dx_j(t)}{dt} = B_j + S_j f(t) - D_j x_j(t)
\]

- Can be used as a model of gene transcription: Barenco et al., 2006.
- \(x_j(t)\) – concentration of gene \(j\)’s mRNA
- \(f(t)\) – concentration of active transcription factor
- Model parameters: baseline \(B_j\), sensitivity \(S_j\) and decay \(D_j\)
- Application: identifying co-regulated genes (targets)
- Problem: how do we fit the model when \(f(t)\) is not observed?
RBF covariance function for \( f(t) \)

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

- Joint distribution for \( x_1(t), x_2(t) \) and \( f(t) \).

Here:

<table>
<thead>
<tr>
<th>( D_1 )</th>
<th>( S_1 )</th>
<th>( D_2 )</th>
<th>( S_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring \( f(t) \)
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
Artificial Example: Inferring $f(t)$
p53 “Guardian of the Cell”

- Responsible for Repairing DNA damage
- Activates DNA Repair proteins
- Pauses the Cell Cycle (prevents replication of damage DNA)
- Initiates *apoptosis* (cell death) in the case where damage can’t be repaired.
- Large scale feedback loop with NF-κB.
Figure: p53. Left unbound, Right bound to DNA. Images by David S. Goodsell from http://www.rcsb.org/ (see the “Molecule of the Month” feature).
Figure: Repair of DNA damage by p53. Image from Goodsell (1999).
Assume p53 affects targets as a single input module network motif (SIM).

Figure: p53 SIM network motif as modelled by Barenco et al. 2006.
Inferred p53 protein

Gene TNFRSF20b mRNA

Gene DDB2 mRNA

Gene p21 mRNA

Gene BIK mRNA

Gene hPA26 mRNA
Target Ranking for Elk-1.

Elk-1 is phosphorylated by ERK from the EGF signalling pathway.

Predict concentration of Elk-1 from known targets.

Rank other targets of Elk-1.
Elk-1 (MLP covariance)

Jennifer Withers

Transcription factor concentration over time
Elk-1 target selection

Fitted model used to rank potential targets of Elk-1
Solutions to these differential equations is normally as a convolution.

\[ x_i(t) = \int f(u) k_i(u - t) \, du + h_i(t) \]

\[ x_i(t) = \int_0^t f(u) g_i(u) \, du + h_i(t) \]

- Convolution Processes (Higdon, 2002; Boyle and Frean, 2005).
- Convolutions lead to \( N \times d \) size covariance matrices \( O \left( N^3 d^3 \right) \) complexity, \( O \left( N^2 d^2 \right) \) storage.
- Model is conditionally independent over \( \{x_i(t)\}_{i=1}^d \) given \( f(t) \).
**Independence Assumption**

- Can assume conditional independence given given \( \{ f(t_i) \}_{i=1}^k \).
  - Result is very similar to PITC approximation (Quiñonero Candela and Rasmussen, 2005).
  - Reduces to \( O(N^3dk^2) \) complexity, \( O(N^2dk) \) storage.
  - Can also do a FITC style approximation (Snelson and Ghahramani, 2006).
  - Reduces to \( O(Ndk^2) \) complexity, \( O(Ndk) \) storage.
Tide Sensor Network

Mauricio Alvarez

- Network of tide height sensors in the solent — tide heights are correlated.
- Data kindly provided by Alex Rogers (see Rogers et al., 2008).
- \( d = 3 \) and \( N = 1000 \) of the 4320 for the training set.
- Simulate sensor failure by knocking out one sensor for a given time.
- For the other two sensors we used all 1000 training observations.
- Take \( k = 100 \).
Figure: Predictive Mean and variance using independent GPs and the PITC approximation for the tide height signal in the sensor dataset.

(a) Bramblemet Independent

(b) Bramblemet PITC
Cokriging Jura

Mauricio Alvarez

- Jura dataset — concentrations of several heavy metals.
- Prediction 259 data, validation 100 data points.
- Predict primary variables (cadmium and copper) at prediction locations in conjunction with some secondary variables (nickel and zinc for cadmium; lead, nickel and zinc for copper) (Goovaerts, 1997, p. 248,249).
### Swiss Jura Results

<table>
<thead>
<tr>
<th></th>
<th>IGP</th>
<th>P(50)</th>
<th>P(100)</th>
<th>P(200)</th>
<th>P(500)</th>
<th>FGP</th>
<th>CK</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cadmium</strong></td>
<td>0.42</td>
<td>0.44</td>
<td>0.46</td>
<td>0.48</td>
<td>0.50</td>
<td>0.52</td>
<td></td>
</tr>
<tr>
<td><strong>Copper</strong></td>
<td>0.48</td>
<td>0.50</td>
<td>0.52</td>
<td>0.54</td>
<td>0.56</td>
<td>0.58</td>
<td></td>
</tr>
</tbody>
</table>

#### Figure: Mean absolute error. IGP stands for independent GP, P($M$) stands for PITC with $M$ inducing values, FGP stands for full GP and CK stands for ordinary co-kriging.
Models of non-linear regulation

- Non-linear Activation: Michaelis-Menten Kinetics

\[ \frac{dx_i(t)}{dt} = B_i + \frac{Si f(t)}{\gamma_i + f(t)} - D_i x_i(t) \]

used by Rogers and Girolami (2006)

- Non-linear Repression

\[ \frac{dx_i(t)}{dt} = B_i + \frac{Si}{\gamma_i + f(t)} - D_i x_i(t) \]

used by Khanin et al., 2006, PNAS 103
Models of non-linear regulation

- Non-linear Activation: Michaelis-Menten Kinetics

\[ \frac{dx_i(t)}{dt} = B_i + \frac{S_i f(t)}{\gamma_i + f(t)} - D_i x_i(t) \]

used by Rogers and Girolami (2006)

- Non-linear Repression

\[ \frac{dx_i(t)}{dt} = B_i + \frac{S_i}{\gamma_i + f(t)} - D_i x_i(t) \]

used by Khanin et al., 2006, PNAS 103
Models of non-linear regulation

- **Non-linear Activation: Michaelis-Menten Kinetics**

  \[
  \frac{dx_i(t)}{dt} = B_i + \frac{S_i f(t)}{\gamma_i + f(t)} - D_i x_i(t)
  \]

  used by Rogers and Girolami (2006)

- **Non-linear Repression**

  \[
  \frac{dx_i(t)}{dt} = B_i + \frac{S_i}{\gamma_i + f(t)} - D_i x_i(t)
  \]

  used by Khanin et al., 2006, PNAS 103
Consider the following modification to the model,

\[
\frac{dx_j(t)}{dt} = B_j + S_j g(f(t)) - D_j x_j(t),
\]

where \( g(\cdot) \) is a non-linear function. The differential equation can still be solved,

\[
x_j(t) = \frac{B_j}{D_j} + S_j \int_0^t e^{-D_j(t-u)} g_j(f(u)) \, du
\]

Use Laplace’s method (Laplace, 1774),

\[
p(f \mid x) = \mathcal{N}(\hat{f}, A^{-1}) \propto \exp \left( -\frac{1}{2} (f - \hat{f})^T A (f - \hat{f}) \right)
\]

where \( \hat{f} = \text{argmax} p(f \mid x) \) and \( A = -\nabla\nabla \log p(f \mid y) \big|_{f=\hat{f}} \) is the Hessian of the negative posterior at that point.
The Michaelis-Menten activation model uses the following non-linearity

\[ g_j(f(t)) = \frac{e^{f(t)}}{\gamma_j + e^{f(t)}}, \]

where we are using a GP \( f(t) \) to model the log of the TF activity.
Figure: Laplace approximation error bars along with samples from the true posterior distribution.
Sample in Gaussian processes

\[ p(f|x) \propto p(x|f) p(f) \]

Likelihood relates GP to data through

\[ x_j(t) = \alpha_j e^{-D_j t} + \frac{B_j}{D_j} + S_j \int_0^t e^{-D_j(t-u)} g_j(f(u)) du \]

We use control points for fast sampling.
Sampling using control points

- Separate the points in \( \mathbf{f} \) into two groups:
  - few control points \( \mathbf{f}_c \)
  - and the large majority of the remaining points \( \mathbf{f}_\rho = \mathbf{f} \setminus \mathbf{f}_c \)

- Sample the control points \( \mathbf{f}_c \) using a proposal \( q \left( \mathbf{f}_c^{(t+1)} | \mathbf{f}_c^{(t)} \right) \)

- Sample the remaining points \( \mathbf{f}_\rho \) using the conditional GP prior \( p \left( \mathbf{f}_\rho^{(t+1)} | \mathbf{f}_c^{(t+1)} \right) \)

- The whole proposal is

\[
Q \left( \mathbf{f}^{(t+1)} | \mathbf{f}^{(t)} \right) = p \left( \mathbf{f}_\rho^{(t+1)} | \mathbf{f}_c^{(t+1)} \right) q \left( \mathbf{f}_c^{(t+1)} | \mathbf{f}_c^{(t)} \right)
\]

- Its like sampling from the prior \( p(\mathbf{f}) \) but imposing random walk behaviour through the control points.
One transcription factor (p53) that acts as an activator. We consider the Michaelis-Menten kinetic equation

\[
\frac{dx_j(t)}{dt} = B_j + S_j \frac{\exp(f(t))}{\exp(f(t)) + \gamma_j} - D_jx_j(t)
\]

**MCMC details:**
- 7 control points are used (placed in a equally spaced grid)
- Running time 4/5 hours for 2 million sampling iterations plus burn in
- Acceptance rate for \(f\) after burn in was between 15\% – 25\%
Data used by Barenco et al. (2006): Predicted gene expressions for the 1st replica

1. **DDB2 Gene – first Replica**
2. **BIK Gene – first Replica**
3. **TNFRSF10b Gene – first Replica**
4. **Clp1/p21 Gene – first Replica**
5. **p26 sesn1 Gene – first Replica**
Data used by Barenco et al. (2006): Protein concentrations

Linear model (Barenco et al. predictions are shown as crosses)

Nonlinear (Michaelis-Menten kinetic equation)
Our results (grey) compared with Barenco et al. (2006) (black). Note that Barenco et al. use a linear model
Outline

1. Introduction
2. Covariance Functions
3. Convolutions and Computational Complexity
4. Non-linear Response Models
5. Cascaded Differential Equations
6. Discussion and Future Work
7. Acknowledgements
Transcription factor protein also has governing mRNA.
This mRNA can be measured.
In signalling systems this measurement can be misleading because it is activated (phosphorylated) transcription factor that counts.
In development phosphorylation plays less of a role.
Drosophila *Mesoderm* Development

Data from Furlong Lab in EMBL Heidelberg.

- Describe mesoderm development.
Cascaded Differential Equations

Antti Honkela

We take the production rate of active transcription factor to be given by

\[
\frac{df(t)}{dt} = \sigma y(t) - \delta f(t)
\]

\[
\frac{dx_j(t)}{dt} = B_j + S_j f(t) - D_j x_j(t)
\]

The solution for \( f(t) \), setting transient terms to zero, is

\[
f(t) = \sigma \exp(-\delta t) \int_0^t y(u) \exp(\delta u) \, du.
\]
RBF covariance function for $y(t)$

$$f(t) = \sigma \exp(-\delta t) \int_0^t y(u) \exp(\delta u) \, du$$

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

- Joint distribution for $x_1(t)$, $x_2(t)$, $f(t)$ and $y(t)$.

- Here:

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>$D_1$</th>
<th>$S_1$</th>
<th>$D_2$</th>
<th>$S_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>5</td>
<td>5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>
Results for Mef2 using the Cascade model
Outline

1 Introduction

2 Covariance Functions

3 Convolutions and Computational Complexity

4 Non-linear Response Models

5 Cascaded Differential Equations

6 Discussion and Future Work

7 Acknowledgements
Discussion and Future Work

- Integration of probabilistic inference with mechanistic models.
- These results are small simple systems.
- Ongoing work:
  - Scaling up to larger systems
  - Applications to other types of system, e.g. non-steady-state metabolomics, spatial systems etc.
  - Improved approximations.
  - Stochastic differential equations
1 Introduction

2 Covariance Functions

3 Convolutions and Computational Complexity

4 Non-linear Response Models

5 Cascaded Differential Equations

6 Discussion and Future Work

7 Acknowledgements
Acknowledgements

- Investigators: Neil Lawrence and Magnus Rattray
- Researchers: Peo Gao, Antti Honkela, Michalis Titsias, Mauricio Alvarez and Jennifer Withers
- Charles Girardot and Eileen Furlong of EMBL in Heidelberg (mesoderm development in *D. Melanogaster*).
- Martino Barenco and Mike Hubank at the Institute of Child Health in UCL (p53 pathway).

Funded by the BBSRC award “Improved Processing of microarray data using probabilistic models” and EPSRC award “Gaussian Processes for Systems Identification with applications in Systems Biology”


