Approximate Inference in Deep GPs

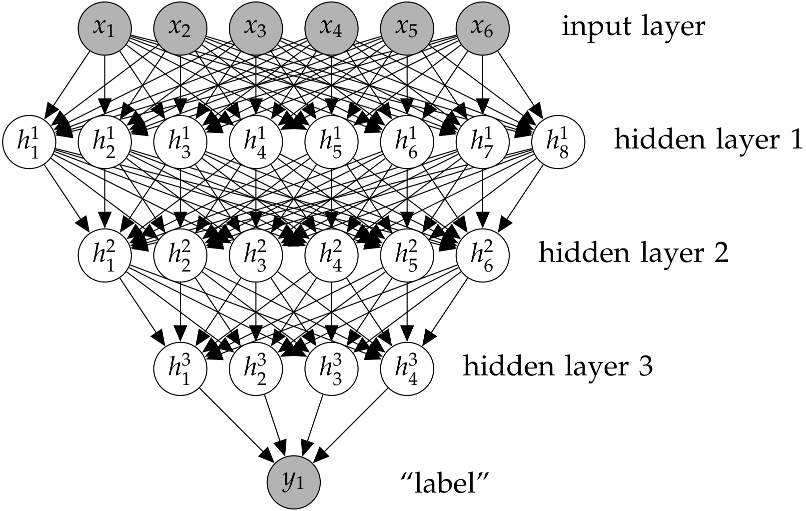
Neil D. Lawrence
Sheffield Institute of Translational Neuroscience and Department of Computer Science, University of Sheffield, U.K.

CSML, University College London

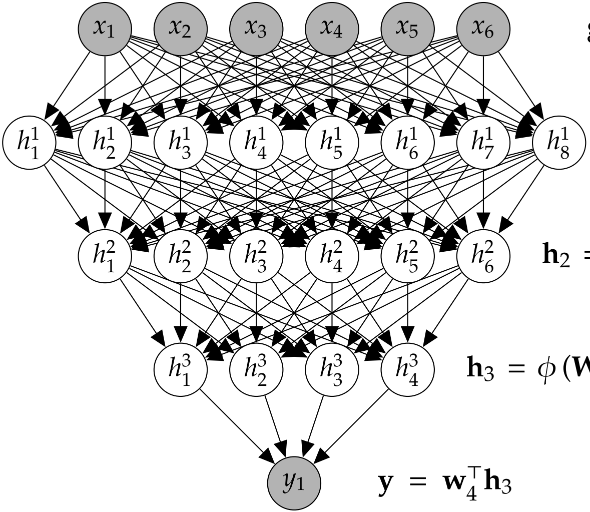
23rd October 2014

Outline

Deep Neural Network



Deep Neural Network



given $\mathbf{x}$

$$\mathbf{h}_1 = \phi(\mathbf{W}_1\mathbf{x})$$

$$\mathbf{h}_2 = \phi(\mathbf{W}_2\mathbf{h}_1)$$

$$\mathbf{h}_3 = \phi(\mathbf{W}_3\mathbf{h}_2)$$

$$\mathbf{y} = \mathbf{w}_4^\top\mathbf{h}_3$$

Mathematically

$$\begin{aligned}
\mathbf{h}_1 &= \phi\left(\mathbf{W}_1\mathbf{x}\right)\\
\mathbf{h}_2 &= \phi\left(\mathbf{W}_2\mathbf{h}_1\right)\\
\mathbf{h}_3 &= \phi\left(\mathbf{W}_3\mathbf{h}_2\right)\\
\mathbf{y} &= \mathbf{w}_4^\top\mathbf{h}_3
\end{aligned}$$

Overfitting

- Potential problem: if number of nodes in two adjacent layers is big, corresponding **W** is also very big and there is the potential to overfit.
- Proposed solution: "dropout".
- Alternative solution: parameterize **W** with its SVD.

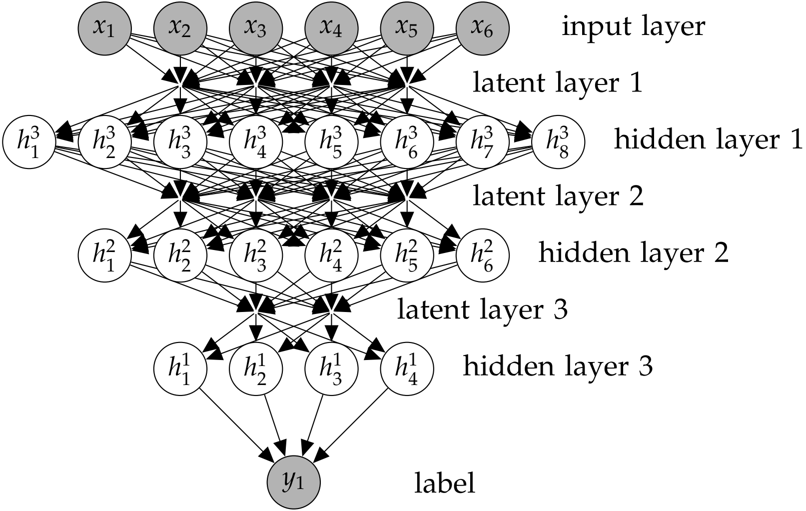
$$\mathbf{W} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^\top$$

or

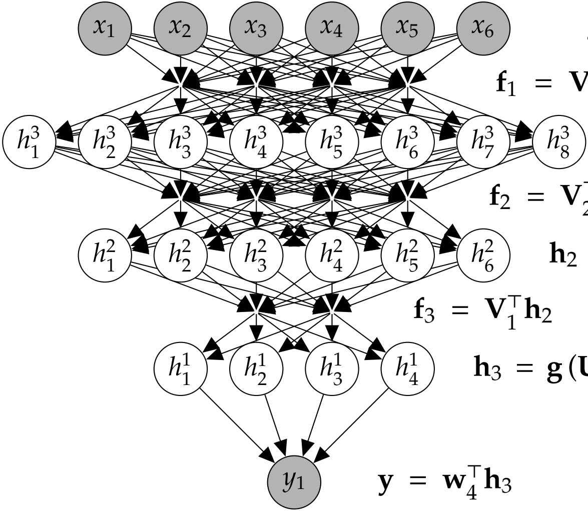
$$\mathbf{W} = \mathbf{U}\mathbf{V}^\top$$

where if $\mathbf{W} \in \Re^{k_1 \times k_2}$ then $\mathbf{U} \in \Re^{k_1 \times q}$ and $\mathbf{V} \in \Re^{k_2 \times q}$, i.e. we have a low rank matrix factorization for the weights.

Deep Neural Network



Deep Neural Network



given $\mathbf{x}$

$$\mathbf{f}_1 = \mathbf{V}_1^\top \mathbf{x}$$

$$\mathbf{h}_1 = \mathbf{g}\,(\mathbf{U}_1 \mathbf{f}_1)$$

$$\mathbf{f}_2 = \mathbf{V}_2^\top \mathbf{h}_3$$

$$\mathbf{h}_2 = \mathbf{g}\,(\mathbf{U}_2 \mathbf{f}_2)$$

$$\mathbf{f}_3 = \mathbf{V}_1^\top \mathbf{h}_2$$

$$\mathbf{h}_3 = \mathbf{g}\,(\mathbf{U}_3 \mathbf{f}_3)$$

$$\mathbf{y} = \mathbf{w}_4^\top \mathbf{h}_3$$

Mathematically

$$
\begin{aligned}
\mathbf{f}_1 &= \mathbf{V}_1^\top \mathbf{x} \\
\mathbf{h}_1 &= \phi\left(\mathbf{U}_1 \mathbf{f}_1\right) \\
\mathbf{f}_2 &= \mathbf{V}_2^\top \mathbf{h}_1 \\
\mathbf{h}_2 &= \phi\left(\mathbf{U}_2 \mathbf{f}_2\right) \\
\mathbf{f}_3 &= \mathbf{V}_3^\top \mathbf{h}_2 \\
\mathbf{h}_3 &= \phi\left(\mathbf{U}_3 \mathbf{f}_3\right) \\
\mathbf{y} &= \mathbf{w}_4^\top \mathbf{h}_3
\end{aligned}
$$

A Cascade of Neural Networks

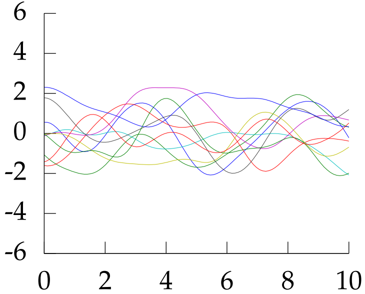
$$
\begin{aligned}
\mathbf{f}_1 &= \mathbf{V}_1^\top \mathbf{x} \\
\mathbf{f}_2 &= \mathbf{V}_2^\top \phi\left(\mathbf{U}_1 \mathbf{f}_1\right) \\
\mathbf{f}_3 &= \mathbf{V}_3^\top \phi\left(\mathbf{U}_2 \mathbf{f}_2\right) \\
\mathbf{y} &= \mathbf{w}_4^\top \mathbf{f}_3
\end{aligned}
$$

Replace Each Neural Network with a Gaussian Process

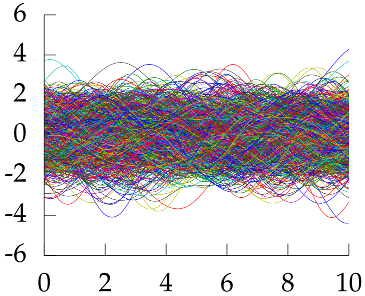
$$
\begin{aligned}
\mathbf{f}_1 &= \mathbf{f}(\mathbf{x}) \\
\mathbf{f}_2 &= \mathbf{f}(\mathbf{f}_1) \\
\mathbf{f}_3 &= \mathbf{f}(\mathbf{f}_2) \\
\mathbf{y} &= \mathbf{f}(\mathbf{f}_3)
\end{aligned}
$$

This is equivalent to Gaussian prior over weights and integrating out all parameters and taking width of each layer to infinity.

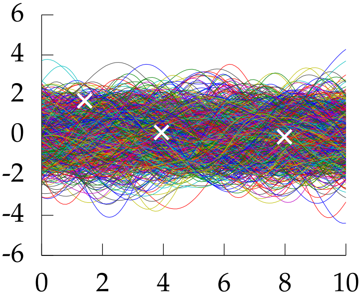
Gaussian Processes: Extremely Short Overview



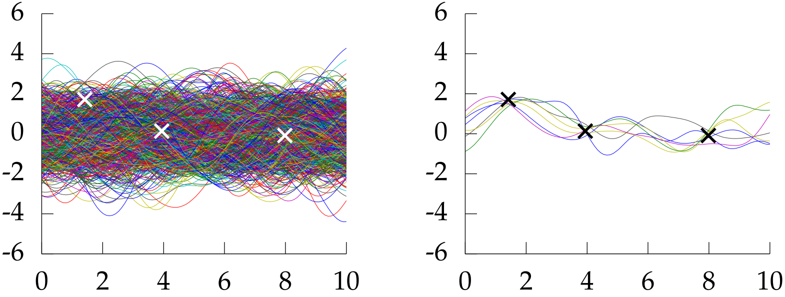
Gaussian Processes: Extremely Short Overview



Gaussian Processes: Extremely Short Overview



Gaussian Processes: Extremely Short Overview



Outline

Mathematically

- Composite *multivariate* function

$$\mathbf{g}(\mathbf{x}) = \mathbf{f}_5(\mathbf{f}_4(\mathbf{f}_3(\mathbf{f}_2(\mathbf{f}_1(\mathbf{x})))))$$

Why Deep?

- Gaussian processes give priors over functions.
- Elegant properties:
  - e.g. *Derivatives* of process are also Gaussian distributed (if they exist).
- For particular covariance functions they are 'universal approximators', i.e. all functions can have support under the prior.
- Gaussian derivatives might ring alarm bells.
- E.g. a priori they don't believe in function 'jumps'.

Process Composition

- From a process perspective: *process composition*.
- A (new?) way of constructing more complex *processes* based on simpler components.

Note: To retain *Kolmogorov consistency* introduce IBP priors over latent variables in each layer (Zhenwen Dai).

Analysis of Deep GPs

- Duvenaud et al. (2014) Duvenaud et al show that the derivative distribution of the process becomes more *heavy tailed* as number of layers increase.

Difficulty for Probabilistic Approaches

- Propagate a probability distribution through a non-linear mapping.
- Normalisation of distribution becomes intractable.

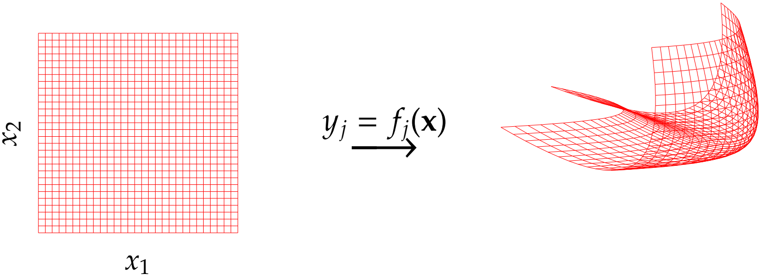


Figure : A three dimensional manifold formed by mapping from a two dimensional space to a three dimensional space.

Difficulty for Probabilistic Approaches

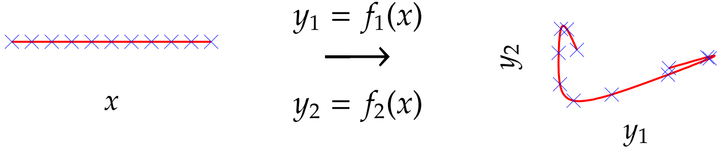


Figure : A string in two dimensions, formed by mapping from one dimension, x, line to a two dimensional space, $[y_1,\ y_2]$ using nonlinear functions $f_1(\cdot)$ and $f_2(\cdot)$.

Difficulty for Probabilistic Approaches

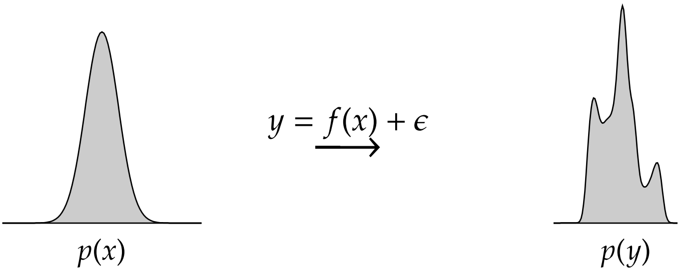


Figure : A Gaussian distribution propagated through a non-linear mapping. $y_i = f(x_i) + \epsilon_i$. $\epsilon \sim \mathcal{N}\left(0, 0.2^2\right)$ and $f(\cdot)$ uses RBF basis, 100 centres between -4 and 4 and $\ell = 0.1$. New distribution over y (right) is multimodal and difficult to normalize.

b

Outline

Introduction

Deep Gaussian Process Models

Flexible Parametric Approximation

Variational Compression

Conclusions

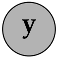
Inducing Variable Approximations

- Date back to (Williams and Seeger, 2001; Smola and Bartlett, 2001; Csató and Opper, 2002; Seeger et al., 2003; Snelson and Ghahramani, 2006). See Quiñonero Candela and Rasmussen (2005) for a review.
- We follow variational perspective of (Titsias, 2009).
- This is an augmented variable method, followed by a collapsed variational approximation (King and Lawrence, 2006; Hensman et al., 2012).

Augmented Variable Model: Not Wrong but Useful?

Augment standard model with a set of m new inducing variables, $\mathbf{u}$.

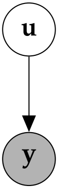
$$p(\mathbf{y}) = \int p(\mathbf{y}, \mathbf{u}) \mathrm{d}\mathbf{u}$$



Augmented Variable Model: Not Wrong but Useful?

Augment standard model with a set of m new inducing variables, $\mathbf{u}$.

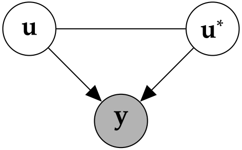
$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{u}$$



Augmented Variable Model: Not Wrong but Useful?

Important: Ensure inducing variables are *also* Kolmogorov consistent (we have m^* other inducing variables we are not *yet* using.)

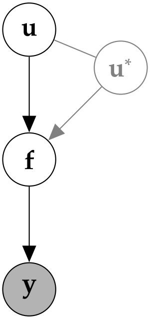
$$p(\mathbf{u}) = \int p(\mathbf{u}, \mathbf{u}^*) \mathrm{d}\mathbf{u}^*$$



Augmented Variable Model: Not Wrong but Useful?

Assume that relationship is through $\mathbf{f}$ (represents 'fundamentals'—push Kolmogorov consistency up to here).

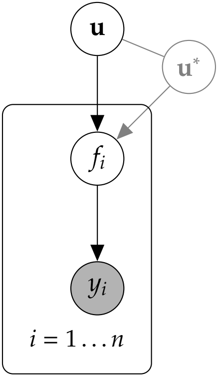
$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{f}\mathrm{d}\mathbf{u}$$



Augmented Variable Model: Not Wrong but Useful?

Convenient to assume factorization (*doesn't* invalidate model—think delta function as worst case).

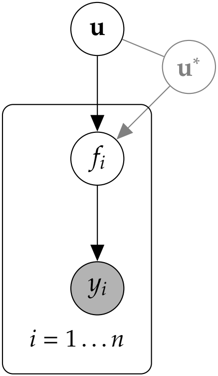
$$p(\mathbf{y}) = \int \prod_{i=1}^{n} p(y_i|f_i)p(\mathbf{f}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{f}\mathrm{d}\mathbf{u}$$



Augmented Variable Model: Not Wrong but Useful?

Focus on integral over $\mathbf{f}$.

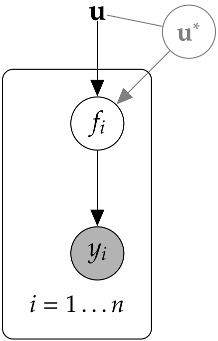
$$p(\mathbf{y}) = \int \int \prod_{i=1}^{n} p(y_i|f_i) p(\mathbf{f}|\mathbf{u}) \mathrm{d}\mathbf{f} p(\mathbf{u}) \mathrm{d}\mathbf{u}$$



Augmented Variable Model: Not Wrong but Useful?

Focus on integral over **f**.

$$p(\mathbf{y}|\mathbf{u}) = \int \prod_{i=1}^{n} p(y_i|f_i)p(\mathbf{f}|\mathbf{u})\mathrm{d}\mathbf{f}$$



Variational Bound on $p(\mathbf{y}|\mathbf{u})$

$$
\begin{aligned}
\log p(\mathbf{y}|\mathbf{u}) &= \log \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\mathbf{u}) \mathrm{d}\mathbf{f} \\
&= \int q(\mathbf{f}) \log \frac{p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\mathbf{u})}{q(\mathbf{f})} \mathrm{d}\mathbf{f} + \mathrm{KL}\left(q(\mathbf{f}) \,\|\, p(\mathbf{f}|\mathbf{y}, \mathbf{u})\right)
\end{aligned}
$$

Variational Bound on $p(\mathbf{y}|\mathbf{u})$

$$\log p(\mathbf{y}|\mathbf{u}) = \log \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})\mathrm{d}\mathbf{f}$$

$$= \int q(\mathbf{f}) \log \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})}{q(\mathbf{f})}\mathrm{d}\mathbf{f} + \mathrm{KL}\left(q(\mathbf{f})\,\|\,p(\mathbf{f}|\mathbf{y},\mathbf{u})\right)$$

(Titsias, 2009)

- Example, set $q(\mathbf{f}) = p(\mathbf{f}|\mathbf{u})$,

$$\log p(\mathbf{y}|\mathbf{u}) \geq \log \int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f})\mathrm{d}\mathbf{f}.$$

$$p(\mathbf{y}|\mathbf{u}) \geq \exp \int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f})\mathrm{d}\mathbf{f}.$$

Optimal Compression in Inducing Variables

- Maximizing lower bound minimizes the KL divergence (information gain):

$$\text{KL}\left(p(\mathbf{f}|\mathbf{u})\,\|\,p(\mathbf{f}|\mathbf{y},\mathbf{u})\right) = \int p(\mathbf{f}|\mathbf{u}) \log \frac{p(\mathbf{f}|\mathbf{u})}{p(\mathbf{f}|\mathbf{y},\mathbf{u})} \mathrm{d}\mathbf{u}$$

- This is minimized when the information stored about $\mathbf{y}$ is stored already in $\mathbf{u}$.
- The bound seeks an *optimal compression* from the *information gain* perspective.
- If $\mathbf{u} = \mathbf{f}$ bound is exact ($\mathbf{f}$ d-separates $\mathbf{y}$ from $\mathbf{u}$).

Choice of Inducing Variables

- Optimizing the bound directly not always practical.
- Free to choose whatever heuristics for the inducing variables.
- Can quantify which heuristics perform better through checking lower bound.

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

$$p(\mathbf{y}|\mathbf{u}) \geq \exp \int p(\mathbf{f}|\mathbf{u}) \log \prod_{i=1}^{n} p(y_i|f_i) \mathrm{d}\mathbf{f}.$$

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

$$p(\mathbf{y}|\mathbf{u}) \geq \exp \int p(\mathbf{f}|\mathbf{u}) \log \prod_{i=1}^{n} p(y_i|f_i) \mathrm{d}\mathbf{f}.$$

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

$$p(\mathbf{y}|\mathbf{u}) \geq \exp \int p(\mathbf{f}|\mathbf{u}) \sum_{i=1}^{n} \log p(y_i|f_i) \mathrm{d}\mathbf{f}.$$

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

$$p(\mathbf{y}|\mathbf{u}) \geq \exp \int p(\mathbf{f}|\mathbf{u}) \sum_{i=1}^{n} \log p(y_i|f_i) \mathrm{d}\mathbf{f}.$$

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

$$p(\mathbf{y}|\mathbf{u}) \geq \exp \sum_{i=1}^{n} \int p(f_i|\mathbf{u}) \log p(y_i|f_i) \mathrm{d}\mathbf{f}.$$

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

$$p(\mathbf{y}|\mathbf{u}) \geq \exp \sum_{i=1}^{n} \int p(f_i|\mathbf{u}) \log p(y_i|f_i) \mathrm{d}\mathbf{f}.$$

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

$$p(\mathbf{y}|\mathbf{u}) \geq \prod_{i=1}^{n} \exp \int p(f_i|\mathbf{u}) \log p(y_i|f_i) \mathrm{d}\mathbf{f}.$$

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

$$p(\mathbf{y}|\mathbf{u}) \geq \prod_{i=1}^{n} \exp \int p(f_i|\mathbf{u}) \log p(y_i|f_i) \mathrm{d}\mathbf{f}.$$

- Then the bound factorizes.

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

$$p(\mathbf{y}|\mathbf{u}) \geq \prod_{i=1}^{n} \exp \left\langle \log p(y_i|f_i) \right\rangle_{p(f_i|\mathbf{u})}$$

- Then the bound factorizes.

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

$$p(\mathbf{y}|\mathbf{u}) \geq \prod_{i=1}^{n} \exp \left\langle \log p(y_i|f_i) \right\rangle_{p(f_i|\mathbf{u})}$$

- Then the bound factorizes.
- Now need a choice of distributions for **f** and **y**|**f** ...

$$\mathbf{f}, \mathbf{u} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K_{ff}} & \mathbf{K_{fu}} \\ \mathbf{K_{uf}} & \mathbf{K_{uu}} \end{bmatrix}\right)$$

$$\mathbf{y}|\mathbf{f} = \prod_i \mathcal{N}\left(f, \sigma^2\right)$$

Outline

Gaussian $p(y_i|f_i)$

For Gaussian likelihoods:

$$\left\langle \log p(y_i|f_i) \right\rangle_{p(f_i|\mathbf{u})} = -\frac{1}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\left(y_i - \left\langle f_i \right\rangle\right)^2 - \frac{1}{2\sigma^2}\left(\left\langle f_i^2 \right\rangle - \left\langle f_i \right\rangle^2\right)$$

Gaussian $p(y_i|f_i)$

For Gaussian likelihoods:

$$\langle \log p(y_i|f_i) \rangle_{p(f_i|\mathbf{u})} = -\frac{1}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\left(y_i - \langle f_i \rangle\right)^2 - \frac{1}{2\sigma^2}\left(\left\langle f_i^2 \right\rangle - \langle f_i \rangle^2\right)$$

Implying:

$$p(y_i|\mathbf{u}) \geq \exp \langle \log c_i \rangle \mathcal{N}\left(y_i| \langle f_i \rangle, \sigma^2\right)$$

Gaussian Process Over $\mathbf{f}$ and $\mathbf{u}$

Define:

$$q_{i,i} = \text{var}_{p(f_i|\mathbf{u})}(f_i) = \left\langle f_i^2 \right\rangle_{p(f_i|\mathbf{u})} - \left\langle f_i \right\rangle_{p(f_i|\mathbf{u})}^2$$

We can write:

$$c_i = \exp\left(-\frac{q_{i,i}}{2\sigma^2}\right)$$

If joint distribution of $p(\mathbf{f}, \mathbf{u})$ is Gaussian then:

$$q_{i,i} = k_{i,i} - \mathbf{k}_{i,\mathbf{u}}^\top \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{k}_{i,\mathbf{u}}$$

c_i is not a function of $\mathbf{u}$ but *is* a function of $\mathbf{X}_{\mathbf{u}}$.

Total Conditional Variance

- The sum of $q_{i,i}$ is the *total conditional variance*.
- If conditional density $p(\mathbf{f}|\mathbf{u})$ is Gaussian then it has covariance
$$\mathbf{Q} = \mathbf{K}_{\mathbf{ff}} - \mathbf{K}_{\mathbf{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{uf}}$$
- $\text{tr}\left(\mathbf{Q}\right) = \sum_i q_{i,i}$ is known as total variance.
- Because it is on conditional distribution we call it *total conditional variance*.

Capacity of a Density

- Measure the 'capacity of a density'.
- Determinant of covariance represents 'volume' of density.
- log determinant is entropy: sum of *log* eigenvalues of covariance.
- trace of covariance is total variance: sum of eigenvalues of covariance.
- $\lambda > \log \lambda$ then total conditional variance upper bounds entropy.

Alternative View

Exponentiated total variance bounds determinant.

$$|\mathbf{Q}| < \exp \operatorname{tr}(\mathbf{Q})$$

Because

$$\prod_{i=1}^{k} \lambda_i < \prod_{i=1}^{k} \exp(\lambda_i)$$

where $\{\lambda_i\}_{i=1}^{k}$ are the *positive* eigenvalues of $\mathbf{Q}$ This in turn implies

$$|\mathbf{Q}| < \prod_{i=1}^{k} \exp(q_{i,i})$$

Communication Channel

- Conditional density $p(\mathbf{f}|\mathbf{u})$ can be seen as a *communication channel*.
- Normally we have:

$$\text{Transmitter} \xrightarrow{\mathbf{u}} \underset{\text{Channel}}{p(\mathbf{f}|\mathbf{u})} \xrightarrow{\mathbf{f}} \text{Receiver}$$

  and we control $p(\mathbf{u})$ (the source density).
- *Here* we can also control the transmission channel $p(\mathbf{f}|\mathbf{u})$.

Lower Bound on Likelihood

Substitute variational bound into marginal likelihood:

$$p(\mathbf{y}) \geq \prod_{i=1}^{n} c_i \int \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f} \right\rangle, \sigma^2\mathbf{I}\right) p(\mathbf{u}) \mathrm{d}\mathbf{u}$$

Note that:

$$\left\langle \mathbf{f} \right\rangle_{p(\mathbf{f}|\mathbf{u})} = \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}$$

is *linearly* dependent on $\mathbf{u}$.

Deterministic Training Conditional

Making the marginalization of $\mathbf{u}$ straightforward. In the Gaussian case:

$$p(\mathbf{u}) = \mathcal{N}\left(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}}\right)$$

$$\int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{u} \geq \prod_{i=1}^{n} c_i \int \mathcal{N}\left(\mathbf{y}|\mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}, \sigma^2\right)\mathcal{N}\left(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}}\right)\mathrm{d}\mathbf{u}$$

Deterministic Training Conditional

Making the marginalization of $\mathbf{u}$ straightforward. In the Gaussian case:

$$p(\mathbf{u}) = \mathcal{N}\left(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}}\right)$$

$$\int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{u} \geq \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}\right)$$

Deterministic Training Conditional

Making the marginalization of $\mathbf{u}$ straightforward. In the Gaussian case:

$$p(\mathbf{u}) = \mathcal{N}\left(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}}\right)$$

$$\int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{u} \geq \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}\right)$$

Maximize log of the bound to find covariance function parameters,

$$L \geq \sum_{i=1}^{n} \log c_i + \log \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f},}\right)$$

Deterministic Training Conditional

Making the marginalization of $\mathbf{u}$ straightforward. In the Gaussian case:

$$p(\mathbf{u}) = \mathcal{N}\left(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}}\right)$$

$$\int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{u} \geq \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}\right)$$

Maximize log of the bound to find covariance function parameters,

$$L \geq \sum_{i=1}^{n} \log c_i + \log \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f},}\right)$$

Deterministic Training Conditional

Making the marginalization of $\mathbf{u}$ straightforward. In the Gaussian case:

$$p(\mathbf{u}) = \mathcal{N}\left(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}}\right)$$

$$\int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{u} \geq \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}\right)$$

Maximize log of the bound to find covariance function parameters,

$$L \approx \log \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f},}\right)$$

- If the bound is normalized, the c_i terms are removed.

Deterministic Training Conditional

Making the marginalization of $\mathbf{u}$ straightforward. In the Gaussian case:

$$p(\mathbf{u}) = \mathcal{N}\left(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}}\right)$$

$$\int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{u} \geq \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}\right)$$

Maximize log of the bound to find covariance function parameters,

- If the bound is normalized, the c_i terms are removed.
- This results in the projected process approximation (Rasmussen and Williams, 2006) or DTC (Quiñonero Candela and Rasmussen, 2005). Proposed by (Smola and Bartlett, 2001; Seeger et al., 2003; Csató and Opper, 2002; Csató, 2002).

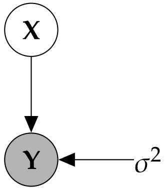
Selecting Data Dimensionality

- GP-LVM Provides probabilistic non-linear dimensionality reduction.
- How to select the dimensionality?
- Need to estimate marginal likelihood.
- In standard GP-LVM it increases with increasing q.

Integrate Mapping Function and Latent Variables

Bayesian GP-LVM

- Start with a standard GP-LVM.

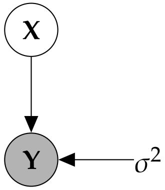


$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

Integrate Mapping Function and Latent Variables

Bayesian GP-LVM

- Start with a standard GP-LVM.
- Apply standard latent variable approach:
  - Define Gaussian prior over *latent space*, **X**.

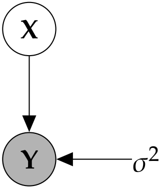


$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

Integrate Mapping Function and Latent Variables

Bayesian GP-LVM

- Start with a standard GP-LVM.
- Apply standard latent variable approach:
  - Define Gaussian prior over *latent space*, **X**.
  - Integrate out *latent variables*.



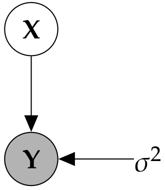
$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

$$p(\mathbf{X}) = \prod_{j=1}^{q} \mathcal{N}\left(\mathbf{x}_{:,j}|\mathbf{0}, \alpha_i^{-2}\mathbf{I}\right)$$

Integrate Mapping Function and Latent Variables

Bayesian GP-LVM

- Start with a standard GP-LVM.
- Apply standard latent variable approach:
  - Define Gaussian prior over *latent space*, $\mathbf{X}$.
  - Integrate out *latent variables*.
  - Unfortunately integration is intractable.



$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_{:,j}|\mathbf{0}, \mathbf{K}\right)$$

$$p(\mathbf{X}) = \prod_{j=1}^{q} \mathcal{N}\left(\mathbf{x}_{:,j}|\mathbf{0}, \alpha_i^{-2}\mathbf{I}\right)$$

$$p(\mathbf{Y}|\boldsymbol{\alpha}) = ??$$

Standard Variational Approach Fails

- Standard variational bound has the form:

$$\mathcal{L} = \left\langle \log p(\mathbf{y}|\mathbf{X}) \right\rangle_{q(\mathbf{X})} + \mathrm{KL}\left(q(\mathbf{X}) \,\|\, p(\mathbf{X}) \right)$$

Standard Variational Approach Fails

- Standard variational bound has the form:

$$\mathcal{L} = \left\langle \log p(\mathbf{y}|\mathbf{X}) \right\rangle_{q(\mathbf{X})} + \mathrm{KL}\left(q(\mathbf{X}) \,\|\, p(\mathbf{X})\right)$$

- Requires expectation of $\log p(\mathbf{y}|\mathbf{X})$ under $q(\mathbf{X})$.

$$\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^\top \left(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}\right)^{-1} \mathbf{y} - \frac{1}{2}\log\left|\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}\right| - \frac{n}{2}\log 2\pi$$

Standard Variational Approach Fails

- Standard variational bound has the form:

$$\mathcal{L} = \left\langle \log p(\mathbf{y}|\mathbf{X}) \right\rangle_{q(\mathbf{X})} + \mathrm{KL}\left(q(\mathbf{X}) \,\|\, p(\mathbf{X})\right)$$

- Requires expectation of $\log p(\mathbf{y}|\mathbf{X})$ under $q(\mathbf{X})$.

$$\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^{\top}\left(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{y} - \frac{1}{2}\log\left|\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}\right| - \frac{n}{2}\log 2\pi$$

- Extremely difficult to compute because $\mathbf{K}_{\mathbf{f},\mathbf{f}}$ is dependent on $\mathbf{X}$ and appears in the inverse.

Variational Bayesian GP-LVM

- Consider collapsed variational bound,

$$p(\mathbf{y}) \geq \prod_{i=1}^{n} c_i \int \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f} \right\rangle, \sigma^2\mathbf{I}\right) p(\mathbf{u}) \mathrm{d}\mathbf{u}$$

Variational Bayesian GP-LVM

- Consider collapsed variational bound,

$$p(\mathbf{y}|\mathbf{X}) \geq \prod_{i=1}^{n} c_i \int \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f} \right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{u}) \mathrm{d}\mathbf{u}$$

Variational Bayesian GP-LVM

- Consider collapsed variational bound,

$$\int p(\mathbf{y}|\mathbf{X})p(\mathbf{X})\mathrm{d}\mathbf{X} \geq \int \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f} \right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{X})\mathrm{d}\mathbf{X} p(\mathbf{u})\mathrm{d}\mathbf{u}$$

Variational Bayesian GP-LVM

- Consider collapsed variational bound,

$$\int p(\mathbf{y}|\mathbf{X})p(\mathbf{X})\mathrm{d}\mathbf{X} \geq \int \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right)p(\mathbf{X})\mathrm{d}\mathbf{X}p(\mathbf{u})\mathrm{d}\mathbf{u}$$

- Apply variational lower bound to the inner integral.

Variational Bayesian GP-LVM

- Consider collapsed variational bound,

$$\int p(\mathbf{y}|\mathbf{X})p(\mathbf{X})\mathrm{d}\mathbf{X} \geq \int \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f} \right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{X})\mathrm{d}\mathbf{X} p(\mathbf{u})\mathrm{d}\mathbf{u}$$

- Apply variational lower bound to the inner integral.

$$\int \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f} \right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{X})\mathrm{d}\mathbf{X}$$

$$\geq \left\langle \sum_{i=1}^{n} \log c_i \right\rangle_{q(\mathbf{X})} + \left\langle \log \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f} \right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) \right\rangle_{q(\mathbf{X})} + \mathrm{KL}\left(q(\mathbf{X}) \,\|\, p(\mathbf{X})\right)$$

Variational Bayesian GP-LVM

- Consider collapsed variational bound,

$$\int p(\mathbf{y}|\mathbf{X})p(\mathbf{X})\mathrm{d}\mathbf{X} \geq \int \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\left\langle\mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right) p(\mathbf{X})\mathrm{d}\mathbf{X}p(\mathbf{u})\mathrm{d}\mathbf{u}$$

- Apply variational lower bound to the inner integral.

$$\begin{aligned}\int \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\left\langle\mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right)p(\mathbf{X})\mathrm{d}\mathbf{X} \\ &\geq \left\langle \sum_{i=1}^{n} \log c_i \right\rangle_{q(\mathbf{X})} \\ &\quad + \left\langle \log \mathcal{N}\left(\mathbf{y}|\left\langle\mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{X})}, \sigma^2\mathbf{I}\right)\right\rangle_{q(\mathbf{X})} \\ &\quad + \mathrm{KL}\left(q(\mathbf{X}) \,\|\, p(\mathbf{X})\right)\end{aligned}$$

- Which is analytically tractable for Gaussian $q(\mathbf{X})$ and some covariance functions.

Required Expectations

- Need expectations under $q(\mathbf{X})$ of:

$$\log c_i = \frac{1}{2\sigma^2}\left[k_{i,i} - \mathbf{k}_{i,\mathbf{u}}^\top \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{k}_{i,\mathbf{u}}\right]$$

and

$$\log \mathcal{N}\left(\mathbf{y}|\left\langle \mathbf{f}\right\rangle_{p(\mathbf{f}|\mathbf{u},\mathbf{Y})}, \sigma^2\mathbf{I}\right) = -\frac{1}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\left(y_i - \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}\right)^2$$

- This requires the expectations

$$\left\langle \mathbf{K}_{\mathbf{f},\mathbf{u}}\right\rangle_{q(\mathbf{X})}$$

and

$$\left\langle \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}\right\rangle_{q(\mathbf{X})}$$

which can be computed analytically for some covariance functions.

Variational Compression

(Damianou and Lawrence, 2013)

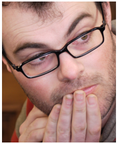
- Augment each layer with inducing variables $\mathbf{u}_i$.
- Apply variational compression,

$$p(\mathbf{y}, \{\mathbf{f}_i\}_{i=1}^{\ell-1}|\{\mathbf{u}_i\}_{i=1}^{\ell}, \mathbf{X}) \geq \tilde{p}(\mathbf{y}|\mathbf{u}_\ell, \mathbf{f}_{\ell-1}) \prod_{i=2}^{\ell-1} \tilde{p}(\mathbf{f}_i|\mathbf{u}_i, \mathbf{f}_{i-1}) \tilde{p}(\mathbf{f}_1|\mathbf{u}_i, \mathbf{X})$$
$$\times \exp\left(\sum_{i=1}^{\ell} -\frac{1}{2\sigma_i^2} \text{tr}\left(\boldsymbol{\Sigma}_i\right)\right) \quad (1)$$

where

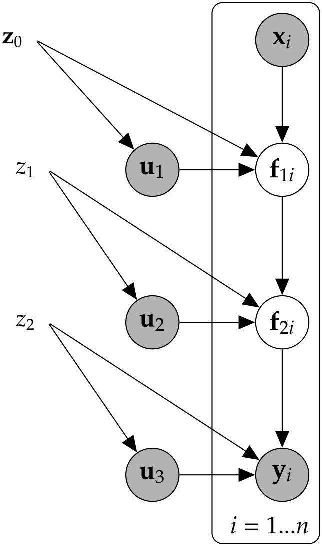
$$\tilde{p}(\mathbf{f}_i|\mathbf{u}_i, \mathbf{f}_{i-1}) = \mathcal{N}\left(\mathbf{f}_i|\mathbf{K}_{\mathbf{f}_i\mathbf{u}_i}\mathbf{K}_{\mathbf{u}_i\mathbf{u}_i}^{-1}\mathbf{u}_i, \sigma_i^2\mathbf{I}\right).$$

Nested Variational Compression



(Hensman and Lawrence, 2014)

- By sustaining explicity distributions over inducing variables James Hensman has developed a nested variatnt of variational compression.
- Exciting thing: it mathematically looks like a deep neural network, but with inducing variables in the place of basis functions.
- Additional complexity control term in the objective function.



$\mathbf{z}_0$
z_1
z_2
$\mathbf{u}_1$
$\mathbf{u}_2$
$\mathbf{u}_3$
$\mathbf{x}_i$
$\mathbf{f}_{1i}$
$\mathbf{f}_{2i}$
$\mathbf{y}_i$
$i = 1...n$

Nested Bound

$$
\begin{aligned}
\log p(\mathbf{y}|\mathbf{X}) \geq &- \frac{1}{\sigma_1^2}\mathrm{tr}\left(\boldsymbol{\Sigma}_1\right) - \sum_{i=2}^{\ell} \frac{1}{2\sigma_i^2}\left(\psi_i - \mathrm{tr}\left(\boldsymbol{\Phi}_i \mathbf{K}_{\mathbf{u}_i\mathbf{u}_i}^{-1}\right)\right) \\
&- \sum_{i=1}^{\ell} \mathrm{KL}\left(q(\mathbf{u}_i) \,\|\, p(\mathbf{u}_i)\right) \\
&- \sum_{i=2}^{\ell} \frac{1}{2\sigma_i^2}\mathrm{tr}\left((\boldsymbol{\Phi}_i - \boldsymbol{\Psi}_i^\top \boldsymbol{\Psi}_i)\mathbf{K}_{\mathbf{u}_i\mathbf{u}_i}^{-1} \left\langle \mathbf{u}_i\mathbf{u}_i^\top \right\rangle_{q(\mathbf{u}_i)} \mathbf{K}_{\mathbf{u}_i\mathbf{u}_i}^{-1}\right) \\
&+ \log \mathcal{N}\left(\mathbf{y}|\boldsymbol{\Psi}_\ell \mathbf{K}_{\mathbf{u}_\ell\mathbf{u}_\ell}^{-1}\mathbf{m}_\ell, \sigma_\ell^2 \mathbf{I}\right) \qquad (2)
\end{aligned}
$$

Nested Bound

$$
\begin{aligned}
\log p(\mathbf{y}|\mathbf{X}) \geq & -\frac{1}{\sigma_1^2}\mathrm{tr}\left(\boldsymbol{\Sigma}_1\right) - \sum_{i=2}^{\ell}\frac{1}{2\sigma_i^2}\left(\psi_i - \mathrm{tr}\left(\boldsymbol{\Phi}_i\mathbf{K}_{\mathbf{u}_i\mathbf{u}_i}^{-1}\right)\right) \\
& - \sum_{i=1}^{\ell}\mathrm{KL}\left(q(\mathbf{u}_i)\,\|\,p(\mathbf{u}_i)\right) \\
& - \sum_{i=2}^{\ell}\frac{1}{2\sigma_i^2}\mathrm{tr}\left((\boldsymbol{\Phi}_i - \boldsymbol{\Psi}_i^\top\boldsymbol{\Psi}_i)\mathbf{K}_{\mathbf{u}_i\mathbf{u}_i}^{-1}\left\langle\mathbf{u}_i\mathbf{u}_i^\top\right\rangle_{q(\mathbf{u}_i)}\mathbf{K}_{\mathbf{u}_i\mathbf{u}_i}^{-1}\right) \\
& + \log\mathcal{N}\left(\mathbf{y}|\boldsymbol{\Psi}_\ell\mathbf{K}_{\mathbf{u}_\ell\mathbf{u}_\ell}^{-1}\mathbf{m}_\ell, \sigma_\ell^2\mathbf{I}\right) \qquad (2)
\end{aligned}
$$

Required Expectations

$$\log \mathcal{N}\left(\mathbf{y}|\boldsymbol{\Psi}_\ell \mathbf{K}_{\mathbf{u}_\ell \mathbf{u}_\ell}^{-1} \mathbf{m}_\ell, \sigma_\ell^2 \mathbf{I}\right)$$

where

Required Expectations

$$\log \mathcal{N}\left(\mathbf{y}|\boldsymbol{\Psi}_\ell \mathbf{K}_{\mathbf{u}_\ell\mathbf{u}_\ell}^{-1}\mathbf{m}_\ell, \sigma_\ell^2\mathbf{I}\right)$$

where

$$\boldsymbol{\Psi}_i = \left\langle \mathbf{K}_{\mathbf{f}_i\mathbf{u}_i} \right\rangle_{q(\mathbf{f}_{i-1})}$$

where elements of $\mathbf{K}_{\mathbf{f}_i\mathbf{u}_i}$ are

$$k_{f_i u_i'}(\mathbf{f}_{i-1}, \mathbf{z}_i')$$

Required Expectations

$$\log \mathcal{N}\left(\mathbf{y}|\boldsymbol{\Psi}_\ell \mathbf{K}_{\mathbf{u}_\ell \mathbf{u}_\ell}^{-1}\mathbf{m}_\ell, \sigma_\ell^2 \mathbf{I}\right)$$

where

$$\boldsymbol{\Psi}_i = \left\langle \mathbf{K}_{\mathbf{f}_i \mathbf{u}_i} \right\rangle_{q(\mathbf{f}_{i-1})}$$

where elements of $\mathbf{K}_{\mathbf{f}_i \mathbf{u}_i}$ are

$$k_{f_i u_i'}(\mathbf{f}_{i-1}, \mathbf{z}_i')$$

And

$$q(\mathbf{f}_1) = \int \tilde{p}(\mathbf{f}_1|\mathbf{u}_1, \mathbf{X}) q(\mathbf{u}_1) \mathrm{d}\mathbf{u}_1,$$

$$q(\mathbf{f}_i) = \int \tilde{p}(\mathbf{f}_i|\mathbf{u}_i, \mathbf{f}_{i-1}) q(\mathbf{u}_i) q(\mathbf{f}_{i-1}) \mathrm{d}\mathbf{u}_i \mathrm{d}\mathbf{f}_i,$$

Required Expectations

$$\log \mathcal{N}\left(\mathbf{y}|\boldsymbol{\Psi}_\ell \mathbf{K}_{\mathbf{u}_\ell \mathbf{u}_\ell}^{-1}\mathbf{m}_\ell, \sigma_\ell^2 \mathbf{I}\right)$$

where

$$\boldsymbol{\Psi}_i = \left\langle \mathbf{K}_{\mathbf{f}_i \mathbf{u}_i} \right\rangle_{q(\mathbf{f}_{i-1})}$$

where elements of $\mathbf{K}_{\mathbf{f}_i \mathbf{u}_i}$ are

$$k_{f_i u_i'}(\mathbf{f}_{i-1}, \mathbf{z}_i')$$

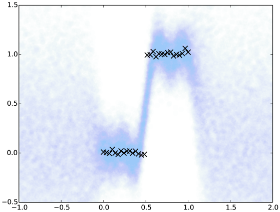
And

$$q(\mathbf{f}_1) = \int \tilde{p}(\mathbf{f}_1|\mathbf{u}_1, \mathbf{X})q(\mathbf{u}_1)\mathrm{d}\mathbf{u}_1,$$

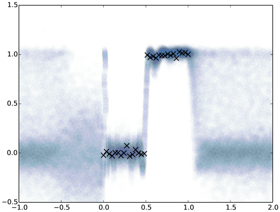
$$q(\mathbf{f}_i) = \int \tilde{p}(\mathbf{f}_i|\mathbf{u}_i, \mathbf{f}_{i-1})q(\mathbf{u}_i)q(\mathbf{f}_{i-1})\mathrm{d}\mathbf{u}_i\mathrm{d}\mathbf{f}_i,$$

cf wake sleep algorithm. recognition network and generation network (Hinton et al., 1995).

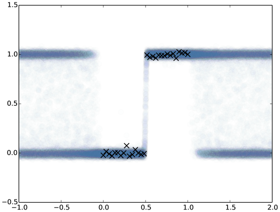
Derivative Tails Increase with Layers: Step Function



(a) GP

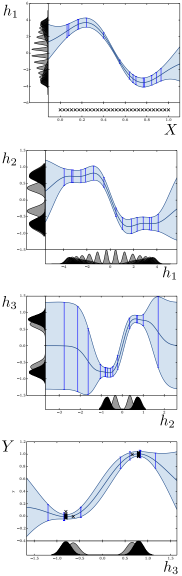


(b) 2 layers

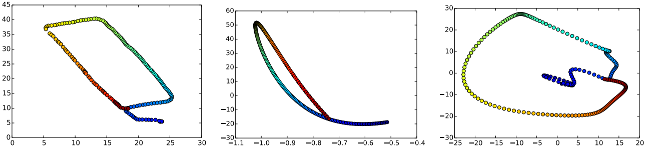


(c) 4 layers

Values in Hidden Layers



Loop Detection in Robotics



(d) True path (e) Hidden layer 1 (f) Hidden layer 2

- Dynamically constrained model
- Correctly detects the loop
- Learns temporal continuity and corner-like features in different layers

Data fit for Loop Closure

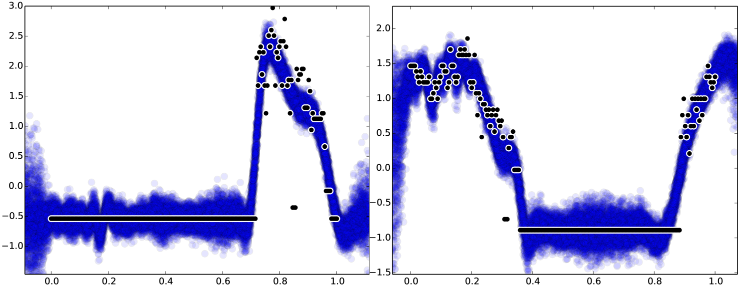
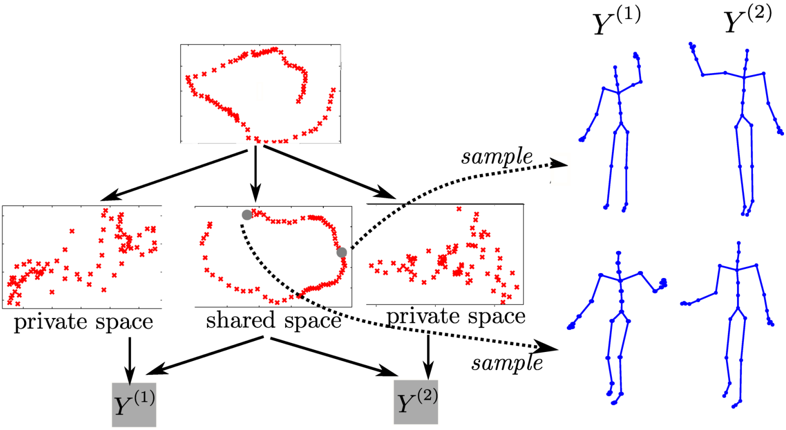


Figure : Example data fits for 2 of the 30 output dimensions

Motion Capture

- ‘High five’ data.
- Model learns structure between two interacting subjects.

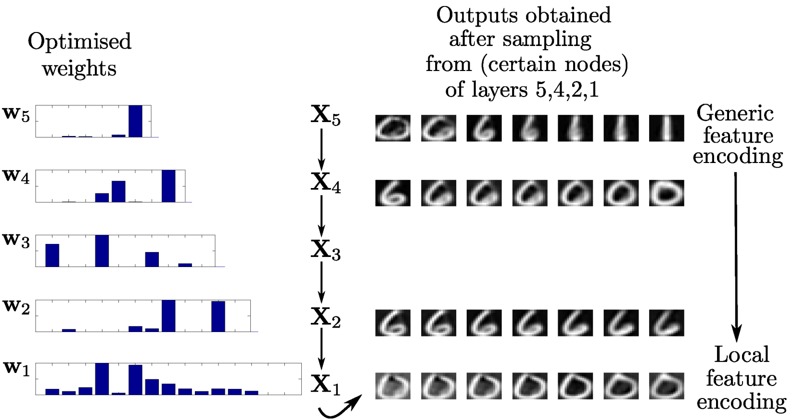
Deep hierarchies – motion capture



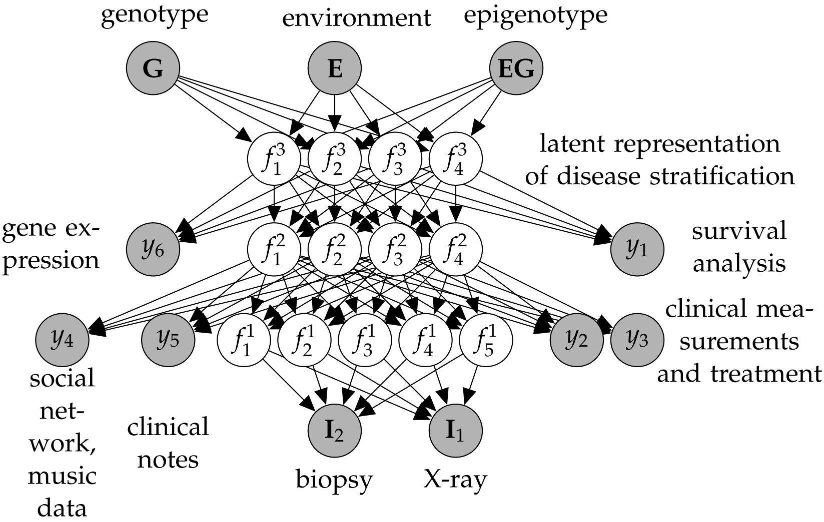
Digits Data Set

- Are deep hierarchies justified for small data sets?
- We can lower bound the evidence for different depths.
- For 150 6s, 0s and 1s from MNIST we found at least 5 layers are required.

Deep hierarchies – MNIST



Deep Health



genotype
environment
epigenotype
G
E
EG
f_1^3 f_2^3 f_3^3 f_4^3
latent representation of disease stratification
gene expression
y_6
f_1^2 f_2^2 f_3^2 f_4^2
y_1
survival analysis
clinical measurements and treatment
y_4 y_5
f_1^1 f_2^1 f_3^1 f_4^1 f_5^1
y_2 y_3
social network, music data
clinical notes
$\mathbf{I}_2$
$\mathbf{I}_1$
biopsy
X-ray

Summary

- Deep Gaussian Processes allow unsupervised and supervised deep learning.
- They can be easily adapted to handle multitask learning.
- Data dimensionality turns out to not be a computational bottleneck.
- Variational compression algorithms show promise for scaling these models to *massive* data sets.

References I

L. Csató. *Gaussian Processes — Iterative Sparse Approximations*. PhD thesis, Aston University, 2002.

L. Csató and M. Opper. Sparse on-line Gaussian processes. *Neural Computation*, 14(3):641–668, 2002.

A. Damianou and N. D. Lawrence. Deep Gaussian processes. In C. Carvalho and P. Ravikumar, editors, *Proceedings of the Sixteenth International Workshop on Artificial Intelligence and Statistics*, volume 31, AZ, USA, 2013. JMLR W&CP 31. [PDF].

D. Duvenaud, O. Rippel, R. Adams, and Z. Ghahramani. Avoiding pathologies in very deep networks. In S. Kaski and J. Corander, editors, *Proceedings of the Seventeenth International Workshop on Artificial Intelligence and Statistics*, volume 33, Iceland, 2014. JMLR W&CP 33.

J. Hensman and N. D. Lawrence. Nested variational compression in deep gaussian processes. Technical report, University of Sheffield,

J. Hensman, M. Rattray, and N. D. Lawrence. Fast variational inference in the conjugate exponential family. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, Cambridge, MA, 2012. [PDF].

G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal. The wake-sleep algorithm for unsupervised neural networks. *Science*, 268:1158–1161, 1995.

N. J. King and N. D. Lawrence. Fast variational inference for Gaussian Process models through KL-correction. In *ECML, Berlin, 2006*, Lecture Notes in Computer Science, pages 270–281, Berlin, 2006. Springer-Verlag. [PDF].

T. K. Leen, T. G. Dietterich, and V. Tresp, editors. *Advances in Neural Information Processing Systems*, volume 13, Cambridge, MA, 2001. MIT Press.

J. Quiñonero Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006. [Google Books] .

M. Seeger, C. K. I. Williams, and N. D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In C. M. Bishop and B. J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, Key West, FL, 3–6 Jan 2003.

A. J. Smola and P. L. Bartlett. Sparse greedy Gaussian process regression. In Leen et al. (2001), pages 619–625.

References II

E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, Cambridge, MA, 2006. MIT Press.

M. K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In D. van Dyk and M. Welling, editors, *Proceedings of the Twelfth International Workshop on Artificial Intelligence and Statistics*, volume 5, pages 567–574, Clearwater Beach, FL, 16-18 April 2009. JMLR W&CP 5.

C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In Leen et al. (2001), pages 682–688.