Deep Gaussian Processes and Variational Propagation of Uncertainty

Andreas Damianou

Department of Computer Science, University of Sheffield, UK

Cambridge, 29/06/2015
Outline

Part 1: A General View
   Deep GPs – structural perspective
   Gaussian processes

Part 2: Deep GPs – Inference, Optimisation, Regularisation
   Motivation
   Bayesian regularization
   Factorised vs non-factorised bound and SVI

Part 3: Further Properties, Extensions, Demonstrations
   Learning rich structure
   Automatic alignment of data-sets
   Supervised learning
   Dynamics
   Partial observations and automatic pipelines

Summary
Outline

Part 1: A General View
  Deep GPs – structural perspective
  Gaussian processes

Part 2: Deep GPs – Inference, Optimisation, Regularisation
  Motivation
  Bayesian regularization
  Factorised vs non-factorised bound and SVI

Part 3: Further Properties, Extensions, Demonstrations
  Learning rich structure
  Automatic alignment of data-sets
  Supervised learning
  Dynamics
  Partial observations and automatic pipelines

Summary
A general \textit{family} of probabilistic models

\[
Y = f_3(f_2(\cdots f_1(X))), \quad H_i = f_i(H_{i-1})
\]
Deep Gaussian processes - Big Picture

Deep GP:
- Directed graphical model
- Non-parametric, non-linear mappings $f$
- Mappings $f$ marginalised out analytically
- Continuous variables
- NOT a GP!

Challenges:
- Marginalise out $H$ (intractable)
- No sampling: analytic approximation of objective
- Regularisation and principled uncertainty handling
- Automation in learning structure
Deep Gaussian processes - Big Picture

Deep GP:
- Directed graphical model
- Non-parametric, non-linear mappings $f$
- Mappings $f$ marginalised out analytically
- Continuous variables
- NOT a GP!

Challenges:
- Marginalise out $H$ (intractable)
- No sampling: analytic approximation of objective
- Regularisation and principled uncertainty handling
- Automation in learning structure
Quick Intro to GPs

- 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:

\[
p\left(f_1, f_2, \cdots, f_s, f_{s+1}, f_{s+2}, \cdots, f_N\right) \sim \mathcal{N}(\mu, K).
\]

with:

\[
\mu = \begin{bmatrix} \mu_A \\ \mu_B \end{bmatrix} \quad \text{and} \quad K = \begin{bmatrix} K_{AA} & K_{AB} \\ K_{BA} & K_{BB} \end{bmatrix}
\]

Marginalisation property:

\[
p(f_A, f_B) \sim \mathcal{N}(\mu, K). \quad \text{Then:} \quad p(f_A) = \int_{f_B} p(f_A, f_B)df_B = \mathcal{N}(\mu_A, K_{AA})
\]
Infinite model... but we *always* work with finite sets!

Let’s start with a multivariate Gaussian:

\[
p(f_1, f_2, \ldots, f_s, f_{s+1}, f_{s+2}, \ldots, f_N) \sim \mathcal{N}(\mu, K).
\]

with:

\[
\mu = \begin{bmatrix} \mu_A \\ \mu_B \end{bmatrix} \quad \text{and} \quad K = \begin{bmatrix} K_{AA} & K_{AB} \\ K_{BA} & K_{BB} \end{bmatrix}
\]

Marginalisation property:

\[
p(f_A, f_B) \sim \mathcal{N}(\mu, K). \quad \text{Then:} \quad p(f_A) = \int_{f_B} p(f_A, f_B)df_B = \mathcal{N}(\mu_A, K_{AA})
\]
Infinite model... but we can work with finite sets!

In the GP context:

\[
\begin{align*}
\mu_\infty &= \begin{bmatrix} 
\mu_x \\
\vdots \\
\end{bmatrix} \\
K_\infty &= \begin{bmatrix} 
K_{xx} & \cdots \\
\vdots & \ddots \\
\end{bmatrix}
\end{align*}
\]

*Posterior* is also a Gaussian process!
Infinite model... but we can work with finite sets!

In the GP context:

\[
\mu_{\infty} = \begin{bmatrix} \mu_x \\ \vdots \\ \vdots \end{bmatrix} \quad \text{and} \quad K_{\infty} = \begin{bmatrix} K_{xx} & \cdots \\ \vdots & \ddots \end{bmatrix}
\]

*Posterior* is also a Gaussian process!
Incorporating Gaussian noise is tractable

- So far we assumed: \( f = f(X) \)
- Assuming that we only observe noisy versions \( y \) of the true outputs \( f \):
  
  \[ y = f(X) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \beta^{-1}) \]
Fitting the data
Fitting the data
Fitting the data
Fitting the data
## Inducing points

<table>
<thead>
<tr>
<th>$h^{(1)}$</th>
<th>$f^{(1)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h^{(2)}$</td>
<td>$f^{(2)}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$h^{(30)}$</td>
<td>$f^{(30)}$</td>
</tr>
<tr>
<td>$h^{(31)}$</td>
<td>$f^{(31)}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$h^{(N)}$</td>
<td>$f^{(N)}$</td>
</tr>
</tbody>
</table>

![Graph showing function $f$ with inducing points $h^{(1)}$ to $h^{(N)}$]
**Inducing points**

<table>
<thead>
<tr>
<th>$h^{(1)}$</th>
<th>$f^{(1)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h^{(2)}$</td>
<td>$f^{(2)}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$h^{(30)}$</td>
<td>$f^{(30)}$</td>
</tr>
<tr>
<td>$\tilde{h}^{(i)}$</td>
<td>$u^{(i)}$</td>
</tr>
<tr>
<td>$h^{(31)}$</td>
<td>$f^{(31)}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$h^{(N)}$</td>
<td>$f^{(N)}$</td>
</tr>
</tbody>
</table>

![Graph](image-url)
## Inducing points

<table>
<thead>
<tr>
<th>$h^{(1)}$</th>
<th>$f^{(1)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h^{(2)}$</td>
<td>$f^{(2)}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$h^{(30)}$</td>
<td>$f^{(30)}$</td>
</tr>
<tr>
<td>$\tilde{h}^{(i)}$</td>
<td>$u^{(i)}$</td>
</tr>
<tr>
<td>$h^{(31)}$</td>
<td>$f^{(31)}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$h^{(N)}$</td>
<td>$f^{(N)}$</td>
</tr>
</tbody>
</table>

![Diagram](image-url)
Outline

Part 1: A General View
Deep GPs – structural perspective
Gaussian processes

Part 2: Deep GPs – Inference, Optimisation, Regularisation
Motivation
Bayesian regularization
Factorised vs non-factorised bound and SVI

Part 3: Further Properties, Extensions, Demonstrations
Learning rich structure
Automatic alignment of data-sets
Supervised learning
Dynamics
Partial observations and automatic pipelines

Summary
Sampling from a deep GP

\[ x \xrightarrow{f_h} h \xrightarrow{f_Y} Y \]

Input

\[-2 \quad -1.5 \quad -1 \quad -0.5 \quad 0 \quad 0.5 \quad 1 \quad 1.5 \quad 2\]

Unobserved

Output
Learning deep GPs according to [Damianou et al., AISTATS 2013]:

- **Analytic variational bound** $\mathcal{F} \leq p(y|x)$
  - Extend the inducing variable trick of [1,2,3]
    
    
    
  
  - *Approximately* marginalise out $h$

- **Automatic structure discovery (nodes, connections, layers)**
  - Use the Automatic / Manifold Relevance Determination trick
Direct marginalisation of $h$ is intractable

Objective: $p(y|x) = \int_{h_2} \left( p(y|h_2) \int_{h_1} p(h_2|h_1)p(h_1|x) \right)$
Direct marginalisation of $h$ is intractable

- **Objective:** $p(y|x) = \int_{h_2} \left( p(y|h_2) \int_{h_1} p(h_2|h_1)p(h_1|x) \right)$

- $p(h_2|x) = \int_{h_1,f_2} p(h_2|f_2)p(f_2|h_1)p(h_1|x)$
Direct marginalisation of $h$ is intractable

- Objective: $p(y|x) = \int_{h_2} \left( p(y|h_2) \int_{h_1} p(h_2|h_1) p(h_1|x) \right)$

- $p(h_2|x) = \int_{h_1, f_2} p(h_2|f_2) p(f_2|h_1) p(h_1|x)$
Direct marginalisation of $h$ is intractable

Objective: \[ p(y|x) = \int_{h_2} \left( p(y|h_2) \int_{h_1} p(h_2|h_1)p(h_1|x) \right) \]

\[ p(h_2|x) = \int_{h_1,f_2} p(h_2|f_2) \underbrace{p(f_2|h_1)} p(h_1|x) \]

contains \((k(h_1, h_1))^{-1}\)
Direct marginalisation of $h$ is intractable

- Objective: $p(y|x) = \int_{h_2} \left( p(y|h_2) \int_{h_1} p(h_2|h_1)p(h_1|x) \right)$

- $p(h_2|x) = \int_{h_1, f_2} p(h_2|f_2)p(f_2|h_1) \ p(h_1|x)$

- $p(h_2|x, \tilde{h}_1) = \int_{h_1, f_2, u_2} p(h_2|f_2)p(f_2|u_2, h_1)p(u_2|\tilde{h}_1)p(h_1|x)$
Direct marginalisation of $h$ is intractable

- **Objective:** $p(y|x) = \int_{h_2} \left( p(y|h_2) \int_{h_1} p(h_2|h_1)p(h_1|x) \right)$

- $p(h_2|x) = \int_{h_1,f_2} p(h_2|f_2)p(f_2|h_1)p(h_1|x)$

- $p(h_2|x, \tilde{h}_1) = \int_{h_1,f_2,u_2} p(h_2|f_2)p(f_2|u_2,h_1)p(u_2|\tilde{h}_1)p(h_1|x)$

- $\log p(h_2|x, \tilde{h}_1) \geq \int_{h_1,f_2,u_2} Q \log \frac{p(h_2|f_2)p(f_2|u_2,h_1)p(u_2|\tilde{h}_1)p(h_1|x)}{Q}$
Direct marginalisation of $h$ is intractable

Objective: $p(y|x) = \int_{h_2} \left( p(y|h_2) \int_{h_1} p(h_2|h_1)p(h_1|x) \right)$

$p(h_2|x) = \int_{h_1,f_2} p(h_2|f_2)p(f_2|h_1)p(h_1|x)$

$p(h_2|x, \tilde{h}_1) = \int_{h_1,f_2,u_2} p(h_2|f_2)p(f_2|u_2,h_1)p(u_2|\tilde{h}_1)p(h_1|x)$

$log\ p(h_2|x, \tilde{h}_1) \geq \int_{h_1,f_2,u_2} Q \log \frac{p(h_2|f_2)p(f_2|u_2,h_1)p(u_2|\tilde{h}_1)p(h_1|x)}{Q = p(f_2|u_2,h_1)q(u_2)q(h_1)}$
Direct marginalisation of $h$ is intractable

- **Objective:**
  \[ p(y|x) = \int_{h_2} \left( p(y|h_2) \int_{h_1} p(h_2|h_1)p(h_1|x) \right) \]

- \[ p(h_2|x) = \int_{h_1, f_2} p(h_2|f_2)p(f_2|h_1)p(h_1|x) \]

- \[ p(h_2|x, \tilde{h}_1) = \int_{h_1, f_2, u_2} p(h_2|f_2)p(f_2|u_2, h_1)p(u_2|\tilde{h}_1)p(h_1|x) \]

- \[ \log p(h_2|x, \tilde{h}_1) \geq \int_{h_1, f_2, u_2} Q \log \frac{p(h_2|f_2)p(f_2|u_2, h_1)p(u_2|\tilde{h}_1)p(h_1|x)}{Q = p(f_2|u_2, h_1)q(u_2)q(h_1)} \]

- \[ \log p(h_2|x, \tilde{h}_1) \geq \int_{h_1, f_2, u_2} Q \log \frac{p(h_2|f_2)p(u_2|\tilde{h}_1)p(h_1|x)}{q(u_2)q(h_1)} \]

  \[ p(u_2|\tilde{h}_1) \text{ contains } k(\tilde{h}_1, \tilde{h}_1)^{-1} \]
Direct marginalisation of $h$ is intractable

- **Objective:**
  
  $$p(y|x) = \int_{h_2} \left( p(y|h_2) \int_{h_1} p(h_2|h_1)p(h_1|x) \right)$$

- $p(h_2|x) = \int_{h_1, f_2} p(h_2|f_2)p(f_2|h_1)p(h_1|x)$

- $p(h_2|x, \tilde{h}_1) = \int_{h_1, f_2, u_2} p(h_2|f_2)p(f_2|u_2, h_1)p(u_2|\tilde{h}_1)p(h_1|x)$

- \[ \log p(h_2|x, \tilde{h}_1) \geq \int_{h_1, f_2, u_2} Q \log \frac{p(h_2|f_2)p(f_2|u_2, h_1)p(u_2|\tilde{h}_1)p(h_1|x)}{Q=p(f_2|u_2, h_1)q(u_2)q(h_1)} \]

- \[ \log p(h_2|x, \tilde{h}_1) \geq \int_{h_1, f_2, u_2} Q \log \frac{p(h_2|f_2)p(u_2|\tilde{h}_1)p(h_1|x)}{q(u_2)q(h_1)} \]

  \[ p(u_2|\tilde{h}_1) \text{ contains } k(\tilde{h}_1, \tilde{h}_1)^{-1} \]

Some extra work required for “linking” between layers:

$q(h_l)$ is involved in layer $l$ and in layer $l + 1$. 
Note: All $q$ distributions (in $Q$) are selected to be Gaussian.

$$\mathcal{F} = \sum_{l=2}^{L+1} \langle \mathcal{L}_l \rangle_Q - \sum_{l=2}^{L+1} \text{KL} (q(u_l) \| p(u_l))$$

where

$$\mathcal{L}_l = \sum_{n=1}^{N} \sum_{q=1}^{Q_l} \log \mathcal{N} \left( h_l^{(n,q)} | k_l^{(n,:)} K^{-1} u_l^{(:,q)}, \beta_l^{-1} I \right) - \frac{\beta_l^{-1} k_l^{(n)}}{2}$$
Properties of the bound (unsupervised case)

Note: All $q$ distributions (in $Q$) are selected to be Gaussian.

$$
\mathcal{F} = \sum_{l=2}^{L+1} \langle \mathcal{L}_l \rangle_Q - \sum_{l=2}^{L+1} \text{KL} (q(u_l) \| p(u_l))
$$

$$
= -\text{KL} (q(h_1) \| p(h_1)) + \sum_{l=2}^{L} \text{Regularisation}
$$

where

$$
\mathcal{L}_l = \sum_{n=1}^{N} \left[ \sum_{q=1}^{Q_l} \log \mathcal{N} \left( h_l^{(n,q)} \mid k_l^{(n,:)} K^{-1} u_l^{(:,q)}, \beta_l^{-1} \mathbf{I} \right) - \frac{\beta_l^{-1} k_l^{(n)}}{2} \right]
$$

- All terms factorise w.r.t data points.
Identify global param., $\theta_{\text{global}}$, as in SVIGP of [Hensman et al., UAI'13]

Unlike $\theta_{\text{global}}$, $h$ are not global variables.

So, estimate $q(h^{(\text{batch})})$ given (current) $\theta_{\text{global}}$ and iterate.

[Hensman, Damianou and Lawrence, AISTATS (Late-breaking) 2014]

Hidden space projections (20K mocap examples):

Global motion features  
Clustered motion features
Identify global param., $\theta_{\text{global}}$, as in SVIGP of [Hensman et al., UAI’13].

Unlike $\theta_{\text{global}}$, $h$ are not global variables.

So, estimate $q(h^{(\text{batch})})$ given (current) $\theta_{\text{global}}$ and iterate.

[Hensman, Damianou and Lawrence, AISTATS (Late-breaking) 2014]

Hidden space projections (20K mocap examples):

Global motion features  Clustersed motion features
Stochastic Variational Inference

- Identify global param., $\theta_{\text{global}}$, as in SVIGP of [Hensman et al., UAI’13]
- Unlike $\theta_{\text{global}}$, $h$ are not global variables.
- So, estimate $q(h^{\text{batch}})$ given (current) $\theta_{\text{global}}$ and iterate.

[Hensman, Damianou and Lawrence, AISTATS (Late-breaking) 2014]

Hidden space projections (20K mocap examples):

Global motion features  
Clustered motion features
Properties of the bound (unsupervised case)

Note: All $q$ distributions (in $Q$) are selected to be Gaussian.

\[
\mathcal{F} = \sum_{l=2}^{L+1} \langle \mathcal{L}_l \rangle_Q - \sum_{l=2}^{L+1} \text{KL} (q(u_l) \parallel p(u_l)) - \text{KL} (q(h_1) \parallel p(h_1)) + \sum_{l=2}^{L} \mathcal{H} (q(h_l))
\]

where

\[
\mathcal{L}_l = \sum_{n=1}^{N} \left[ \sum_{q=1}^{Q_l} \log \mathcal{N} \left( h_l^{(n,q)} | k_l^{(n,:)}, K^{-1} u_l^{(,:q)} , \beta_l^{-1} I \right) - \frac{\beta_l^{-1} k_l^{(n)}}{2} \right]
\]

- All terms factorise w.r.t data points.
Properties of the bound (unsupervised case)

Note: All $q$ distributions (in $Q$) are selected to be Gaussian.

\[
\mathcal{F} = \sum_{l=2}^{L+1} \langle \mathcal{L}_l \rangle_Q - \sum_{l=2}^{L+1} \text{KL} (q(u_l) \| p(u_l)) \\
-\text{KL} (q(h_1) \| p(h_1)) + \sum_{l=2}^{L} \mathcal{H} (q(h_l))
\]

where

\[
\mathcal{L}_l = \sum_{n=1}^{N} \left[ \sum_{q=1}^{Q_l} \log \mathcal{N} \left( h_l^{(n,q)} \mid k_l^{(n,:)} K^{-1} u_l^{(:,q)}, \beta_l^{-1} I \right) - \frac{\beta_l^{-1} k_l^{(n)}}{2} \right]
\]

- All terms factorise w.r.t data points.
- We can additionally collapse $q(u)$
Collapsing $q(u)$ eliminates many variational parameters and makes bound “tighter”...

...but this introduces coupling and breaks the factorisation.

Likely we can still distribute the computations efficiently (e.g. by extending the work of [1, 2])


Outline

Part 1: A General View
  Deep GPs – structural perspective
  Gaussian processes

Part 2: Deep GPs – Inference, Optimisation, Regularisation
  Motivation
  Bayesian regularization
  Factorised vs non-factorised bound and SVI

Part 3: Further Properties, Extensions, Demonstrations
  Learning rich structure
  Automatic alignment of data-sets
  Supervised learning
  Dynamics
  Partial observations and automatic pipelines

Summary
Automatic structure discovery: outline

Tools:
- ARD: Eliminate unnecessary nodes/connections
- MRD: Conditional independencies
- Approximating evidence: Number of layers (?)
Automatic structure discovery: outline

Tools:

- ARD: Eliminate unnecessary nodes/connections
- MRD: Conditional independencies
- Approximating evidence: Number of layers (?)
Automatic dimensionality detection

- Achieved by employing automatic relevance determination (ARD) priors for the mapping $f$.

- $f \sim \mathcal{GP}(0, k_f)$ with:

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

- Example:
Deep GP: MNIST example

Optimised weights

Outputs obtained after sampling from (certain nodes) of layers 1, 2, 4, 5

Generic feature encoding

Local feature encoding

https://youtu.be/E8-vxt8wxBU (video demonstration)

[Damianou and Lawrence, AISTATS 2013]
Observations come into two different views: $Y$ and $Z$.

The latent space is segmented into parts private to $Y$, private to $Z$ and shared between $Y$ and $Z$.

Used for data consolidation and discovering commonalities.

[Damianou et al. ICML 2012]
Example 1: Motion capture / silhouette

Example 2: Faces data  
[Link to video](https://youtu.be/rIPX3CIOhKY)
Deep GPs: Another multi-view example
Alignment of views (e.g. video-audio, measurements-timestamps)

NP-hard problem.
Automatic Alignment of Data-sets

Greedy approach:

- Given fully aligned instances collected in $D_0 = \{Y, Z\}$ train a factorised MRD model

- Determine the segmentation $X = [X^Y, X^{Y,Z}, X^Z]$

- $D \leftarrow D_0, \quad D_\ast \leftarrow \{Y_\ast, Z_\ast\}$

- For each test instance $y_\ast$:
  - Compute $x_\ast \approx p(x_\ast | y_\ast, D)$
  - $z_\ast = \arg\max_z p(z | x_\ast^{Y,Z}, x_\ast^Z, D), \quad z \in D_\ast$
  - Update the global parameters of the model given $\{y_\ast, z_\ast\}$
  - $D \leftarrow [D, \{y_\ast, z_\ast\}], \quad D_\ast \leftarrow D_\ast - \{y_\ast, z_\ast\}$. 
The variational distribution on the top layer now is *coupled across datapoints*:

\[
q(H_1) = \prod_{q=1}^{Q_1} \mathcal{N}(h_{1q}|m_{1q}, S_{1q}^{(q)})
\]

(and small other changes in the bound...)

- Now \(S_1\) is a *full* \(N \times N\) matrix!
- Reparametrisation
  
  [Opper and Archambeau 2009, Damianou et al. 2011]:

\[
S_{1q} = \left(K_x^{-1} + \lambda^{(q)}\right)^{-1}
\]

- Coupling the inputs gives rise to a powerful model for *multivariate timeseries / system identification*. 
The variational distribution on the top layer now is coupled across datapoints:

$$q(H_1) = \prod_{q=1}^{Q_1} \mathcal{N}\left(h^{(q)}_1 | m^{(q)}_1, S^{(q)}_1\right)$$

(and small other changes in the bound...)

Now $S_1$ is a full $N \times N$ matrix!

Reparametrisation

[Opper and Archambeau 2009, Damianou et al. 2011]:

$$S^{(q)}_1 = \left(K_x^{-1} + \lambda^{(q)}\right)^{-1}$$

Coupling the inputs gives rise to a powerful model for multivariate timeseries / system identification.
The variational distribution on the top layer now is **coupled across datapoints**:

\[
q(H_1) = \prod_{q=1}^{Q_1} \mathcal{N}\left(h_1^{(q)}|m_1^{(q)}, S_1^{(q)}\right)
\]

(and small other changes in the bound...)

Now \(S_1\) is a **full** \(N \times N\) matrix!

Reparametrisation

[Opper and Archambeau 2009, Damianou et al. 2011]:

\[
S_1^{(q)} = \left(K_x^{-1} + \lambda^{(q)}\right)^{-1}
\]

Coupling the inputs gives rise to a powerful model for **multivariate timeseries / system identification**.
The variational distribution on the top layer now is coupled across datapoints:

\[ q(H_1) = \prod_{q=1}^{Q_1} \mathcal{N} \left( h_1^{(q)} | m_1^{(q)}, S_1^{(q)} \right) \]

(and small other changes in the bound...)

Now \( S_1 \) is a full \( N \times N \) matrix!

Reparametrisation

[Opper and Archambeau 2009, Damianou et al. 2011]:

\[ S_1^{(q)} = \left( K^{-1} + \lambda^{(q)} \right)^{-1} \]

Coupling the inputs gives rise to a powerful model for multivariate timeseries / system identification.
Dynamics

- Deterministic inputs in top layer ⇒ can consider any kernel!
- Dynamics are encoded in the covariance matrix $K = k(t, t)$.
- We can consider special forms for $K$.

Model individual sequences

Model periodic data

[https://www.youtube.com/watch?v=i9TEoYxaBxQ](https://www.youtube.com/watch?v=i9TEoYxaBxQ) (miss-America)

[https://www.youtube.com/watch?v=mUY1XHPnoCU](https://www.youtube.com/watch?v=mUY1XHPnoCU) (dog-treadmill)

[https://www.youtube.com/watch?v=fHDWloJtgk8](https://www.youtube.com/watch?v=fHDWloJtgk8) (mocap)

[Damianou et al., NIPS 2011]
Deep GP variants

Deep GP - Supervised

Deep GP - Unsupervised

Multi-view

Warped GP

Temporal

Autoencoder
Partial observations: automating the learning pipeline

Semi-described and semi-supervised learning

[Damianou et al., UAI 2015]
Consider: observed, $\phi$, and unobserved set, $\mathcal{U}$ from $X$

**Variational constraints:**

$$q(H|X, \{\phi, \mathcal{U}\}) = q(H_\phi|X_\phi)q(H_\mathcal{U}|X_\mathcal{U})$$

$$= \prod_{n \in \phi} \mathcal{N}(h^{(n)}_\phi|x^{(n)}_\phi, \epsilon I) \prod_{n \in \mathcal{U}} \mathcal{N}(h^{(n)}_\mathcal{U}|\mu^{(n)}_\mathcal{U}, S^{(n)}_\mathcal{U}), \epsilon \to 0$$

**Algorithm (sketch):**

- Train on the fully observed set
- Impute unobserved values and obtain uncertainties $S_\mathcal{U}$
- The predicted uncertainty now becomes input uncertainty in a variationally constrained model
- Recalibrate the new model which accounts for input uncertainty
Partial observations are successfully taken into account, yielding better results in regression/classification.
Autoencoder: Brendan faces
Deep GPs in iCub’s “brain”

Use deep GPs as an advanced, automatic perception (data representation) module.

http://youtu.be/Z5K0csC5gZ4 (iCub – face recognition demo)

[Damianou et al., Living Machines 2015]
Not so close to A.I. singularity...

But Bayesian non-parametrics are promising for building expressive and intuitive models of perception (data representation) while decreasing dependence on the human expert (e.g. automatic signal decomposition in MRD). Uncertainty propagation is a promising and intuitive way for communicating “messages” between stages of algorithmic pipelines and within components of probabilistic models.
Not so close to A.I singularity...

But Bayesian non-parametrics are promising for building expressive and intuitive models of perception (data representation) while decreasing dependence on the human expert (e.g. automatic signal decomposition in MRD). Uncertainty propagation is a promising and intuitive way for communicating “messages” between stages of algorithmic pipelines and within components of probabilistic models.
A deep GP is a more general model than a GP.

Supervised / unsupervised learning or anywhere in between.

A variational bound can be derived by special treatment of inducing variables.

Strongly regularised model ⇒ discovers rich structure.

Many variants: multi-view, temporal, autoencoders ...

Future: make it scalable with distributed computations / recognition models.

Future: how does it compare to / complement more traditional deep models?
Thanks to Neil Lawrence, James Hensman, Michalis Titsias, Carl Henrik Ek.
References:

- N. D. Lawrence (2006) “Probabilistic dimensional reduction with the Gaussian process latent variable model” (talk)
BACKUP SLIDES
MRD weights

\[
\begin{align*}
\mathbf{X}_1 & \quad \mathbf{X}_2 \quad \ldots \quad \mathbf{X}_Q \\
& \downarrow \quad \downarrow \quad \ldots \quad \downarrow \\
\mathbf{Y} & \quad & \mathbf{Z} \\
\omega_1^Y & \quad \omega_2^Y & \quad \omega_Q^Y \\
& \quad \downarrow \quad \uparrow \quad \downarrow \\
\omega_1^Z & \quad \omega_2^Z & \quad \omega_Q^Z
\end{align*}
\]