Deep non-parametric learning with Gaussian processes

Andreas Damianou

Department of Computer Science, University of Sheffield, UK

School of Computing Science, Glasgow, 10/06/2015

Outline

Part 1: A general view
- Deep modelling and deep GPs

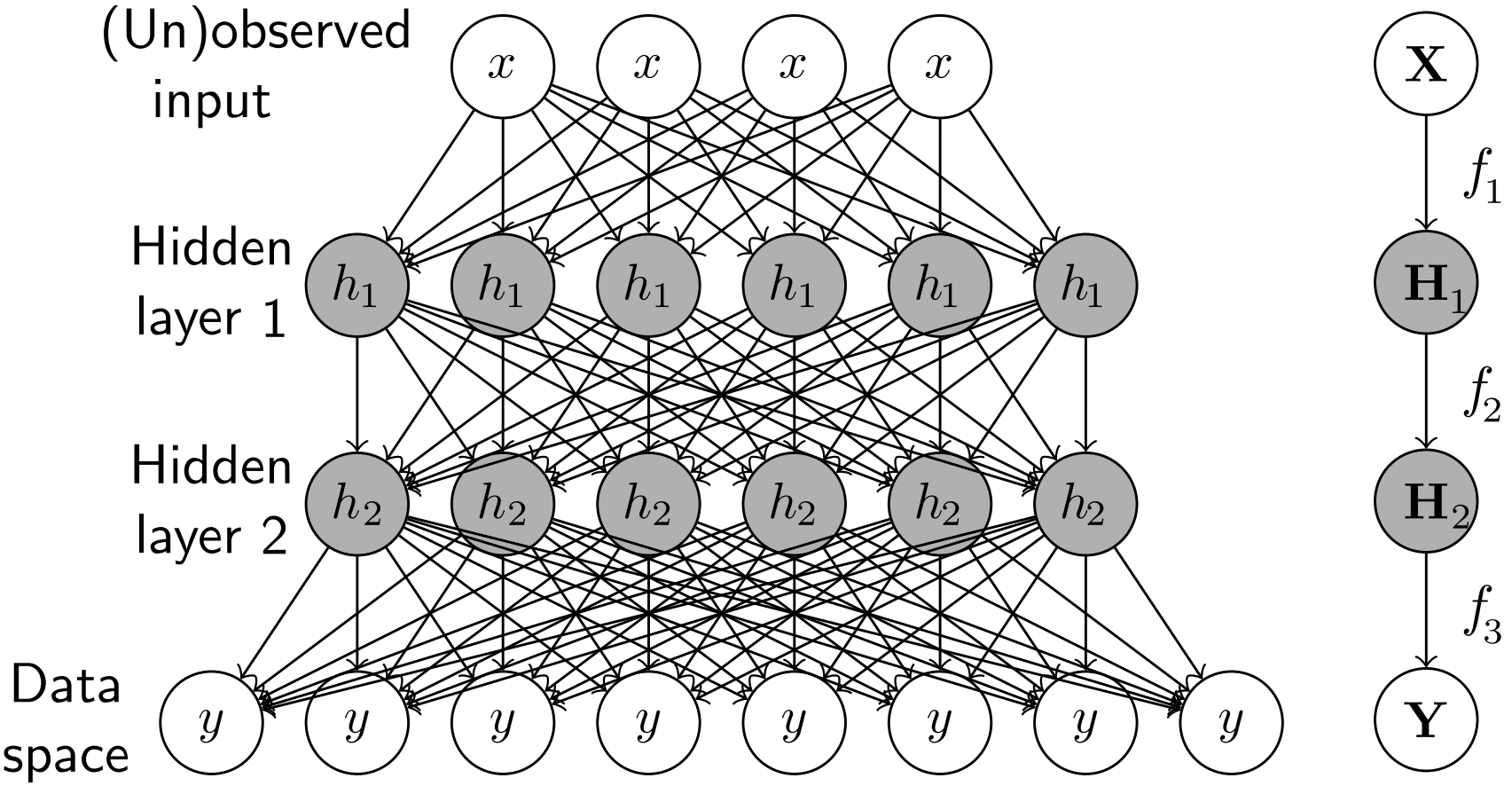
Part 2: Gaussian processes
- GPs as infinite dimensional Gaussian distributions
- Unsupervised GPs: GP-LVM

Part 3: Deep Gaussian processes
- Bayesian regularization
- Inducing Points
- Structure: ARD and MRD (multi-view)

Summary

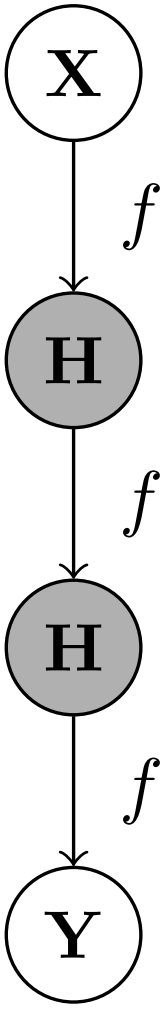
Outline

Deep learning (directed graph)



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

Deep Gaussian processes - Big Picture



Deep GP:

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

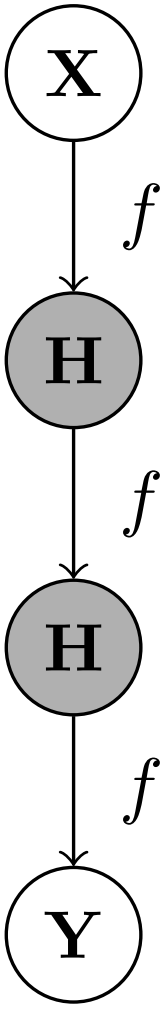
Challenges:

- 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!

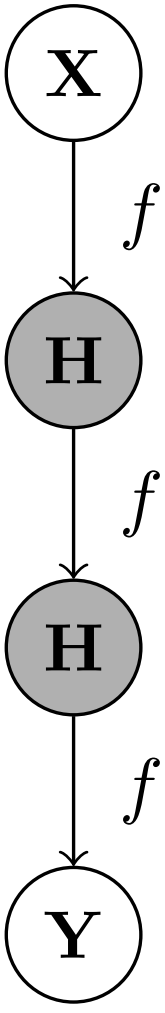
Challenges:

- 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

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

?

Outline

Introducing Gaussian Processes:

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

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

Let's start with a multivariate Gaussian:

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

with:

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{bmatrix} \quad \text{and} \quad \mathbf{K} = \begin{bmatrix} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ \mathbf{K}_{BA} & \mathbf{K}_{BB} \end{bmatrix}$$

Marginalisation property:

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

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

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

Let's start with a multivariate Gaussian:

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

with:

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{bmatrix} \quad \text{and} \quad \mathbf{K} = \begin{bmatrix} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ \mathbf{K}_{BA} & \mathbf{K}_{BB} \end{bmatrix}$$

Marginalisation property:

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

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

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

In the GP context:

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

Posterior is also 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:

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

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

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:

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

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

Posterior is also Gaussian!

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

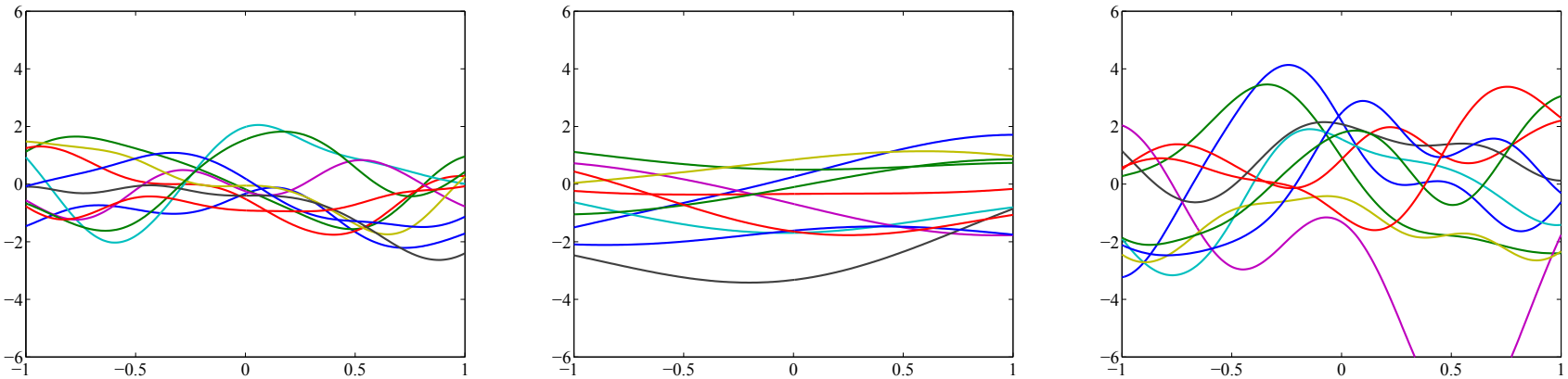
In the GP context this can be used for inter/extrapolation:

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

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

Covariance samples and hyperparameters

- $k(x, x') = \alpha \exp\left(-\frac{\gamma}{2}(x - x')^\top (x - x')\right)$
- The hyperparameters of the cov. function define the properties (and NOT an explicit form) of the sampled functions

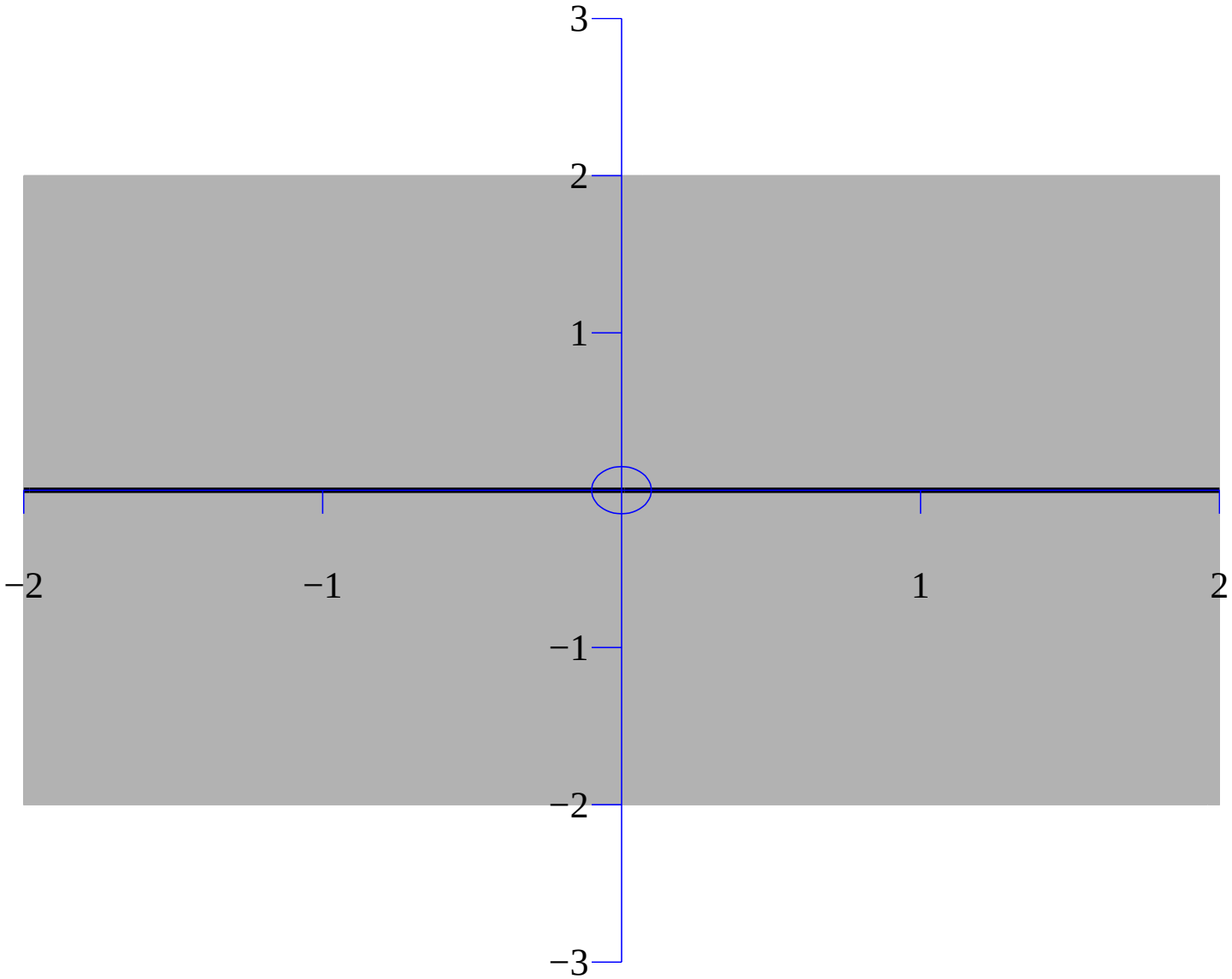


Incorporating Gaussian noise is tractable

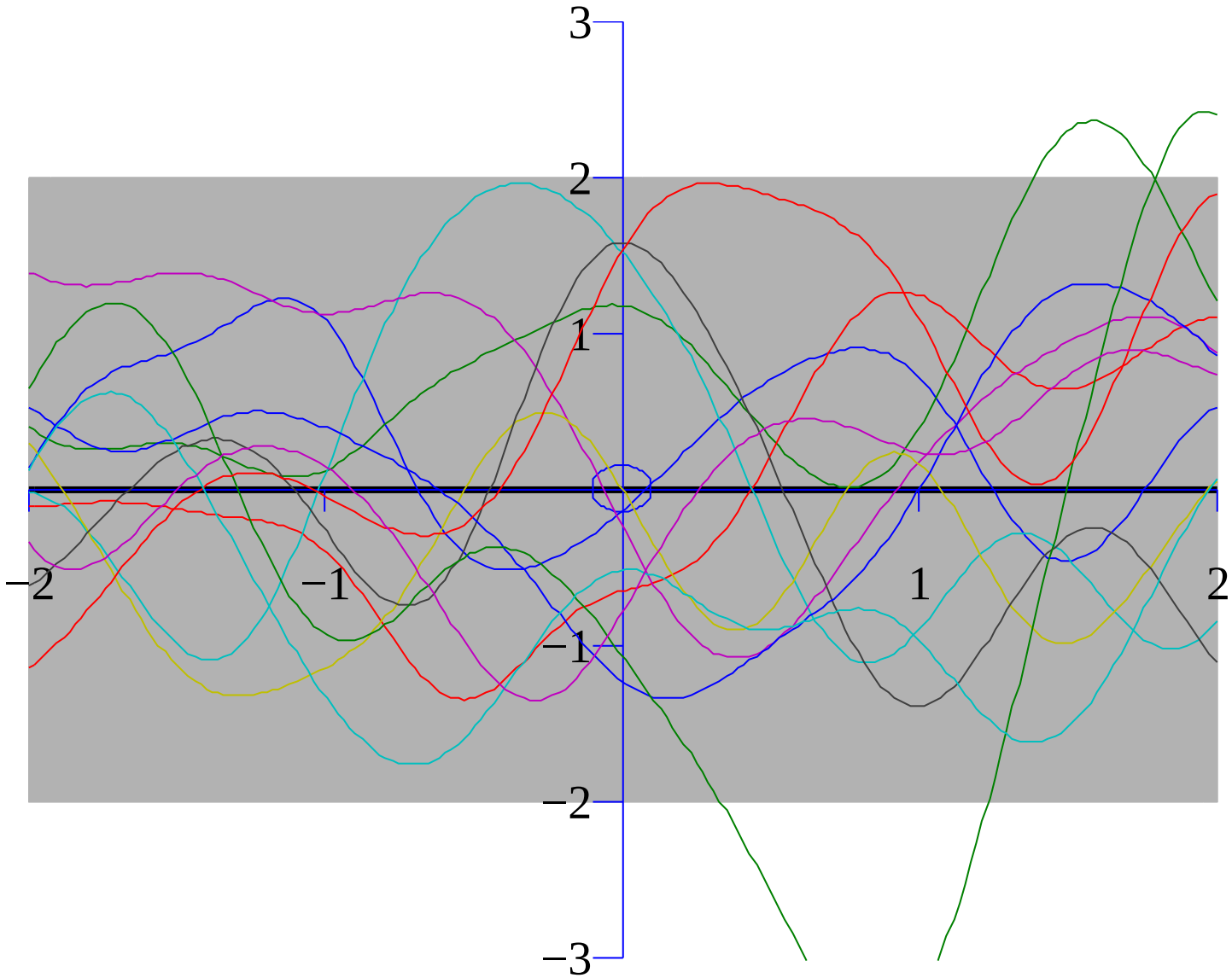
- So far we assumed: $\mathbf{f} = f(\mathbf{X})$
- Assuming that we only observe noisy versions $\mathbf{y}$ of the true outputs $\mathbf{f}$:

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

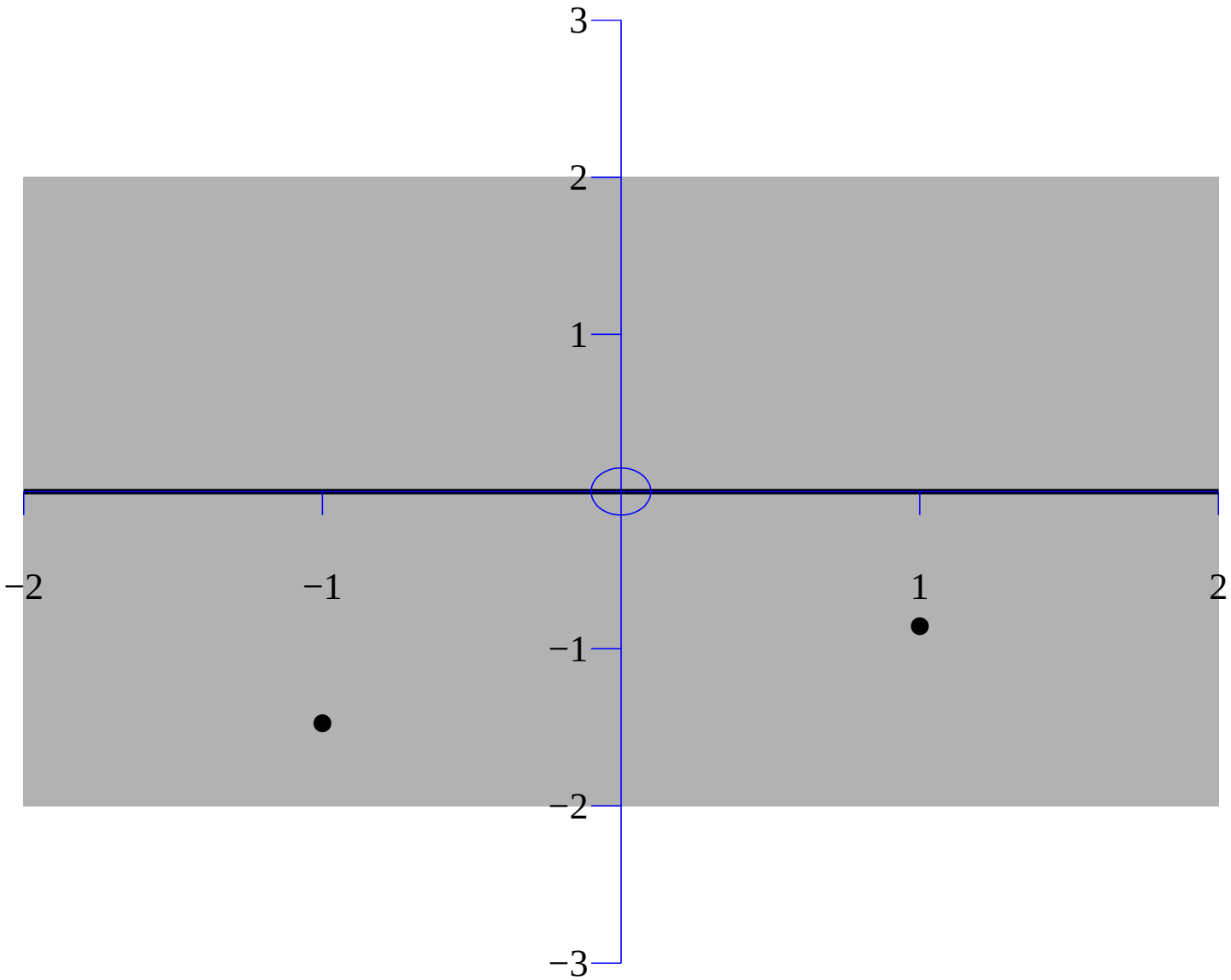
Fitting the data *(shaded area is* **uncertainty***)*



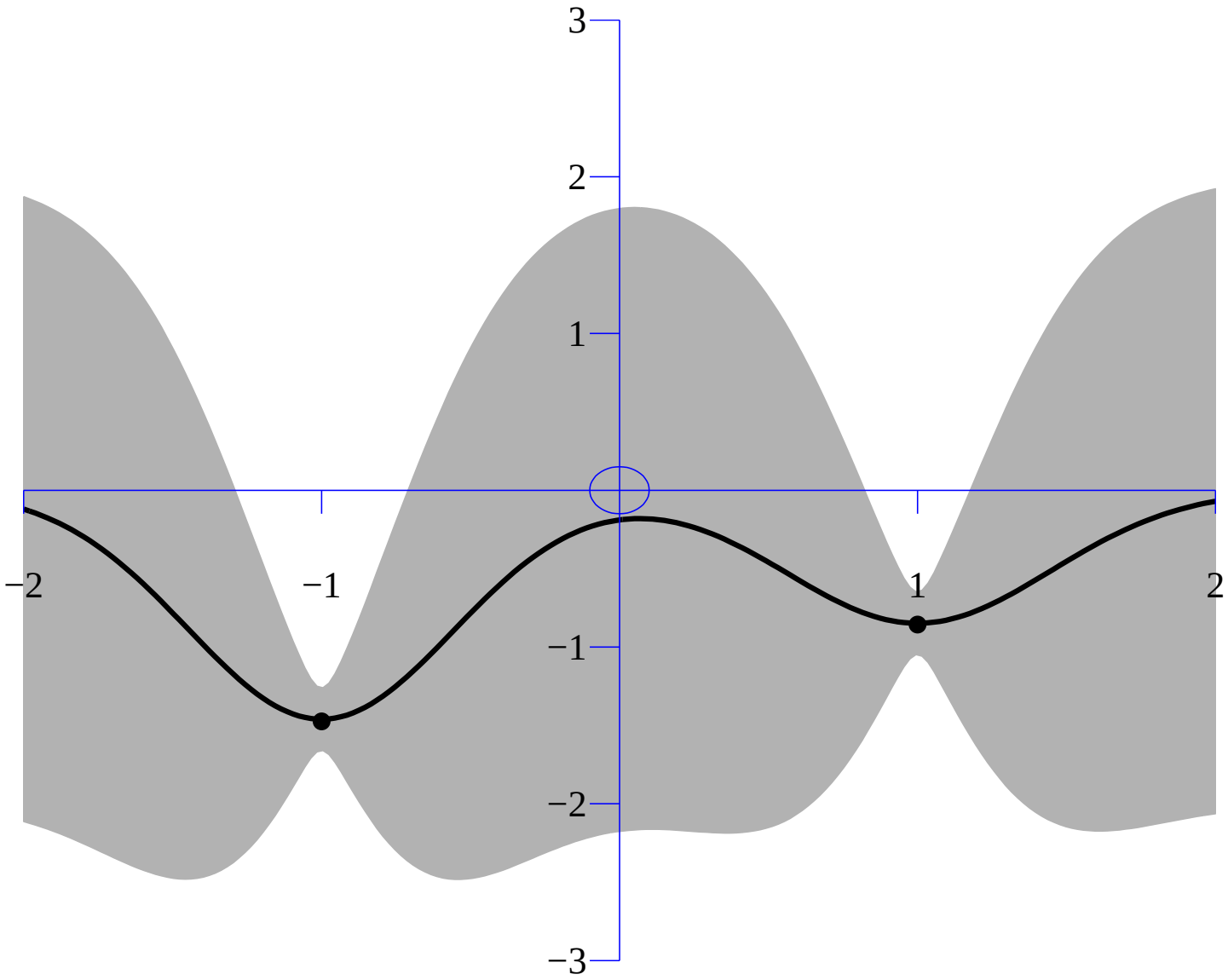
Fitting the data - Prior Samples



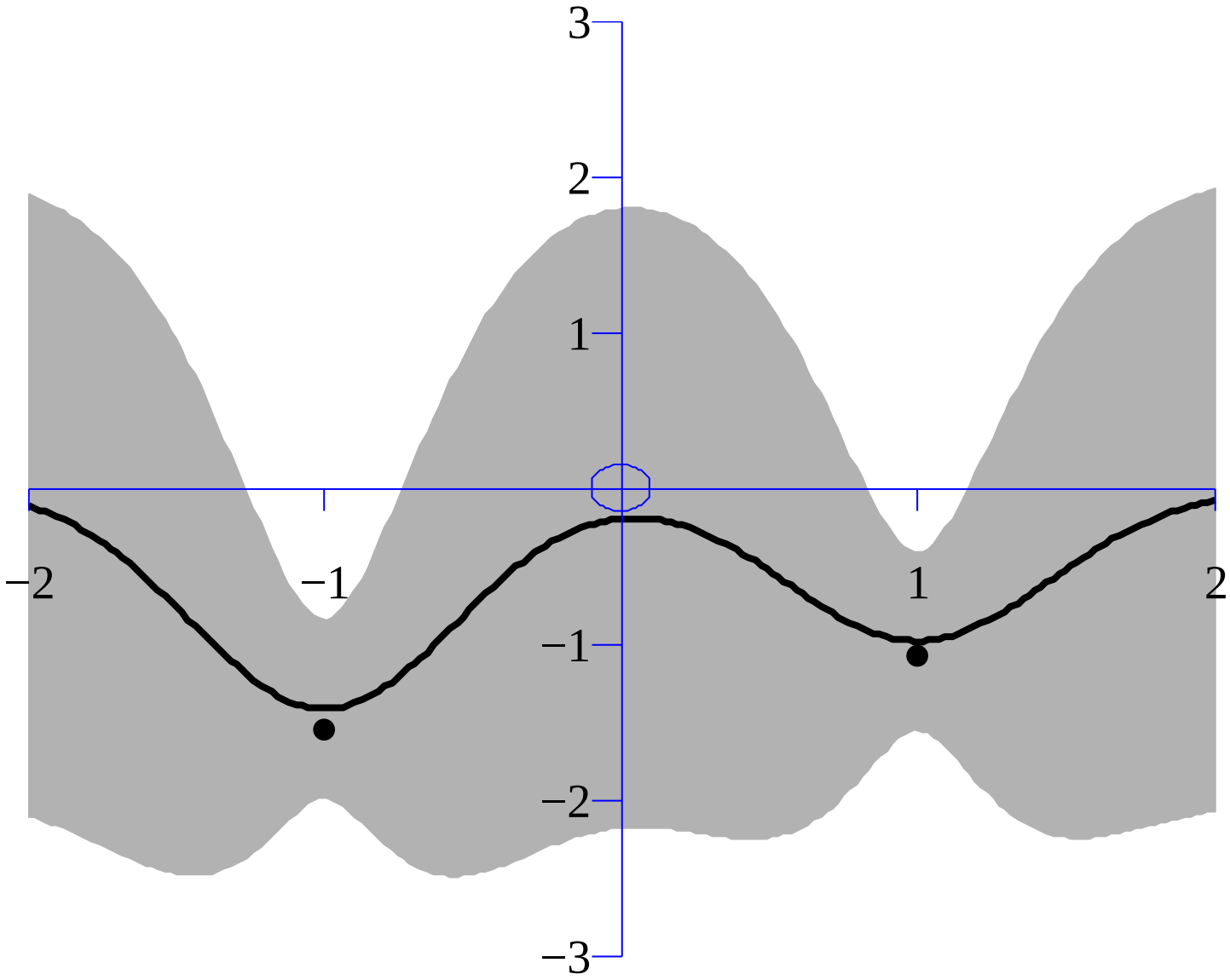
Fitting the data



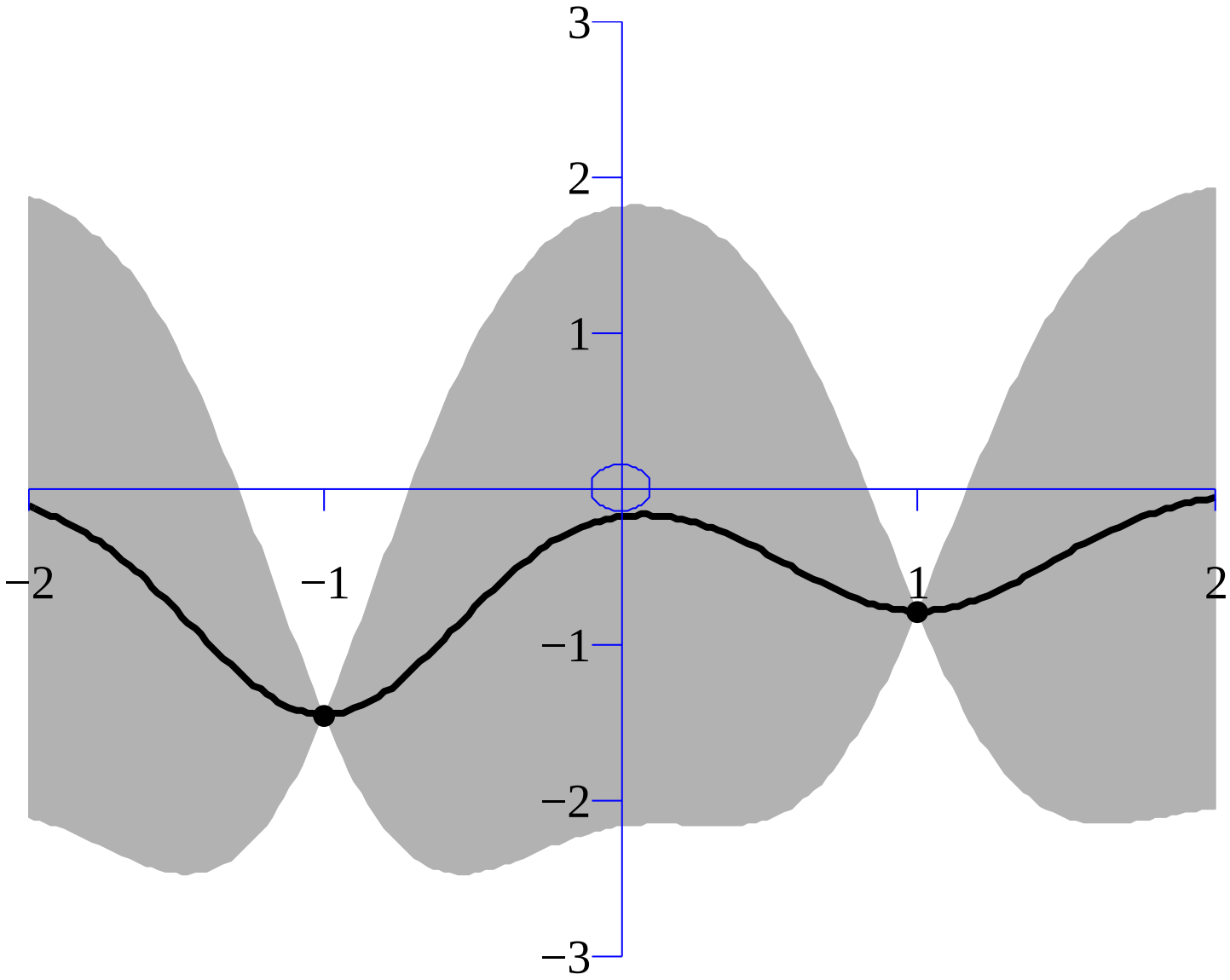
Fitting the data



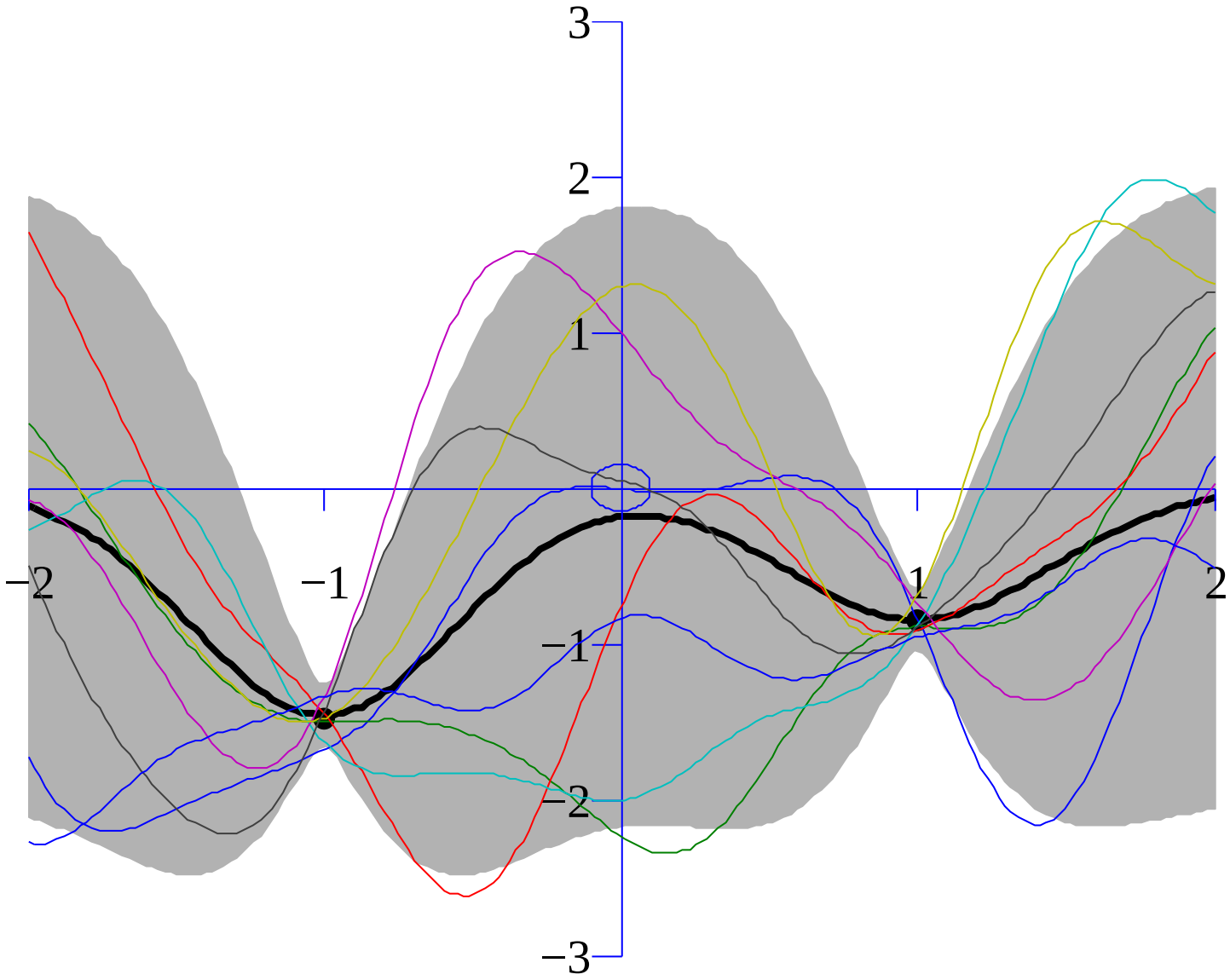
Fitting the data - more noise



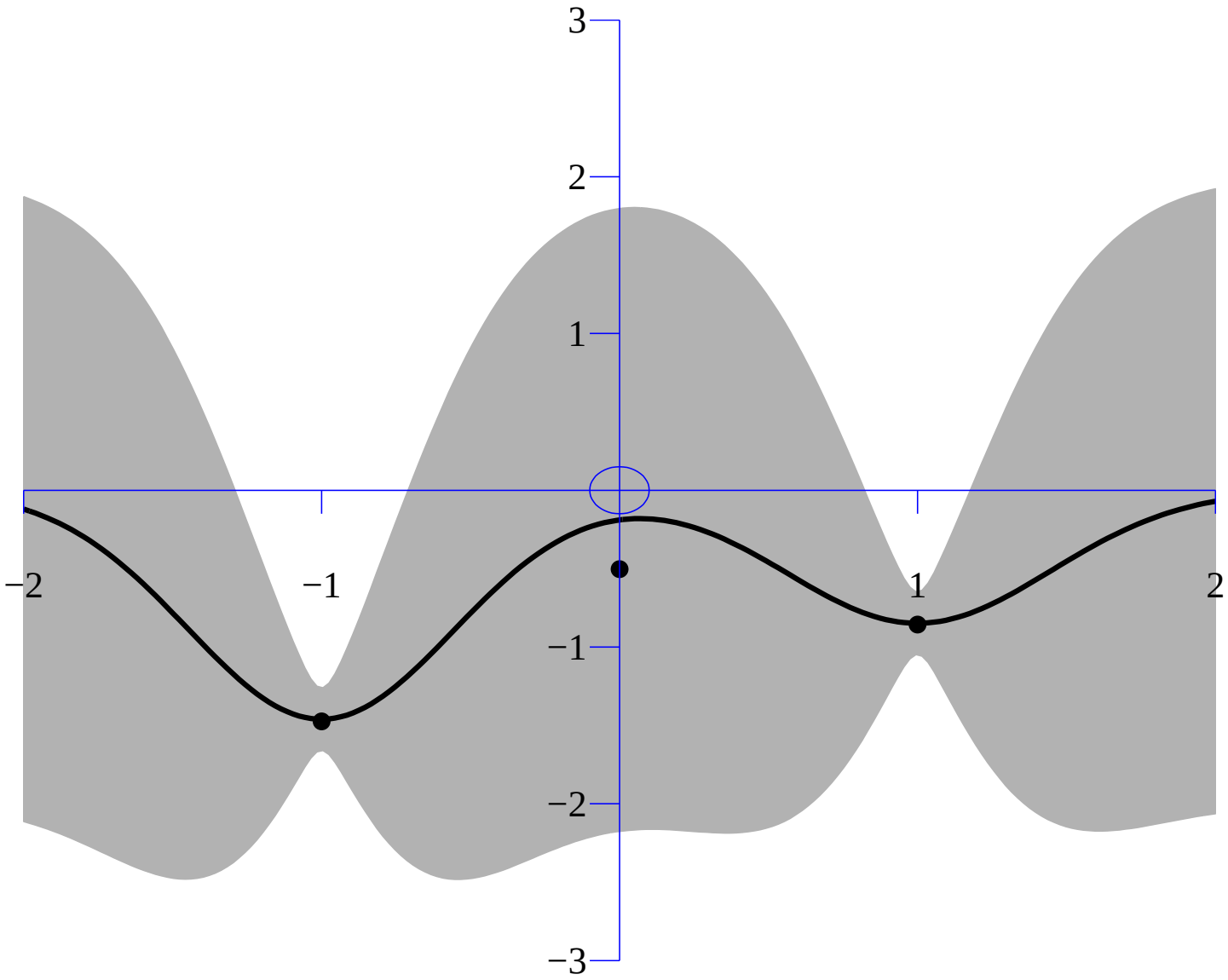
Fitting the data - no noise



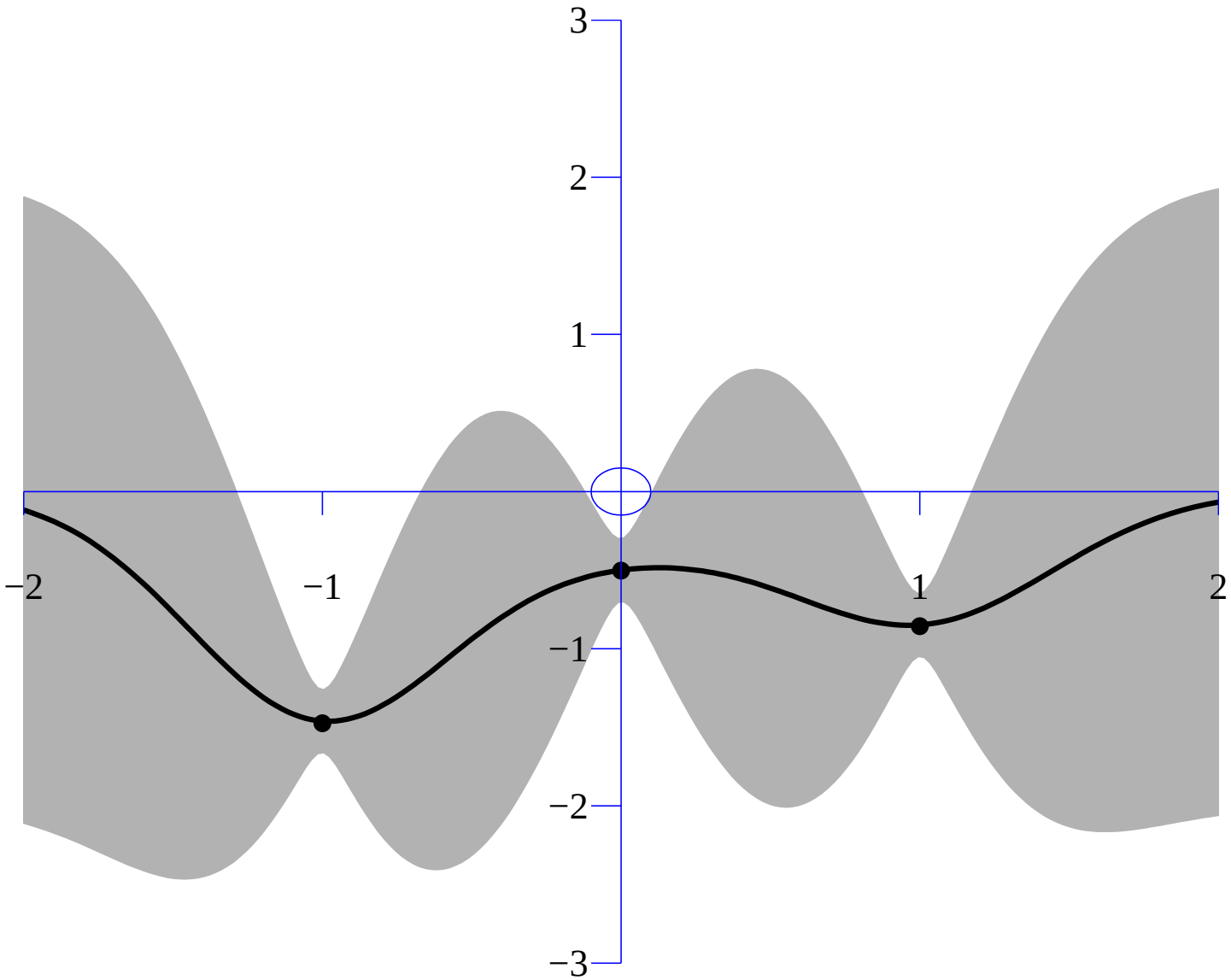
Fitting the data - Posterior samples



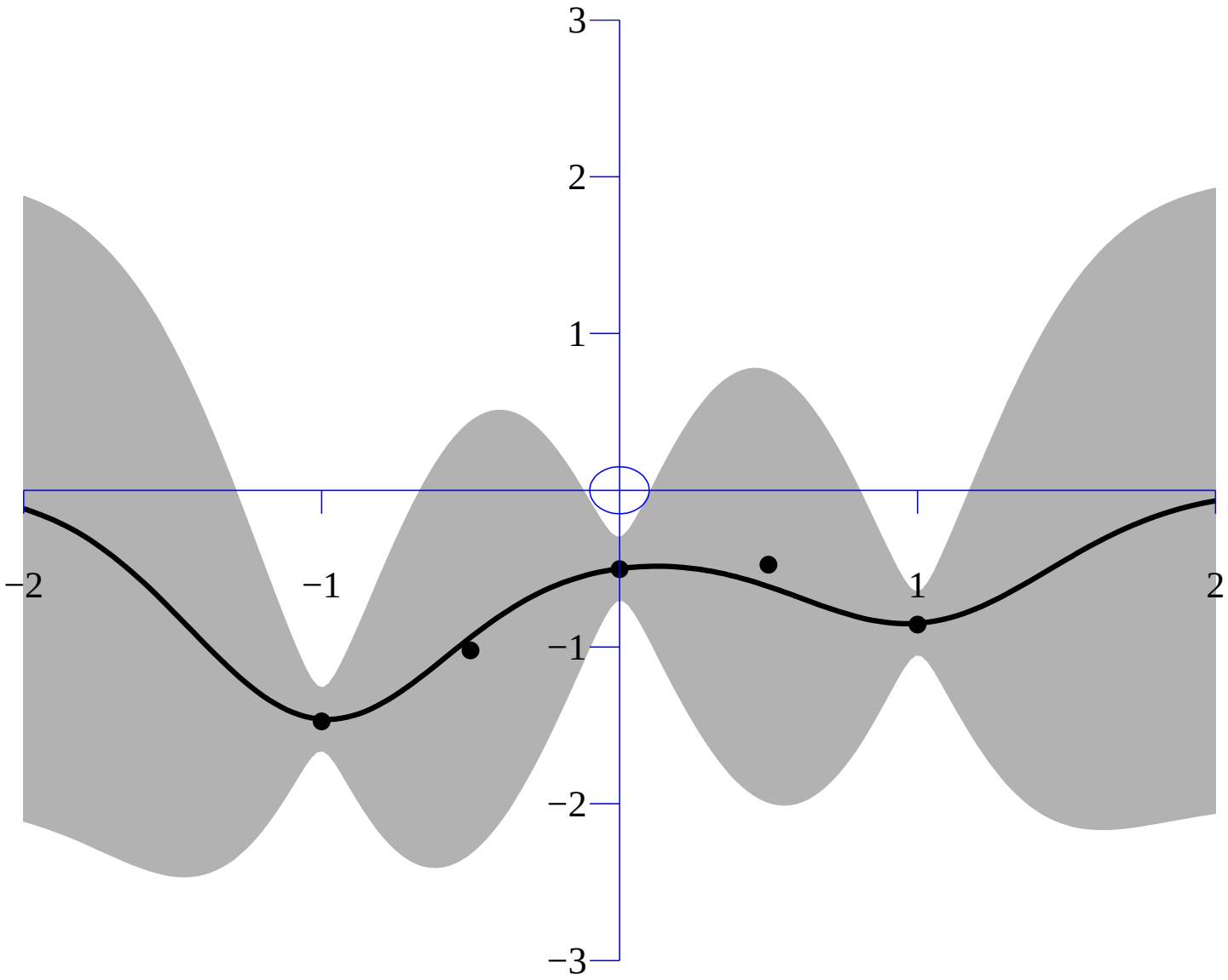
Fitting the data



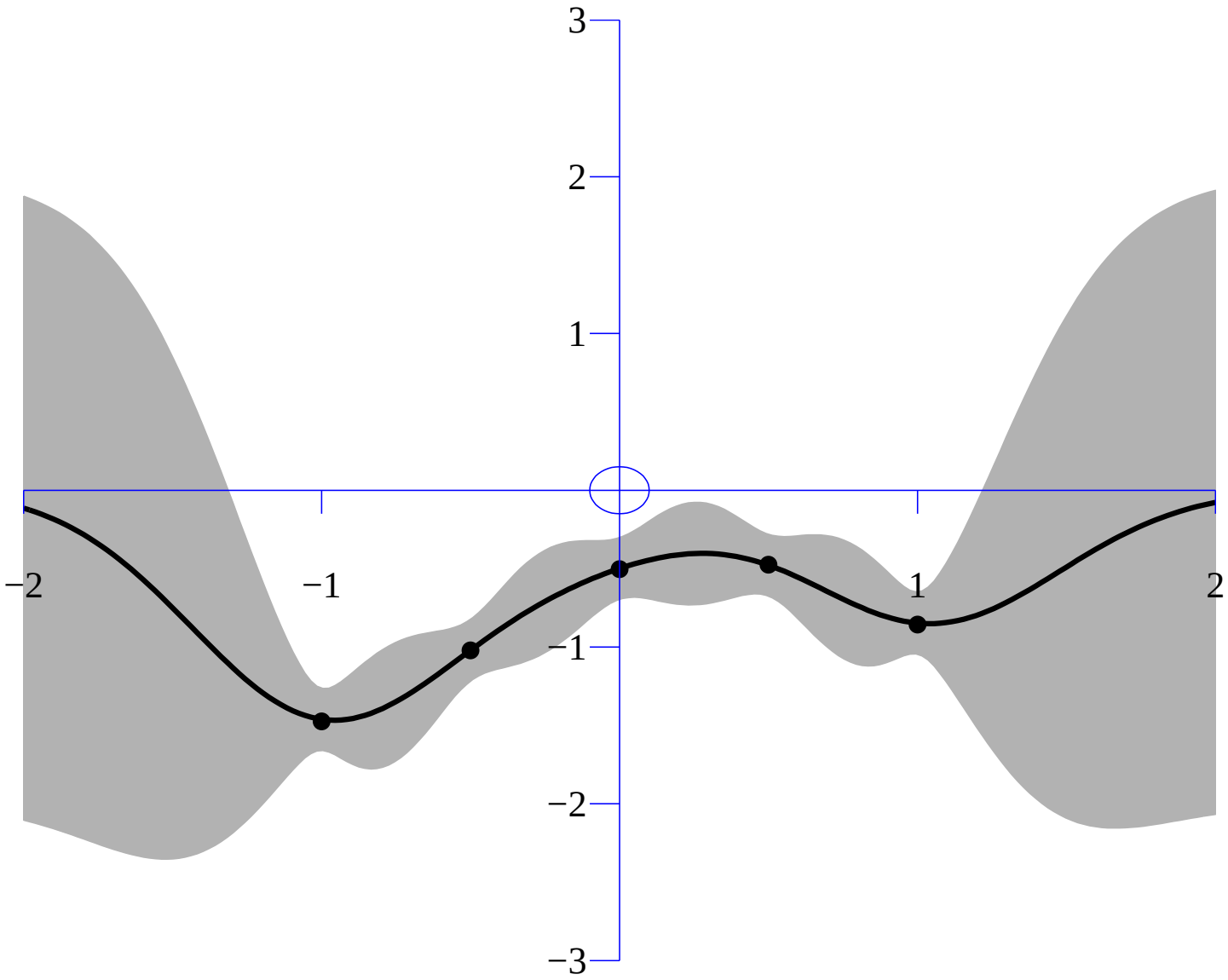
Fitting the data



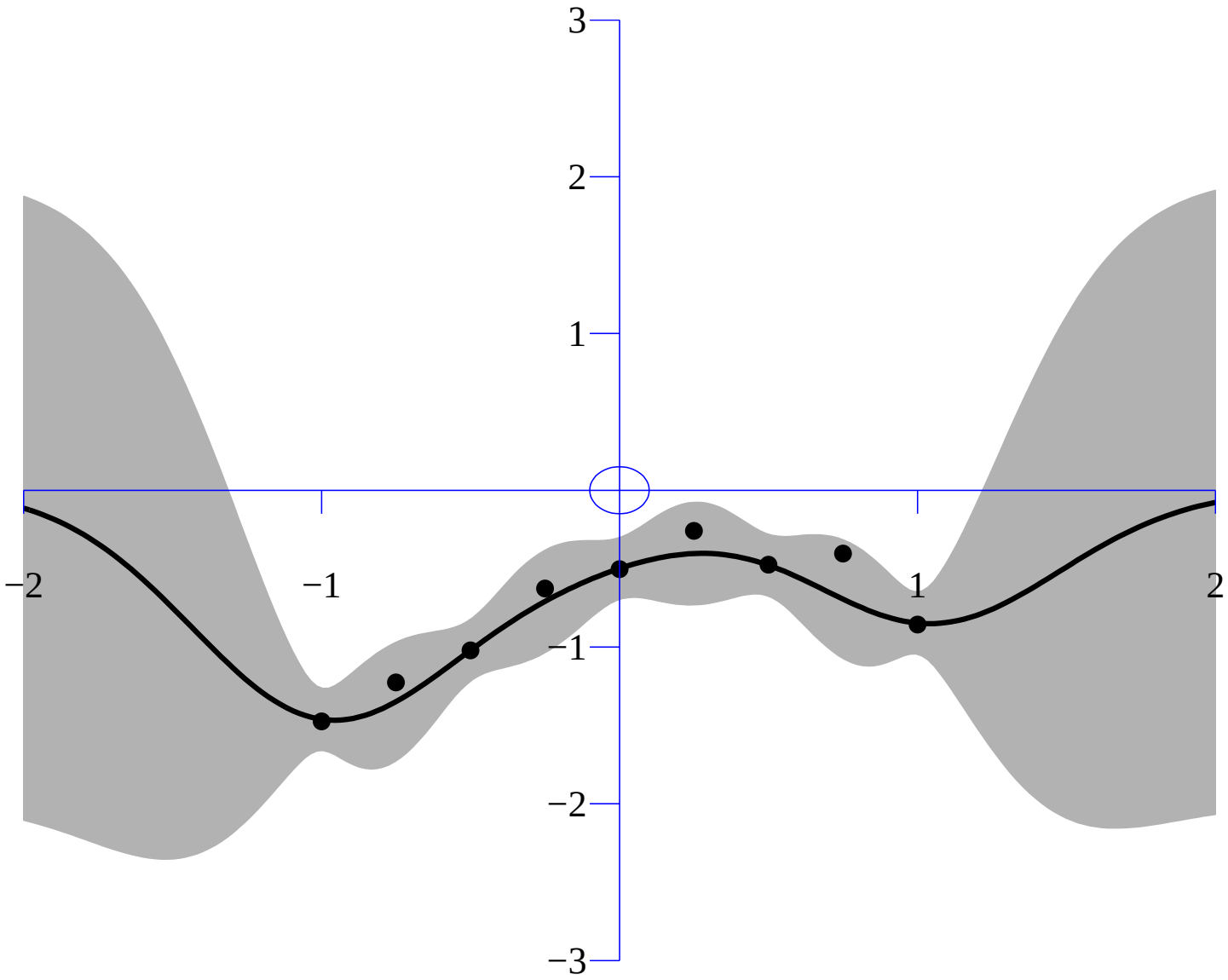
Fitting the data



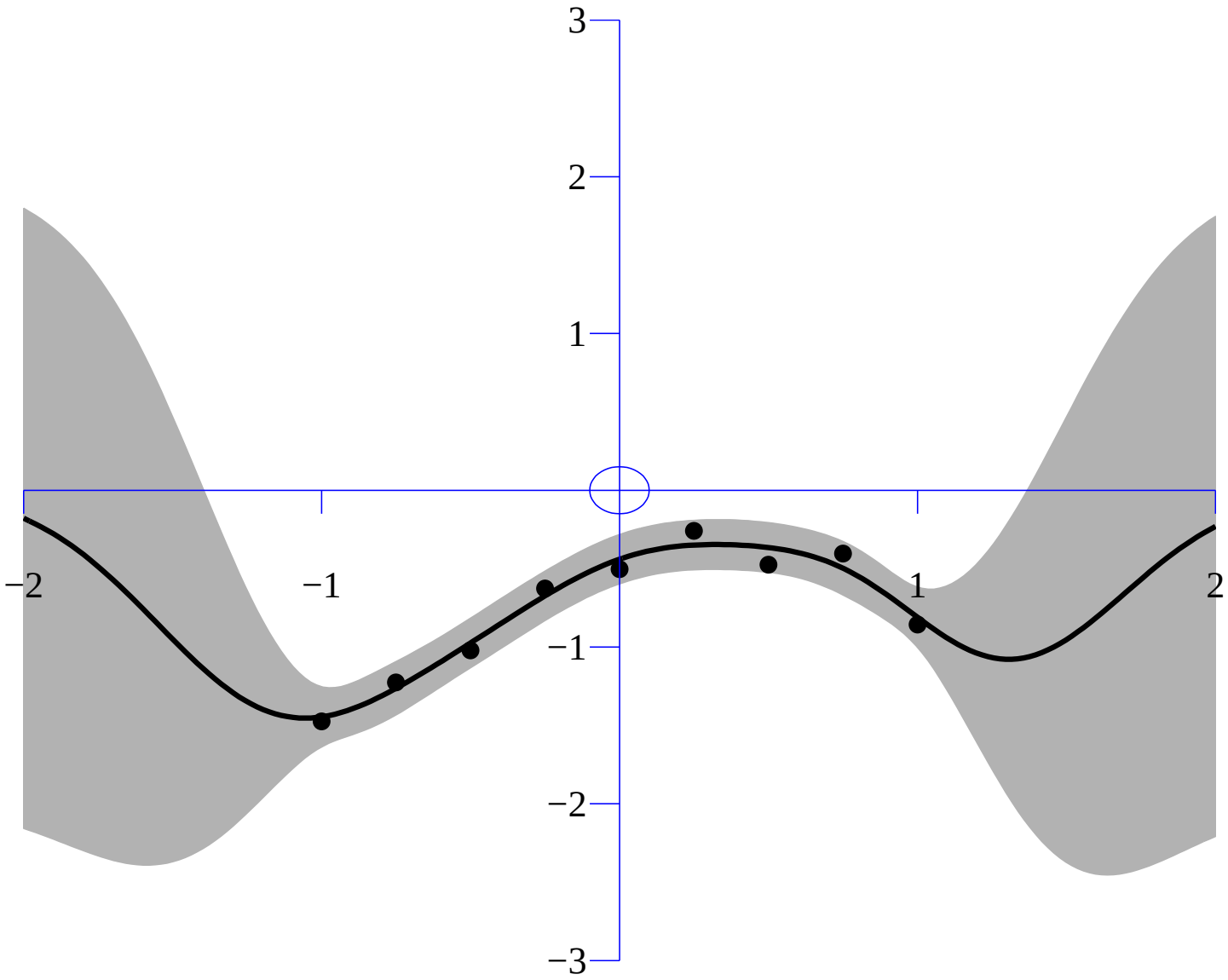
Fitting the data



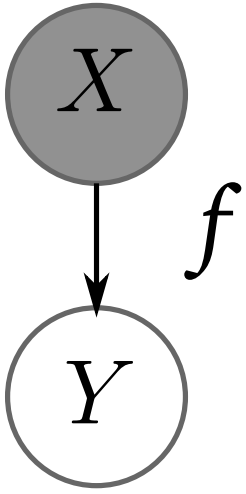
Fitting the data



Fitting the data



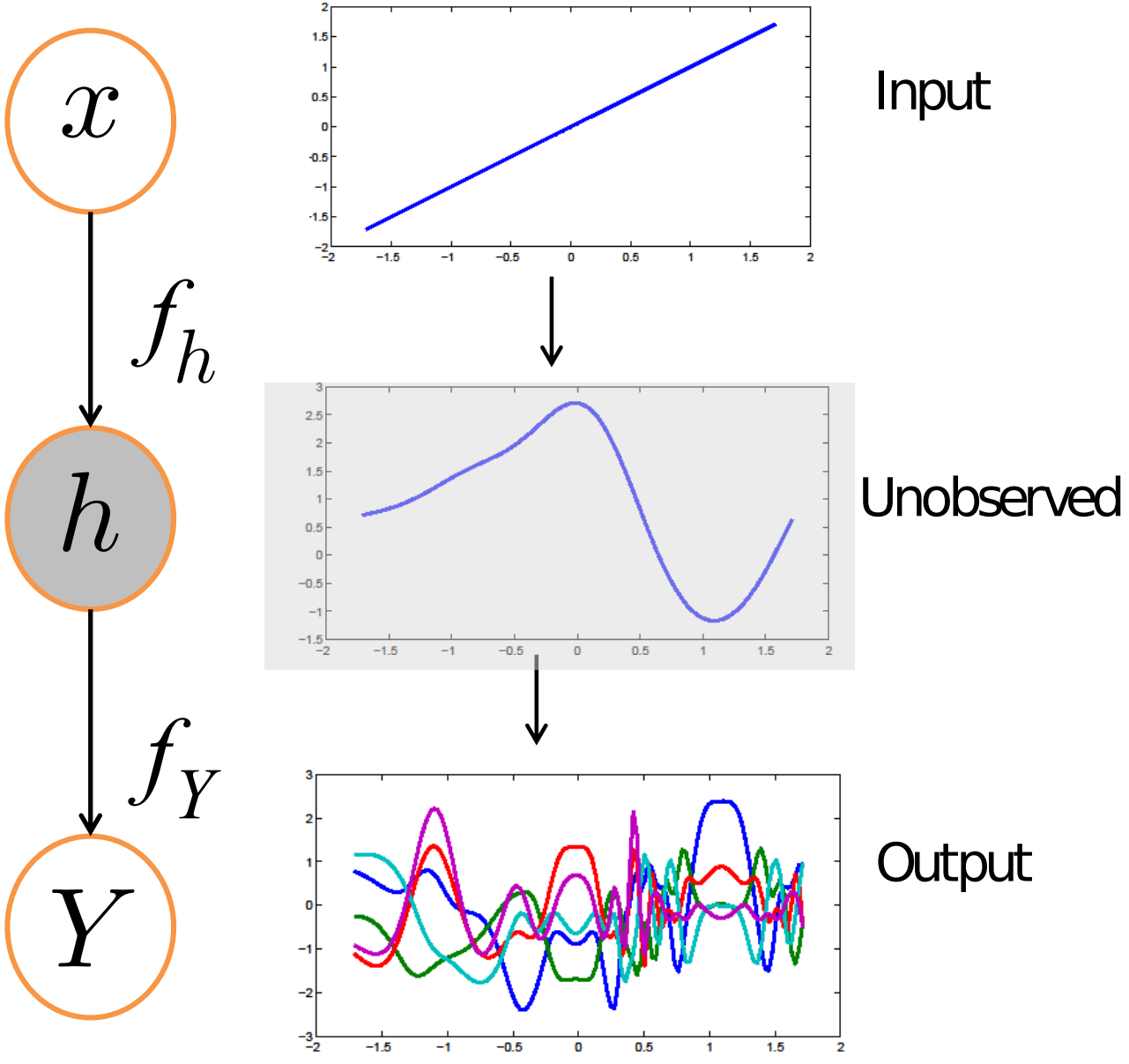
Unsupervised learning: GP-LVM



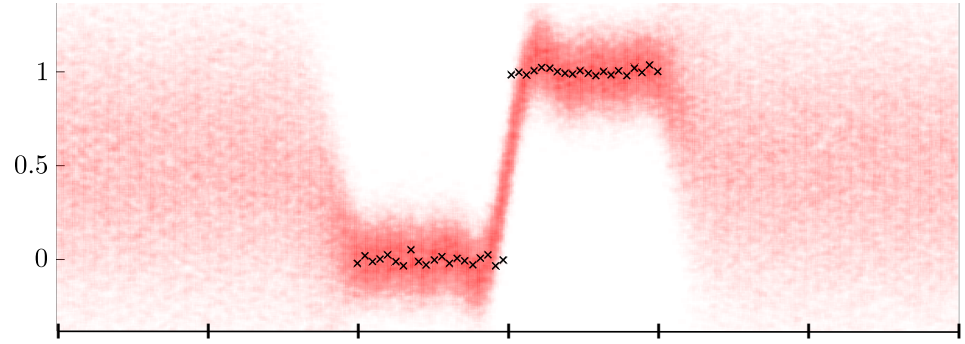
- If $\mathbf{X}$ is unobserved, treat it as a parameter and optimize over it.

Outline

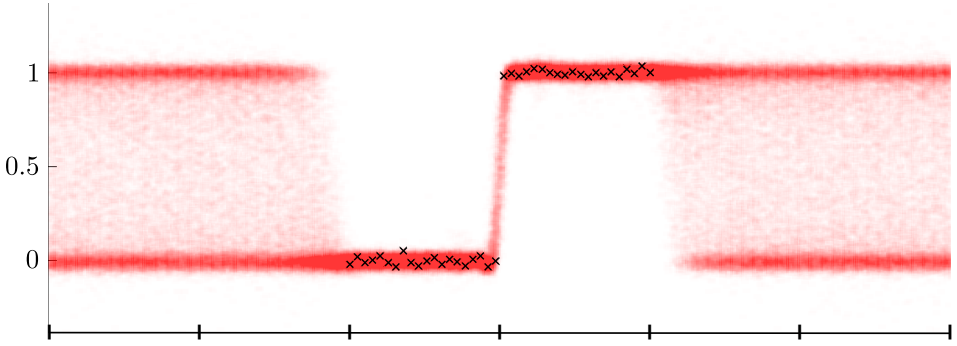
Sampling from a deep GP



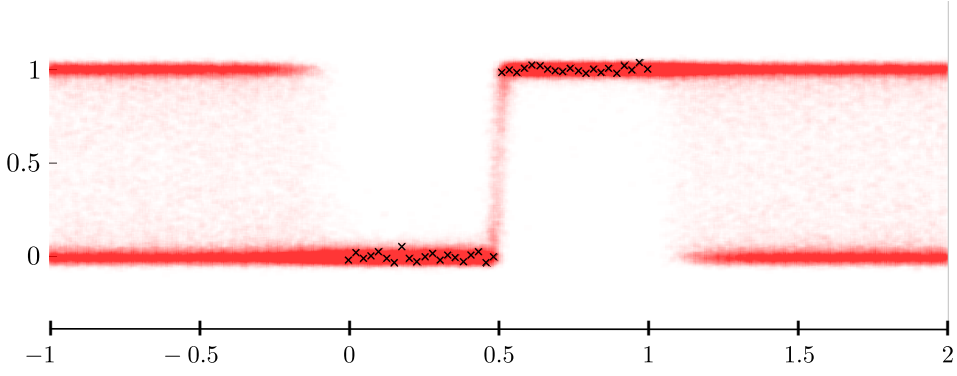
Deep GP: Step function (credits for idea to J. Hensman)



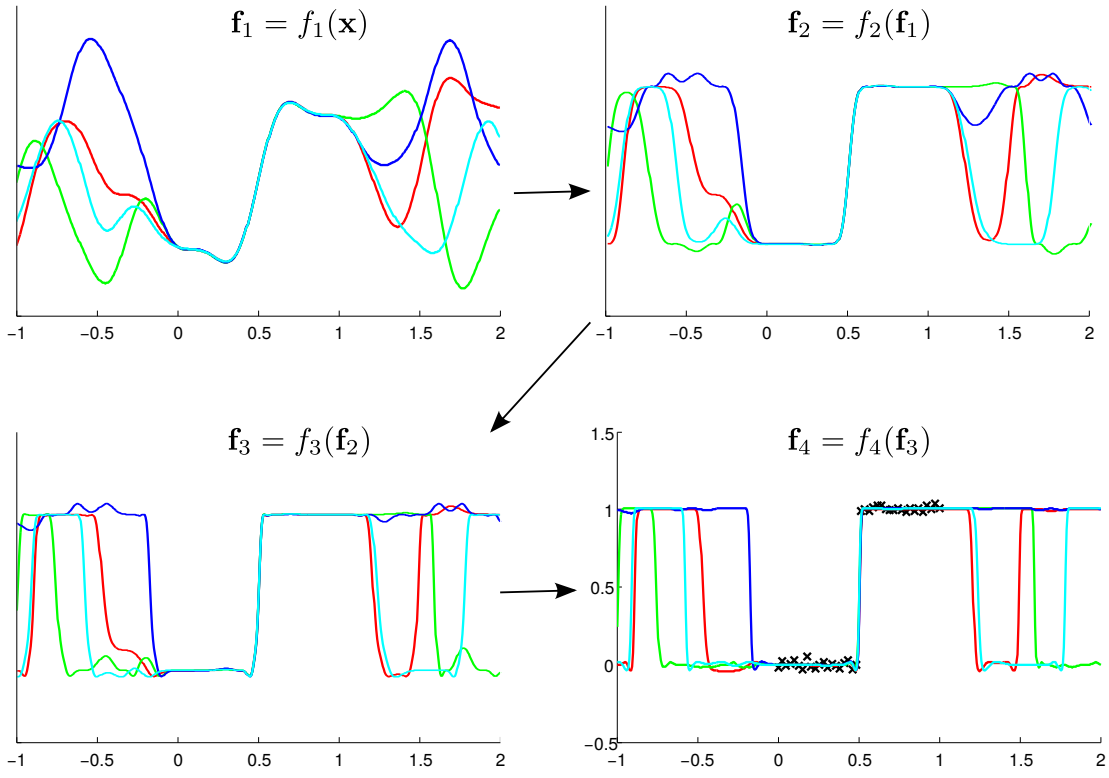
(standard GP)



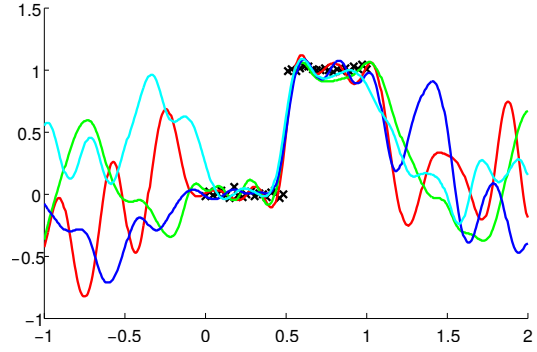
(deep GP - 1 hidden layer)



(deep GP - 3 hidden layers)

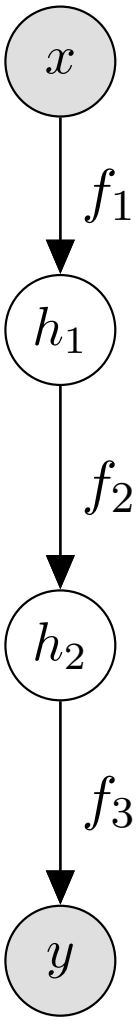


Deep GP with three hidden plus one warping layer



Standard GP

MAP optimisation?



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

Direct marginalisation of h is intractable (O_o)

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

Direct marginalisation of h is intractable (O_o)

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

Direct marginalisation of h is intractable (O_o)

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

Direct marginalisation of h is intractable (O_o)

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

Direct marginalisation of h is intractable (O_o)

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

Direct marginalisation of h is intractable (O_o)

- New objective: $p(y|x) = \int_{h_2} \left(p(y|h_2) \int_{h_1} p(h_2|h_1) p(h_1|x) \right)$
- $p(h_2|x) = \int_{h_1, f_2} p(h_2|f_1) p(f_1|h_1) \; p(h_1|x)$
- $p(h_2|x, \tilde{h}_1) = \int_{h_1, f_2, u_2} p(h_2|f_2) p(f_2|u_2, h_1) p(u_1|\tilde{h}_1) p(h_1|x)$
- $\log p(h_2|x, \tilde{h}_1) \geq \int_{h_1, f_2, u_2} \mathcal{Q} \log \frac{p(h_2|f_2) \cancel{p(f_2|u_2, h_1)} p(u_2|\tilde{h}_1) p(h_1|x)}{\mathcal{Q} = \cancel{p(f_2|u_2, h_1)} q(u_2) q(h_1)}$

Direct marginalisation of h is intractable (O_o)

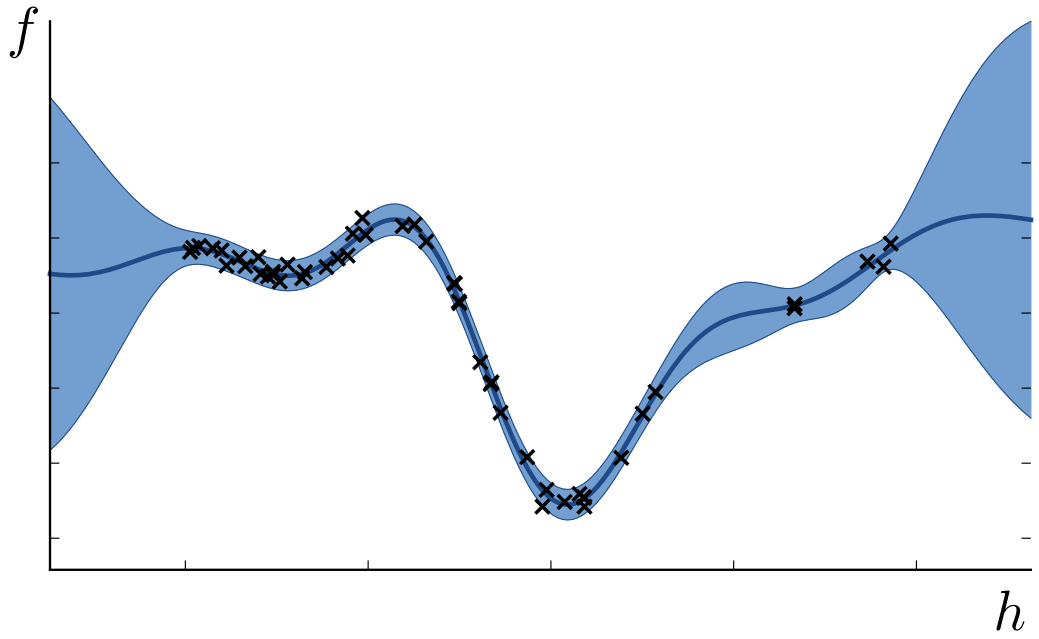
- New objective: $p(y|x) = \int_{h_2} \left(p(y|h_2) \int_{h_1} p(h_2|h_1) p(h_1|x) \right)$
- $p(h_2|x) = \int_{h_1, f_2} p(h_2|f_2) p(f_2|h_1) \; p(h_1|x)$
- $p(h_2|x, \tilde{h}_1) = \int_{h_1, f_2, u_2} p(h_2|f_2) p(f_2|u_2, h_1) p(u_2|\tilde{h}_1) p(h_1|x)$
- $\log p(h_2|x, \tilde{h}_1) \geq \int_{h_1,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

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

The above trick is applied to all layers simultaneously.

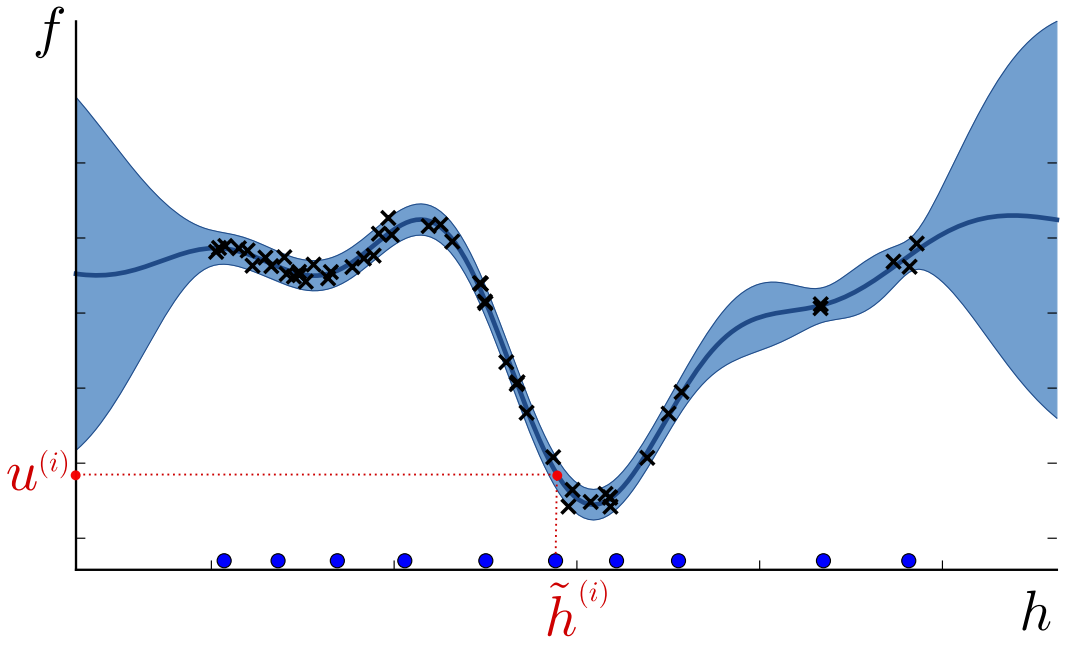
Inducing points: sparseness, tractability and Big Data

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



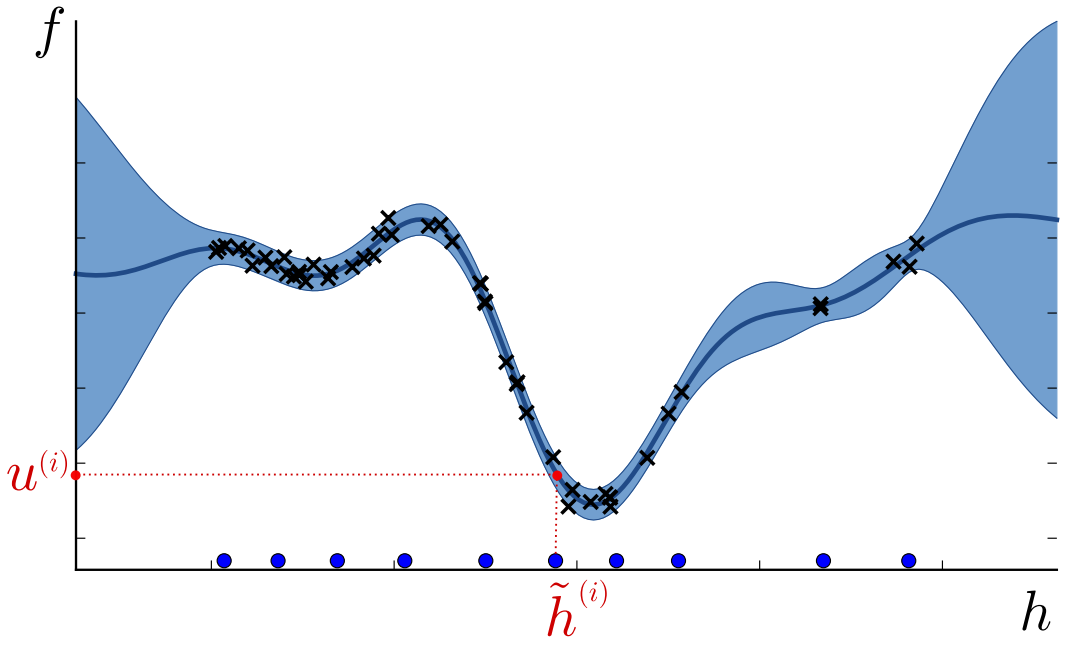
Inducing points: sparseness, tractability and Big Data

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



Inducing points: 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**
- But this also induces **tractability** in our models, due to the conditional independencies assumed
- Viewing them as **global variables**
  $\Rightarrow$ extension to **Big Data** [Hensman et al., UAI 2013]

Factorised vs non-factorised bound

- Preliminary bound

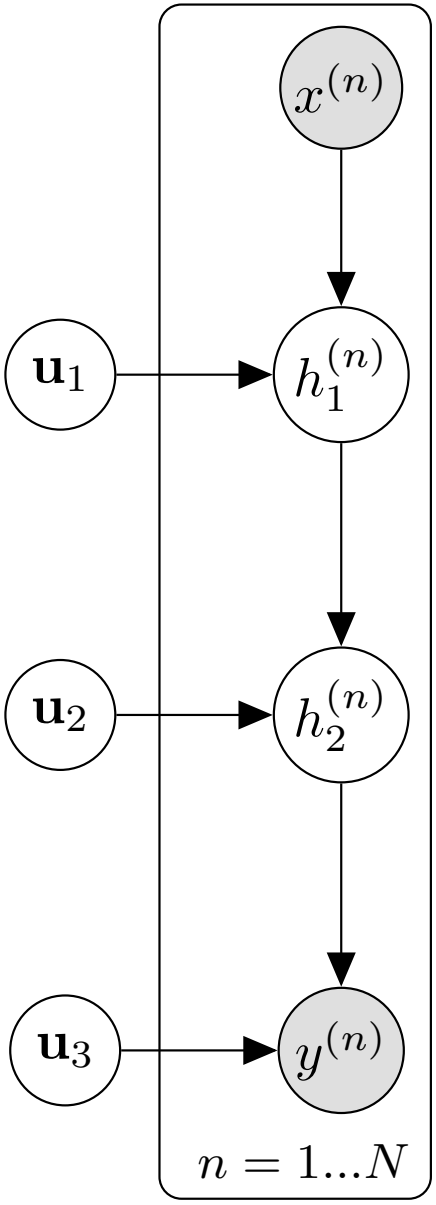
$$\mathcal{L} \leq \log p(\mathbf{Y}, \{\mathbf{H}_l\}_{l=1}^{L} | \{\mathbf{U}_l\}_{l=1}^{L+1}, \mathbf{X})$$

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

$$= \sum_{n=1}^{N} \sum_{l=1}^{L} \sum_{q=1}^{Q_l} \mathcal{L}_l^{n,q}$$

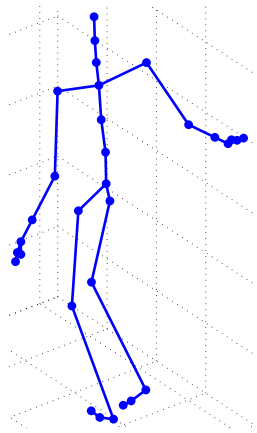
- Fully factorised.

SVI for factorised deep GPs

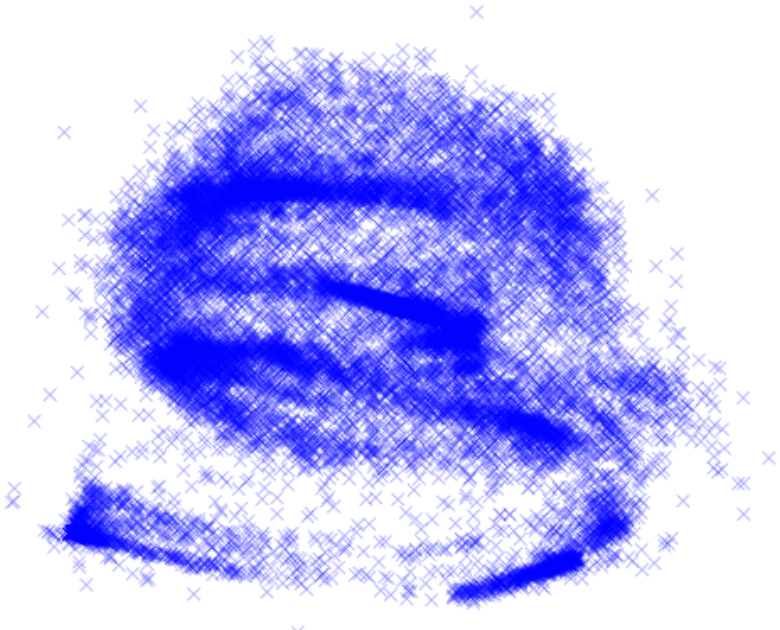


- We can additionally marginalise out $\mathbf{h}$ and maintain factorisation.
- We can consider SVI.
- Unlike $\boldsymbol{\theta}_u$ and $\boldsymbol{\theta}$, $\mathbf{h}$ are *not* global variables.
- So, estimate $\mathbf{h}^{(batch)}$ given the current $\boldsymbol{\theta}_t$
- Adjusting the step-length for SVI is tricky.

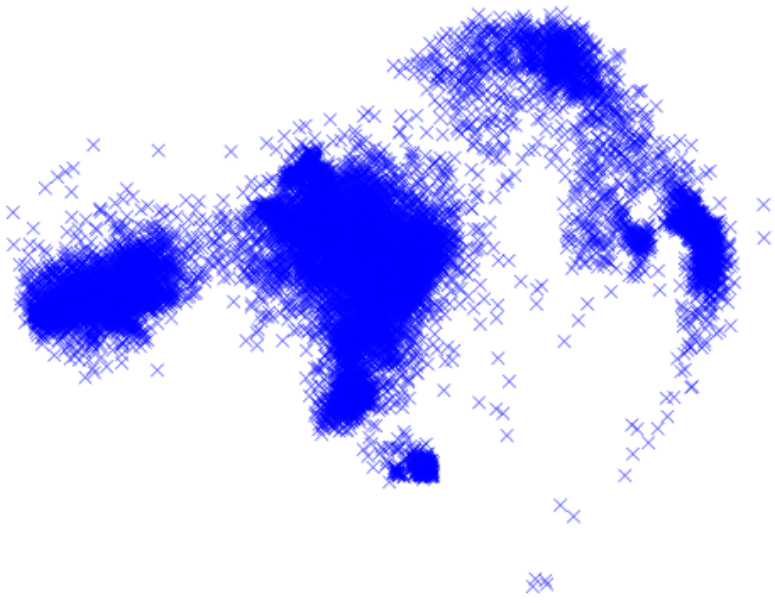
SVI - 18K mocap examples



Hidden space projections:

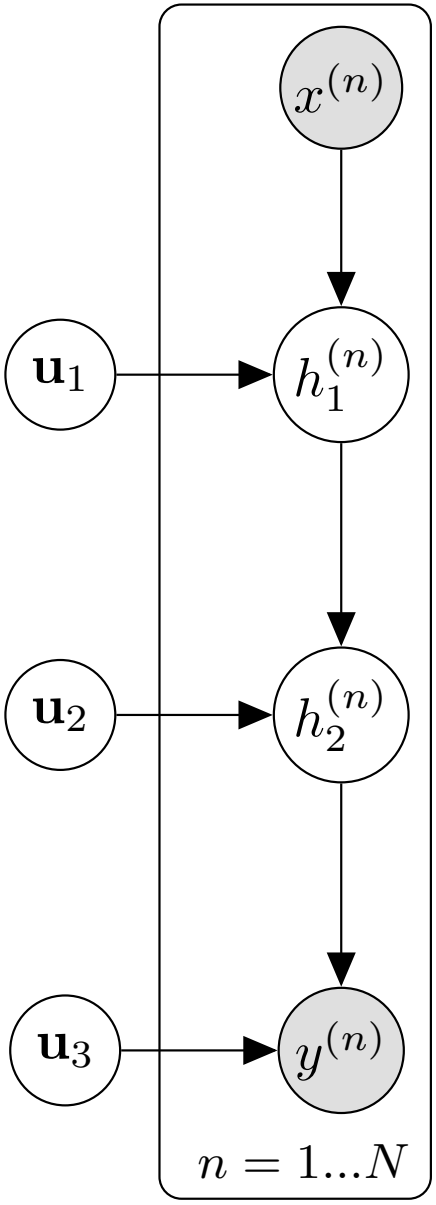


Global motion features

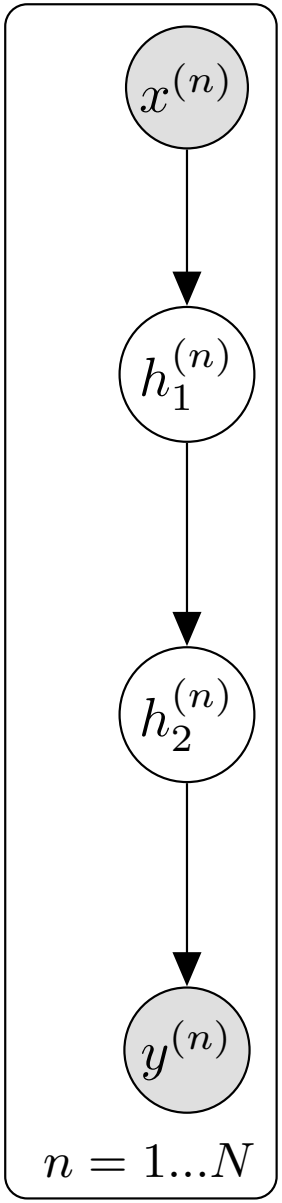


Clustered motion features

Integrate out $\mathbf{u}$



Integrate out $\mathbf{u}$



- Integrating out $\mathbf{u} \rightarrow$ factorisation is maintained.
- "Effect" of $\mathbf{u}$ manifested through $q(\mathbf{u})$

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

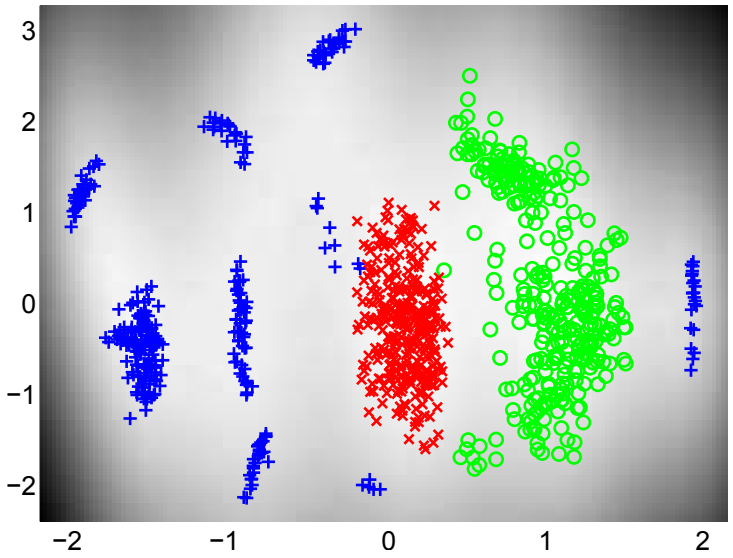
- Collapsing $\mathbf{u}$'s distribution eliminates many variational parameters
- But this introduces coupling and breaks the factorisation
- But we can still distribute the computations efficiently (work by Y. Gal, Z. Dai)

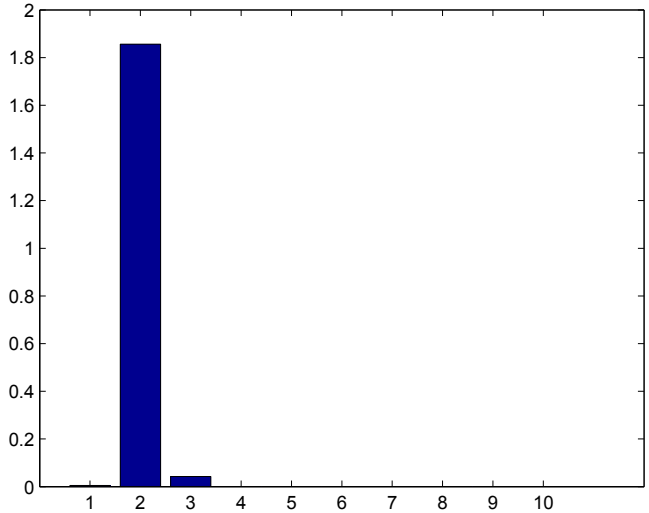
Automatic dimensionality detection

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

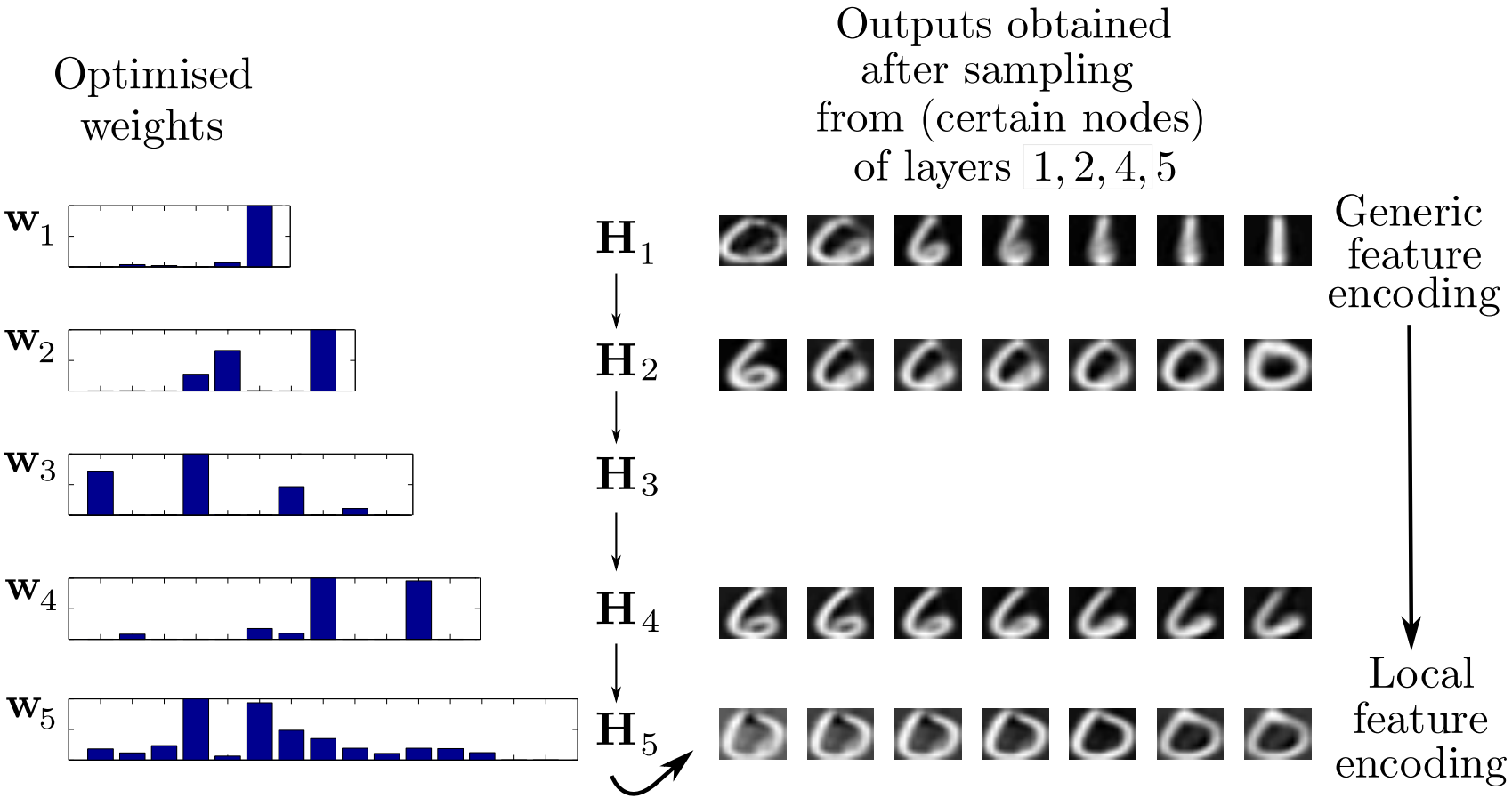
$$k_f\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \sigma^2 \exp\left(-\frac{1}{2}\sum_{q=1}^{Q} w^{(q)}\left(x^{(i,q)} - x^{(j,q)}\right)^2\right)$$

- Example:

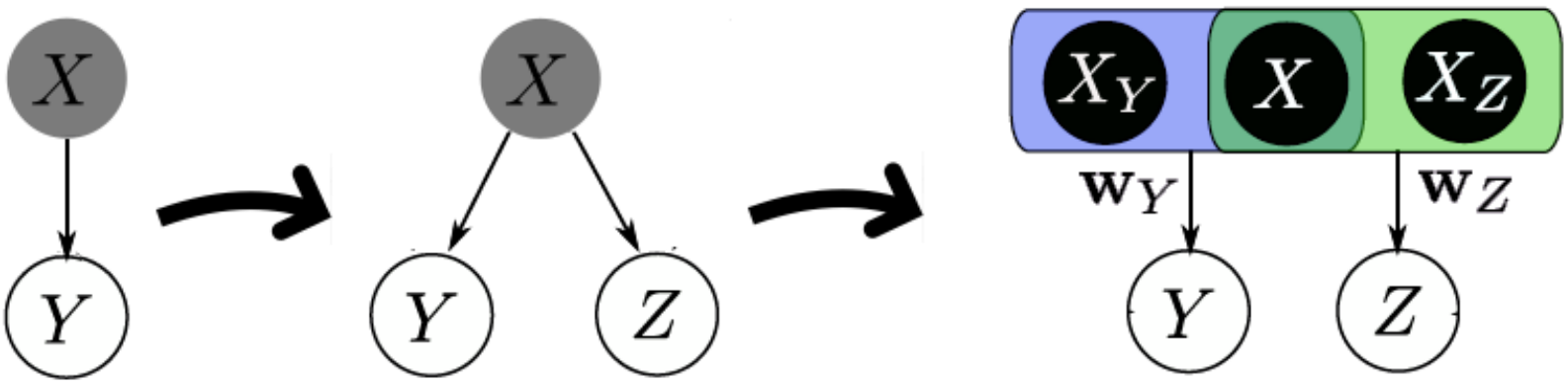




Deep GP: MNIST example



Manifold Relevance Determination

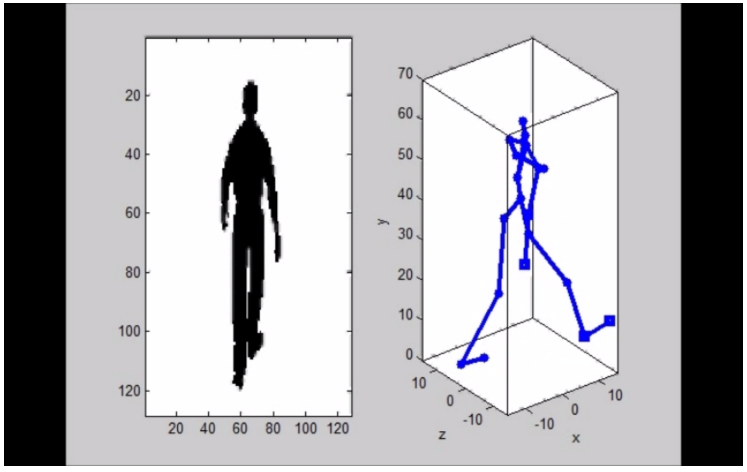


- Observations come into two different *views*: Y and Z.
- The latent space is segmented into parts private to Y, private to Z and shared between Y and Z.
- Used for data consolidation and discovering commonalities.

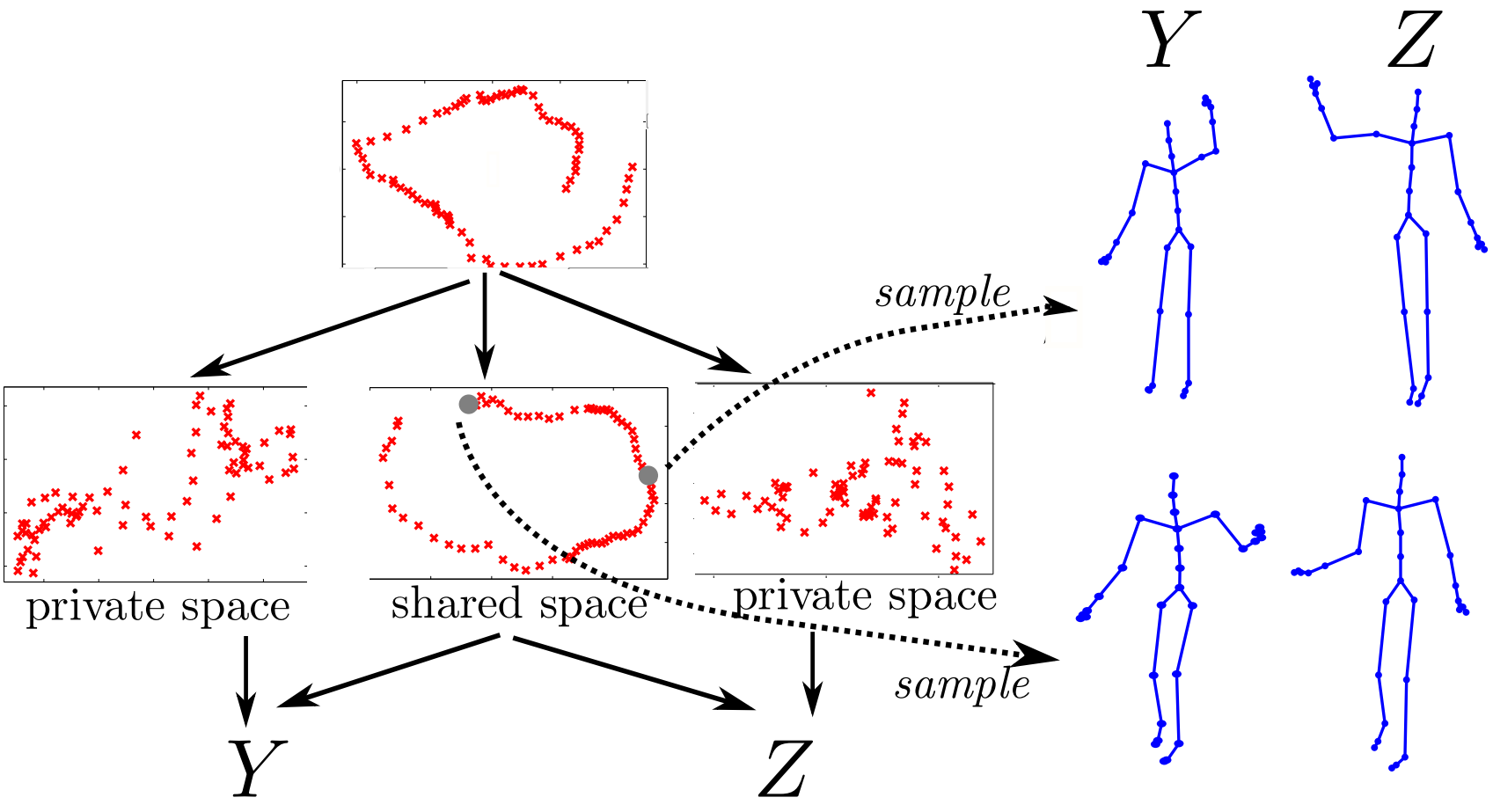
MRD examples

Yale faces

Motion capture / silhouette



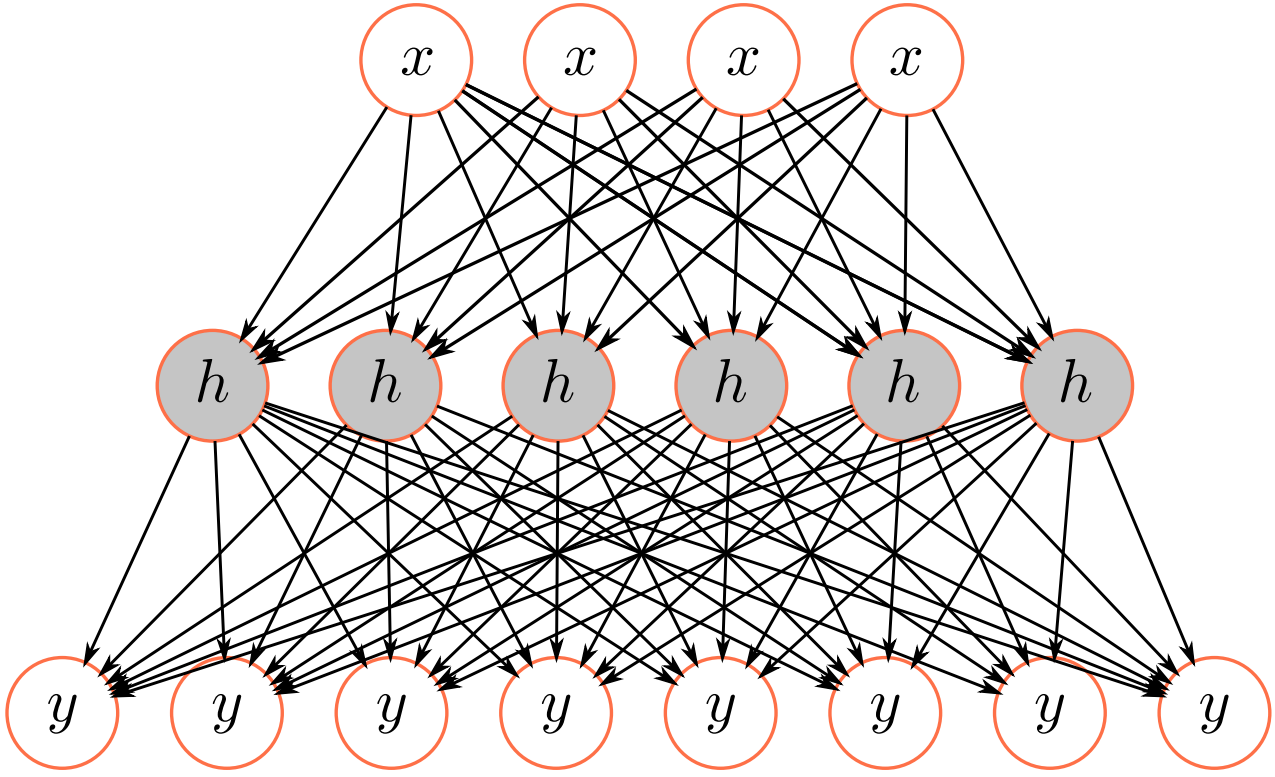
Deep GPs: Another multi-view example



Automatic structure discovery

Tools:

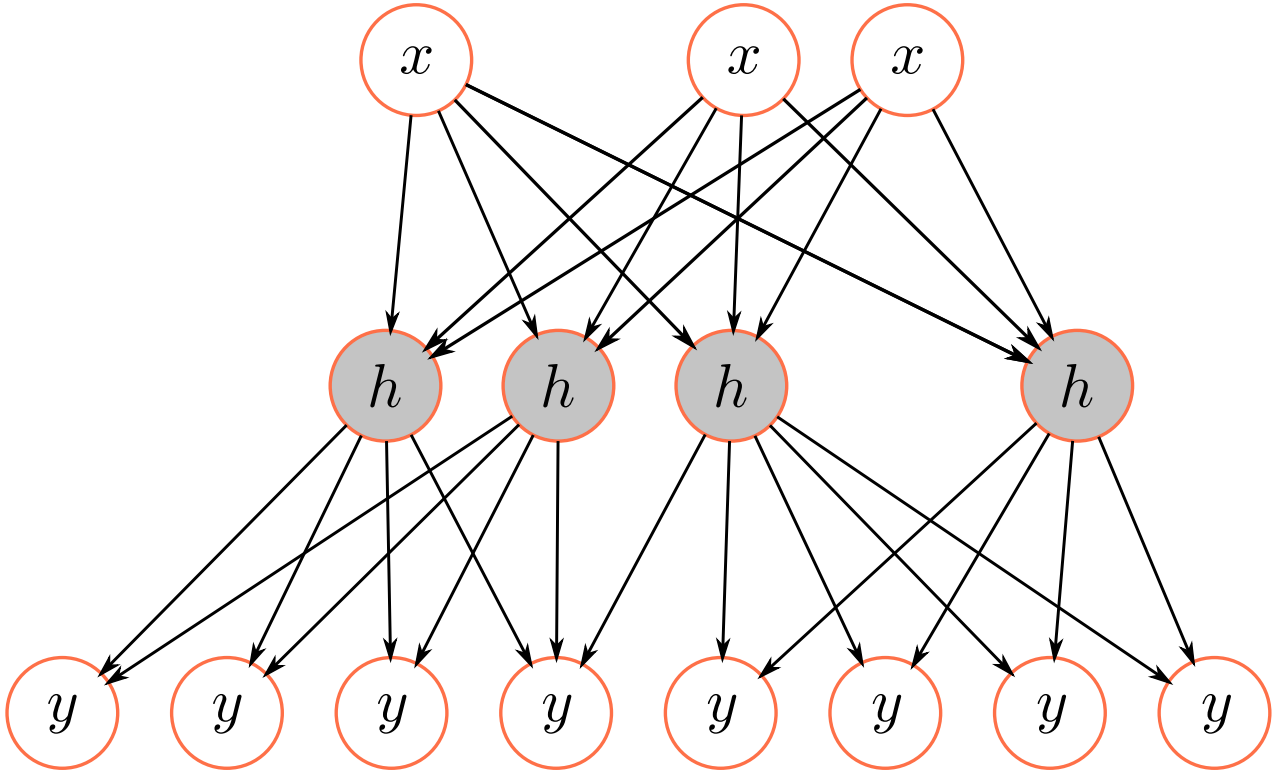
- ARD: Eliminate uncessary nodes/connections
- MRD: Conditional independencies
- Approximating evidence: Number of layers (?)



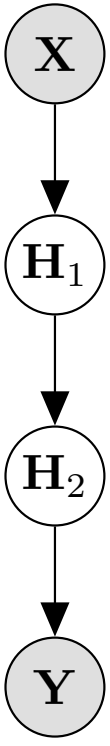
Automatic structure discovery

Tools:

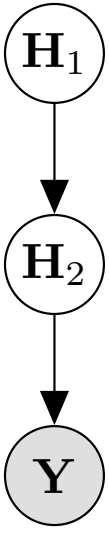
- ARD: Eliminate uncessary nodes/connections
- MRD: Conditional independencies
- Approximating evidence: Number of layers (?)



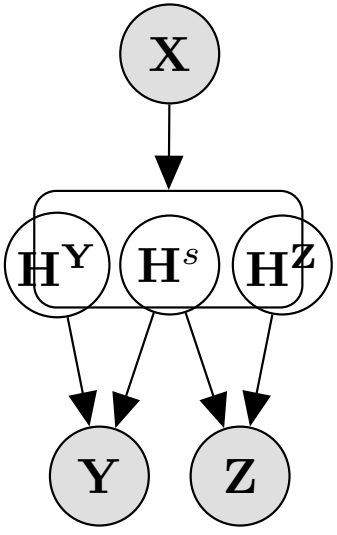
Deep GP variants



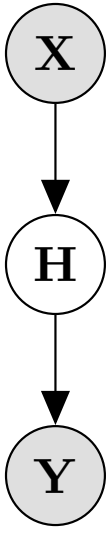
Deep GP - Supervised



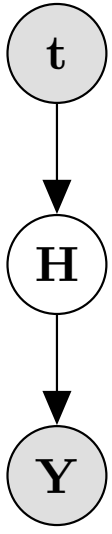
Deep GP - Unsupervised



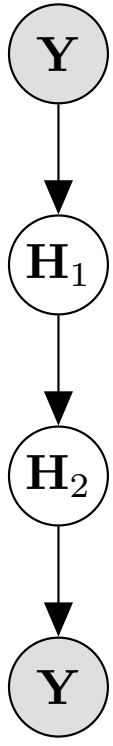
Multi-view



Warped GP



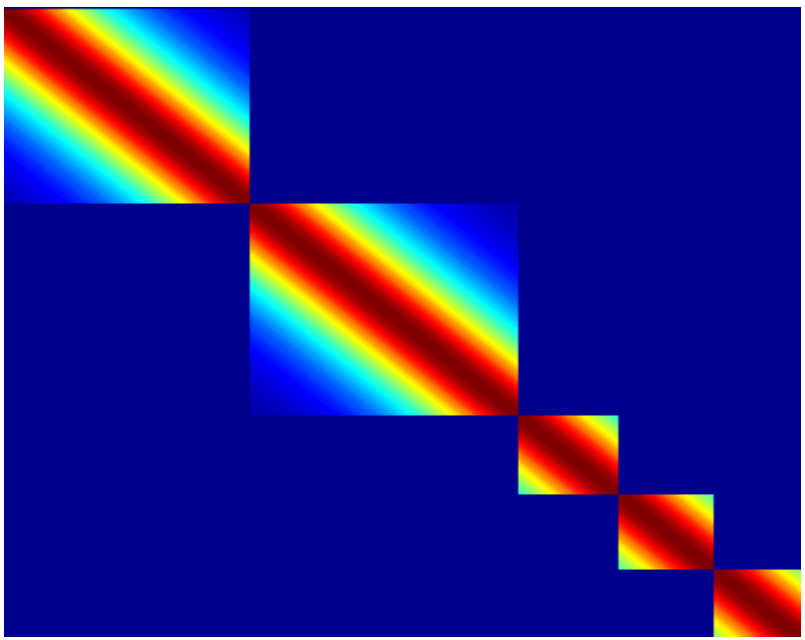
Temporal



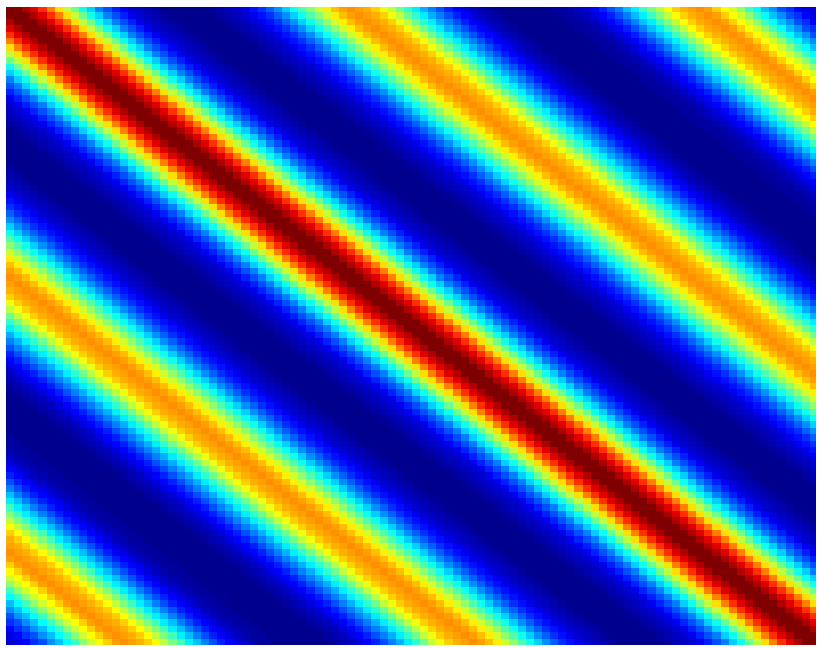
Autoencoder

Dynamics

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



Model individual sequences



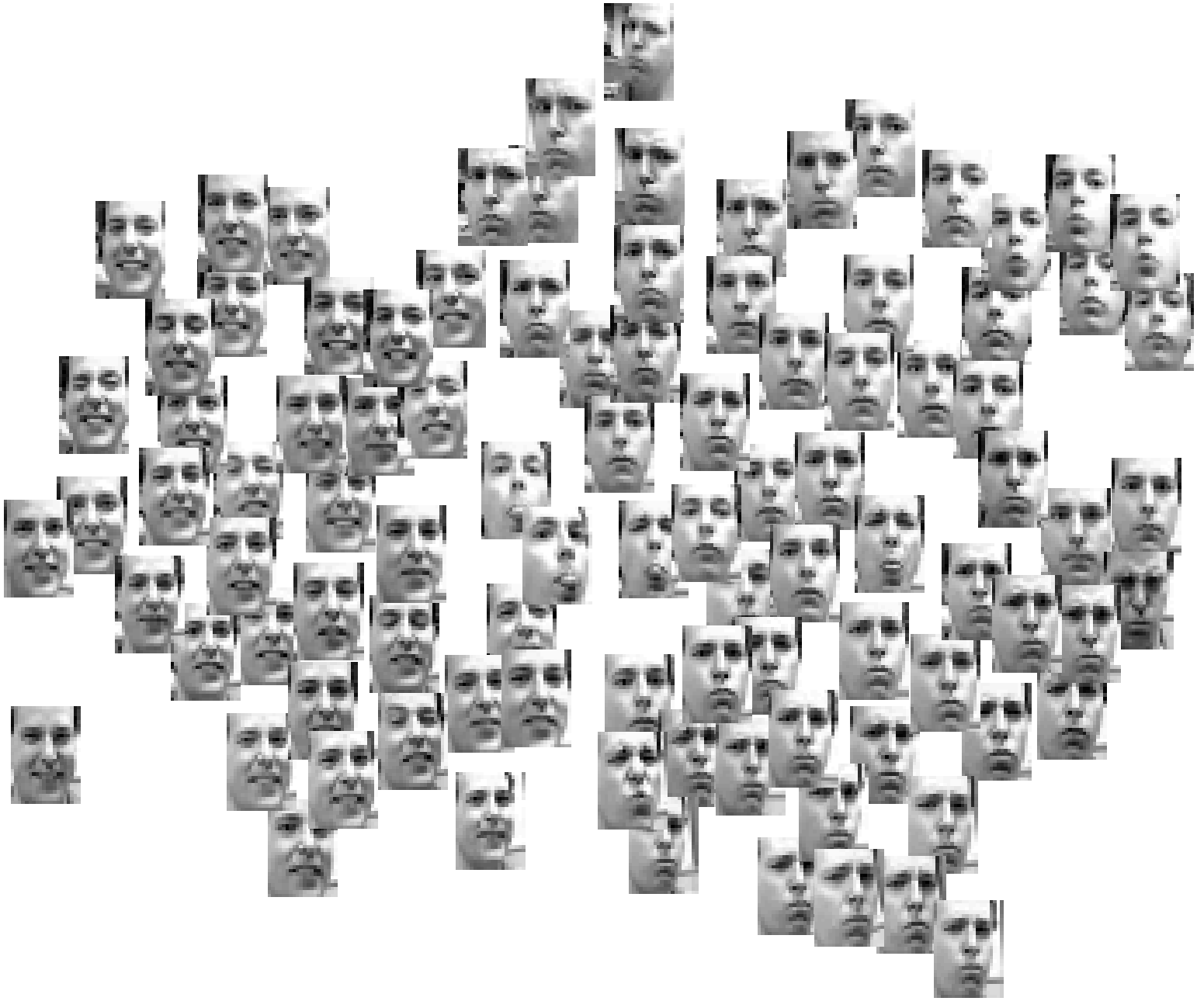
Model periodic data

- ▸ https://www.youtube.com/watch?v=i9TEoYxaBxQ (missa)
- ▸ https://www.youtube.com/watch?v=mUY1XHPnoCU (dog)
- ▸ https://www.youtube.com/watch?v=fHDWloJtgk8 (mocap)

Autoencoder example: Brendan faces

Run demo...

Autoencoder: Brendan faces (credits for idea to J. Hensman)



Summary

- A deep GP is a more general model than a GP.
- Supervised or unsupervised learning.
- 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: make it scalable with distributed computations.
- Future: how does it compare to / complement more traditional deep models?

Thanks

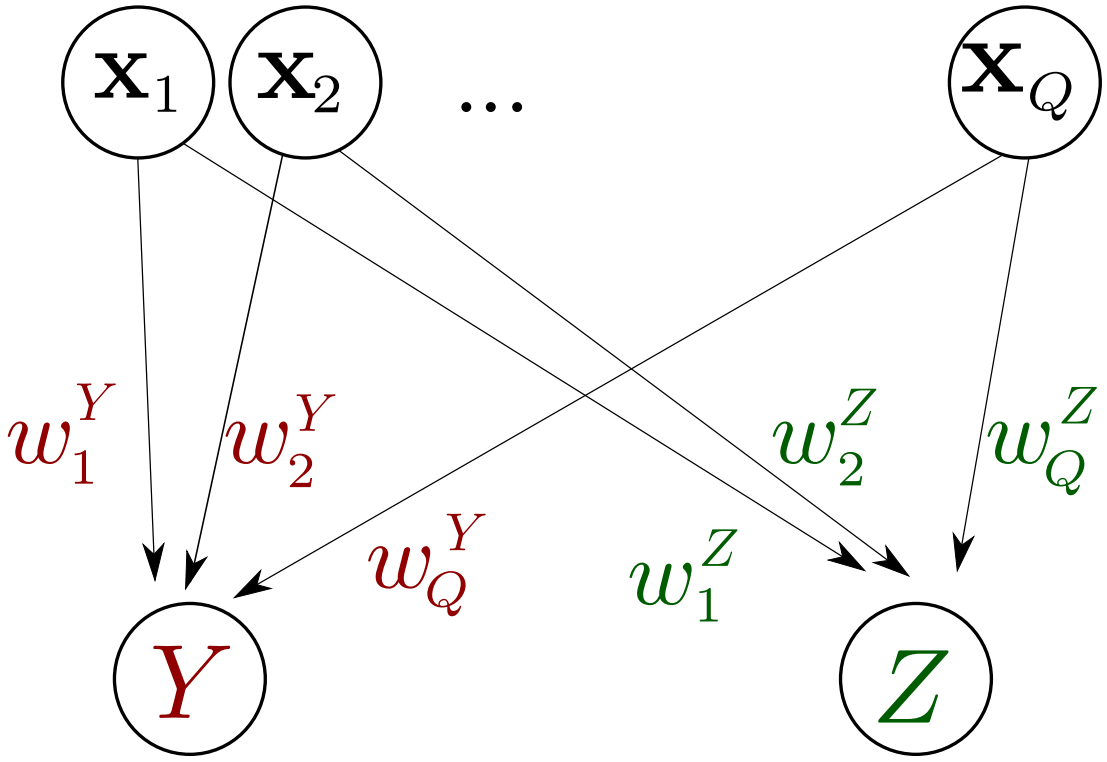
Thanks to Neil Lawrence, James Hensman, Michalis Titsias, Carl Henrik Ek.

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

BACKUP SLIDES

MRD weights



Dimensionality reduction: Linear vs non-linear

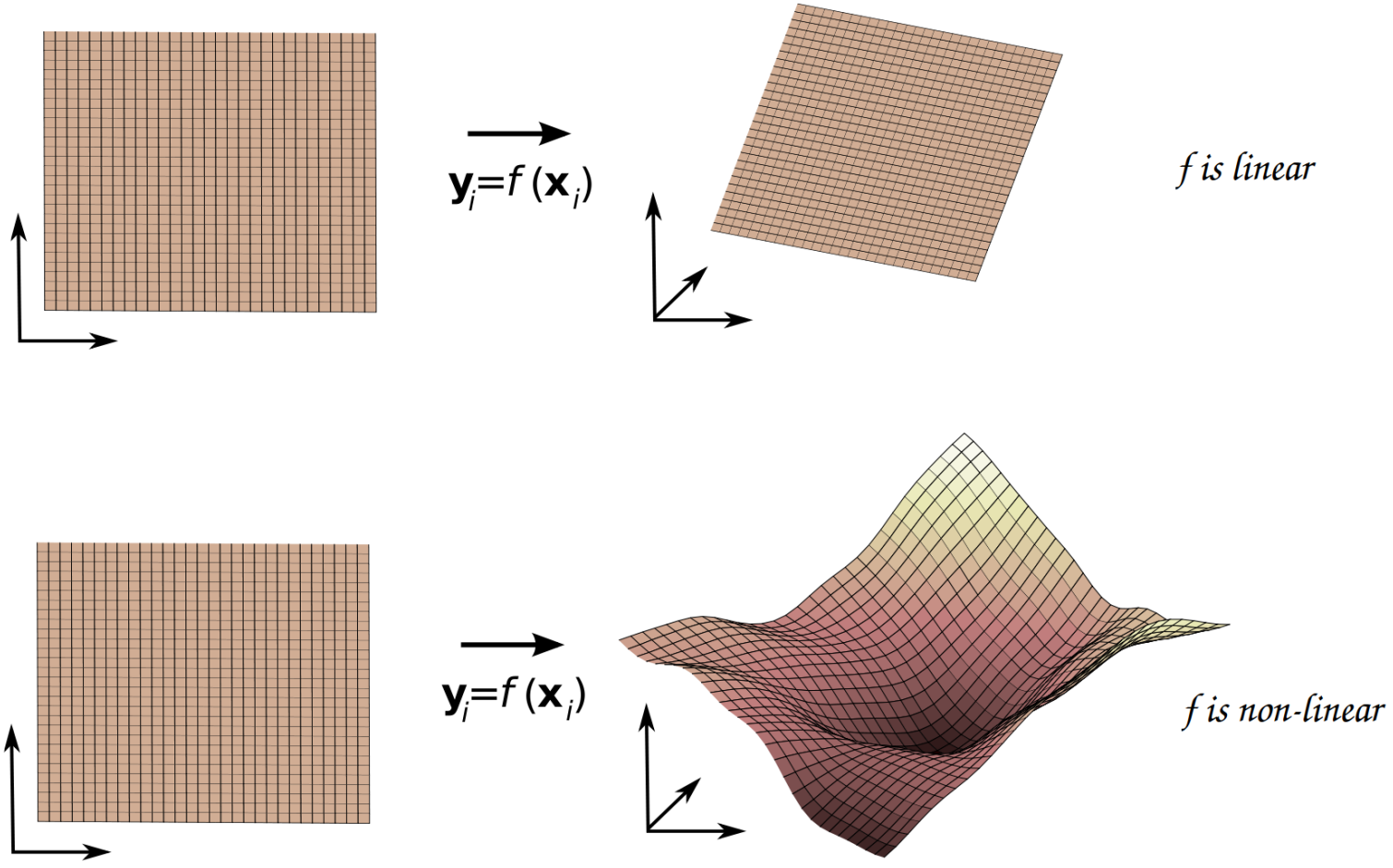


Image from: "Dimensionality Reduction the Probabilistic Way", N. Lawrence, ICML tutorial 2008