Modelling and consolidating complex data with Gaussian process models

Andreas Damianou

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

ICS-FORTH, Heraklion, Greece, 10/06/2014

Outline

Part 1: Gaussian processes

- GPs for nonparametric, nonlinear regression

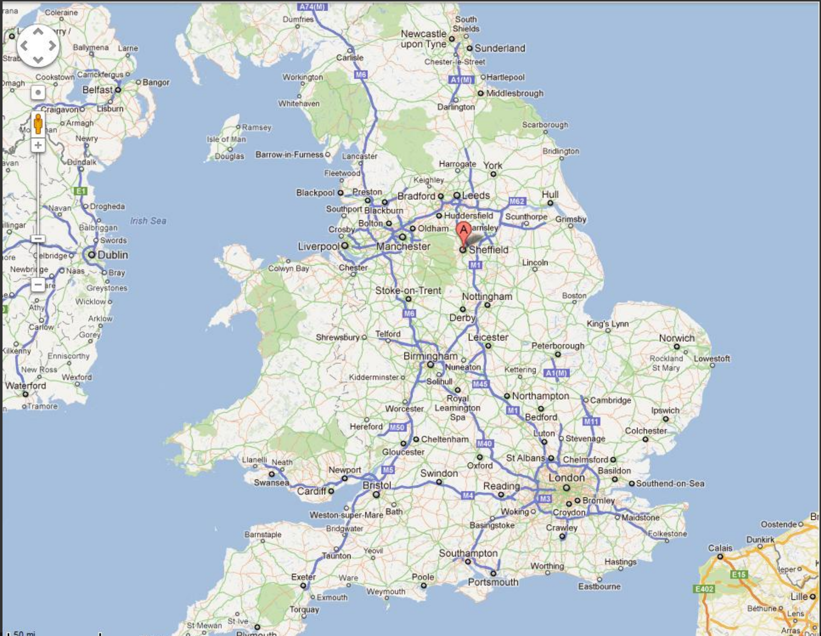
Introducing latent spaces: GP-LVM

Multiple views: MRD

Summary

Part 2: Deep Gaussian processes

Sheffield

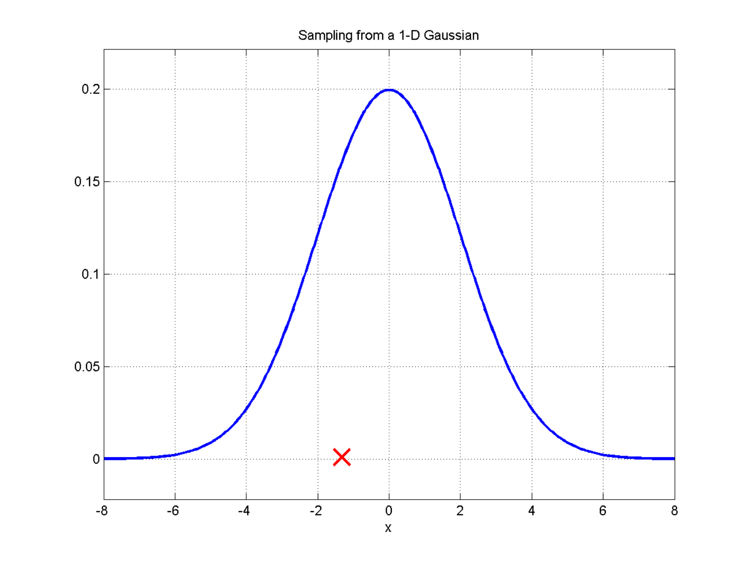


Sheffield
Manchester
Leeds
Liverpool
Birmingham
London
Nottingham
Derby
Leicester
Bradford
Huddersfield
Oldham
Lincoln
Hull
York
Stoke-on-Trent
Newcastle upon Tyne
Sunderland
Middlesbrough
Bristol
Cardiff
Southampton
Portsmouth
Norwich
Dublin
Belfast
Irish Sea

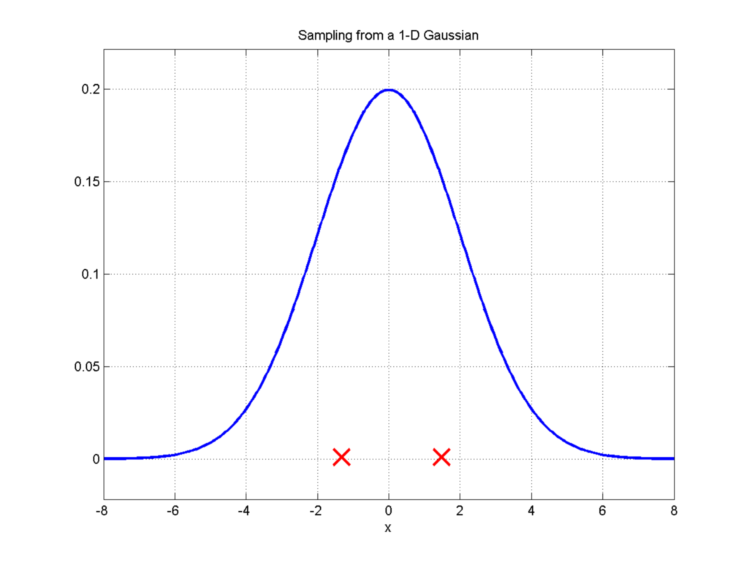
Introducing Gaussian Processes:

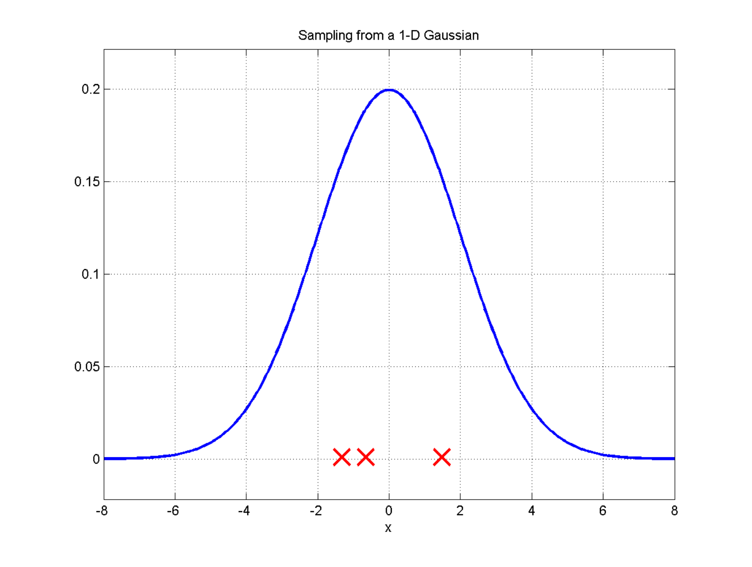
- A Gaussian distribution depends on a mean and a covariance vector / matrix.
- A Gaussian process depends on a mean and a covariance function.

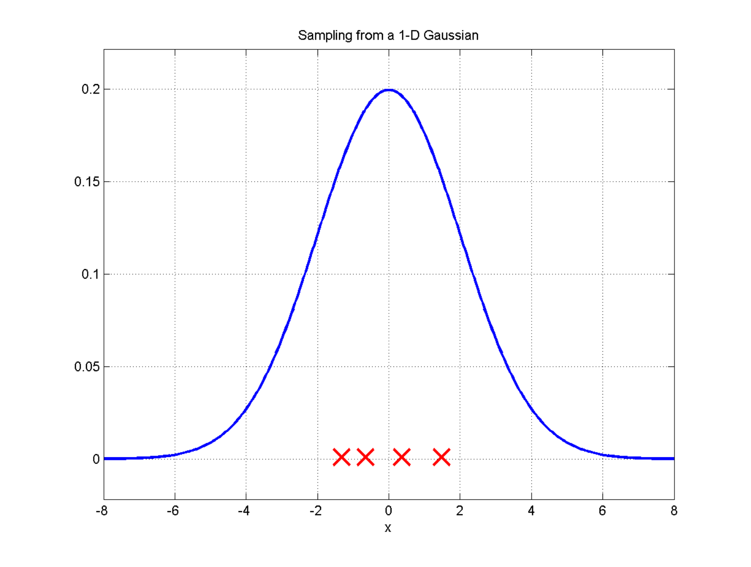
Next: Demo, from Gaussian distributions to Gaussian processes.

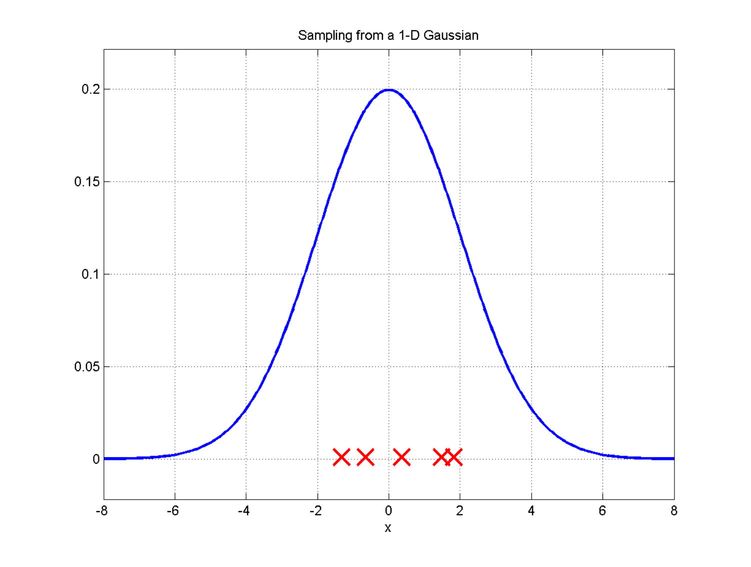


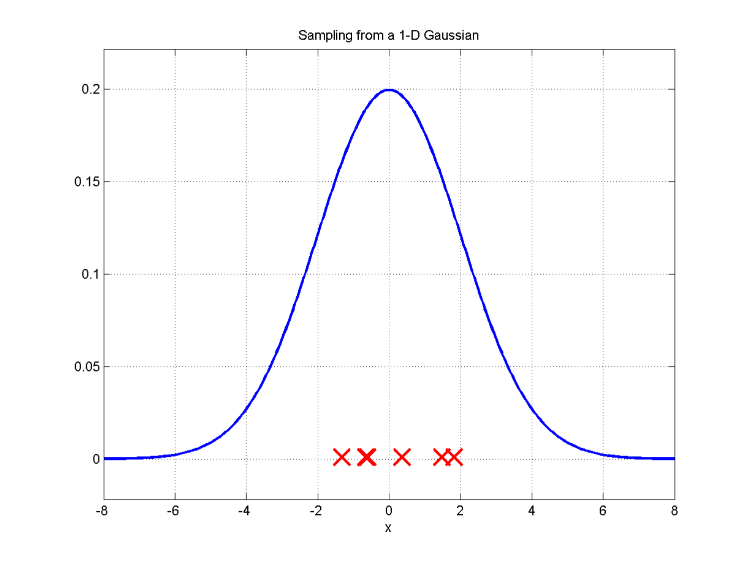
Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x


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

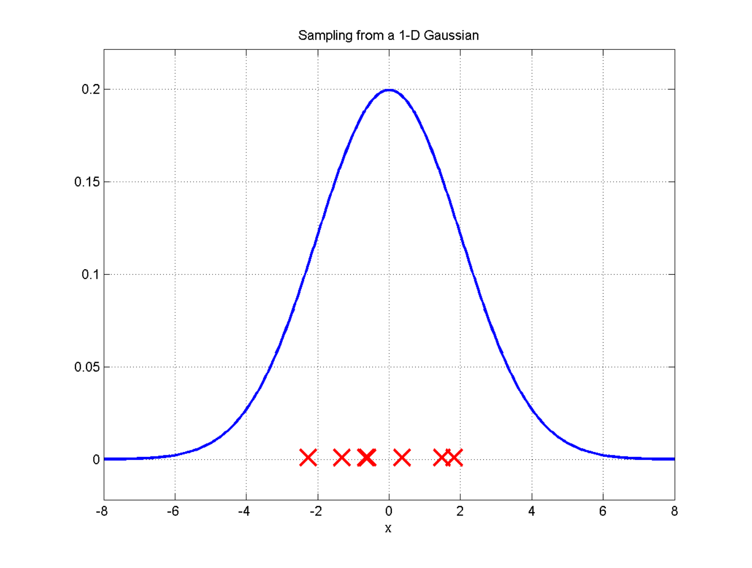

Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x


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

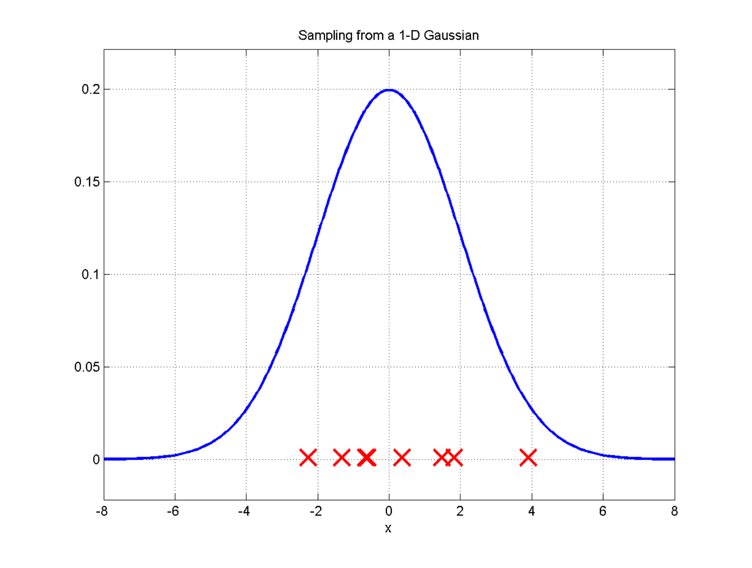

Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x

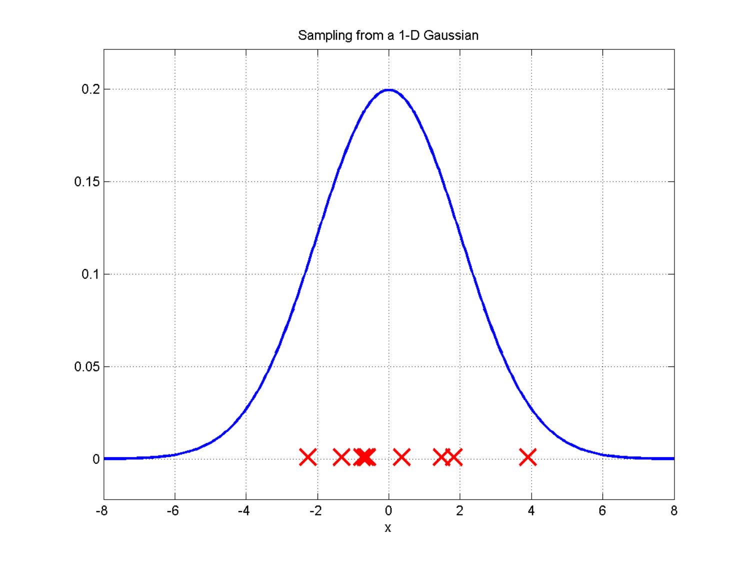


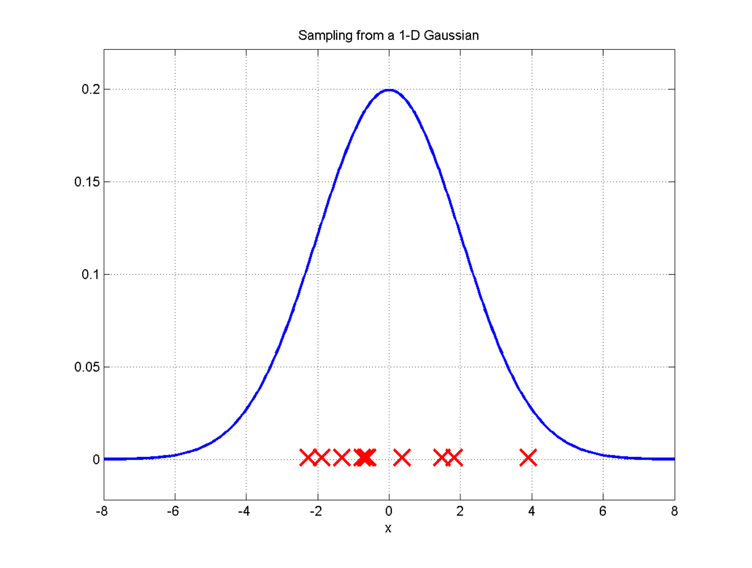
Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x



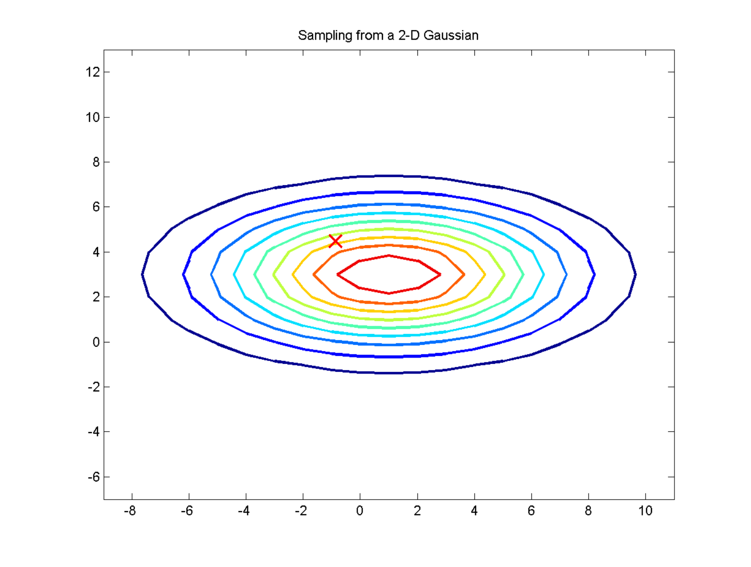
Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x

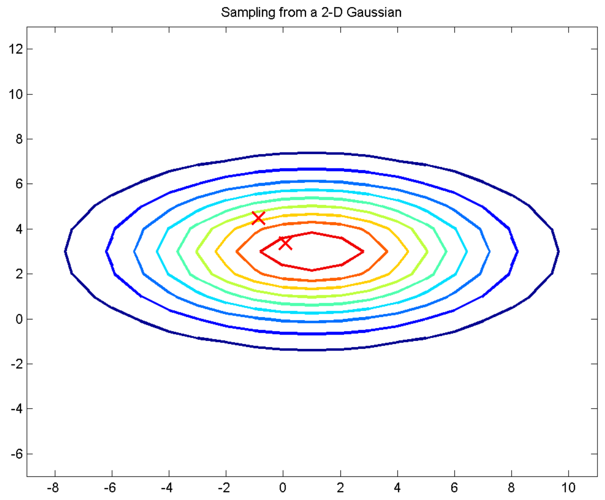

Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x

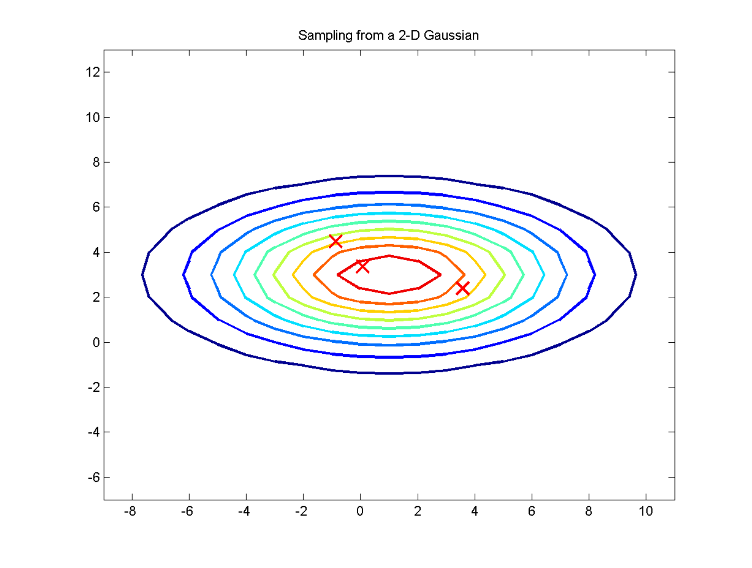

Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x

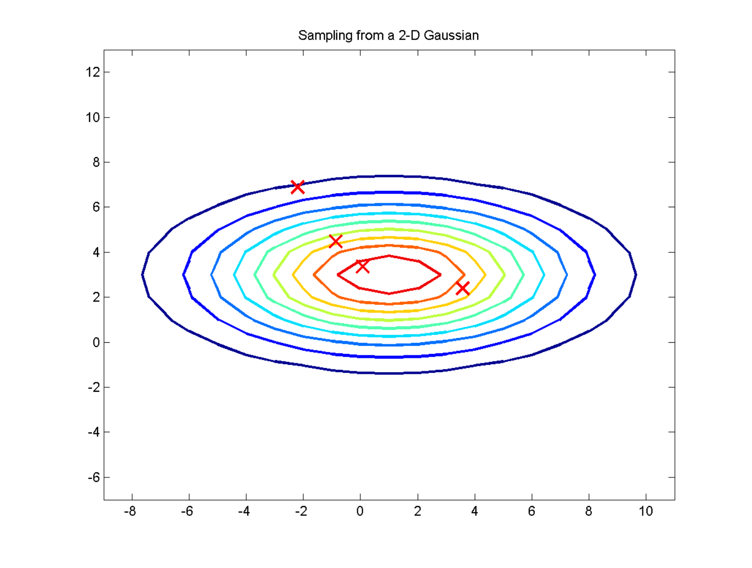


Sampling from a 1-D Gaussian
0.2
0.15
0.1
0.05
0
-8
-6
-4
-2
0
2
4
6
8
x

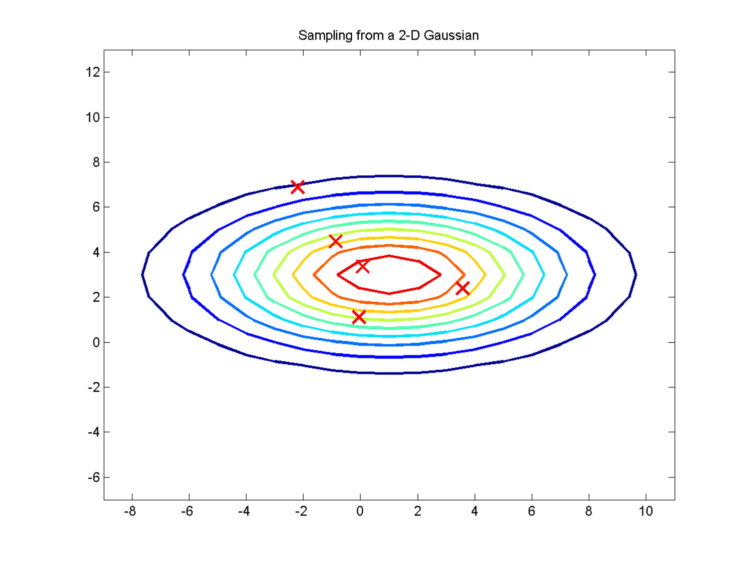

Sampling from a 2-D Gaussian
12
10
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8
10


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

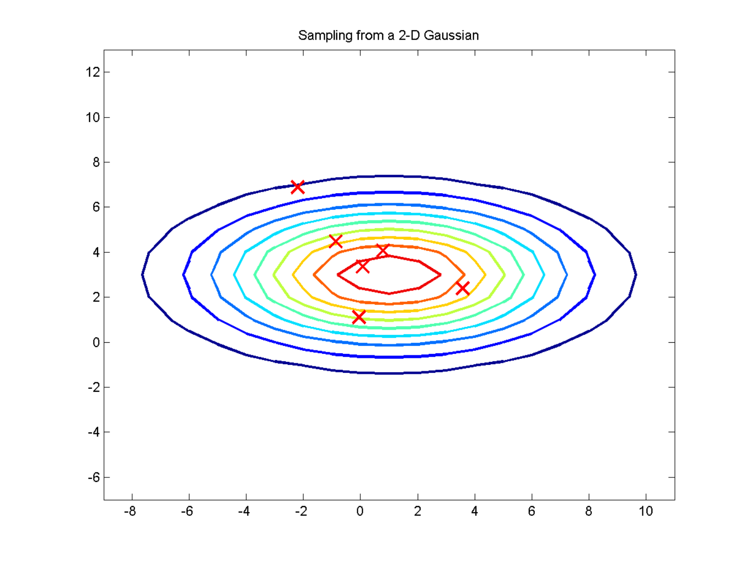

Sampling from a 2-D Gaussian
12
10
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8
10



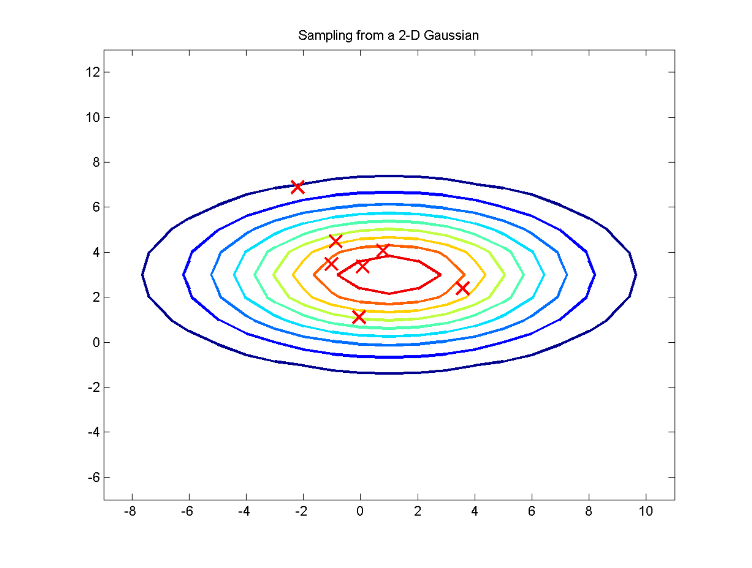
Sampling from a 2-D Gaussian
12
10
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8
10

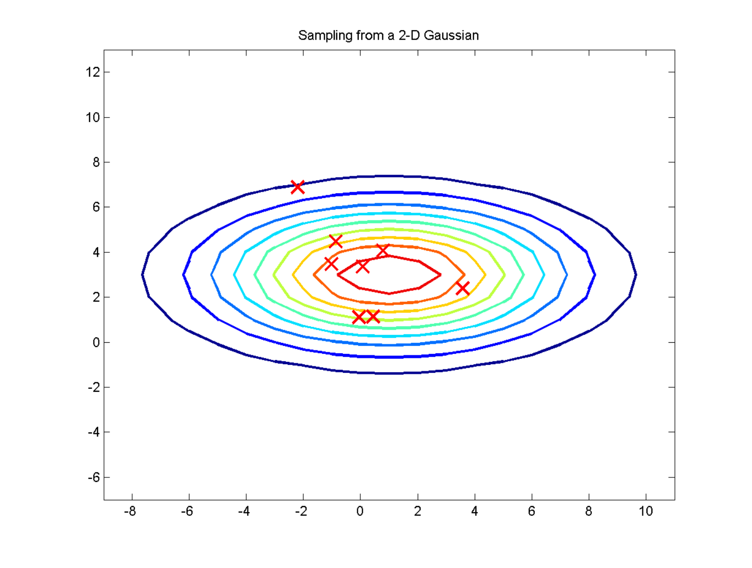


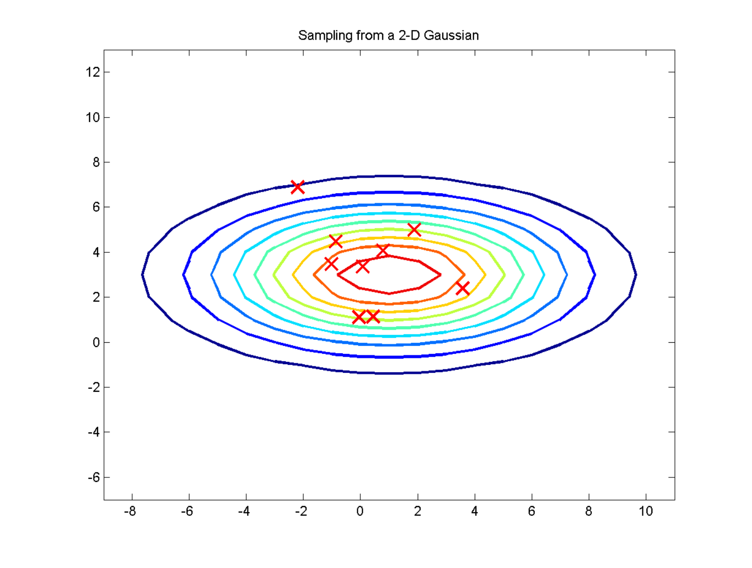
Sampling from a 2-D Gaussian
12
10
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8
10



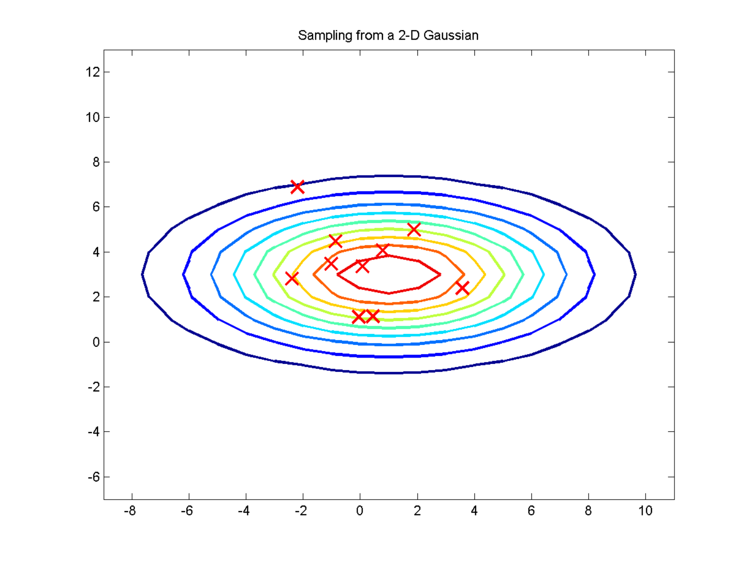
Sampling from a 2-D Gaussian
12
10
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8
10

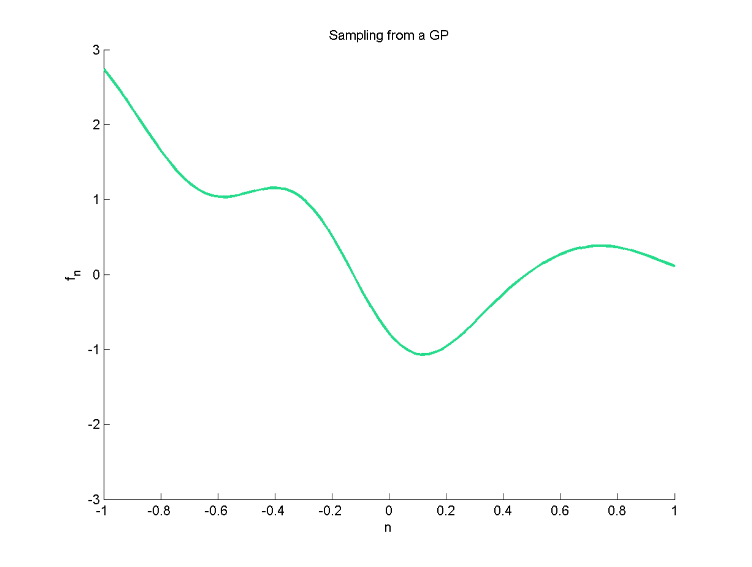

Sampling from a 2-D Gaussian
12
10
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8
10

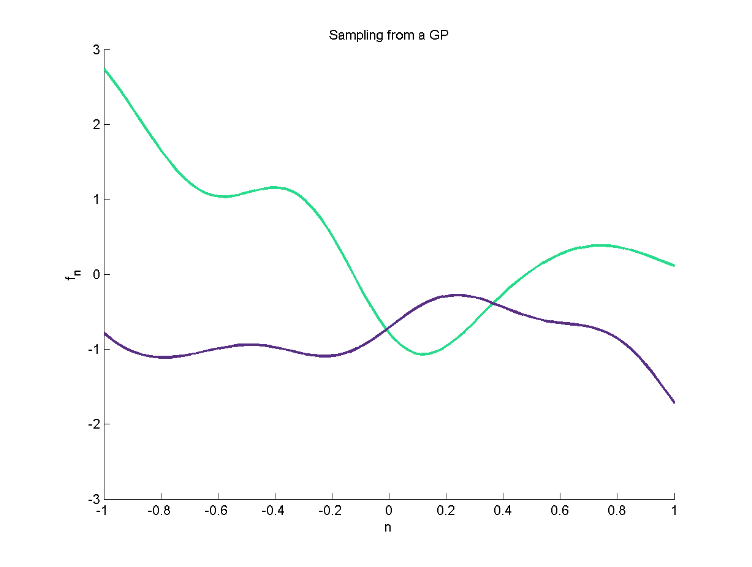

Sampling from a 2-D Gaussian
12
10
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8
10



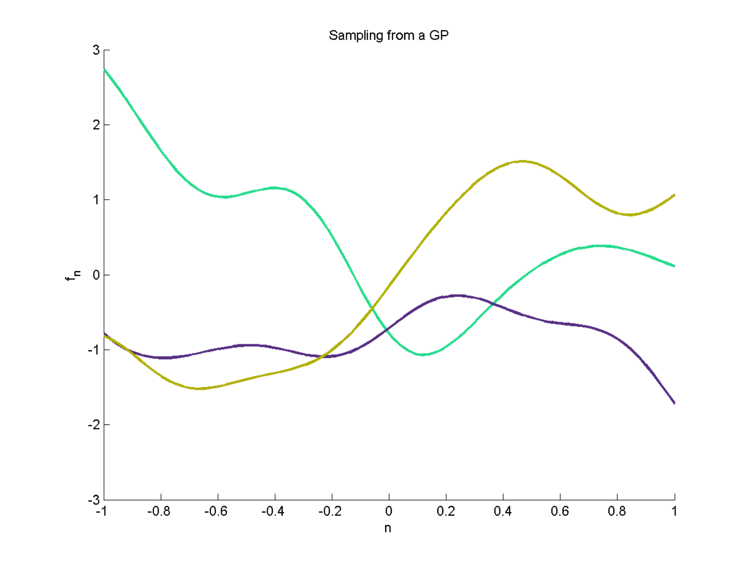
Sampling from a 2-D Gaussian
12
10
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8
10

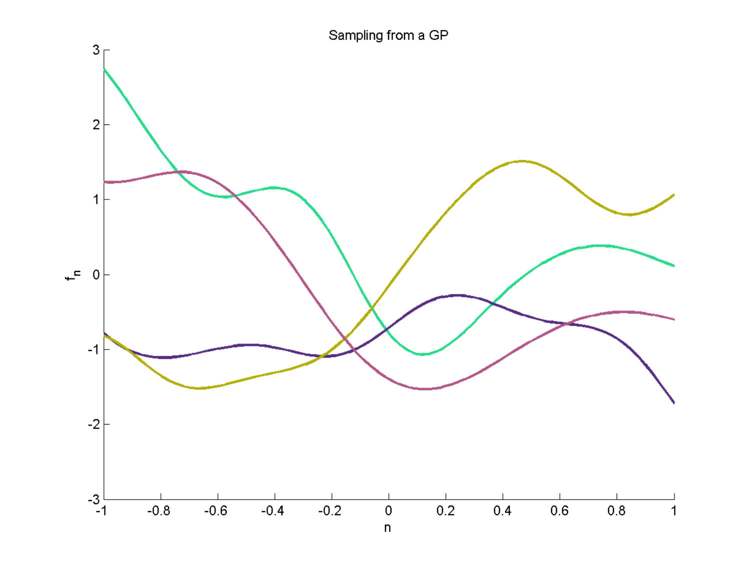

Sampling from a 2-D Gaussian
12
10
8
6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6
8
10

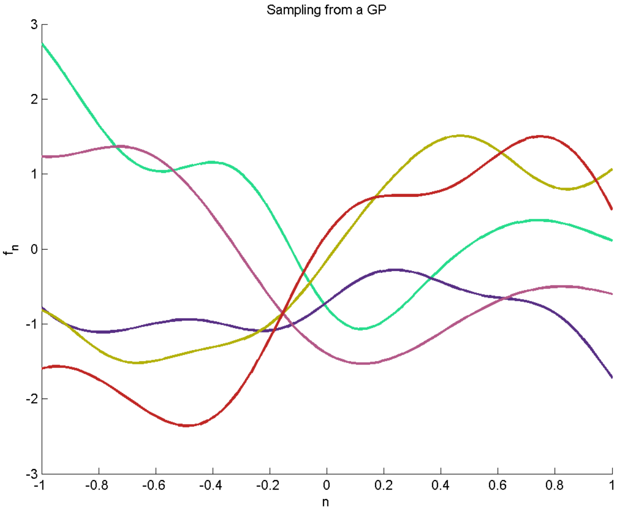

Sampling from a GP
3
2
1
0
-1
-2
-3
f_n
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
n



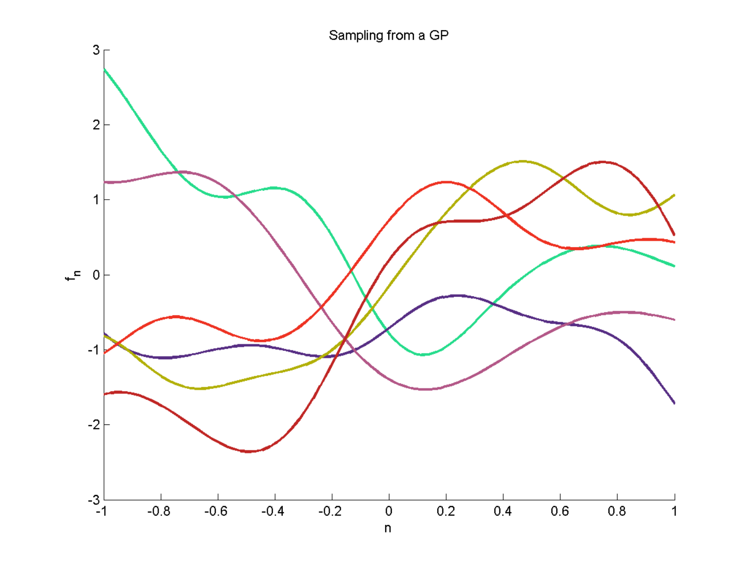
Sampling from a GP
3
2
1
0
-1
-2
-3
f_n
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
n

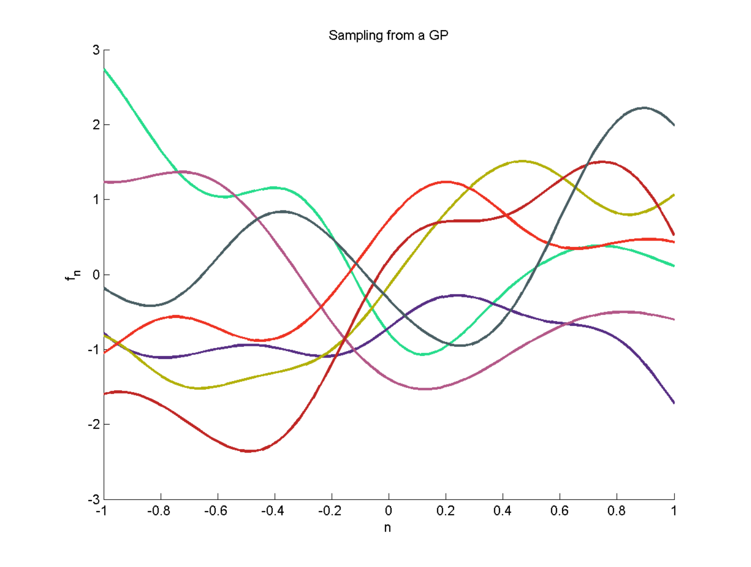

Sampling from a GP
3
2
1
0
-1
-2
-3
f_n
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
n

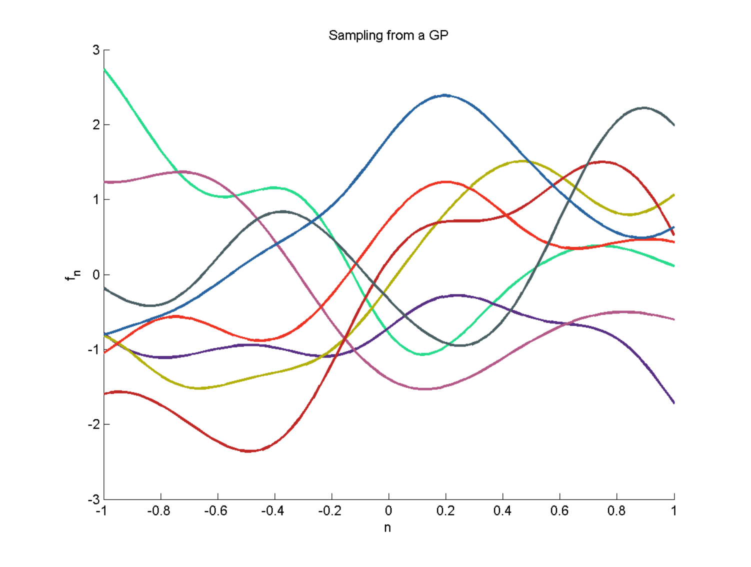

Sampling from a GP
3
2
1
0
-1
-2
-3
f_n
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
n



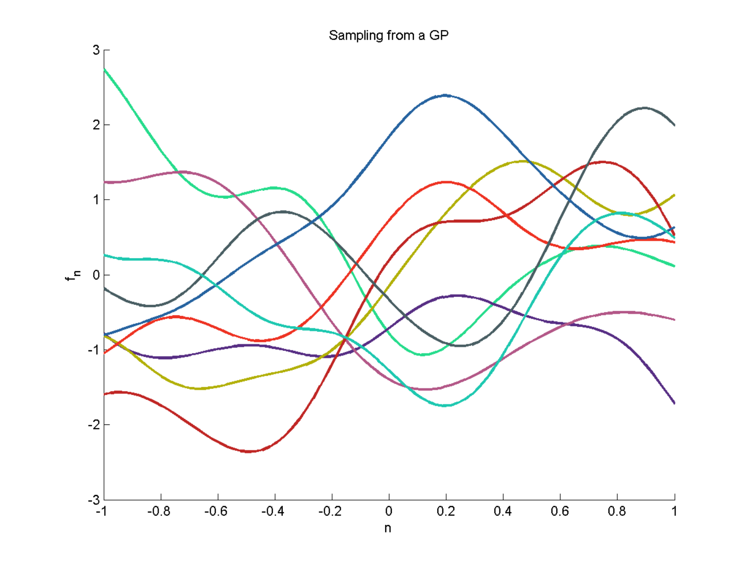
Sampling from a GP
3
2
1
0
-1
-2
-3
f_n
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
n


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

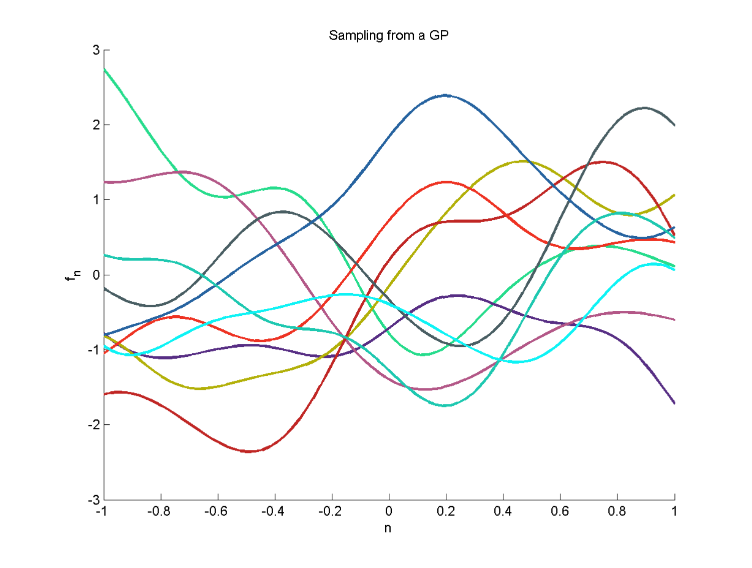

Sampling from a GP
3
2
1
0
-1
-2
-3
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
n
f_n



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
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
n
f_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!

$$p(f_A, f_B) \sim \mathcal{N}(\mu, \mathbf{K}).$$

with:

$$\mu = \begin{bmatrix} \mu_A \\ \mu_B \end{bmatrix} \text{ and } \mathbf{K} = \begin{bmatrix} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ \mathbf{K}_{BA} & \mathbf{K}_{BB} \end{bmatrix}$$

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

$$p(f_A, f_B) \sim \mathcal{N}(\mu, \mathbf{K}). \quad \text{then:}$$

$$p(f_A) = \int_{f_B} p(f_A, f_B) \mathsf{d} f_B = \mathcal{N}(\mu_A, \mathbf{K}_{AA})$$

$$p(f_B) = \int_{f_A} p(f_A, f_B) \mathsf{d} f_A = \mathcal{N}(\mu_B, \mathbf{K}_{BB})$$

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

$$p(f_A, f_B) \sim \mathcal{N}(\mu, \mathbf{K}). \quad \text{then:}$$

$$p(f_A) = \int_{f_B} p(f_A, f_B)\mathsf{d}f_B = \mathcal{N}(\mu_A, \mathbf{K}_{AA})$$

$$p(f_B) = \int_{f_A} p(f_A, f_B)\mathsf{d}f_A = \mathcal{N}(\mu_B, \mathbf{K}_{BB})$$

In the GP context:

$$\begin{aligned} p(\overbrace{f_1, f_2, \cdots, f_N}^{\text{training data}}) &= p\left(f(x_1), f(x_2), \cdots, f(x_N)\right) \\ &= \int_{\mathbb{R}-\{X\}} p\left(f\left(\{x_i \in \mathbb{R}\}\right)\right) \\ &= \mathcal{N}(\mu_X, \mathbf{K}_{XX}) \end{aligned}$$

Posterior is also Gaussian!

$p(f_A, f_B) \sim \mathcal{N}(\mu, \mathbf{K}).$ then:

$$p(f_A|f_B) = \mathcal{N}(\cdots, \cdots)$$

$$p(f_B|f_A) = \mathcal{N}(\cdots, \cdots)$$

Posterior is also Gaussian!

$$p(f_A, f_B) \sim \mathcal{N}(\mu, \mathbf{K}). \quad \text{then:}$$
$$p(f_A | f_B) = \mathcal{N}(\cdots, \cdots)$$
$$p(f_B | f_A) = \mathcal{N}(\cdots, \cdots)$$

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

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

Another view: from lin. regression to GPs

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

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

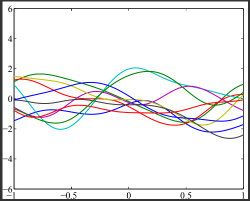
- Gaussian process: $\quad\quad y = f(x) + \epsilon$:

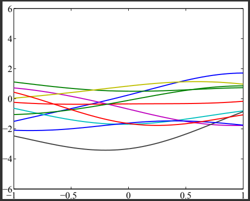
$$p(y|x) = \int_f p(y|f,x) \quad\quad p(f|x) =$$
$$= \int_f \mathcal{N}(f, \sigma^2) \;\; \mathcal{N}(\mu(x), k(X,X))$$

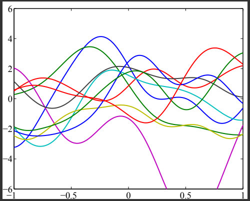
where k is any valid *covariance function*.

Covariance samples and hyperparameters

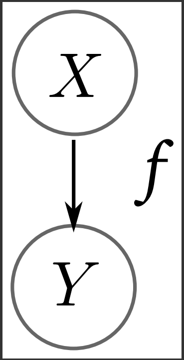
- The hyperparameters of the cov. function define the properties (and NOT an explicit form) of the sampled functions





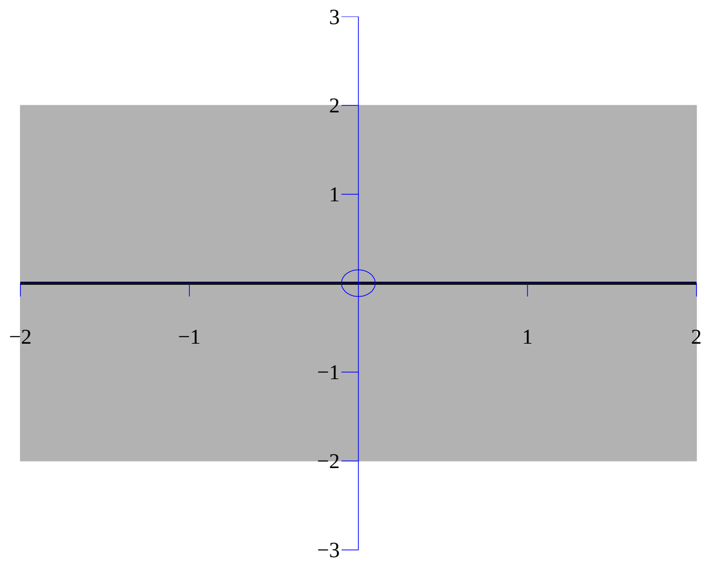


Formally...

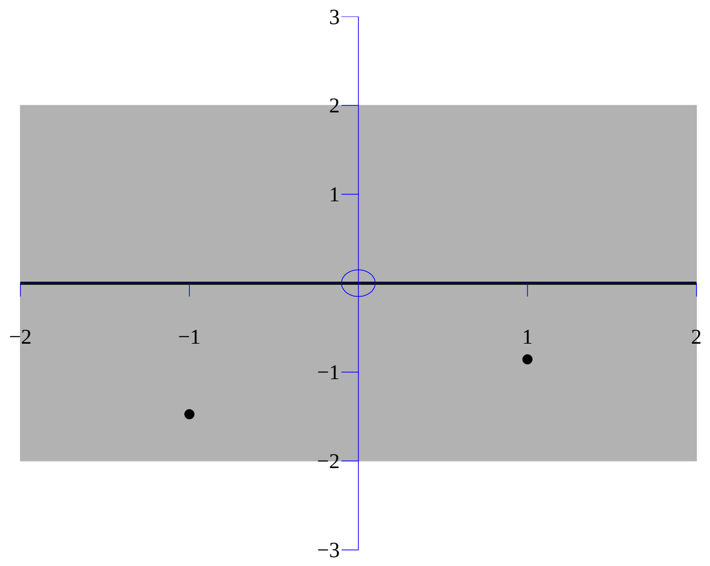


- We write: $f \sim \mathcal{GP}(0, k(x, x))$
- Optimize w.r.t the parameters of k

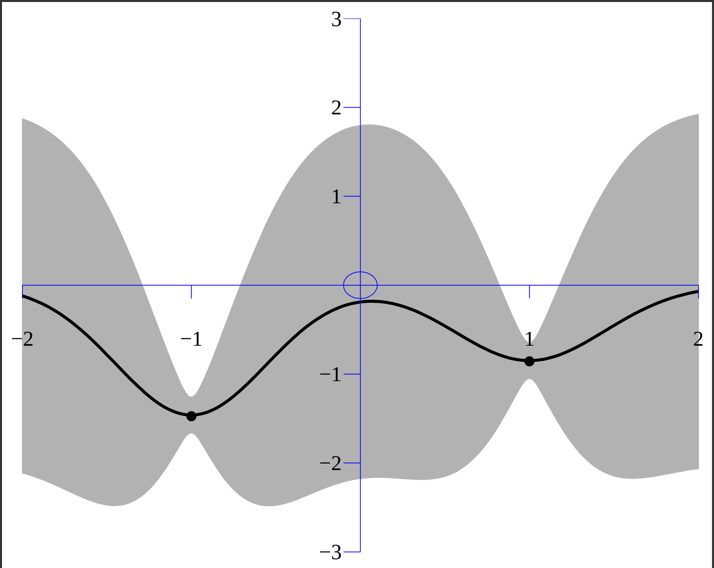
Fitting the data



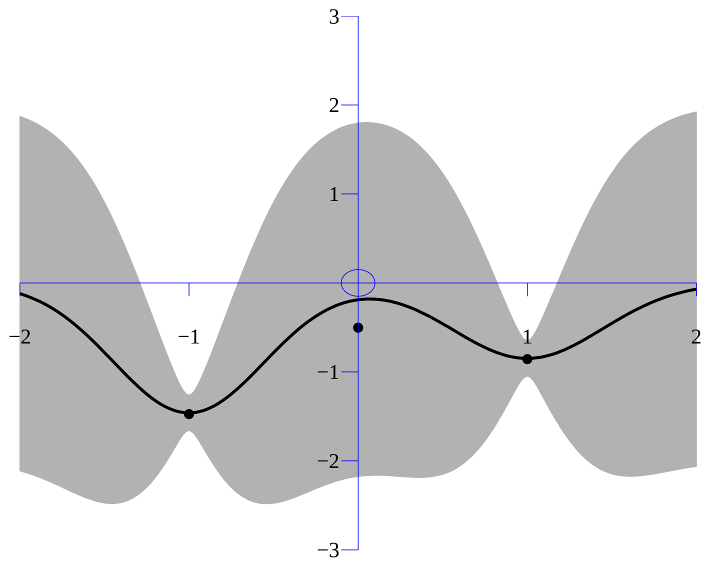
Fitting the data



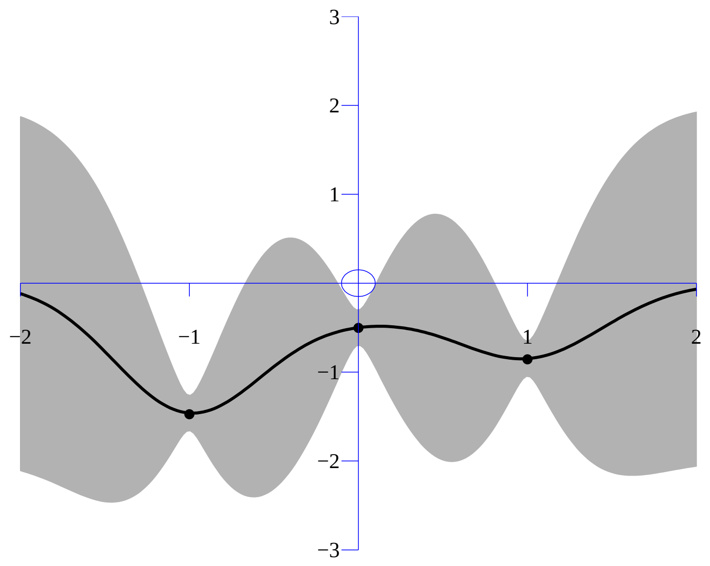
Fitting the data



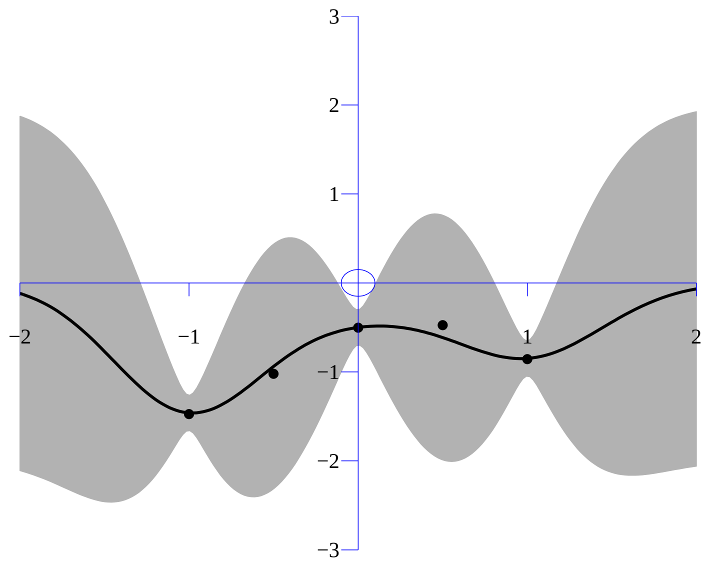
Fitting the data



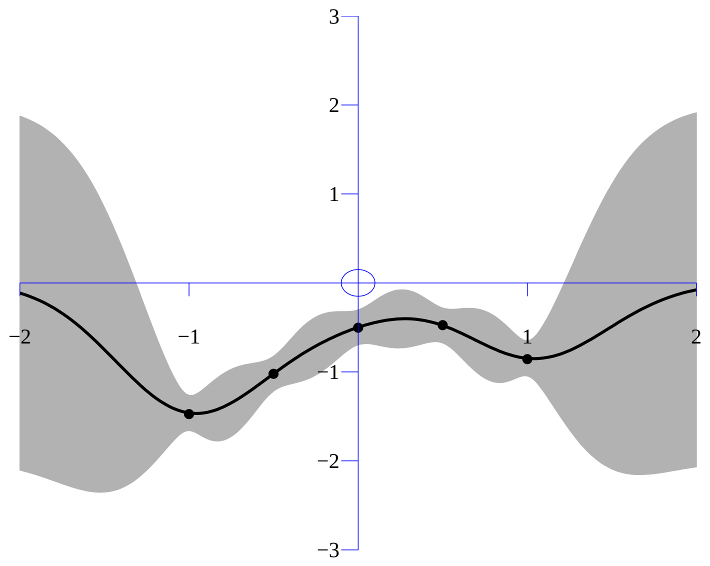
Fitting the data



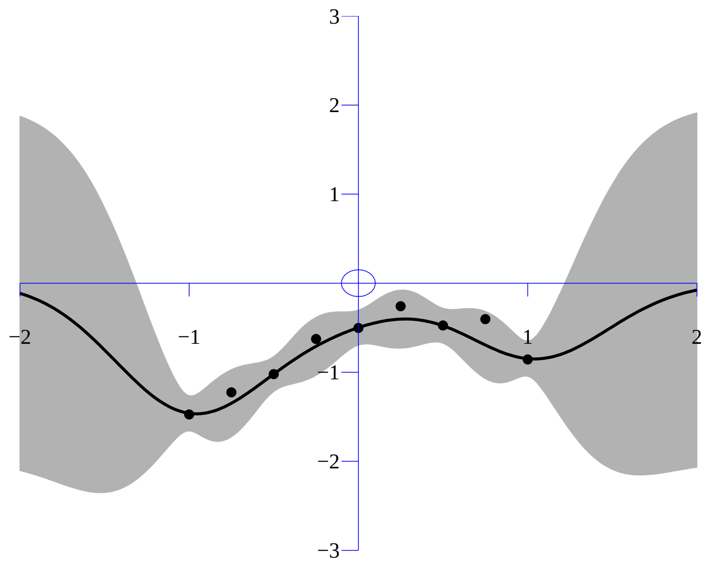
Fitting the data



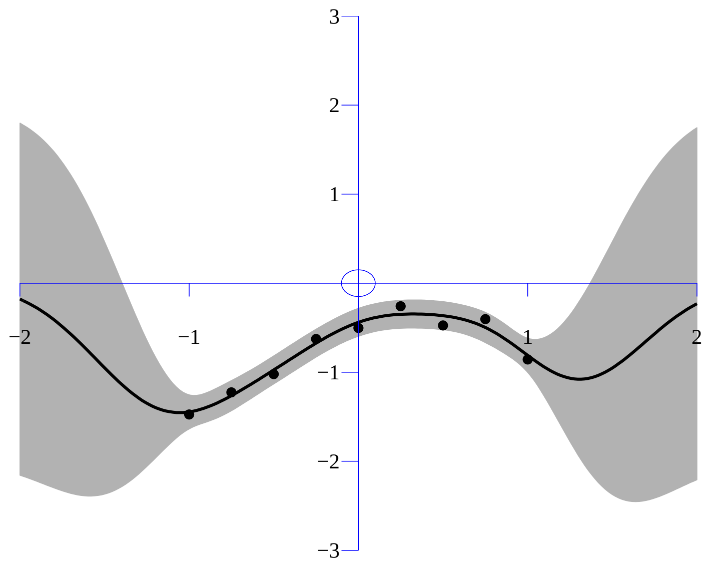
Fitting the data



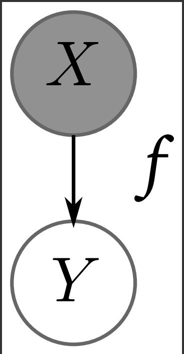
Fitting the data



Fitting the data

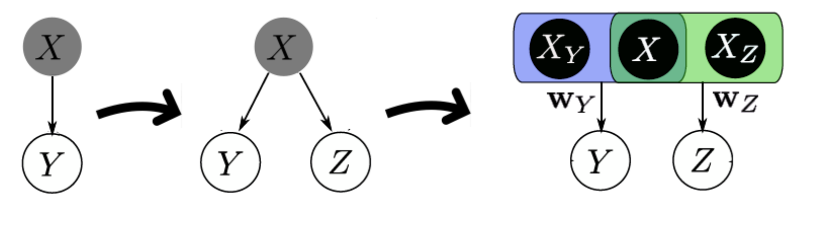


Unsupervised learning: GP-LVM

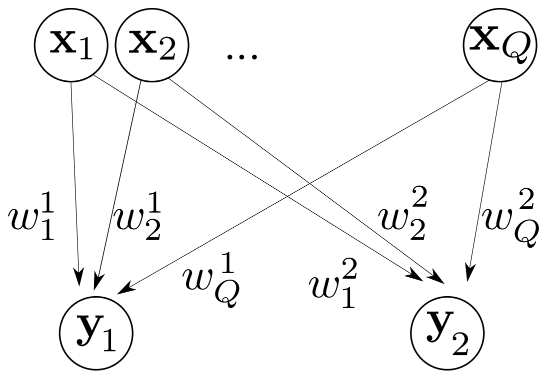


- If X is unobserved, treat it as a parameter and optimize over it.
- X is called the *latent space* assumed to have generated the (noisy) data.
- GP-LVM is interpreted as non-linear PPCA.

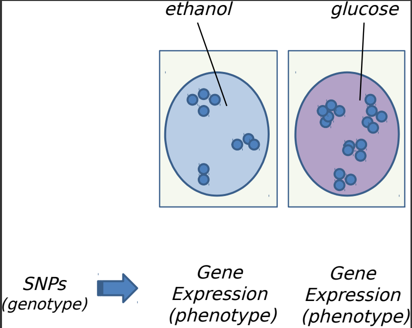
Manifold Relevance Determination



- 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.


$\mathbf{x}_1$
$\mathbf{x}_2$
...
$\mathbf{x}_Q$
w_1^1
w_2^1
w_2^2
w_Q^2
w_Q^1
w_1^2
$\mathbf{y}_1$
$\mathbf{y}_2$

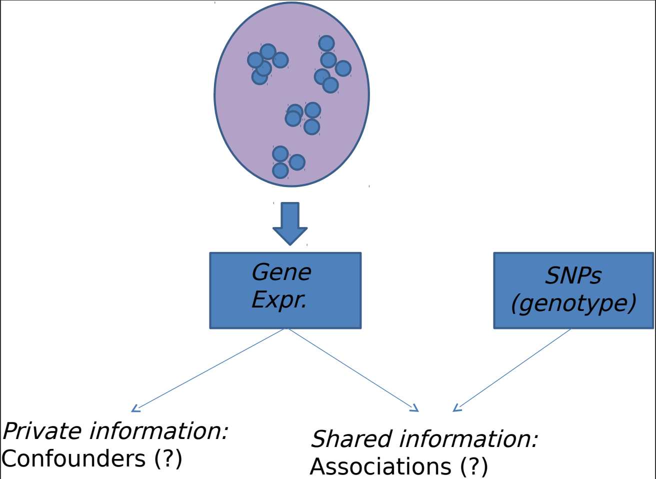
Consolidating complementary experimental data



Shared information: biological signal / confounders
Private information: environmental confounders

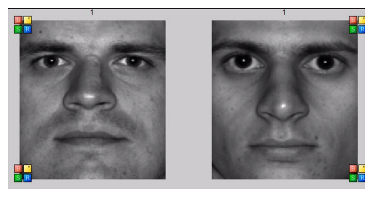
Confounders: Statistical relationships that do not reflect the true causality in the data

Discovering commonalities in heterogeneous data

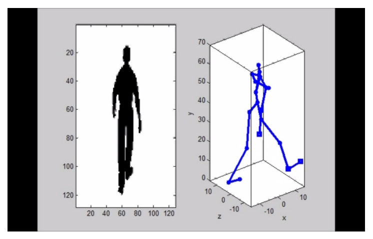


Example 3

Yale faces



Motion capture / silhouette



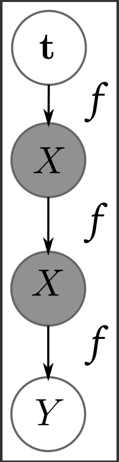
Summary

- Observed data are noisy. Assuming a latent space representation helps in modelling/analysis.
- The emerging structure of the latent space helps in data understanding.
- All of our assumptions can be naturally taken into account in the latent space.
- We can obtain temporal,multi-view, deep models.

Thanks

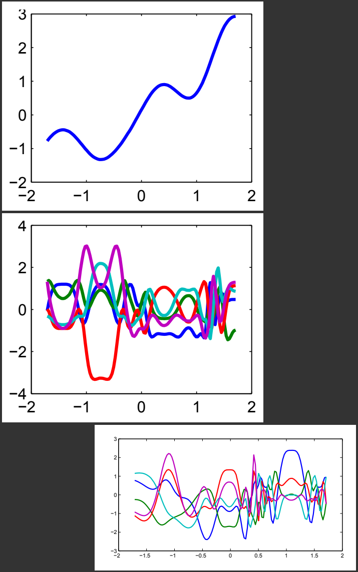
Thanks to Neil Lawrence, James Hensman, Michalis Titsias, Carl Henrik Ek.

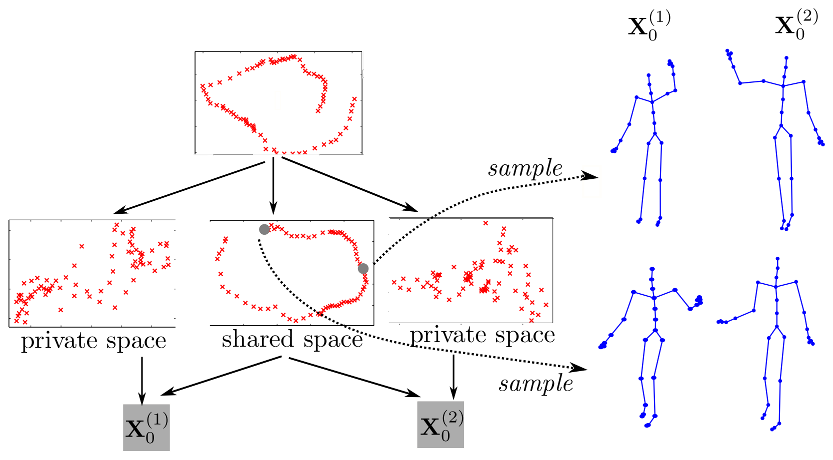
Deep Gaussian processes



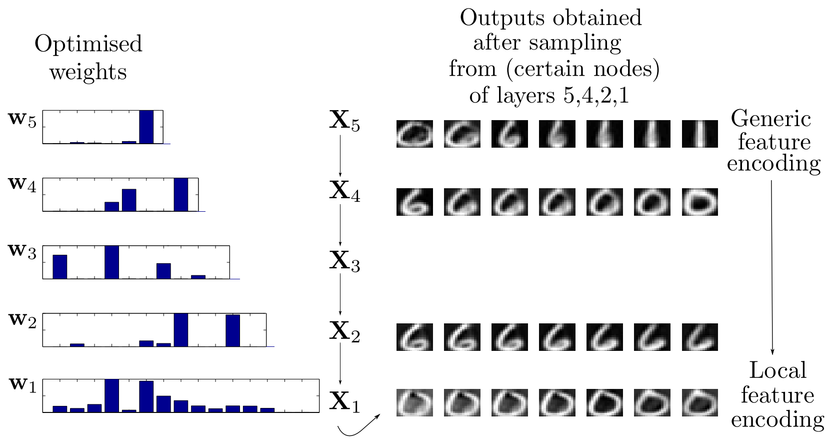
- Construct latent spaces hierarchically:
  $f = f_1(f_2(f_3 \cdots (t)))$
- Supervised / unsupervised
- A deep GP is NOT a GP! Can learn much more complicated functions!

Sampling from a deep GP

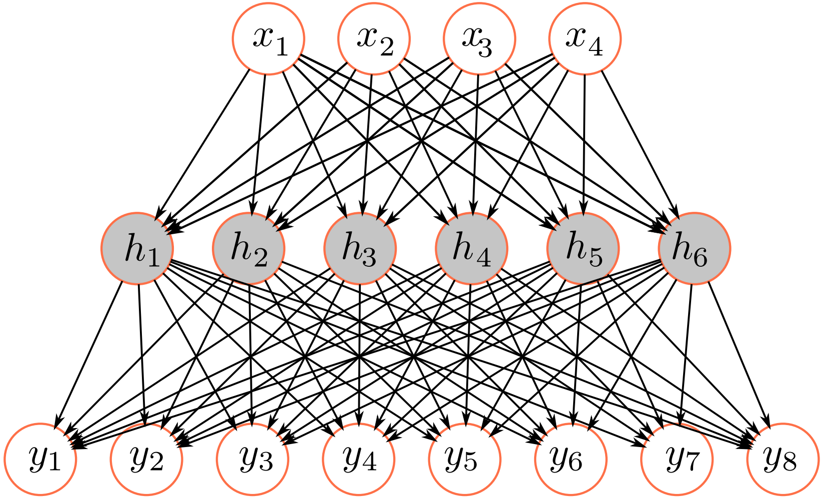


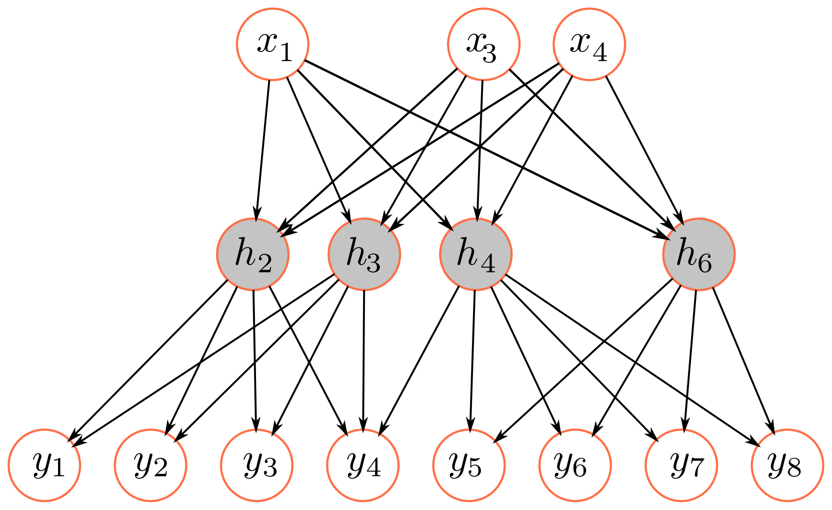


$\mathbf{X}_0^{(1)}$
$\mathbf{X}_0^{(2)}$
private space
shared space
private space
sample
sample
$\mathbf{X}_0^{(1)}$
$\mathbf{X}_0^{(2)}$



Optimised weights
$\mathbf{w}_5$
$\mathbf{w}_4$
$\mathbf{w}_3$
$\mathbf{w}_2$
$\mathbf{w}_1$
$\mathbf{X}_5$
$\mathbf{X}_4$
$\mathbf{X}_3$
$\mathbf{X}_2$
$\mathbf{X}_1$
Outputs obtained after sampling from (certain nodes) of layers 5,4,2,1
Generic feature encoding
Local feature encoding


x_1
x_2
x_3
x_4
h_1
h_2
h_3
h_4
h_5
h_6
y_1
y_2
y_3
y_4
y_5
y_6
y_7
y_8


x_1
x_3
x_4
h_2
h_3
h_4
h_6
y_1
y_2
y_3
y_4
y_5
y_6
y_7
y_8

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.
- 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