Deep Gaussian Processes and Variational Propagation of Uncertainty

Andreas Damianou

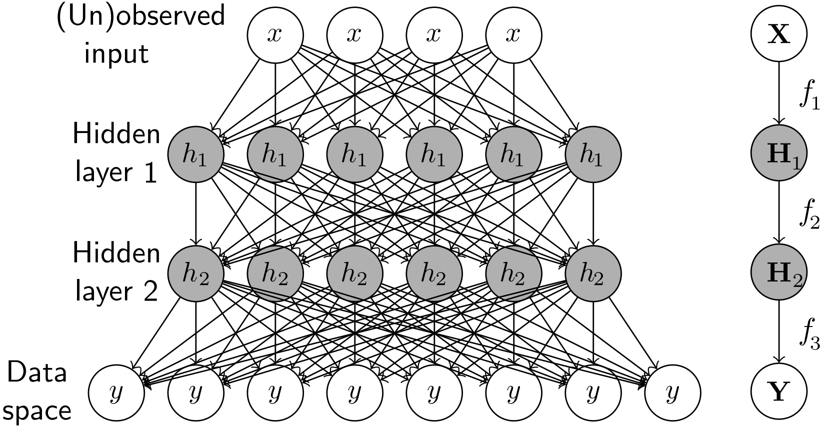
Department of Computer Science, University of Sheffield, UK

Cambridge, 29/06/2015

Outline

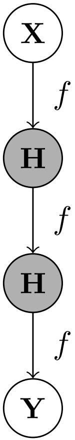
Outline

A general *family* of probabilistic models



$$\mathbf{Y} = f_3(f_2(\cdots f_1(\mathbf{X}))), \qquad \mathbf{H}_i = f_i(\mathbf{H}_{i-1})$$

Deep Gaussian processes - Big Picture



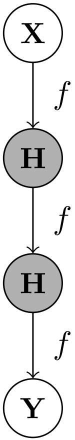
Deep GP:

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

Challenges:

- Marginalise out $\mathbf{H}$ (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 $\mathbf{H}$ (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:

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

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 can 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 a Gaussian process!

Infinite model... but we can work with finite sets!

In the GP context:

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

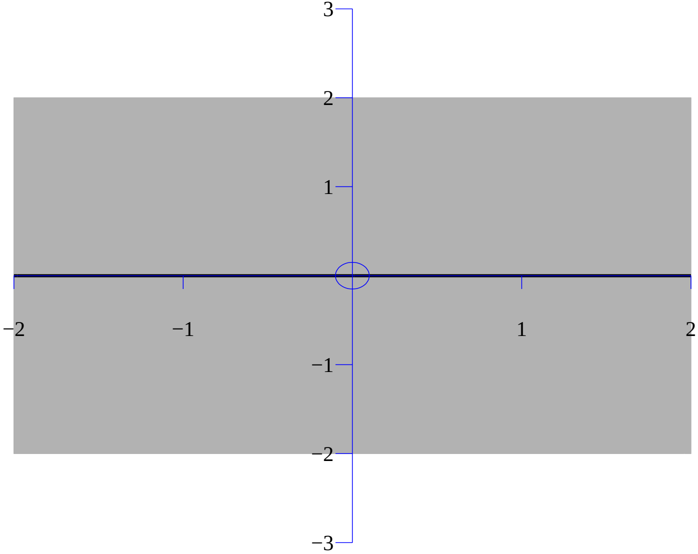
Posterior is also a Gaussian process!

Incorporating Gaussian noise is tractable

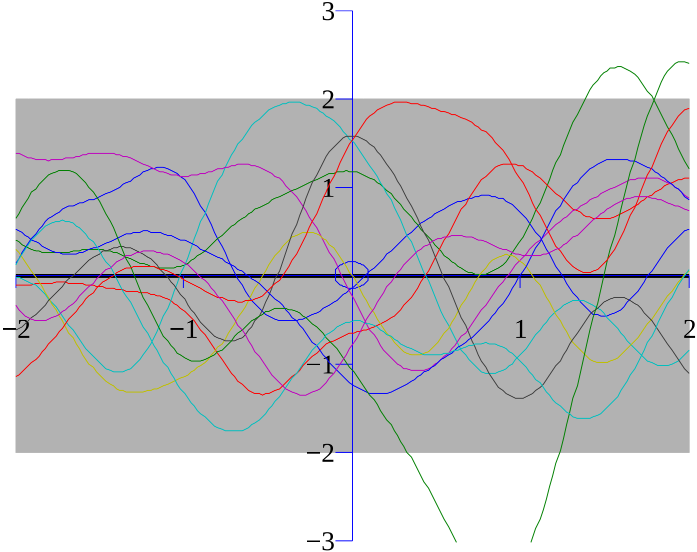
- So far we assumed: $\mathbf{f} = f(\mathbf{X})$
- Assuming that we only observe noisy versions $\mathbf{y}$ of the true outputs $\mathbf{f}$:

$$\mathbf{y} = f(\mathbf{X}) + \epsilon, \;\; \epsilon \sim \mathcal{N}(0, \beta^{-1})$$

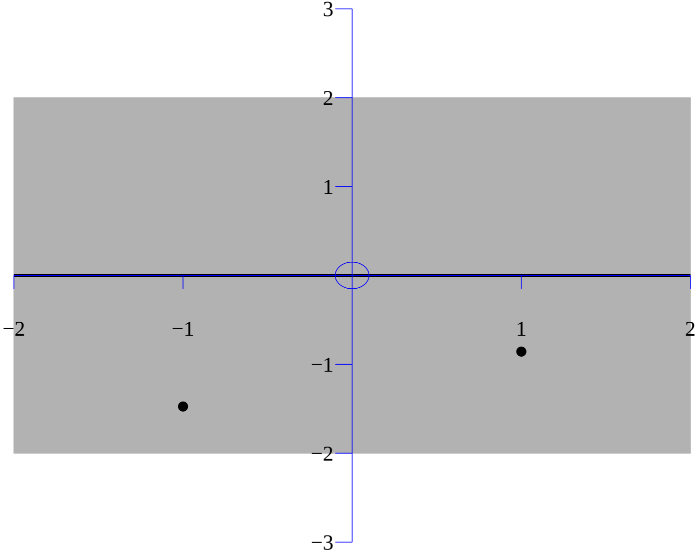
Fitting the data



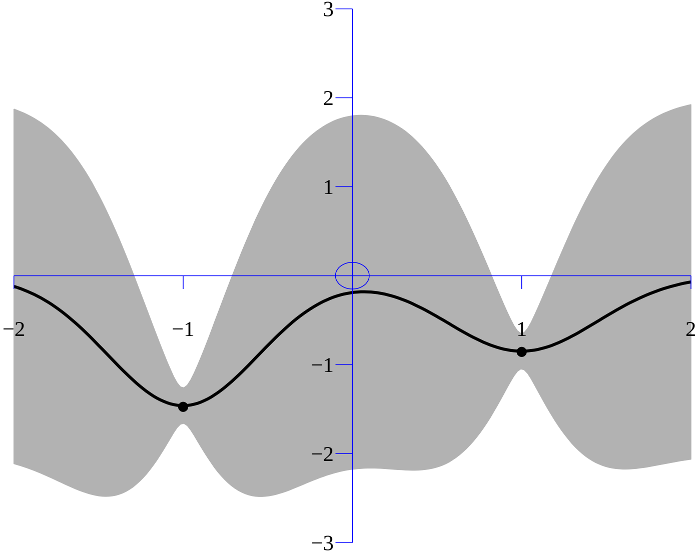
Fitting the data - Prior Samples



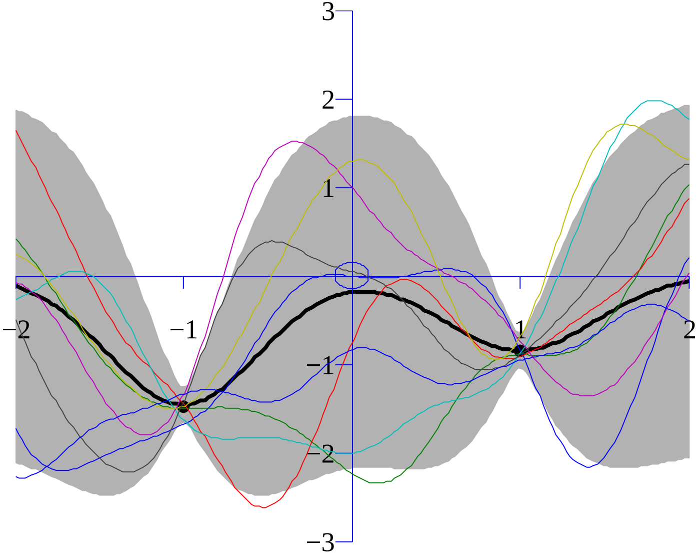
Fitting the data



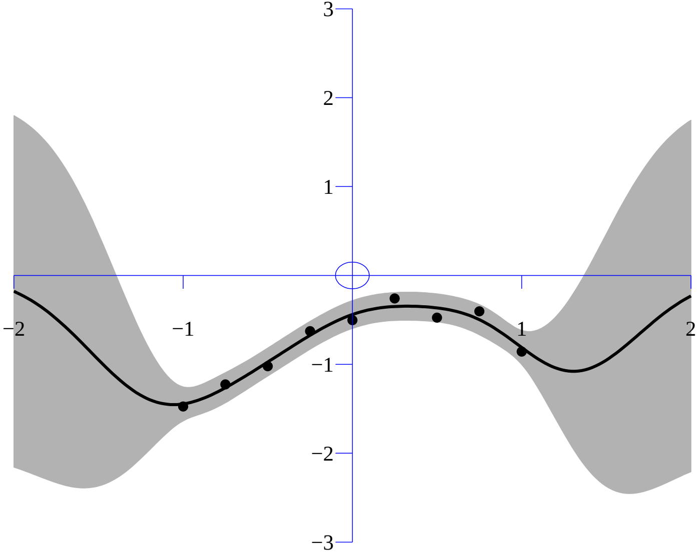
Fitting the data



Fitting the data - Posterior samples

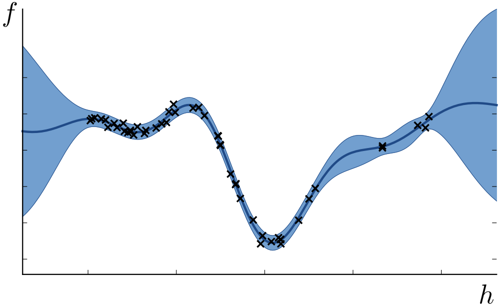


Fitting the data



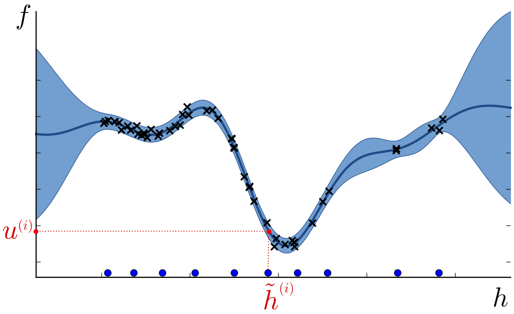
Inducing points

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



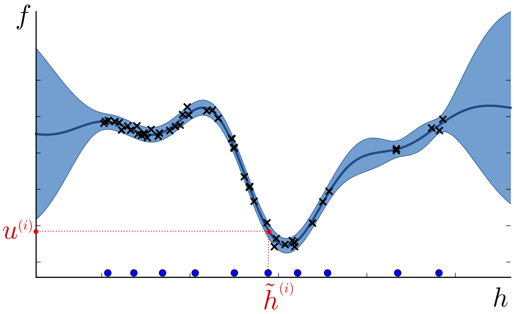
Inducing points

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



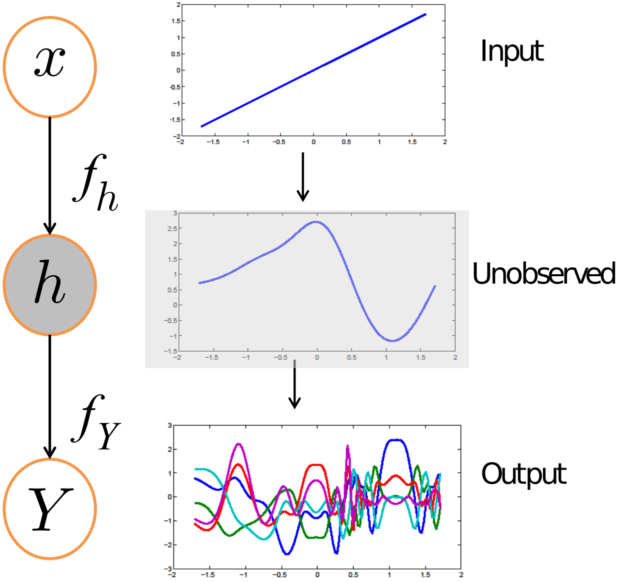
Inducing points

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



Outline

Sampling from a deep GP



Regularization solution: approximate Bayesian framework

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]

    [1] M. Titsias. "Variational Learning of Inducing Variables in Sparse GPs", AISTATS 2009

    [2] M. Titsias, N. Lawrence. "Bayesian GP-LVM", AISTATS 2010

    [3] A. Damianou*, M. Titsias*, N. Lawrence. "Variational Inference for Latent Variables and Uncertain Inputs in Gaussian Processes". JMLR 2015 (under review)
  - *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: $p(y|x) = \int_{h_2} \left(p(y|h_2) \int_{h_1} p(h_2|h_1) p(h_1|x) \right)$
- $p(h_2|x) \quad = \int_{h_1, f_2} \quad p(h_2|f_2) p(f_2|h_1) \quad p(h_1|x)$

Direct marginalisation of h is intractable

- Objective: $p(y|x) = \int_{h_2} \left(p(y|h_2) \int_{h_1} p(h_2|h_1) p(h_1|x) \right)$
- $p(h_2|x) \quad = \int_{h_1, f_2} \quad p(h_2|f_2) p(f_2|h_1) \quad p(h_1|x)$

Direct marginalisation of h is intractable

- Objective: $p(y|x) = \int_{h_2} \left(p(y|h_2) \int_{h_1} p(h_2|h_1) p(h_1|x) \right)$
- $p(h_2|x) \quad = \int_{h_1, f_2} p(h_2|f_2) \underbrace{p(f_2|h_1)}_{\substack{\text{contains} \\ (k(h_1,h_1))^{-1}}} p(h_1|x)$

Direct marginalisation of h is intractable

- Objective: $p(y|x) = \int_{h_2} \left(p(y|h_2) \int_{h_1} p(h_2|h_1) p(h_1|x) \right)$
- $p(h_2|x) \quad = \int_{h_1, f_2} \quad p(h_2|f_2) p(f_2|h_1) \quad p(h_1|x)$
- $p(h_2|x, \tilde{h}_1) = \int_{h_1, f_2, u_2} p(h_2|f_2) p(f_2|u_2, h_1) p(u_2|\tilde{h}_1) p(h_1|x)$

Direct marginalisation of h is intractable

- Objective: $p(y|x) = \int_{h_2} \Big(p(y|h_2) \int_{h_1} p(h_2|h_1) p(h_1|x) \Big)$
- $p(h_2|x) \quad = \int_{h_1, f_2} \quad p(h_2|f_2) p(f_2|h_1) \quad p(h_1|x)$
- $p(h_2|x, \tilde{h}_1) = \int_{h_1, f_2, u_2} p(h_2|f_2) p(f_2|u_2, h_1) p(u_2|\tilde{h}_1) p(h_1|x)$
- $\log p(h_2|x, \tilde{h}_1) \geq \int_{h_1, f_2, u_2} \mathcal{Q} \log \frac{p(h_2|f_2) p(f_2|u_2, h_1) p(u_2|\tilde{h}_1) p(h_1|x)}{\mathcal{Q}}$

Direct marginalisation of h is intractable

- Objective: $p(y|x) = \int_{h_2} \left(p(y|h_2) \int_{h_1} p(h_2|h_1) p(h_1|x) \right)$
- $p(h_2|x) \quad = \int_{h_1, f_2} \quad p(h_2|f_2) p(f_2|h_1) \quad p(h_1|x)$
- $p(h_2|x, \tilde{h}_1) = \int_{h_1, f_2, u_2} p(h_2|f_2) p(f_2|u_2, h_1) p(u_2|\tilde{h}_1) p(h_1|x)$
- $\log p(h_2|x, \tilde{h}_1) \geq \int_{h_1, f_2, u_2} \mathcal{Q} \log \frac{p(h_2|f_2) \cancel{p(f_2|u_2, h_1)} p(u_2|\tilde{h}_1) p(h_1|x)}{\mathcal{Q} = \cancel{p(f_2|u_2, h_1)} q(u_2) q(h_1)}$

Direct marginalisation of h is intractable

- Objective: $p(y|x) = \int_{h_2} \Big(p(y|h_2)\int_{h_1} p(h_2|h_1)p(h_1|x)\Big)$
- $p(h_2|x) \quad = \int_{h_1,f_2} \; p(h_2|f_2)p(f_2|h_1) \; p(h_1|x)$
- $p(h_2|x,\tilde{h}_1) = \int_{h_1,f_2,u_2} p(h_2|f_2)p(f_2|u_2,h_1)p(u_2|\tilde{h}_1)p(h_1|x)$
- $\log p(h_2|x,\tilde{h}_1) \geq \int_{h_1,f_2,u_2} \mathcal{Q} \log \frac{p(h_2|f_2)\cancel{p(f_2|u_2,h_1)}p(u_2|\tilde{h}_1)p(h_1|x)}{\mathcal{Q}=\cancel{p(f_2|u_2,h_1)}q(u_2)q(h_1)}$
- $\log p(h_2|x,\tilde{h}_1) \geq \int_{h_1,f_2,u_2} \mathcal{Q} \log \frac{p(h_2|f_2)p(u_2|\tilde{h}_1)p(h_1|x)}{q(u_2)q(h_1)}$

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

Direct marginalisation of h is intractable

- Objective: $p(y|x) = \int_{h_2} \Big(p(y|h_2) \int_{h_1} p(h_2|h_1) p(h_1|x) \Big)$
- $p(h_2|x) \quad = \int_{h_1, f_2} \; p(h_2|f_2) p(f_2|h_1) \; p(h_1|x)$
- $p(h_2|x, \tilde{h}_1) = \int_{h_1, f_2, u_2} p(h_2|f_2) p(f_2|u_2, h_1) p(u_2|\tilde{h}_1) p(h_1|x)$
- $\log p(h_2|x, \tilde{h}_1) \geq \int_{h_1, f_2, u_2} \mathcal{Q} \log \frac{p(h_2|f_2)\cancel{p(f_2|u_2, h_1)}p(u_2|\tilde{h}_1)p(h_1|x)}{\mathcal{Q} = \cancel{p(f_2|u_2, h_1)}q(u_2)q(h_1)}$
- $\log p(h_2|x, \tilde{h}_1) \geq \int_{h_1, f_2, u_2} \mathcal{Q} \log \frac{p(h_2|f_2)p(u_2|\tilde{h}_1)p(h_1|x)}{q(u_2)q(h_1)}$

$p(u_2|\tilde{h}_1)$ 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 $l+1$.

Properties of the bound (unsupervised case)

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

$$\mathcal{F} = \overbrace{\sum_{l=2}^{L+1} \left\langle \mathcal{L}_l \right\rangle_{\mathcal{Q}}}^{\text{Data fit}} - \sum_{l=2}^{L+1} \mathsf{KL}\left(q(\mathbf{u}_l) \,\|\, p(\mathbf{u}_l)\right)$$

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

where

$$\mathcal{L}_l = \sum_{n=1}^{N} \left[\sum_{q=1}^{Q_l} \log \mathcal{N}\left(h_l^{(n,q)} | \mathbf{k}_l^{(n,:)} \mathbf{K}^{-1} \mathbf{u}_l^{(:,q)}, \beta_l^{-1} \mathbf{I}\right) \overbrace{- \frac{\beta_l^{-1} \tilde{\mathbf{k}}_l^{(n)}}{2}}^{\text{Regularisation}} \right]$$

Properties of the bound (unsupervised case)

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

$$\mathcal{F} = \overbrace{\sum_{l=2}^{L+1} \left\langle \mathcal{L}_l \right\rangle_{\mathcal{Q}}}^{\text{Data fit}} - \sum_{l=2}^{L+1} \mathsf{KL}\left(q(\mathbf{u}_l) \parallel p(\mathbf{u}_l)\right)$$

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

where

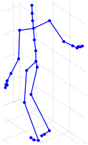
$$\mathcal{L}_l = \sum_{n=1}^{N} \left[\sum_{q=1}^{Q_l} \log \mathcal{N}\left(h_l^{(n,q)} | \mathbf{k}_l^{(n,:)} \mathbf{K}^{-1} \mathbf{u}_l^{(:,q)}, \beta_l^{-1}\mathbf{I}\right) \overbrace{- \frac{\beta_l^{-1} \tilde{\mathbf{k}}_l^{(n)}}{2}}^{\text{Regularisation}} \right]$$

- All terms factorise w.r.t data points.

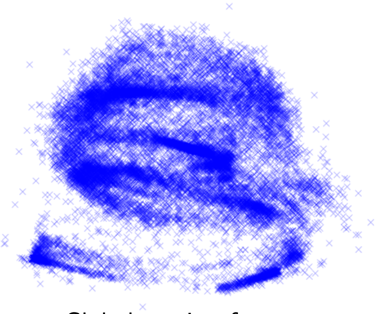
Stochastic Variational Inference

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

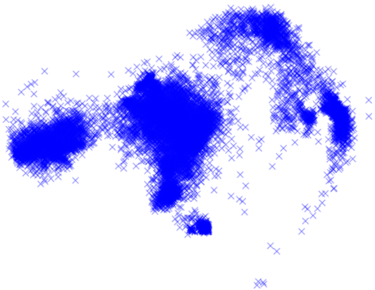
[Hensman, Damianou and Lawrence, AISTATS (Late-breaking) 2014]



Hidden space projections (20K mocap examples):



Global motion features

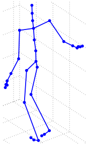


Clustered motion features

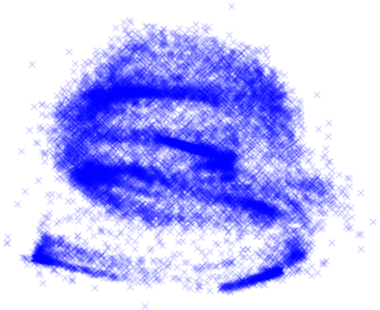
Stochastic Variational Inference

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

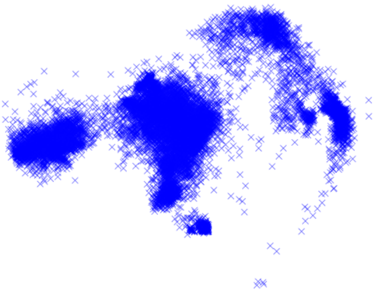
[Hensman, Damianou and Lawrence, AISTATS (Late-breaking) 2014]



Hidden space projections (20K mocap examples):



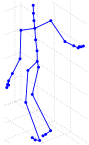
Global motion features



Clustered motion features

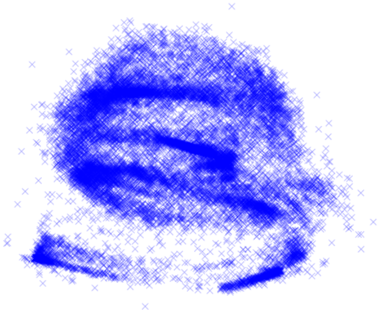
Stochastic Variational Inference

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

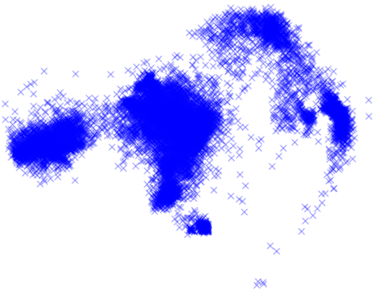


[Hensman, Damianou and Lawrence, AISTATS (Late-breaking) 2014]

Hidden space projections (20K mocap examples):



Global motion features



Clustered motion features

Properties of the bound (unsupervised case)

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

$$\mathcal{F} = \overbrace{\sum_{l=2}^{L+1} \left\langle \mathcal{L}_l \right\rangle_{\mathcal{Q}}}^{\text{Data fit}} - \sum_{l=2}^{L+1} \mathsf{KL}\left(q(\mathbf{u}_l) \parallel p(\mathbf{u}_l)\right)$$

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

where

$$\mathcal{L}_l = \sum_{n=1}^{N} \left[\sum_{q=1}^{Q_l} \log \mathcal{N}\left(h_l^{(n,q)} | \mathbf{k}_l^{(n,:)} \mathbf{K}^{-1} \mathbf{u}_l^{(:,q)}, \beta_l^{-1} \mathbf{I} \right) \overbrace{- \frac{\beta_l^{-1} \tilde{\mathbf{k}}_l^{(n)}}{2}}^{\text{Regularisation}} \right]$$

- All terms factorise w.r.t data points.

Properties of the bound (unsupervised case)

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

$$\mathcal{F} = \overbrace{\sum_{l=2}^{L+1} \langle \mathcal{L}_l \rangle_{\mathcal{Q}}}^{\text{Data fit}} - \sum_{l=2}^{L+1} \mathsf{KL}\left(q(\mathbf{u}_l) \,\|\, p(\mathbf{u}_l)\right)$$

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

where

$$\mathcal{L}_l = \sum_{n=1}^{N} \left[\sum_{q=1}^{Q_l} \log \mathcal{N}\left(h_l^{(n,q)} | \mathbf{k}_l^{(n,:)} \mathbf{K}^{-1} \mathbf{u}_l^{(:,q)}, \beta_l^{-1} \mathbf{I}\right) \overbrace{- \frac{\beta_l^{-1} \tilde{\mathbf{k}}_l^{(n)}}{2}}^{\text{Regularisation}} \right]$$

- All terms factorise w.r.t data points.
- We can additionally collapse $q(\mathbf{u})$

"Collapse" $q(\mathbf{u})$

- Collapsing $q(\mathbf{u})$ 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])

[1] Y. Gal, M. van der Wilk, C. E. Rasmussen, NIPS 2014

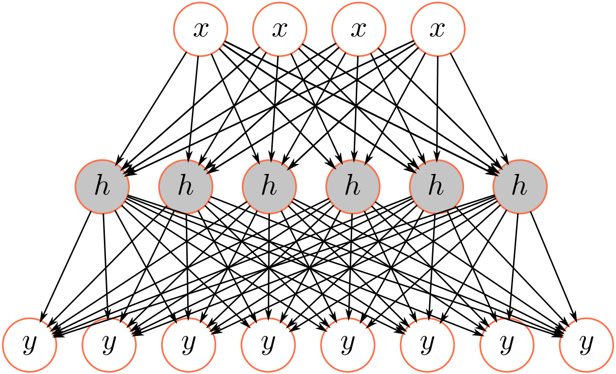
[2] Z. Dai, A. Damianou, J. Hensman, N. Lawrence, NIPS workshops, 2014

Outline

Automatic structure discovery: outline

Tools:

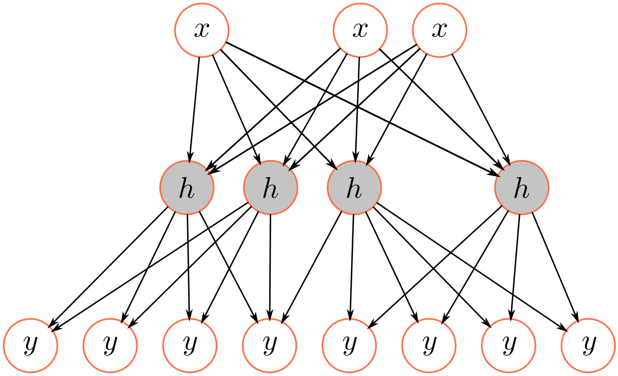
- ARD: Eliminate uncessary nodes/connections
- MRD: Conditional independencies
- Approximating evidence: Number of layers (?)



Automatic structure discovery: outline

Tools:

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

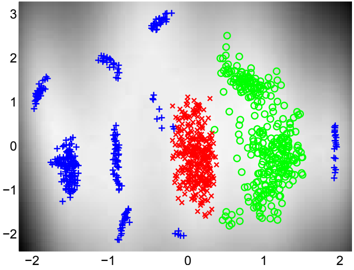


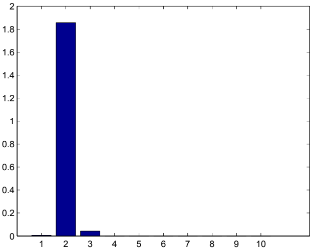
Automatic dimensionality detection

- Achieved by employing *automatic relevance determination (ARD)* priors for the mapping f.
- $f \sim \mathcal{GP}(\mathbf{0}, k_f)$ with:

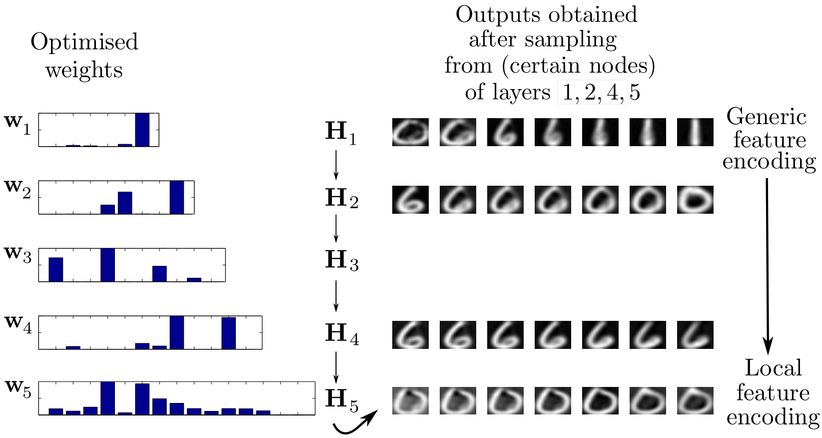
$$k_f\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \sigma^2 \exp\left(-\frac{1}{2}\sum_{q=1}^{Q} w^{(q)}\left(x^{(i,q)} - x^{(j,q)}\right)^2\right)$$

- Example:





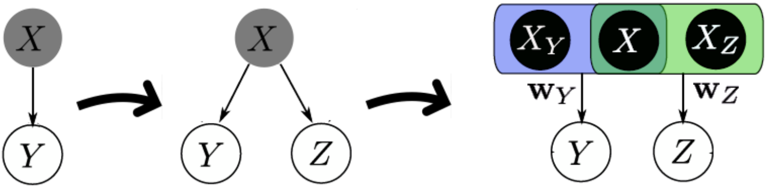
Deep GP: MNIST example



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

[Damianou and Lawrence, AISTATS 2013]

Manifold Relevance Determination

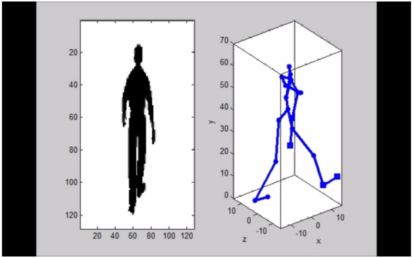


- Observations come into two different *views*: Y and Z.
- The latent space is segmented into parts private to Y, private to Z and shared between Y and Z.
- Used for data consolidation and discovering commonalities.

[Damianou et al. ICML 2012]

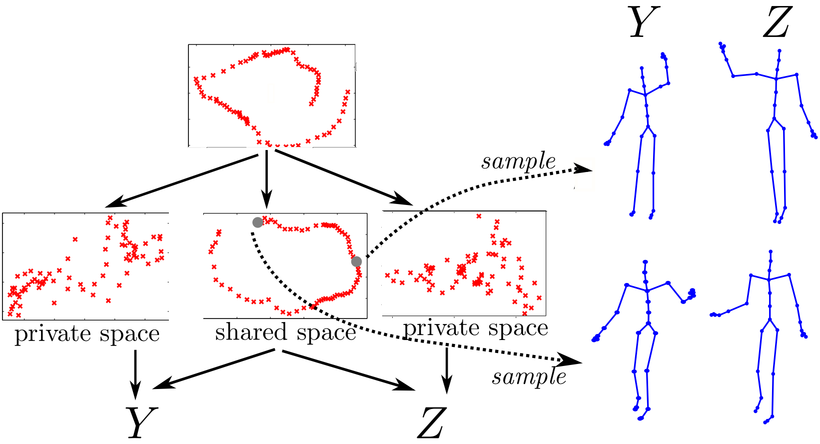
MRD examples

Example 1: Motion capture / silhouette



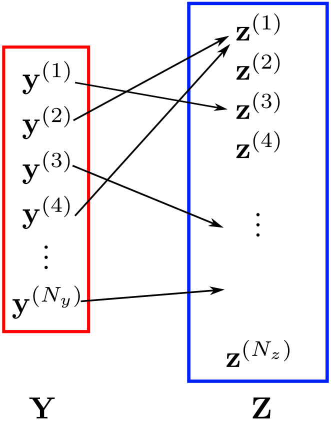
Example 2: Faces data https://youtu.be/rIPX3CIOhKY

Deep GPs: Another multi-view example



Automatic Alignment of Data-sets (work in progress...)

Alignment of views (e.g. video-audio, measurements-timestamps)



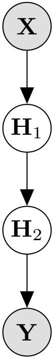
NP-hard problem.

Automatic Alignment of Data-sets

Greedy approach:

- Given fully aligned instances collected in $\mathbf{D}_0 = \{\mathbf{Y}, \mathbf{Z}\}$ train a factorised MRD model
- Determine the segmentation $\mathbf{X} = \left[\mathbf{X}^Y, \mathbf{X}^{Y,Z}, \mathbf{X}^Z\right]$
- $\mathbf{D} \leftarrow \mathbf{D}_0, \quad \mathbf{D}_* \leftarrow \{\mathbf{Y}_*, \mathbf{Z}_*\}$
- For each test instance $\mathbf{y}_*$:
  - Compute $\mathbf{x}_* \approx p(\mathbf{x}_* | \mathbf{y}_*, \mathbf{D})$
  - $\mathbf{z}_* = \underset{\mathbf{z}}{\operatorname{argmax}}\, p(\mathbf{z} | \mathbf{x}_*^{Y,Z}, \mathbf{x}_*^Z, \mathbf{D}), \quad \mathbf{z} \in \mathbf{D}_*$
  - Update the global parameters of the model given $\{\mathbf{y}_*, \mathbf{z}_*\}$
  - $\mathbf{D} \leftarrow [\mathbf{D}, \{\mathbf{y}_*, \mathbf{z}_*\}], \quad \mathbf{D}_* \leftarrow \mathbf{D}_* - \{\mathbf{y}_*, \mathbf{z}_*\}$.

Supervised learning



- The variational distribution on the *top layer* now is *coupled across datapoints:*

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

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

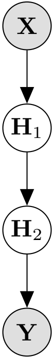
- Now $\mathbf{S}_1$ is a *full* $N \times N$ matrix!
- Reparametrisation

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

$$\mathbf{S}_1^{(q)} = \left(\mathbf{K}_x^{-1} + \boldsymbol{\lambda}^{(q)}\right)^{-1}$$

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

Supervised learning



- The variational distribution on the *top layer* now is *coupled across datapoints:*

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

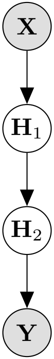
  (and small other changes in the bound...)

- Now $\mathbf{S}_1$ is a *full* $N \times N$ matrix!
- Reparametrisation [Opper and Archambeau 2009, Damianou et al. 2011]:

$$\mathbf{S}_1^{(q)} = \left(\mathbf{K}_x^{-1} + \boldsymbol{\lambda}^{(q)}\right)^{-1}$$

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

Supervised learning



- The variational distribution on the *top layer* now is *coupled across datapoints:*

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

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

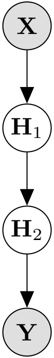
- Now $\mathbf{S}_1$ is a *full* $N \times N$ matrix!
- Reparametrisation

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

$$\mathbf{S}_1^{(q)} = \left(\mathbf{K}_x^{-1} + \boldsymbol{\lambda}^{(q)}\right)^{-1}$$

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

Supervised learning



- The variational distribution on the *top layer* now is *coupled across datapoints:*

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

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

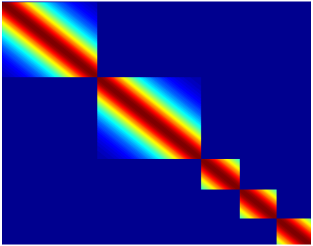
- Now $\mathbf{S}_1$ is a *full* $N \times N$ matrix!
- Reparametrisation [Opper and Archambeau 2009, Damianou et al. 2011]:

$$\mathbf{S}_1^{(q)} = \left(\mathbf{K}_x^{-1} + \boldsymbol{\lambda}^{(q)}\right)^{-1}$$

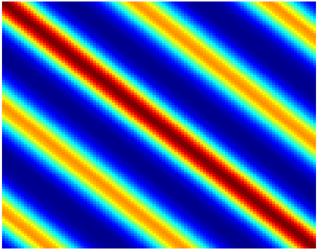
- Coupling the inputs gives rise to a powerful model for *multivariate timeseries / system identification*.

Dynamics

- Deterministic inputs in top layer $\Rightarrow$ can consider *any* kernel!
- 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

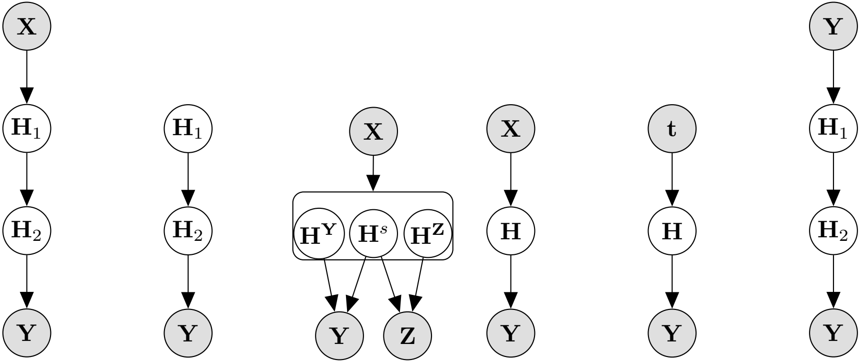


Model periodic data

- https://www.youtube.com/watch?v=i9TEoYxaBxQ (miss-America)
- https://www.youtube.com/watch?v=mUY1XHPnoCU (dog-treadmill)
- https://www.youtube.com/watch?v=fHDWloJtgk8 (mocap)

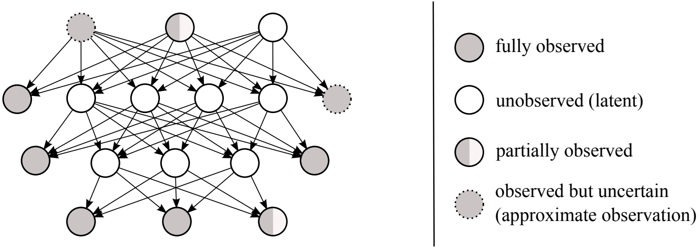
[Damianou et al., NIPS 2011]

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]

Partially observed inputs

Consider: observed, $\mathcal{O}$, and unobserved set, $\mathcal{U}$ from $\mathbf{X}$

Variational constraints:

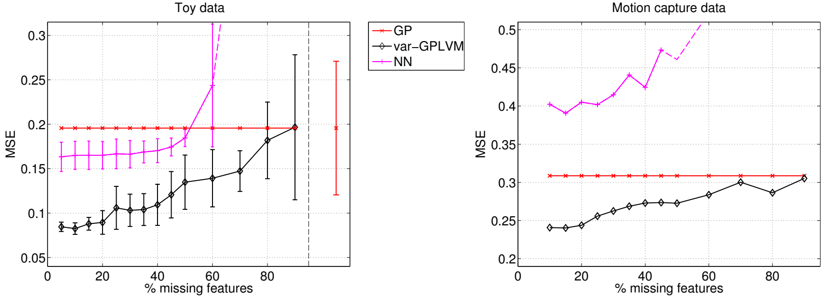
$$
\begin{aligned}
q(\mathbf{H}|\mathbf{X}, \{\mathcal{O}, \mathcal{U}\}) &= q(\mathbf{H}_{\mathcal{O}}|\mathbf{X}_{\mathcal{O}})q(\mathbf{H}_{\mathcal{U}}|\mathbf{X}_{\mathcal{U}}) \\
&= \prod_{n \in \mathcal{O}} \mathcal{N}\left(\mathbf{h}_{\mathcal{O}}^{(n)}|\mathbf{x}_{\mathcal{O}}^{(n)}, \epsilon \mathbf{I}\right) \prod_{n \in \mathcal{U}} \mathcal{N}\left(\mathbf{h}_{\mathcal{U}}^{(n)}|\boldsymbol{\mu}_{\mathcal{U}}^{(n)}, \mathbf{S}_{\mathcal{U}}^{(n)}\right), \quad \epsilon \to 0
\end{aligned}
$$

Algorithm (sketch):

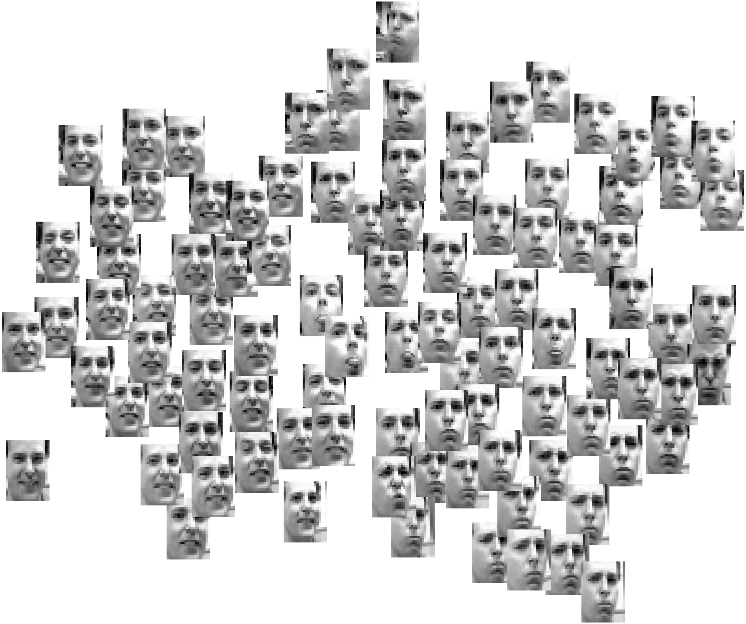
- Train on the fully observed set
- Impute unobserved values and obtain uncertainties $\mathbf{S}_{\mathcal{U}}$
- The predicted uncertainty now becomes input uncertainty in a variationally constrained model
- Recalibrate the new model which accounts for input uncertainty

Results

Partial observations are successfully taken into account, yielding better results in regression/classification.

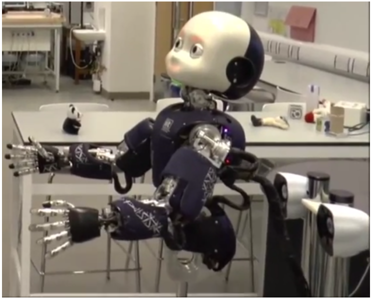


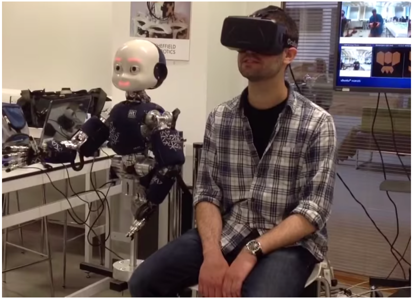
Autoencoder: Brendan faces



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.

Summary

- 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 $\Rightarrow$ 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

Thanks to Neil Lawrence, James Hensman, Michalis Titsias, Carl Henrik Ek.

References:

- N. D. Lawrence (2006) “The Gaussian process latent variable model” Technical Report no CS-06-03, The University of Sheffield, Department of Computer Science
- N. D. Lawrence (2006) “Probabilistic dimensional reduction with the Gaussian process latent variable model” (talk)
- C. E. Rasmussen (2008), “Learning with Gaussian Processes”, Max Planck Institute for Biological Cybernetics, Published: Feb. 5, 2008 (Videolectures.net)
- Carl Edward Rasmussen and Christopher K. I. Williams. Gaussian Processes for Machine Learning. MIT Press, Cambridge, MA, 2006. ISBN 026218253X.
- M. Titsias (2009), “Variational learning of inducing variables in sparse Gaussian processes”, AISTATS 2009
- A. Damianou, M. K. Titsias and N. D. Lawrence (2011), “Variational Gaussian process dynamical systems”, NIPS 2011
- A. Damianou, C. H. Ek, M. K. Titsias and N. D. Lawrence (2012), “Manifold Relevance Determination”, ICML 2012
- A. Damianou and N. D. Lawrence (2013), “Deep Gaussian processes”, AISTATS 2013
- A. Damianou and N. D. Lawrence (2015), “Semi-described and semi-supervised learning with Gaussian processes”, UAI 2015
- A. Damianou, C. H. Ek, L. Boorman, N. Lawrence, T. Prescott (2015), “A top-down approach for a synthetic autobiographical memory system”, Living Machines 2015
- Z. Dai, A. Damianou, J. Hensman and N. Lawrence. (2014) “Gaussian Process Models with Parallelization and GPU acceleration”, NIPS workshop on SE for ML, 2014
- A. Damianou*, M. Titsias*, N. Lawrence. “Variational Inference for Latent Variables and Uncertain Inputs in Gaussian Processes”. JMLR 2015 (under review)
- J. Hensman, A. Damianou and N. Lawrence (2014) “Deep Gaussian Processes for Large Datasets”, Late Breaking Poster, AISTATS 2014
- J. Hensman (2013), “Gaussian processes for Big Data”, UAI 2013

BACKUP SLIDES

MRD weights