Deep Gaussian processes

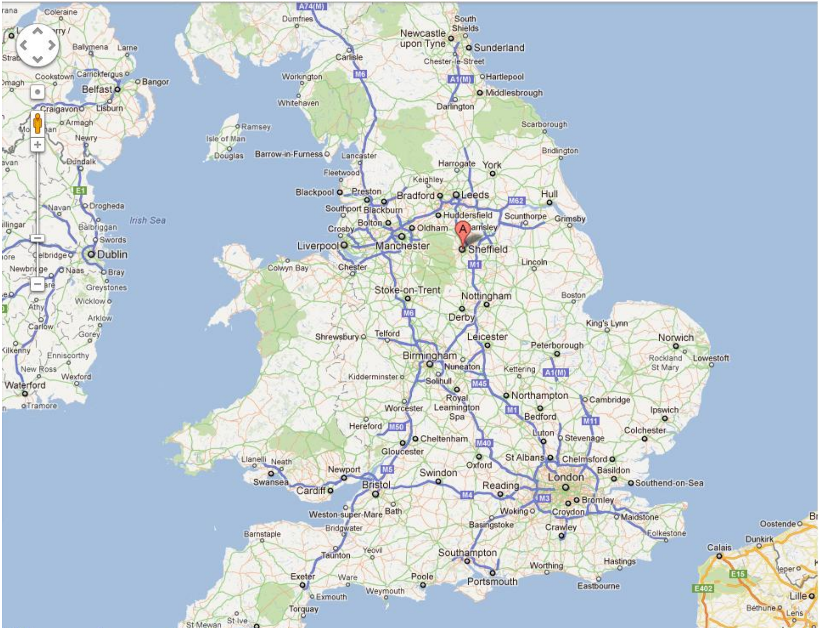
Andreas Damianou

Department of Neuro- and Computer Science, University of Sheffield, UK

Deep Learning Meetup, London, 24/06/2014

Outline

2h away from London!

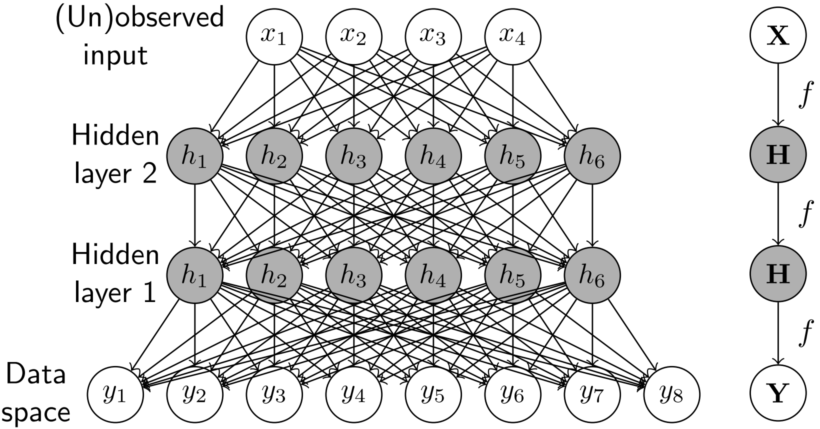


Great collaborators!

- Prof. Neil Lawrence
- Dr James Hensman
- Dr Michalis Titsias
- Dr Carl Henrik Ek

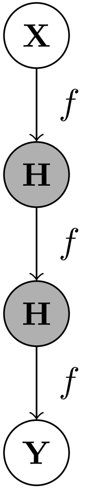
Outline

Deep learning



$$\mathbf{Y} = f(f(\cdots f(\mathbf{X})))$$

Deep Gaussian processes - Big Picture



Deep GP:

- Directed graphical model
- Non-parametric, non-linear mappings f
- Mappings f marginalised out analytically
- Likelihood is a non-linear function of the inputs
- Continuous variables
- NOT a GP!

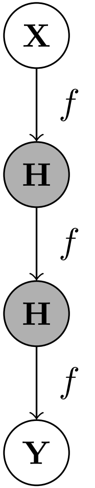
Challenges:

- Marginalise out $\mathbf{H}$
- No sampling: analytic approximation of objective

Solution:

- Variational approximation
- This also gives access to the *model evidence*

Deep Gaussian processes - Big Picture



Deep GP:

- Directed graphical model
- Non-parametric, non-linear mappings f
- Mappings f marginalised out analytically
- Likelihood is a non-linear function of the inputs
- Continuous variables
- NOT a GP!

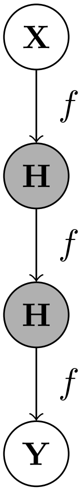
Challenges:

- Marginalise out $\mathbf{H}$
- No sampling: analytic approximation of objective

Solution:

- Variational approximation
- This also gives access to the *model evidence*

Deep Gaussian processes - Big Picture



Deep GP:

- Directed graphical model
- Non-parametric, non-linear mappings f
- Mappings f marginalised out analytically
- Likelihood is a non-linear function of the inputs
- Continuous variables
- NOT a GP!

Challenges:

- Marginalise out $\mathbf{H}$
- No sampling: analytic approximation of objective

Solution:

- Variational approximation
- This also gives access to the *model evidence*

Gesture challenge: human vs computer

A human brain is good at one-shot learning...
a computer struggles...

Gesture challenge: human vs computer

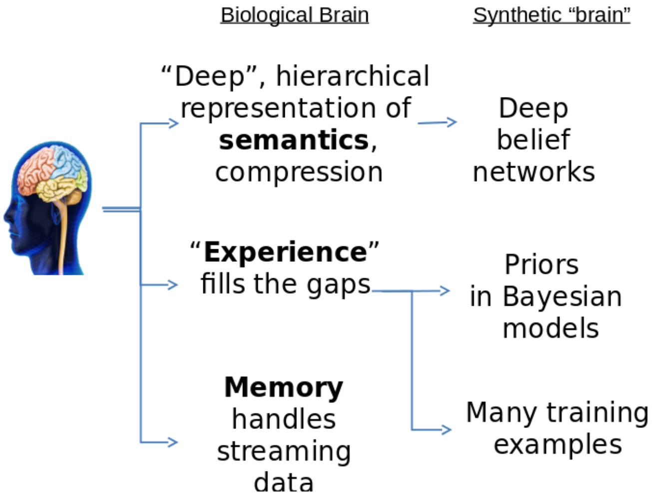
A human brain is good at one-shot learning...
a computer struggles...

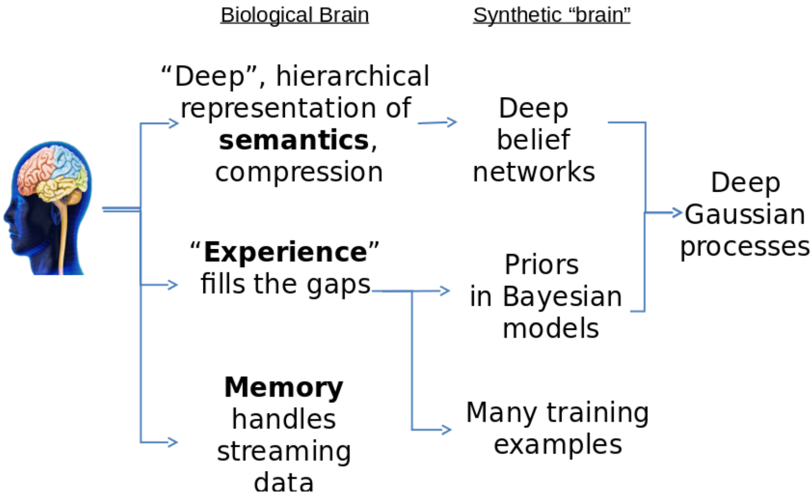
Biological Brain

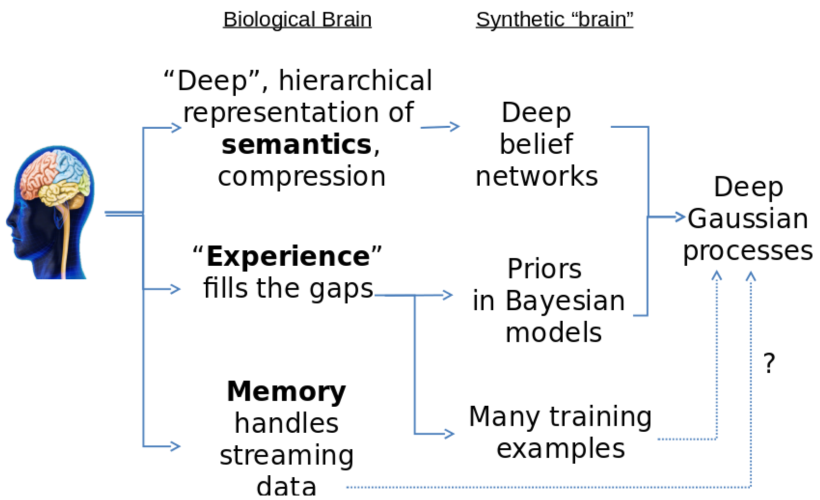
"Deep", hierarchical representation of **semantics**, compression

"**Experience**" fills the gaps

Memory handles streaming data


Biological Brain
Synthetic "brain"
"Deep", hierarchical representation of **semantics**, compression
Deep belief networks
"**Experience**" fills the gaps
Priors in Bayesian models
Memory handles streaming data
Many training examples


Biological Brain
Synthetic "brain"
"Deep", hierarchical representation of **semantics**, compression
Deep belief networks
"**Experience**" fills the gaps
Priors in Bayesian models
Memory handles streaming data
Many training examples
Deep Gaussian processes

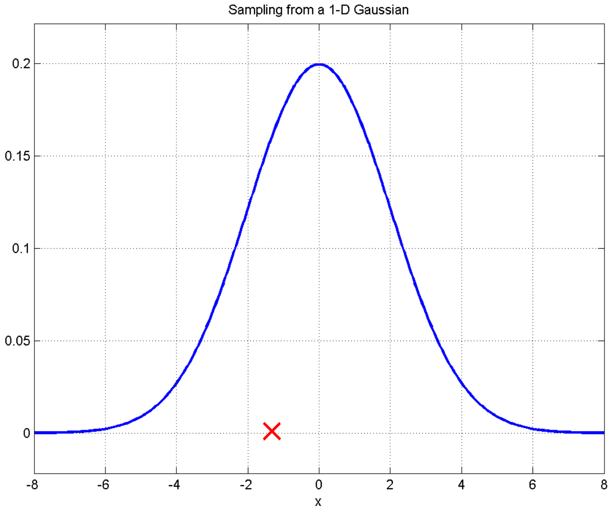

Biological Brain
Synthetic "brain"
"Deep", hierarchical representation of **semantics**, compression
"**Experience**" fills the gaps
Memory handles streaming data
Deep belief networks
Priors in Bayesian models
Many training examples
Deep Gaussian processes
?

Outline

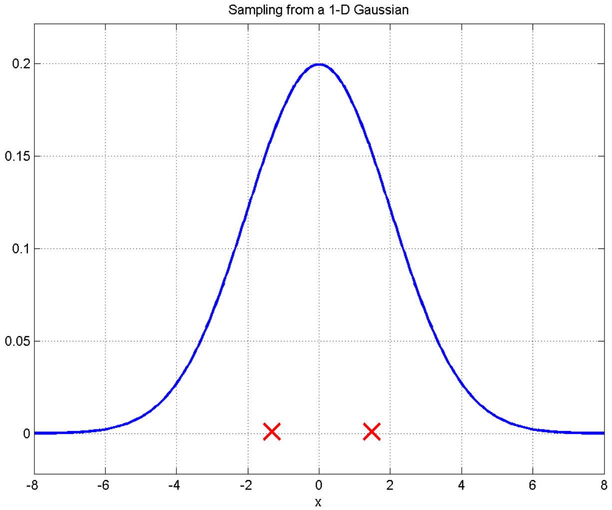
Introducing Gaussian Processes:

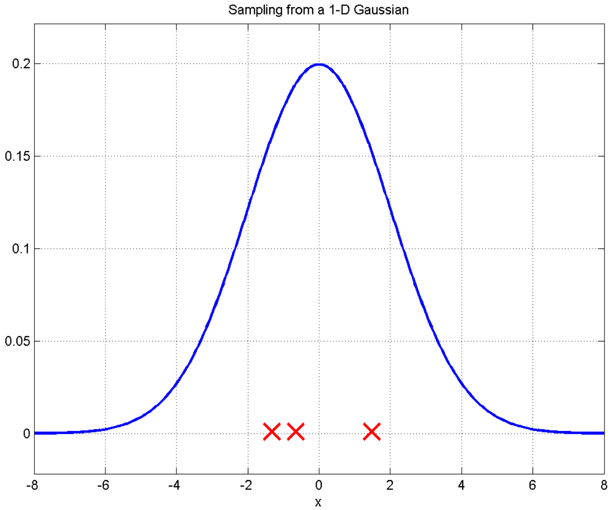
- A Gaussian distribution depends on a mean and a covariance vector / matrix.
- A Gaussian process depends on a mean and a covariance function.

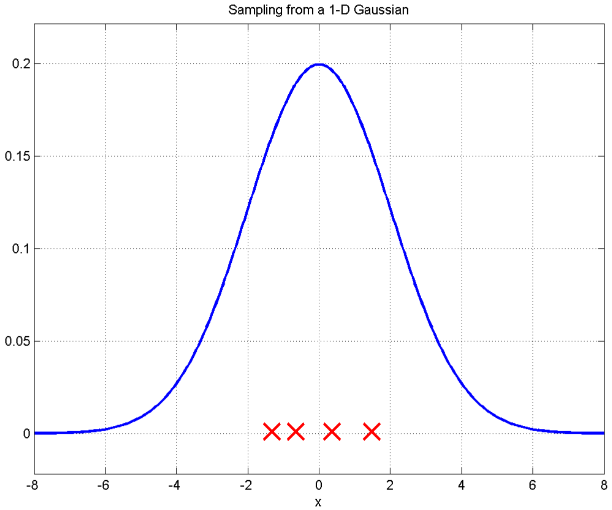
Next: Demo, from Gaussian distributions to Gaussian processes.

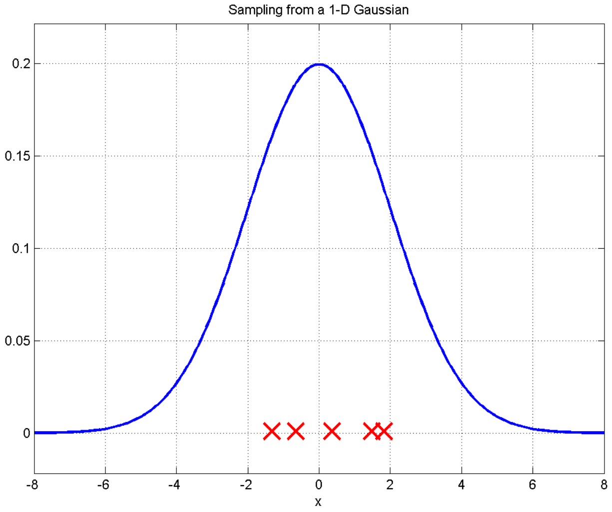


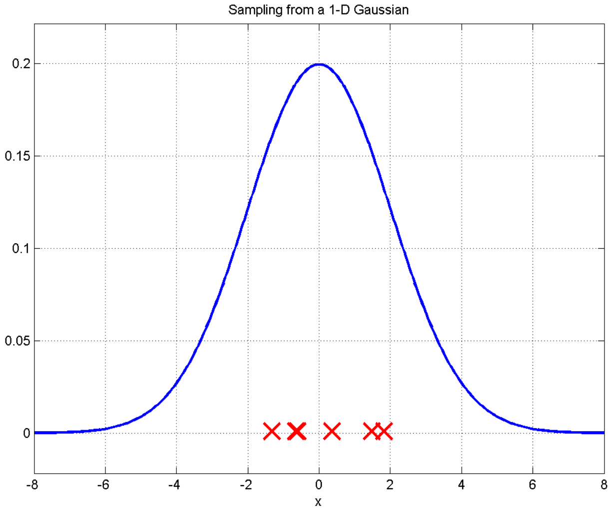
Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x

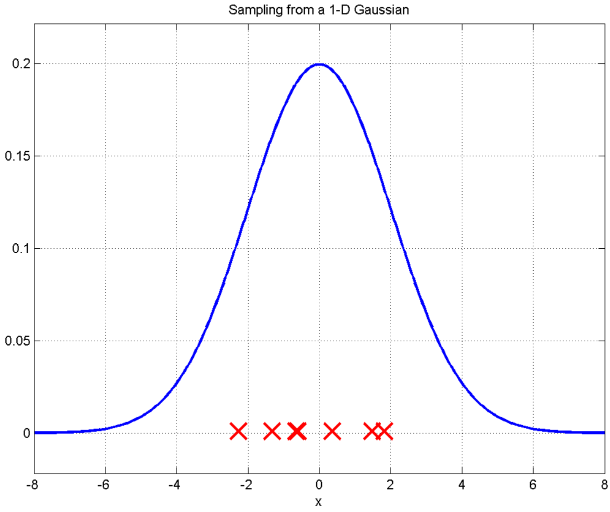

Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x

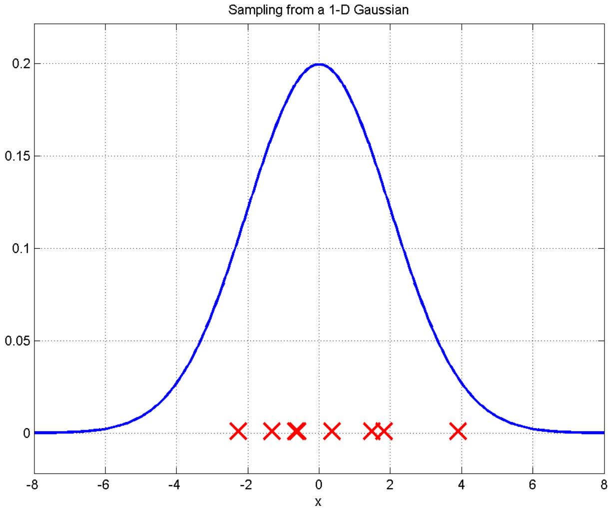

Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x


Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x

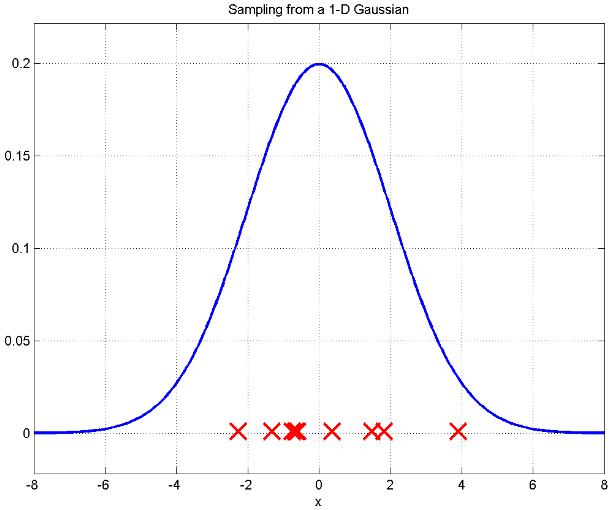

Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x


Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x

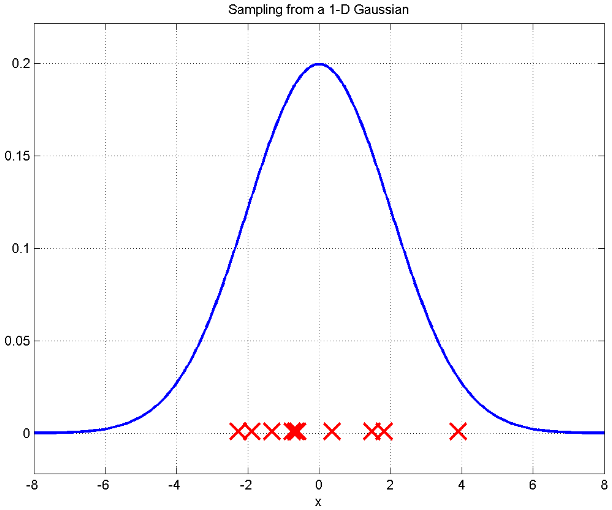

Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x



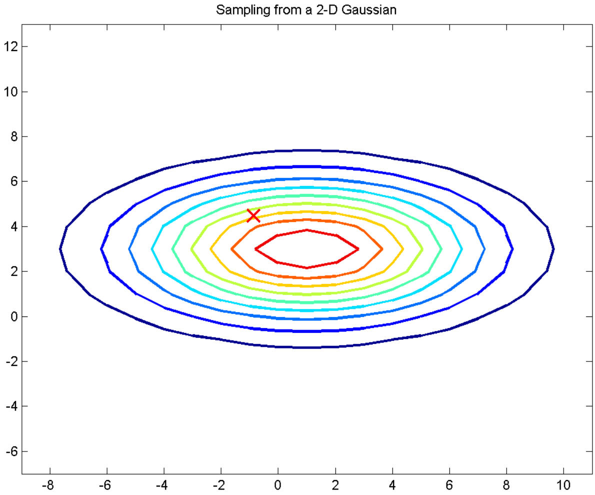
Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x

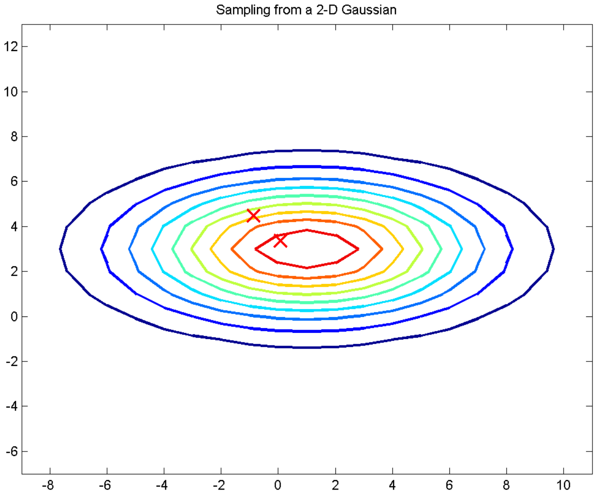


Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x

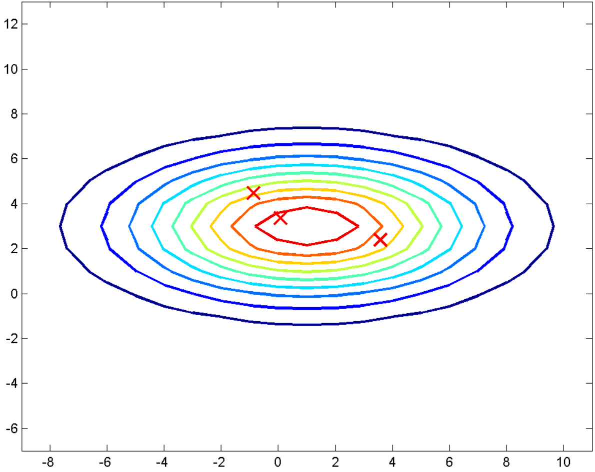


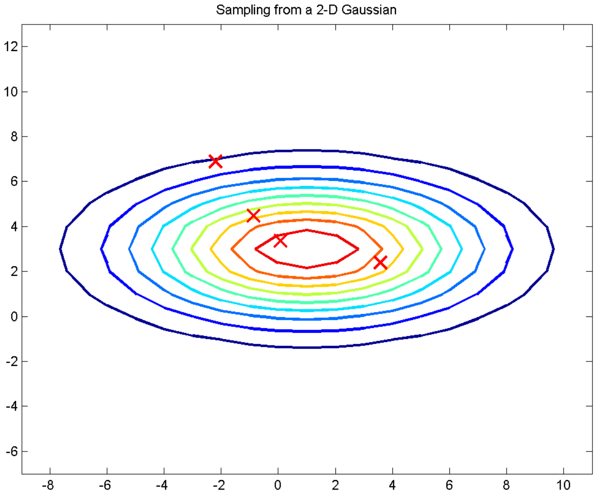
Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x

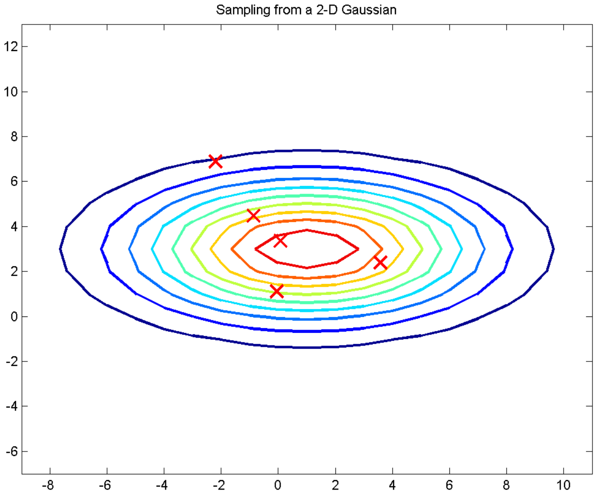

Sampling from a 2-D Gaussian
12
10
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8
10


Sampling from a 2-D Gaussian
12
10
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8
10

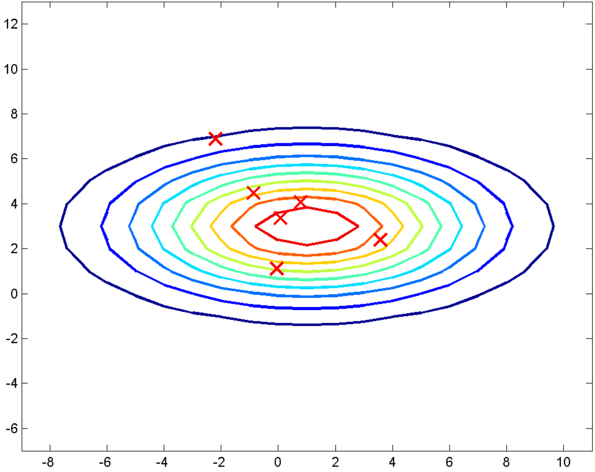
Sampling from a 2-D Gaussian

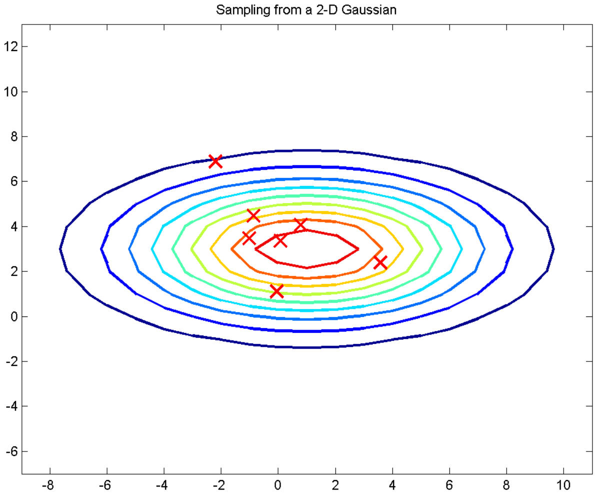



Sampling from a 2-D Gaussian
12
10
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8
10

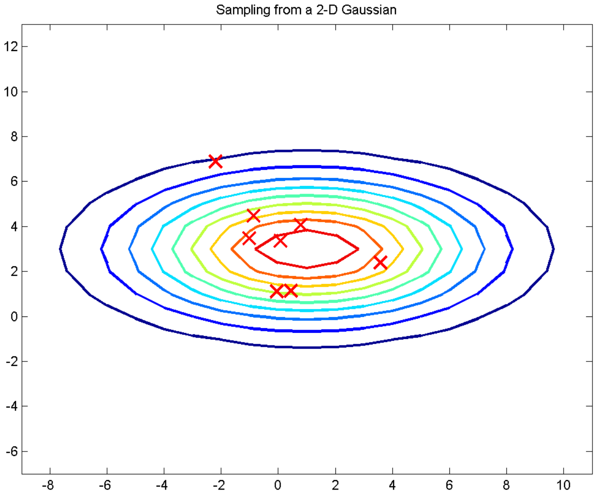

Sampling from a 2-D Gaussian
12
10
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8
10

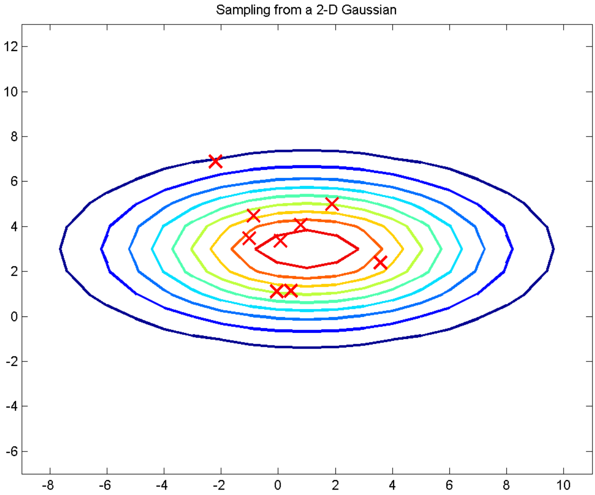
Sampling from a 2-D Gaussian

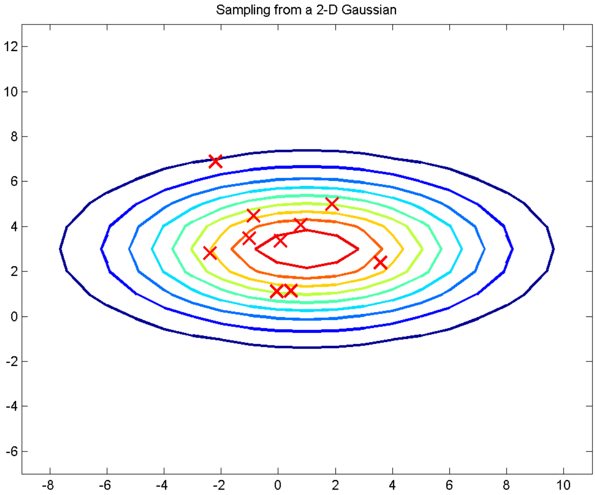


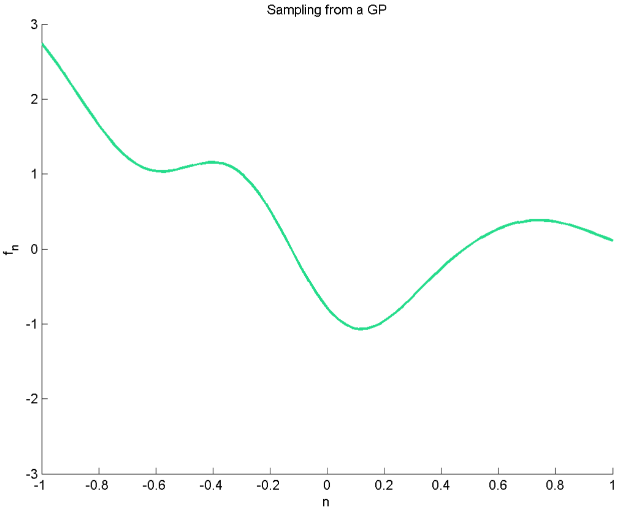


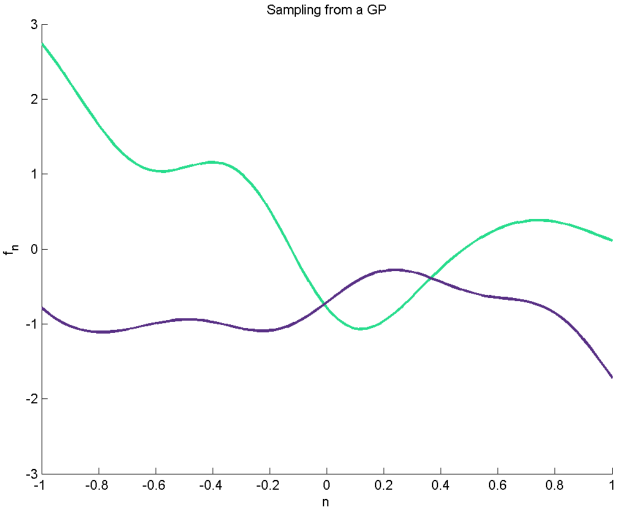
Sampling from a 2-D Gaussian
12
10
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8
10


Sampling from a 2-D Gaussian
12
10
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8
10

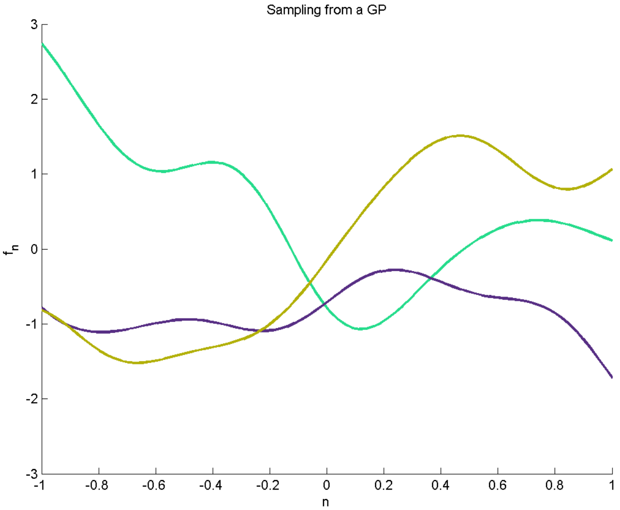

Sampling from a 2-D Gaussian
12
10
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8
10

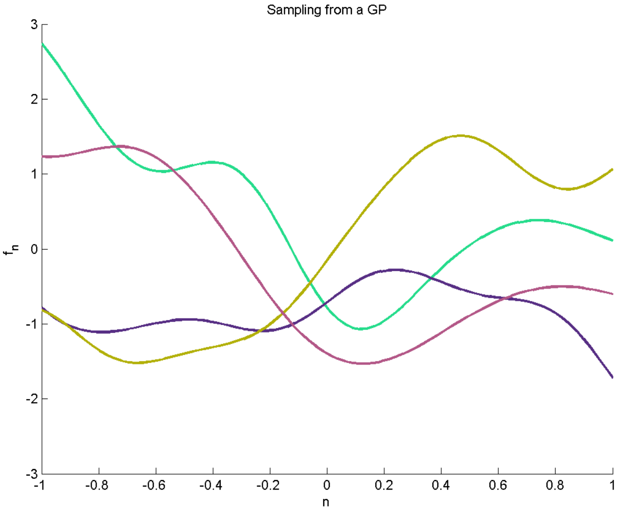

Sampling from a 2-D Gaussian
12
10
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8
10


Sampling from a GP
3
2
1
0
-1
-2
-3
f_n
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
n

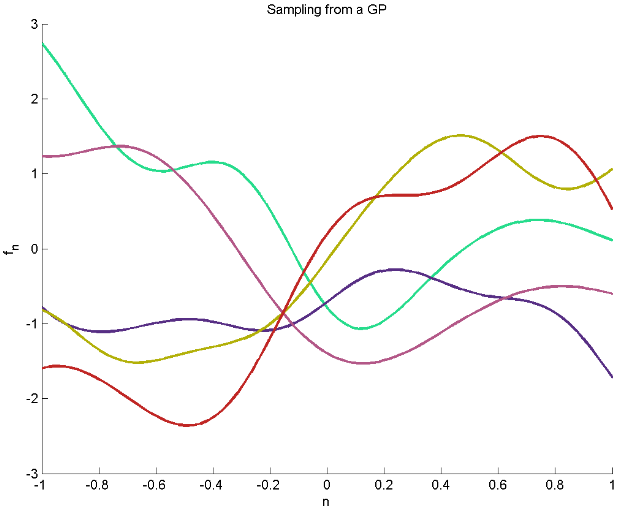


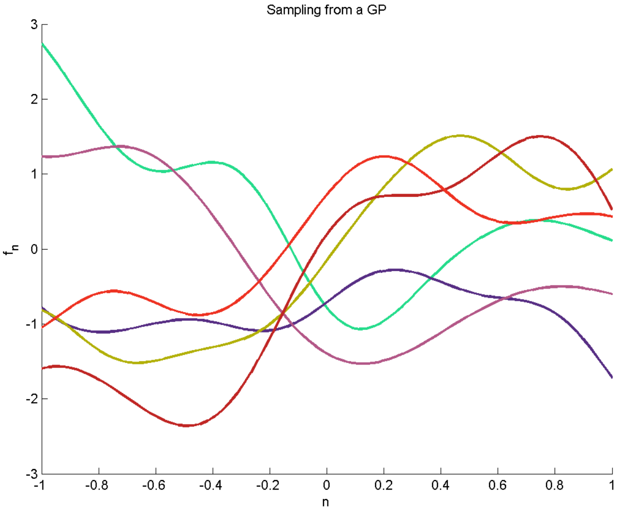
Sampling from a GP
3
2
1
0
-1
-2
-3
f_n
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
n


Sampling from a GP
3
2
1
0
-1
-2
-3
f_n
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
n

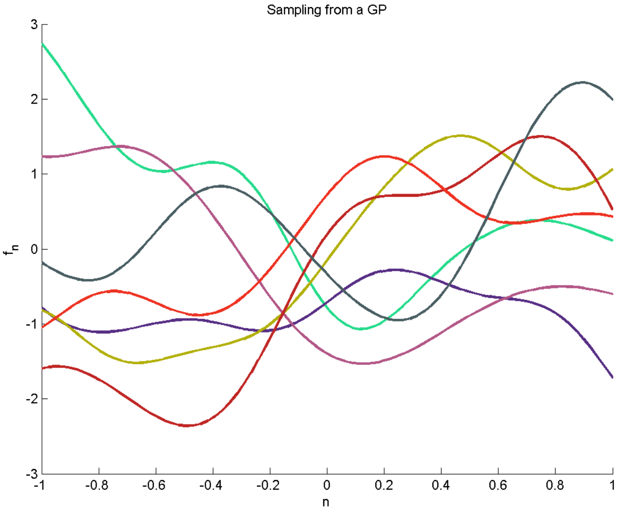


Sampling from a GP
3
2
1
0
-1
-2
-3
f_n
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
n

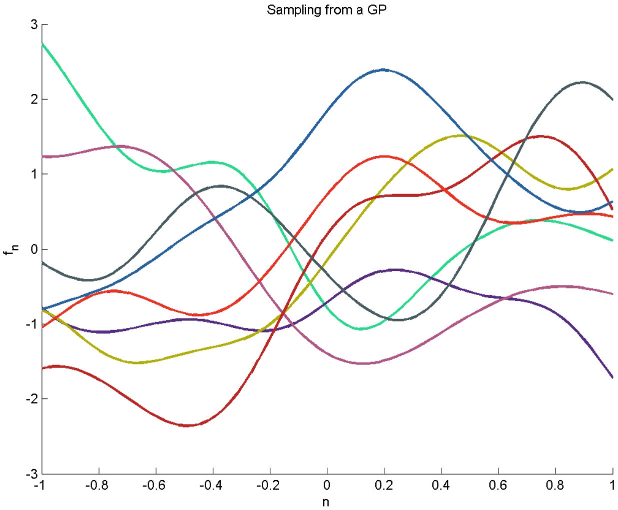

Sampling from a GP
f_n
n
3
2
1
0
-1
-2
-3
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

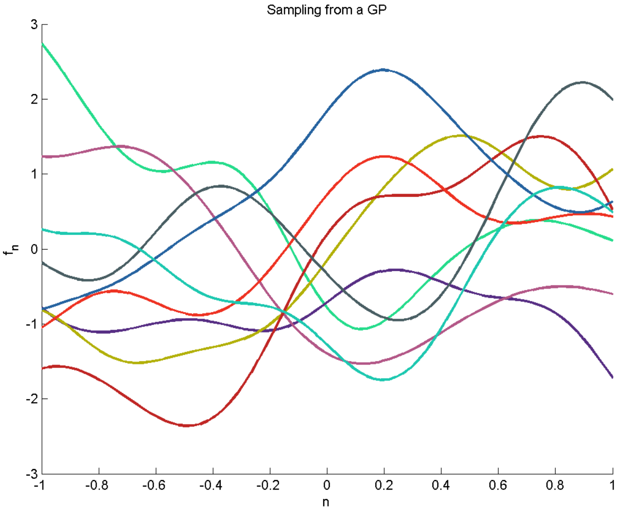


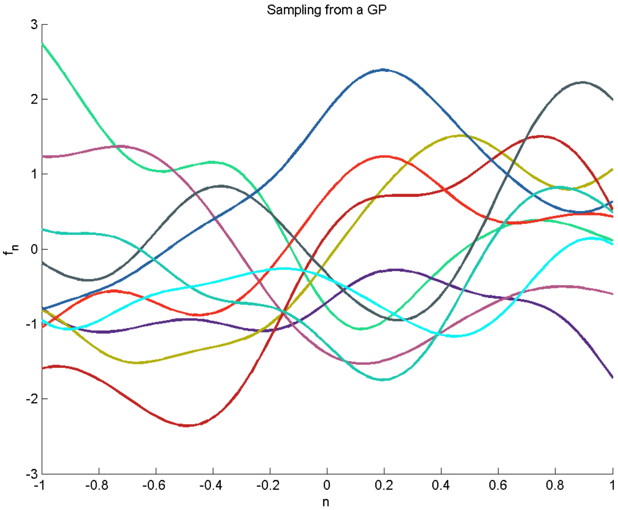
Sampling from a GP
3
2
1
0
-1
-2
-3
f_n
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
n



Sampling from a GP
3
2
1
0
-1
-2
-3
f_n
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
n


Sampling from a GP
3
2
1
0
-1
-2
-3
f_n
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
n


Sampling from a GP
3
2
1
0
-1
-2
-3
f_n
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
n



Sampling from a GP
3
2
1
0
-1
-2
-3
f_n
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
n

Infinite model... but we *always* work with finite sets!

Let's start with a multivariate Gaussian:

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

OR: $p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$.

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:

$$p(\mathbf{f}_A) = \int_{\mathbf{f}_B} p(\mathbf{f}_A, \mathbf{f}_B) \mathrm{d}\mathbf{f}_B = \mathcal{N}(\boldsymbol{\mu}_A, \mathbf{K}_{AA})$$

Infinite model... but we *always* work with finite sets!

Let's start with a multivariate Gaussian:

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

OR: $p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$.

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:

$$p(\mathbf{f}_A) = \int_{\mathbf{f}_B} p(\mathbf{f}_A, \mathbf{f}_B) \mathrm{d}\mathbf{f}_B = \mathcal{N}(\boldsymbol{\mu}_A, \mathbf{K}_{AA})$$

Infinite model... but we *always* work with finite sets!

In the GP context:

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

where:

Training data:

$$\begin{aligned} \mathbf{X} &= [x_1, \cdots, x_N] \\ \mathbf{f} &= [f_1, \cdots, f_N] = [f(x_1), \cdots, f(x_N)] \end{aligned}$$

Posterior is also Gaussian!

$$p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}). \quad \text{Then:}$$
$$p(\mathbf{f}_A | \mathbf{f}_B) = \mathcal{N}(\cdots, \cdots)$$

In the GP context this can be used for inter/extrapolation:

$$p(f_* | f_1, \cdots, f_N) = p(f(x_*) | f(x_1), \cdots, f(x_N)) \sim \mathcal{N}$$

Posterior is also Gaussian!

$$p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}). \quad \text{Then:}$$
$$p(\mathbf{f}_A | \mathbf{f}_B) = \mathcal{N}(\cdots, \cdots)$$

In the GP context this can be used for inter/extrapolation:

$$p(f_* | f_1, \cdots, f_N) = p(f(x_*) | f(x_1), \cdots, f(x_N)) \sim \mathcal{N}$$

More about the GP posterior

- For test points $\mathbf{X}_*$ we can predict their values $\mathbf{f}_*$.
- Assuming a zero-mean GP prior, $\mathbf{f}$ and $\mathbf{f}_*$ follow a joint Gaussian:
$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_{\mathbf{XX}} & \mathbf{K}_{\mathbf{XX}_*} \\ \mathbf{K}_{\mathbf{X}_*\mathbf{X}} & \mathbf{K}_{\mathbf{X}_*\mathbf{X}_*} \end{bmatrix}\right)$$
- The conditional $p(\mathbf{f}_*|\mathbf{f}, \mathbf{X}, \mathbf{X}_*)$ is Gaussian with:

$$\boldsymbol{\mu} = \mathbf{K}_{\mathbf{XX}_*}\mathbf{K}_{\mathbf{XX}}^{-1}\mathbf{f}$$
$$\boldsymbol{\Sigma} = \mathbf{K}_{\mathbf{XX}} - \mathbf{K}_{\mathbf{XX}_*}\mathbf{K}_{\mathbf{XX}}^{-1}\mathbf{K}_{\mathbf{X}_*\mathbf{X}}$$

- But where is $\mathbf{K}_{..}$ coming from?

More about the GP posterior

- For test points $\mathbf{X}_*$ we can predict their values $\mathbf{f}_*$.
- Assuming a zero-mean GP prior, $\mathbf{f}$ and $\mathbf{f}_*$ follow a joint Gaussian:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_{\mathbf{XX}} & \mathbf{K}_{\mathbf{XX}_*} \\ \mathbf{K}_{\mathbf{X}_*\mathbf{X}} & \mathbf{K}_{\mathbf{X}_*\mathbf{X}_*} \end{bmatrix}\right)$$

- The conditional $p(\mathbf{f}_*|\mathbf{f}, \mathbf{X}, \mathbf{X}_*)$ is Gaussian with:

$$\boldsymbol{\mu} = \mathbf{K}_{\mathbf{XX}_*}\mathbf{K}_{\mathbf{XX}}^{-1}\mathbf{f}$$
$$\boldsymbol{\Sigma} = \mathbf{K}_{\mathbf{XX}} - \mathbf{K}_{\mathbf{XX}_*}\mathbf{K}_{\mathbf{XX}}^{-1}\mathbf{K}_{\mathbf{X}_*\mathbf{X}}$$

- But where is $\mathbf{K}_{..}$ coming from?

Covariance functions

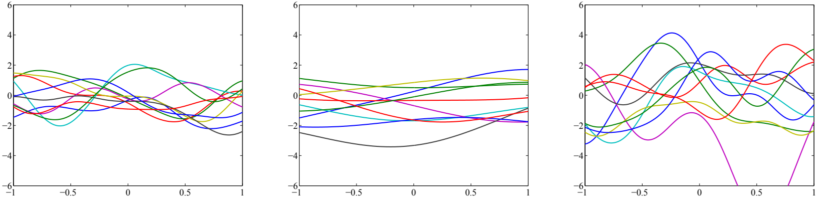
- Assumptions about *properties* of $f \Rightarrow$ define a parametric form for k, e.g:

$$k(x, x') = \alpha \exp\left(-\frac{\gamma}{2}(x - x')^\top (x - x')\right)$$

- However, a GP prior with this cov. function defines a whole *family* of functions
- The parameters $\{\alpha, \gamma\}$ are *hyperparameters*.
- We write: $f \sim \mathcal{GP}(0, k(x, x'))$

Covariance samples and hyperparameters

- The hyperparameters of the cov. function define the properties (and NOT an explicit form) of the sampled functions

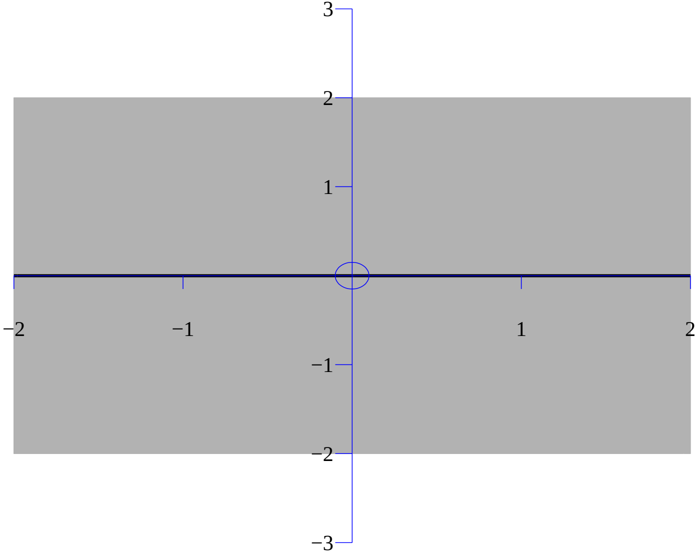


Incorporating Gaussian noise is tractable

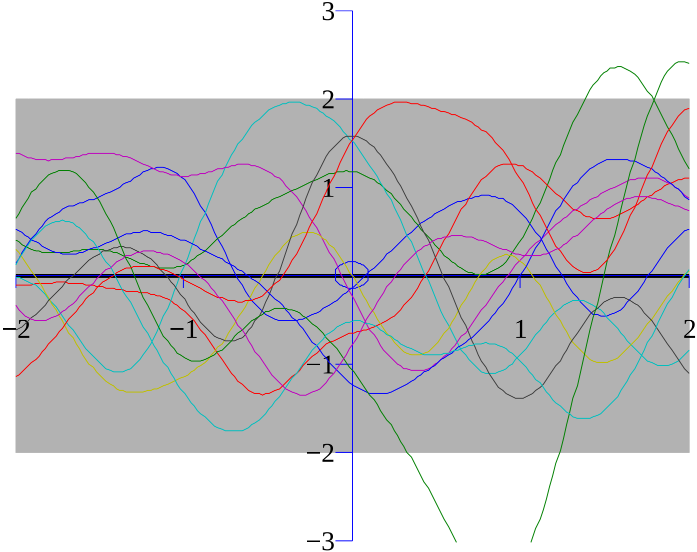
- So far we assumed: $\mathbf{f} = f(\mathbf{X})$
- Assuming that we only observe noisy versions $\mathbf{y}$ of the true outputs $\mathbf{f}$:

$$\mathbf{y} = f(\mathbf{X}) + \epsilon, \;\; \epsilon \sim \mathcal{N}(0, \sigma^2)$$

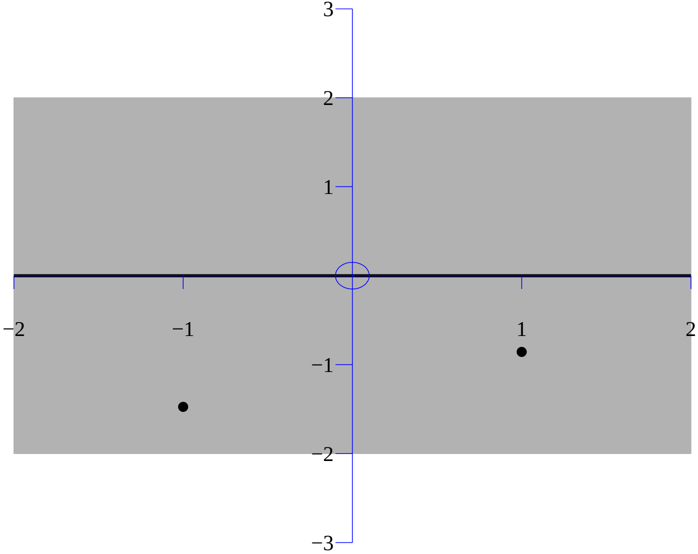
Fitting the data



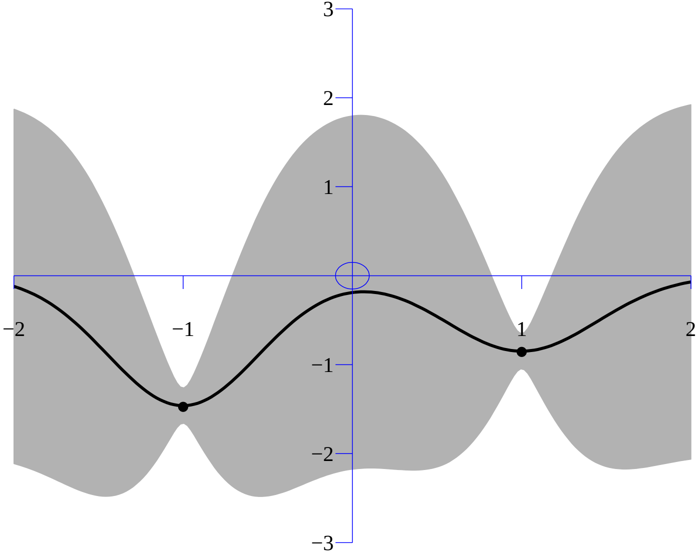
Fitting the data - Prior Samples



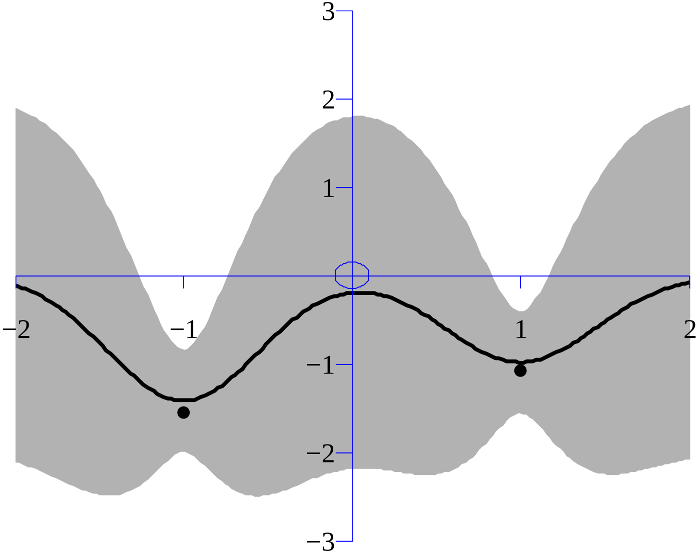
Fitting the data



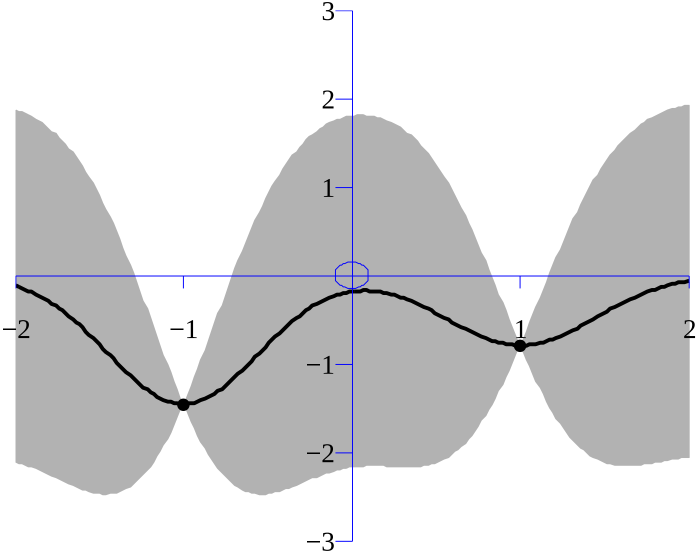
Fitting the data



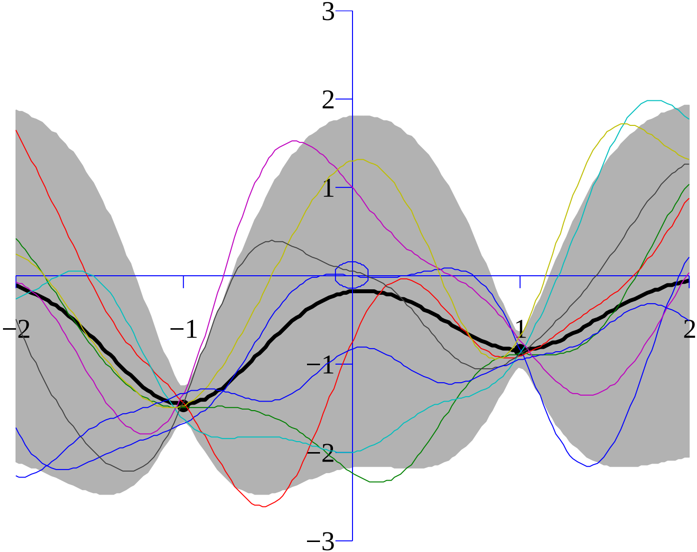
Fitting the data - more noise



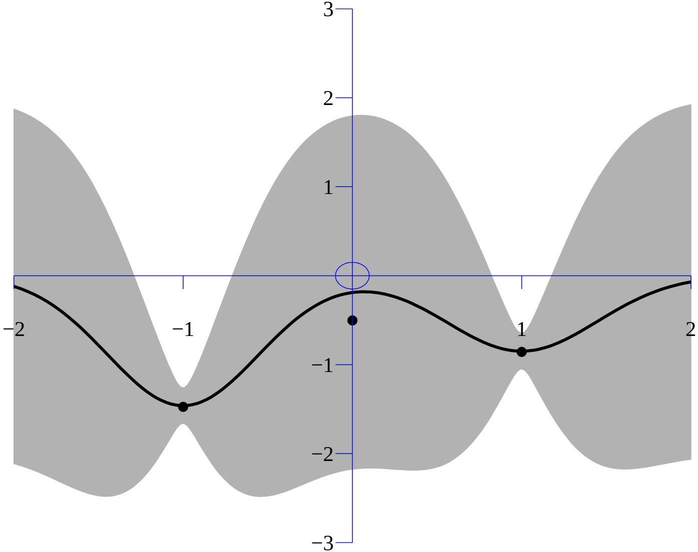
Fitting the data - no noise



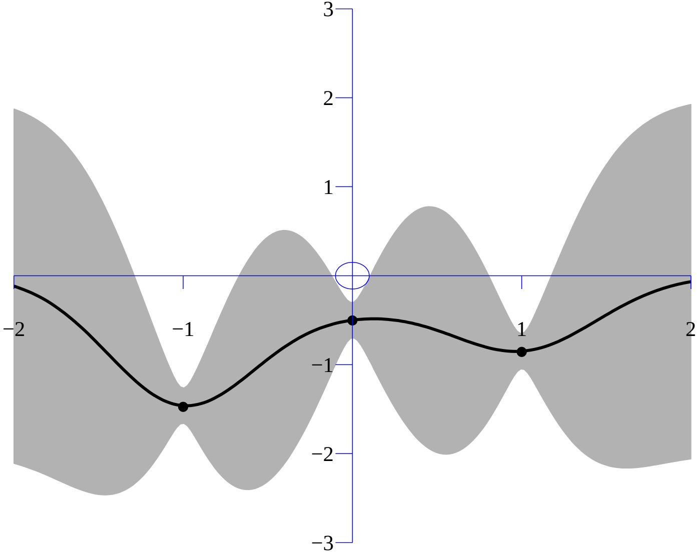
Fitting the data - Posterior samples



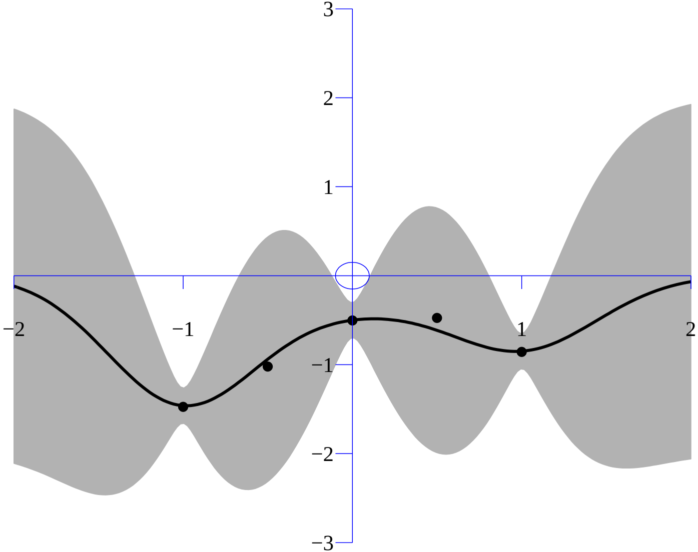
Fitting the data



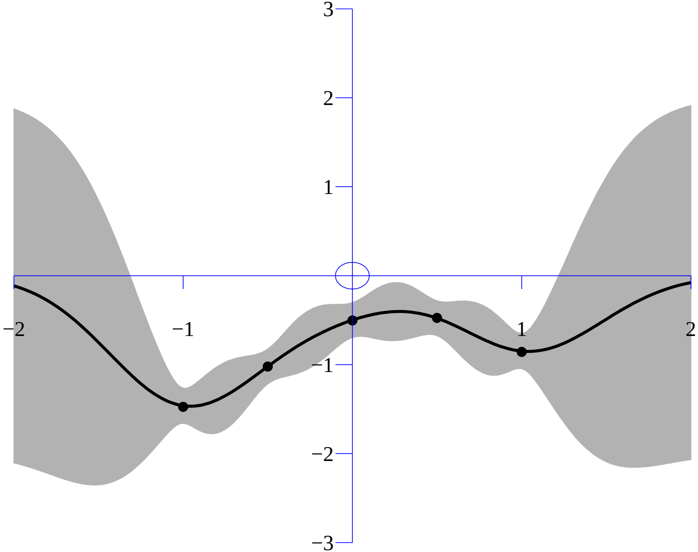
Fitting the data



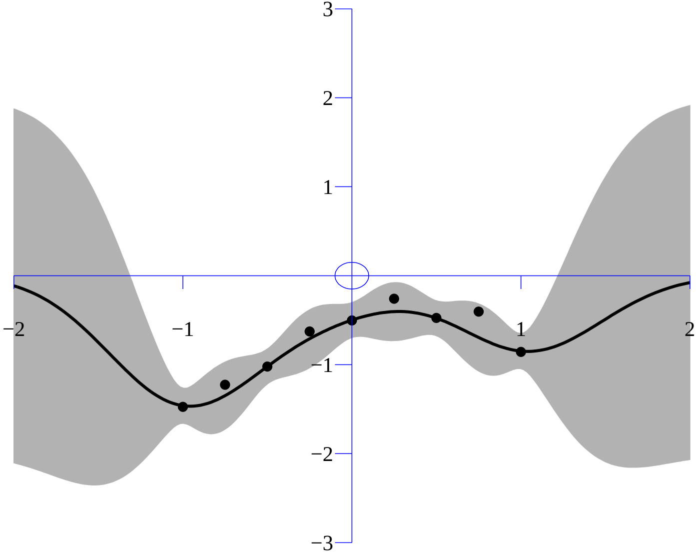
Fitting the data



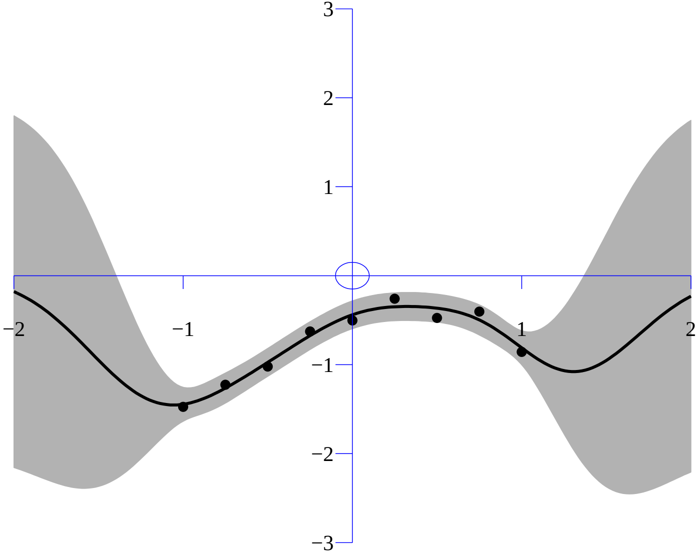
Fitting the data



Fitting the data



Fitting the data



Another view: from lin. regression to GPs

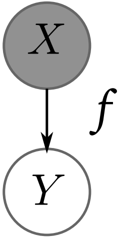
- Bayesian linear regression: $y = \phi(x)w + \epsilon$

$$p(y|x) = \int_w \quad p(y|w,x) \quad p(w) =$$
$$= \int_w \mathcal{N}(\phi(x)w, \sigma^2) \; \mathcal{N}(0, \sigma_w^2)$$

- Gaussian process: $y = f(x) + \epsilon$:

$$p(y|x) = \int_f p(y|f,x) \quad p(f|x) =$$
$$= \int_f \mathcal{N}(f, \sigma^2) \; \mathcal{N}(\mu(x), k(x,x))$$

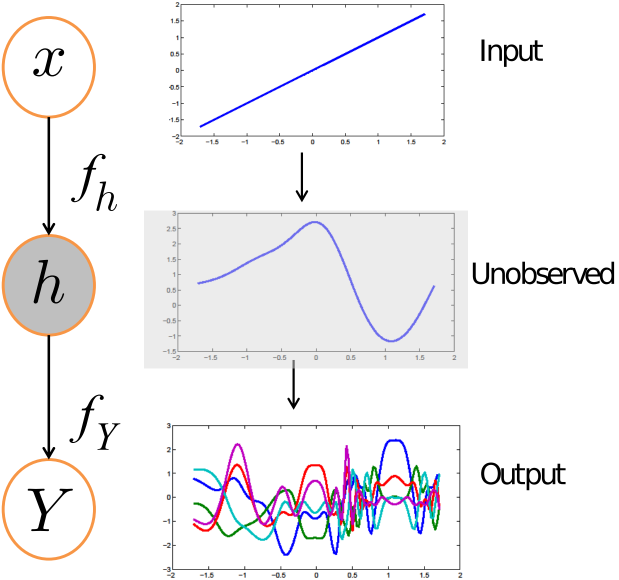
Unsupervised learning: GP-LVM



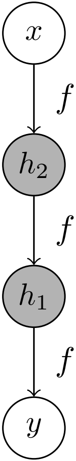
- If $\mathbf{X}$ is unobserved, treat it as a parameter and optimize over it.
- GP-LVM is interpreted as non-linear PPCA.

Outline

Sampling from a deep GP



MAP optimisation?



- Joint $= p(y|h_1)p(h_1|h_2)p(h_2|x)$
- MAP optimization is extremely problematic because:
  - Dimensionality of hs has to be decided a priori
  - Prone to overfitting, if h are treated as parameters
  - Deep structures are not supported by the model's objective but have to be forced [Lawrence & Moore '07]

Regularization solution: approximate Bayesian framework

- Analytic variational bound $\mathcal{F} \leq p(y|x)$
  - Extend Titsias' method for *variational learning of inducing variables in Sparse GPs*.
  - *Approximately* marginalise out h
- Automatic structure discovery (nodes, connections, layers)
  - Use the Automatic / Manifold Relevance Determination trick
- ...

Direct marginalisation of h is intractable (O_o)

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

Direct marginalisation of h is intractable (O_o)

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

Direct marginalisation of h is intractable (O_o)

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

Direct marginalisation of h is intractable (O_o)

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

- $p(h_1|x) = \int_{h_2, f_2} p(h_1|f_2) \underbrace{p(f_2|h_2)}_{\text{contains } (k(h_2,h_2))^{-1}} p(h_2|x)$

Direct marginalisation of h is intractable (O_o)

- New objective: $p(y|x) = \int_{h_1} \left(p(y|h_1) \int_{h_2} p(h_1|h_2) p(h_2|x) \right)$
- $p(h_1|x) = \int_{h_2, f_2} p(h_1|f_2) \underbrace{p(f_2|h_2)}_{\text{contains } \mathbf{K}^{-1}_{h_2,h_2}} p(h_2|x)$

Direct marginalisation of h is intractable (O_o)

- New objective: $p(y|x) = \int_{h_1} \left(p(y|h_1) \int_{h_2} p(h_1|h_2) p(h_2|x) \right)$
- $p(h_1|x) = \int_{h_2, f_2} p(h_1|f_2) p(f_2|h_2) p(h_2|x)$
- $p(h_1|x, \tilde{h}_2) = \int_{h_2, f_2, u_2} p(h_1|f_2) p(f_2|u_2, h_2) p(u_2|\tilde{h}_2) p(h_2|x)$

Direct marginalisation of h is intractable (O_o)

- New objective: $p(y|x) = \int_{h_1} \left(p(y|h_1) \int_{h_2} p(h_1|h_2) p(h_2|x) \right)$
- $p(h_1|x) \quad = \int_{h_2, f_2} \quad p(h_1|f_2) p(f_2|h_2) \quad p(h_2|x)$
- $p(h_1|x, \tilde{h}_2) = \int_{h_2, f_2, u_2} p(h_1|f_2) p(f_2|u_2, h_2) p(u_2|\tilde{h}_2) p(h_2|x)$
- $\log p(h_1|x, \tilde{h}_2) \geq \int_{h_2, f_2, u_2} \mathcal{Q} \log \frac{p(h_1|f_2) p(f_2|u_2, h_2) p(u_2|\tilde{h}_2) p(h_2|x)}{\mathcal{Q}}$

Direct marginalisation of h is intractable (O_o)

- New objective: $p(y|x) = \int_{h_1} \left(p(y|h_1) \int_{h_2} p(h_1|h_2) p(h_2|x) \right)$
- $p(h_1|x) \quad = \int_{h_2, f_2} \quad p(h_1|f_2) p(f_2|h_2) \quad p(h_2|x)$
- $p(h_1|x, \tilde{h}_2) = \int_{h_2, f_2, u_2} p(h_1|f_2) p(f_2|u_2, h_2) p(u_2|\tilde{h}_2) p(h_2|x)$
- $\log p(h_1|x, \tilde{h}_2) \geq \int_{h_2, f_2, u_2} \mathcal{Q} \log \frac{p(h_1|f_2) \cancel{p(f_2|u_2, h_2)} p(u_2|\tilde{h}_2) p(h_2|x)}{\mathcal{Q} = \cancel{p(f_2|u_2, h_2)} q(u_2) q(h_2)}$

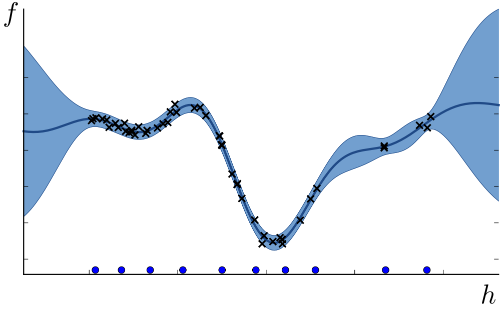
Direct marginalisation of h is intractable (O_o)

- New objective: $p(y|x) = \int_{h_1} \left(p(y|h_1) \int_{h_2} p(h_1|h_2) p(h_2|x) \right)$
- $p(h_1|x) \quad = \int_{h_2,f_2} p(h_1|f_2) p(f_2|h_2) \; p(h_2|x)$
- $p(h_1|x,\tilde{h}_2) = \int_{h_2,f_2,u_2} p(h_1|f_2) p(f_2|u_2,h_2) p(u_2|\tilde{h}_2) p(h_2|x)$
- $\log p(h_1|x,\tilde{h}_2) \geq \int_{h_2,f_2,u_2} \mathcal{Q} \log \frac{p(h_1|f_2) \cancel{p(f_2|u_2,h_2)} p(u_2|\tilde{h}_2) p(h_2|x)}{\mathcal{Q} = \cancel{p(f_2|u_2,h_2)} q(u_2) q(h_2)}$
- $\log p(h_1|x,\tilde{h}_2) \geq \int_{h_2,f_2,u_2} \mathcal{Q} \log \frac{p(h_1|f_2) p(u_2|\tilde{h}_2) p(h_2|x)}{\mathcal{Q} = q(u_2) q(h_2)}$

$p(u_2|\tilde{h}_2)$ contains $\mathbf{K}^{-1}_{\tilde{h}_2 \tilde{h}_2}$

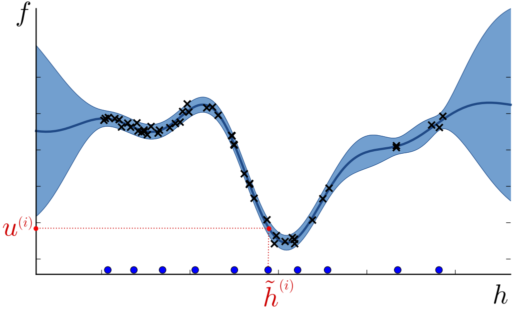
Inducing points: sparseness, tractability and Big Data

h_1	$\mathbf{f}_1$
h_2	$\mathbf{f}_2$
$\ldots$	$\ldots$
h_{30}	$\mathbf{f}_{30}$
h_{31}	$\mathbf{f}_{31}$
$\ldots$	$\ldots$
h_N	$\mathbf{f}_N$



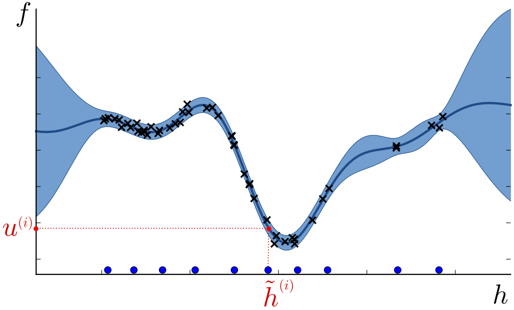
Inducing points: sparseness, tractability and Big Data

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: sparseness, tractability and Big Data

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	$\mathbf{f}_N$



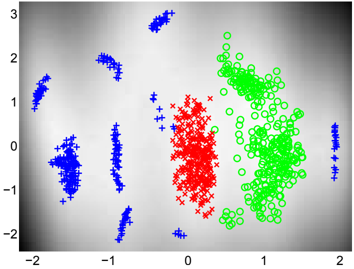
- Inducing points originally introduced for faster **(sparse) GPs**
- Our manipulation allows to **compress information** from the inputs of every layer
- This induces **tractability**
- Viewing them as **global variables**
  $\Rightarrow$ extension to **Big Data** [Hensman et al., UAI 2013]

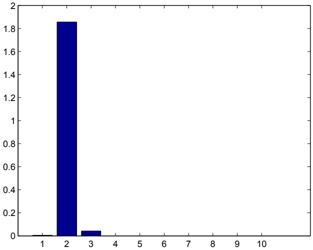
Automatic dimensionality detection

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

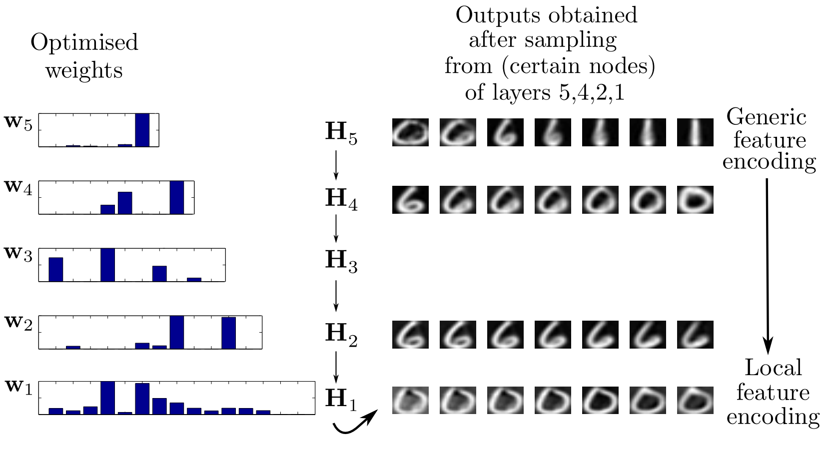
$$k_f\left(\mathbf{x}_i, \mathbf{x}_j\right) = \sigma^2 \exp\left(-\frac{1}{2}\sum_{q=1}^{Q} w_q \left(x_{i,q} - x_{j,q}\right)^2\right)$$

- Example:



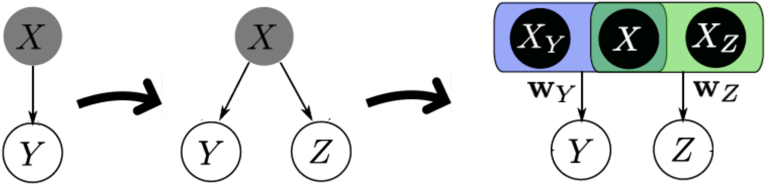


Deep GP: MNIST example



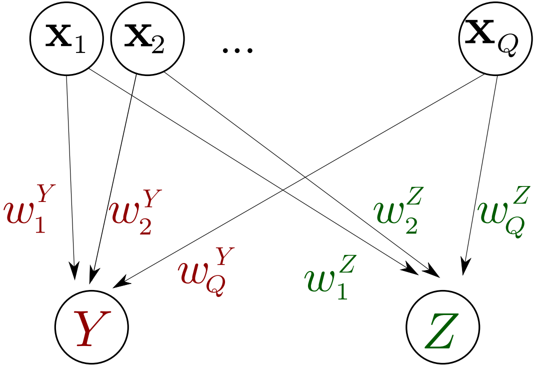
http://staffwww.dcs.sheffield.ac.uk/people/A.Damianou/research/index.html#DeepGPs

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.

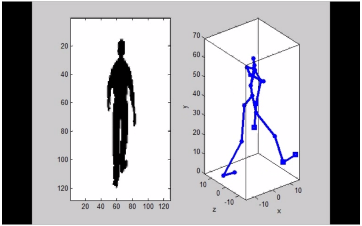
MRD weights



MRD examples

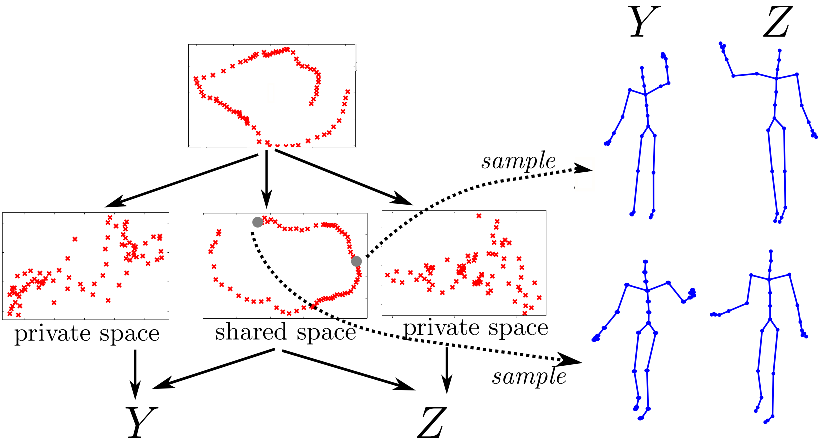
Yale faces

Motion capture / silhouette



▸ http://staffwww.dcs.sheffield.ac.uk/people/A.Damianou/research/index.html#MRD

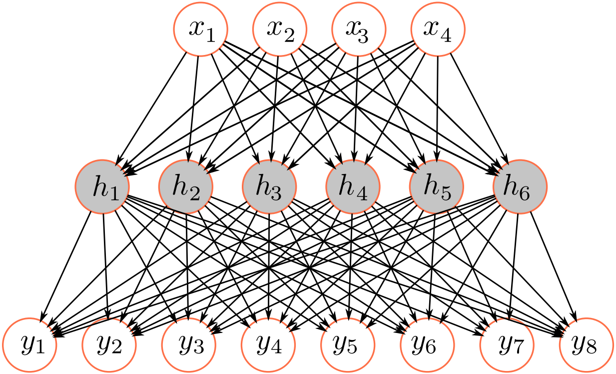
Deep GPs: Another multi-view example



Automatic structure discovery

Tools:

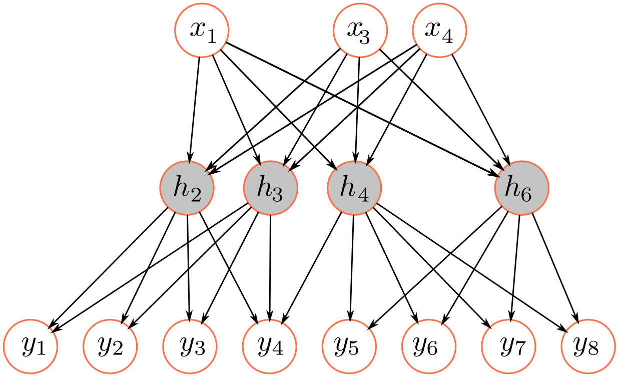
- ARD: Eliminate uncessary nodes/connections
- MRD: Conditional independencies
- Approximating evidence: Number of layers (?)



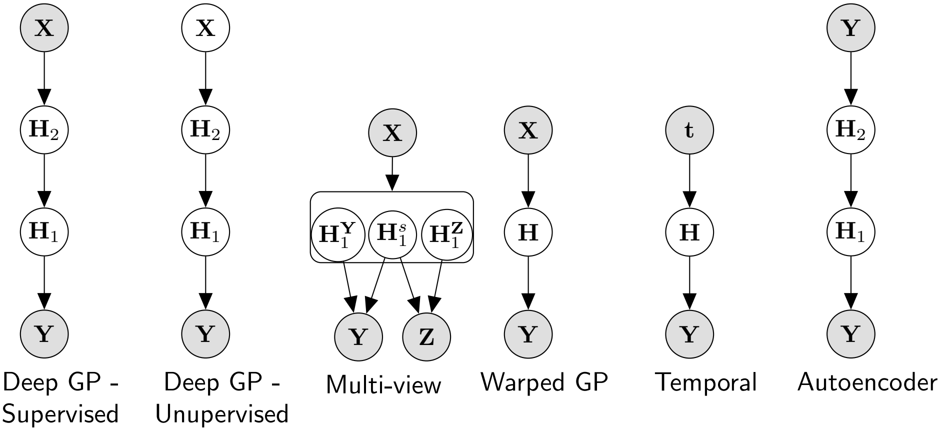
Automatic structure discovery

Tools:

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

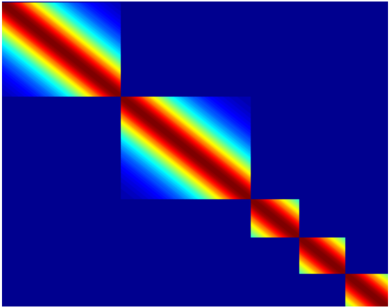


Deep GP variants

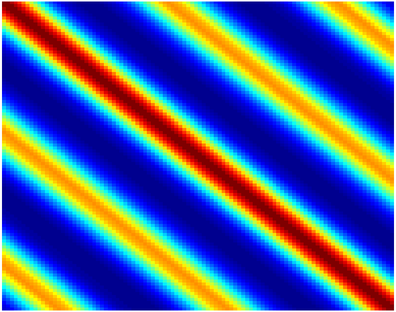


Temporal model: VGPDS

- Dynamics are encoded in the covariance matrix $K_x = k_x(\mathbf{t}, \mathbf{t})$.
- We can consider special forms for K_x.



Model individual sequences



Model periodic data

- Show videos...
- https://www.youtube.com/watch?v=i9TEoYxaBxQ
- https://www.youtube.com/watch?v=mUY1XHPnoCU

Autoencoder example

Run demo...

Summary

- A deep GP is not a GP.
- Sampling is straight-forward. Regularization and training needs to be worked out.
- The solution is a special treatment of auxiliary variables.
- Many variants: multi-view, temporal, autoencoders ...
- 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. K. Titsias (2009), "Variational learning of inducing variables in sparse Gaussian processes", AISTATS 2009
- A. C. Damianou, M. K. Titsias and N. D. Lawrence (2011), "Variational Gaussian process dynamical systems", NIPS 2011
- A. C. Damianou, C. H. Ek, M. K. Titsias and N. D. Lawrence (2012), "Manifold Relevance Determination", ICML 2012
- A. C. Damianou and N. D. Lawrence (2013), "Deep Gaussian processes", AISTATS 2013
- J. Hensman (2013), "Gaussian processes for Big Data", UAI 2013