Bayesian Nonparametric Nonlinear System Identification

Roger Frigola
rf342@cam.ac.uk

Andrew McHutchon
ajm257@cam.ac.uk

University of Cambridge
Machine Learning Group

25th September 2013

Bayesian System Identification. Why?

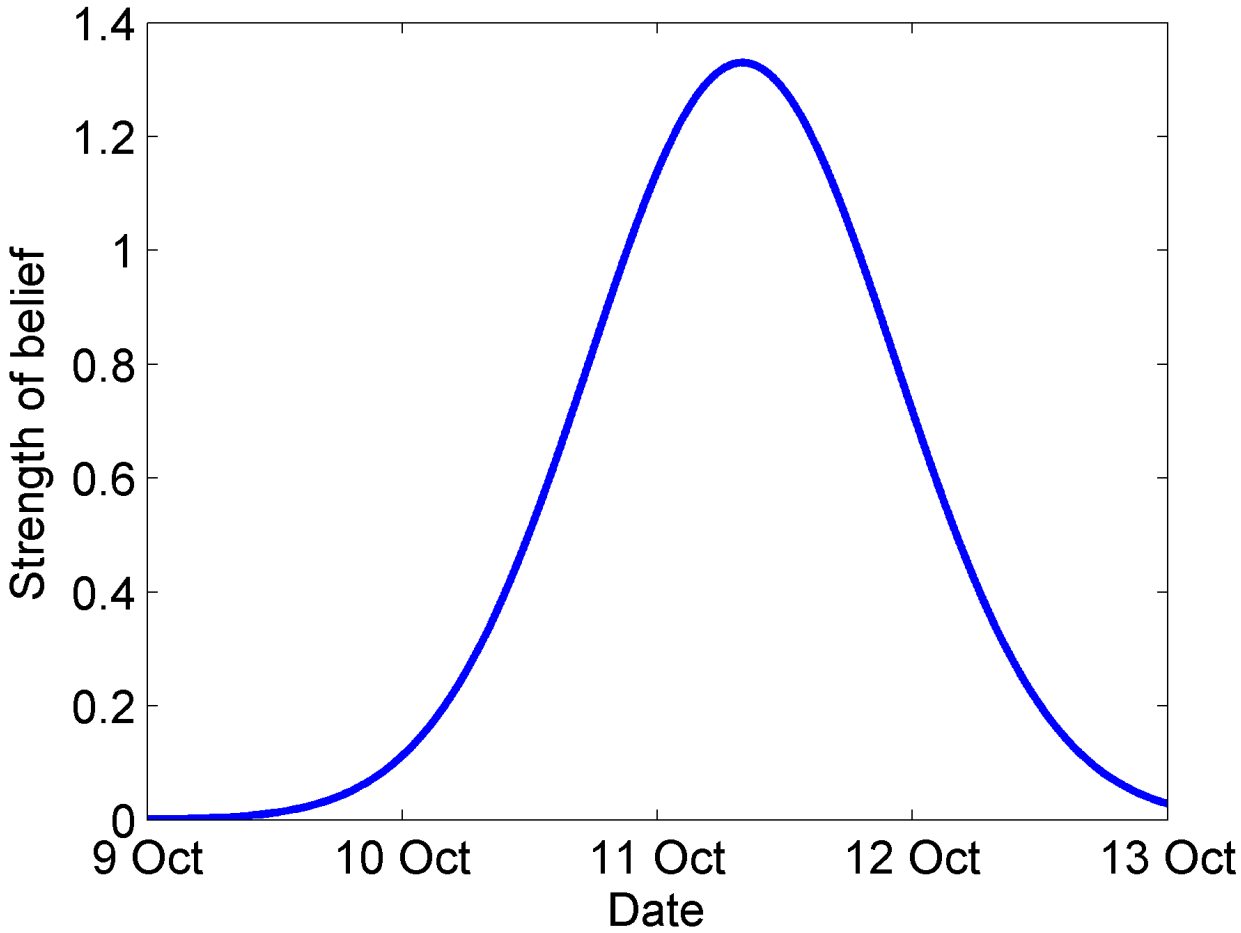
Introduction to Bayesian Modelling and Inference

Uses probability to *quantify uncertainty*.

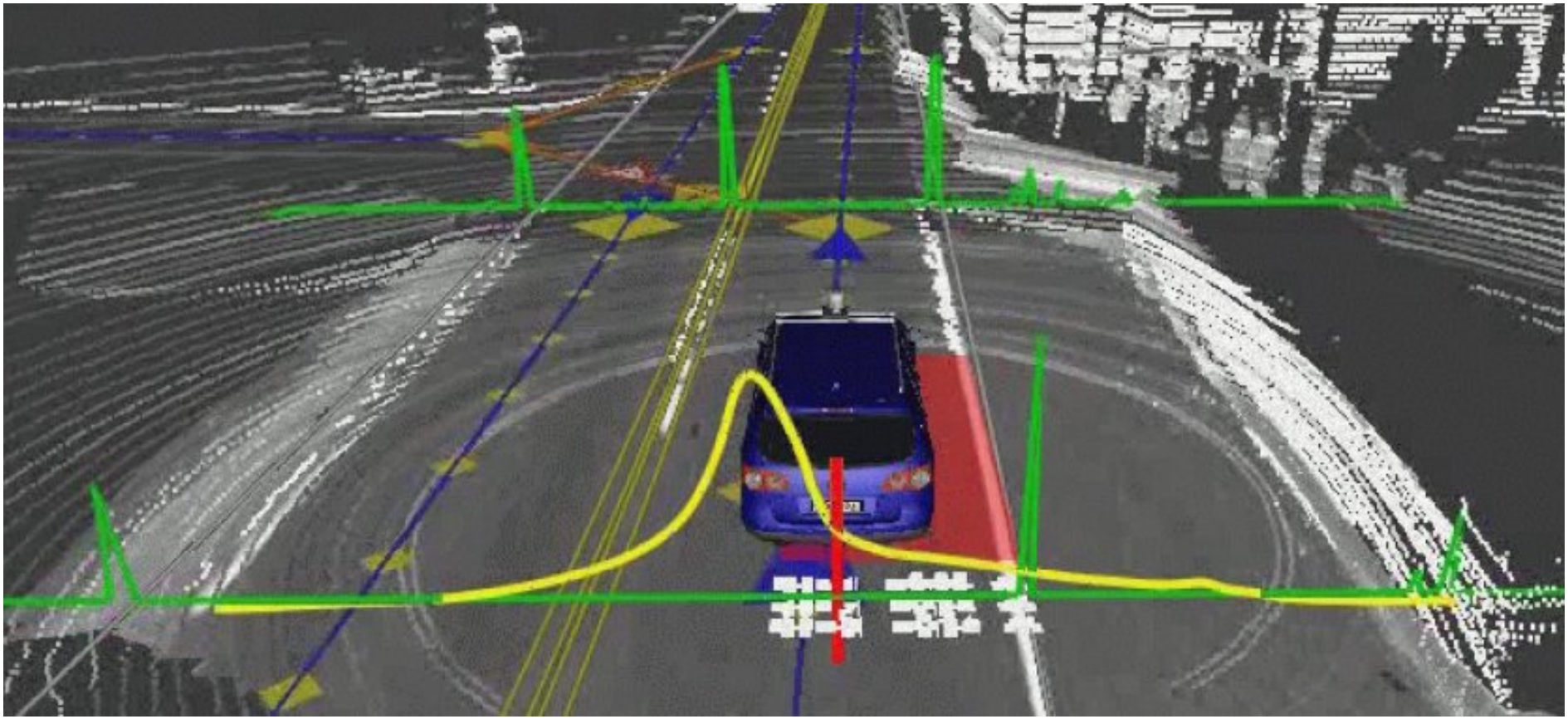
Related to information rather than repeated trials.

Uncertainty is subjective, it depends on what we have seen.

Subjective Uncertainty



Subjective Uncertainty



Stanford's self-driving car for the DARPA Urban Challenge (2007).

Bayesian Inference of Power and Drag

Infer power and drag based on noisy *acceleration measurements* using a simple inertial and aerodynamic model

$$a_t = \frac{1}{mV_t} P - \frac{\rho V_t^2 S}{2m} C_d$$

e.g. with Gaussian noise

$$y_t = \mathcal{N}(a_t, \sigma^2)$$

$$\mathcal{D} = \{y_1, \ldots, y_N, V_1, \ldots, V_N\}$$

Bayesian Inference of Power and Drag

Infer power and drag based on noisy *acceleration measurements* using a simple inertial and aerodynamic model

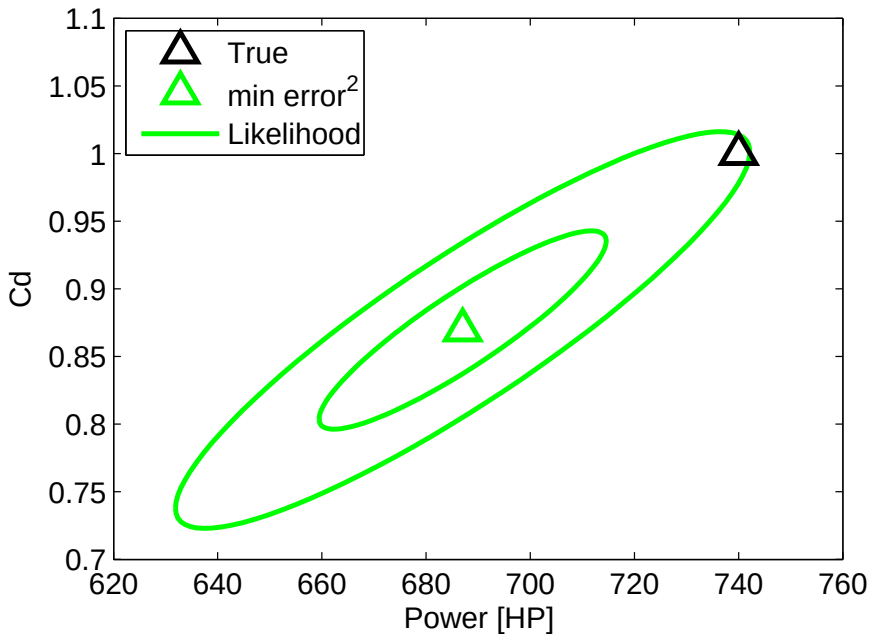
GOAL: find probability distribution of unknown parameters given data

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta)\ p(\theta)}{p(\mathcal{D})}$$

$$p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta)\ p(\theta)$$

Bayesian Inference of Power and Drag

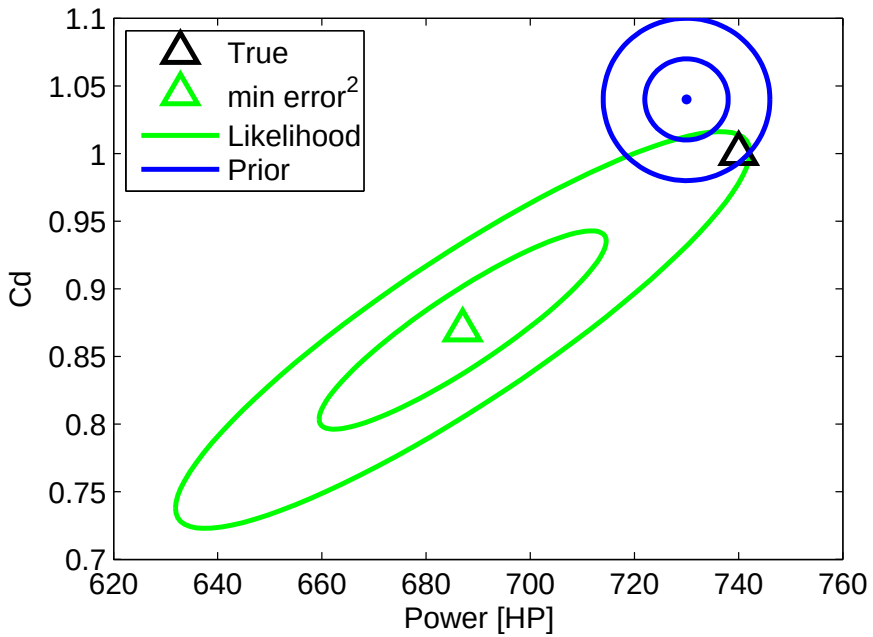
Infer power and drag based on noisy *acceleration measurements* using a simple inertial and aerodynamic model



$$p(\mathcal{D} \mid \theta)$$

Bayesian Inference of Power and Drag

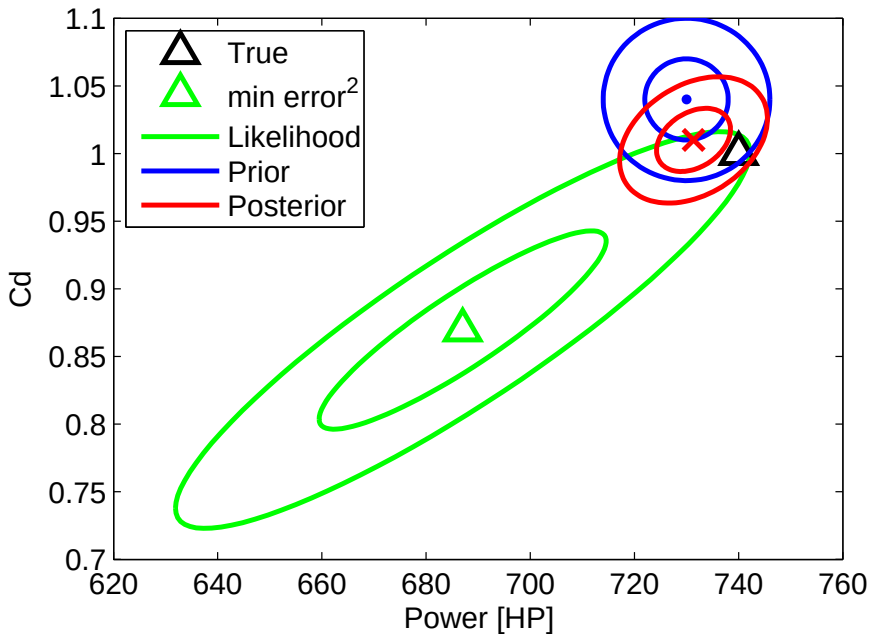
Infer power and drag based on noisy *acceleration measurements* using a simple inertial and aerodynamic model



$p(\theta)$

Bayesian Inference of Power and Drag

Infer power and drag based on noisy *acceleration measurements* using a simple inertial and aerodynamic model



$$p(\theta \mid \mathcal{D})$$

Bayesian Inference

One could say that we have used the prior as a regulariser to solve

$$\theta^* = \arg\min_{\theta} \; L(\theta, \mathcal{D}) + J(\theta)$$

But, we were simply looking for the posterior: $p(\theta \mid \mathcal{D})$.

No optimisation!

The posterior represents our uncertainty and tells us how to average different models.

Bayesian Inference

What is the outcome of Bayesian inference?

Thomas' indoor localisation example.

Posterior over parameters $\rightarrow$ posterior over identified systems.

In fact, we can find posteriors over many different kinds of objects: functions, genetic trees, English language sentences, etc.

Bayesian Inference: Making Predictions

The posterior represents our uncertainty over the parameters.

Any prediction can be found by *averaging* over the posterior

$$p(\text{LapTime} \mid \mathcal{D}) = \int p(\text{LapTime}, \theta \mid \mathcal{D})\ d\theta$$
$$= \int p(\text{LapTime} \mid \theta)\ p(\theta \mid \mathcal{D})\ d\theta.$$

Making Decisions Under Uncertainty

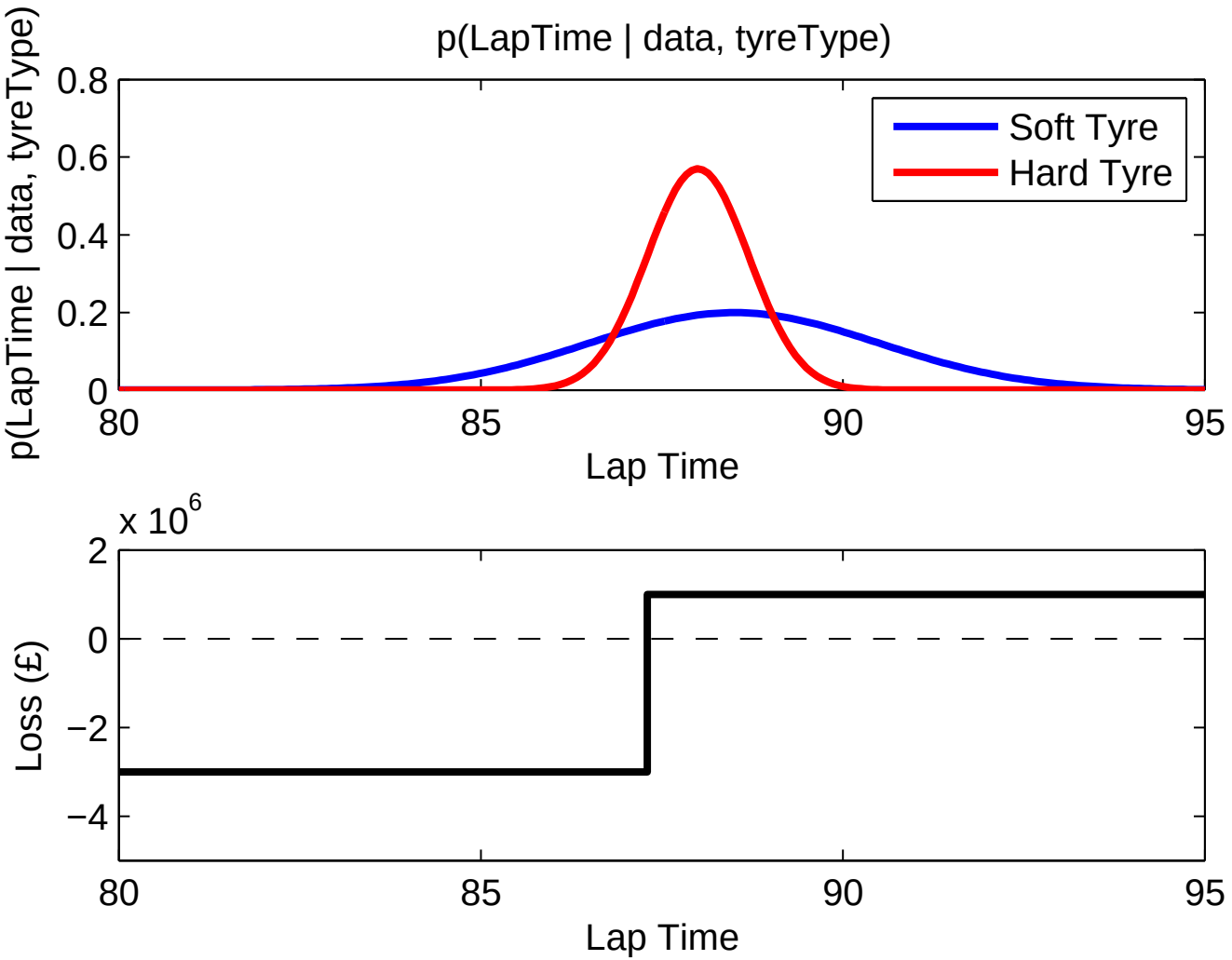
Bayesian inference provides probability distributions.

But, often, we can only take one action.

One solution: take the action that minimises the expected loss (aka risk) under the uncertainty provided by Bayesian inference.

$$a_{\text{opt}} = \arg\min_{a} \int \text{Loss}(a, \theta)\, p(\theta \mid \mathcal{D})\, d\theta$$

Making Decisions Under Uncertainty



Expected loss for hard tyre: $£3.65 \cdot 10^5$

Expected loss for soft tyre: $£-0.98 \cdot 10^5$

Over-fitting and Counting Parameters

Bayesian methods do not overfit because there is *no fitting!*

Inference is based on integration, i.e. averaging.

There is no statistical price to pay for adding more parameters.

Nonidentifiability is not a problem when making predictions.

From Formula 1 to Functions

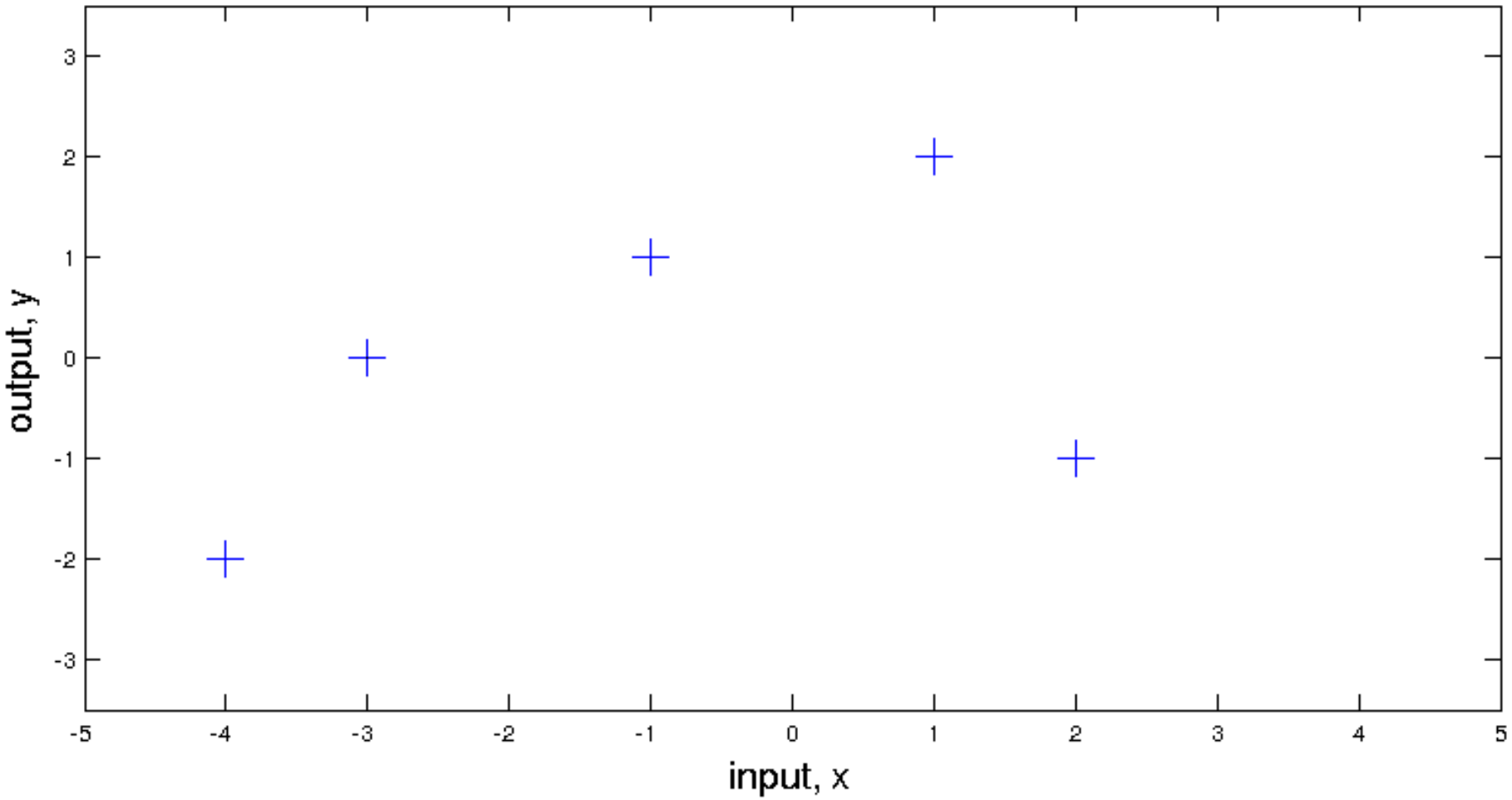
$$y = f(x) \tag{1}$$

Task: Model f in a Bayesian manner

From Formula 1 to Functions

$$y = f(x) \tag{1}$$

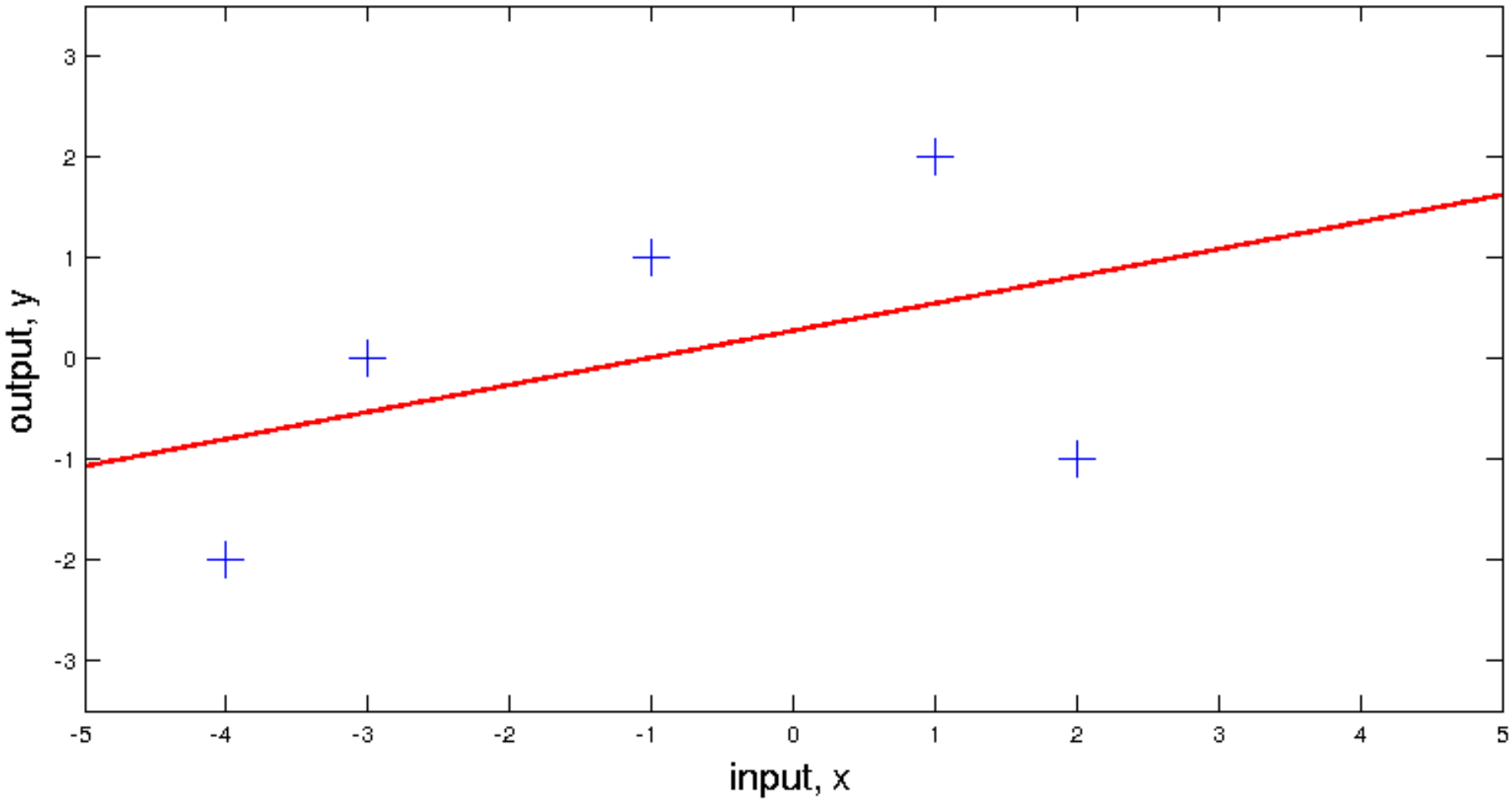
Task: Model f in a Bayesian manner



From Formula 1 to Functions

$$y = f(x) \tag{1}$$

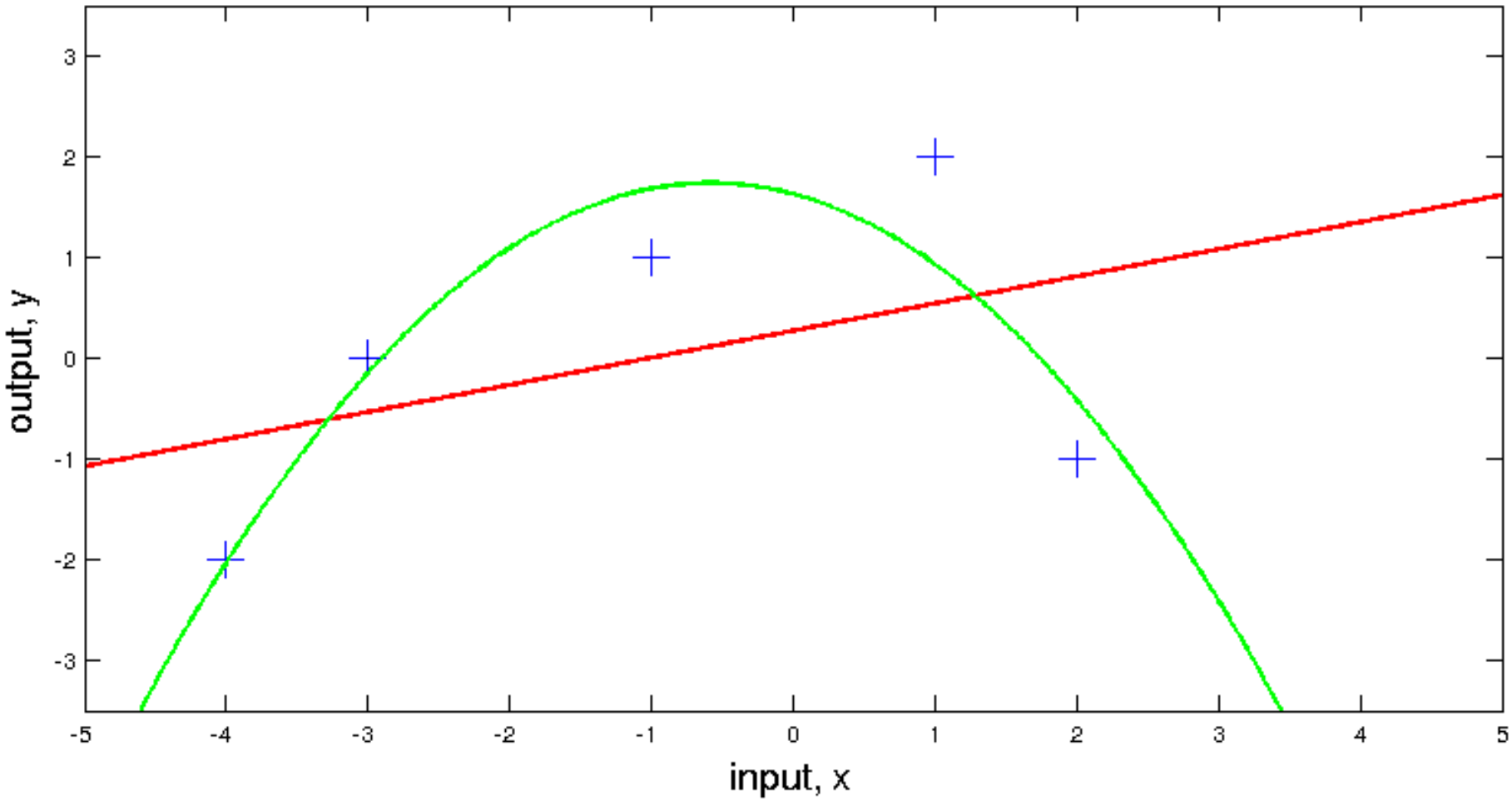
Task: Model f in a Bayesian manner



From Formula 1 to Functions

$$y = f(x) \tag{1}$$

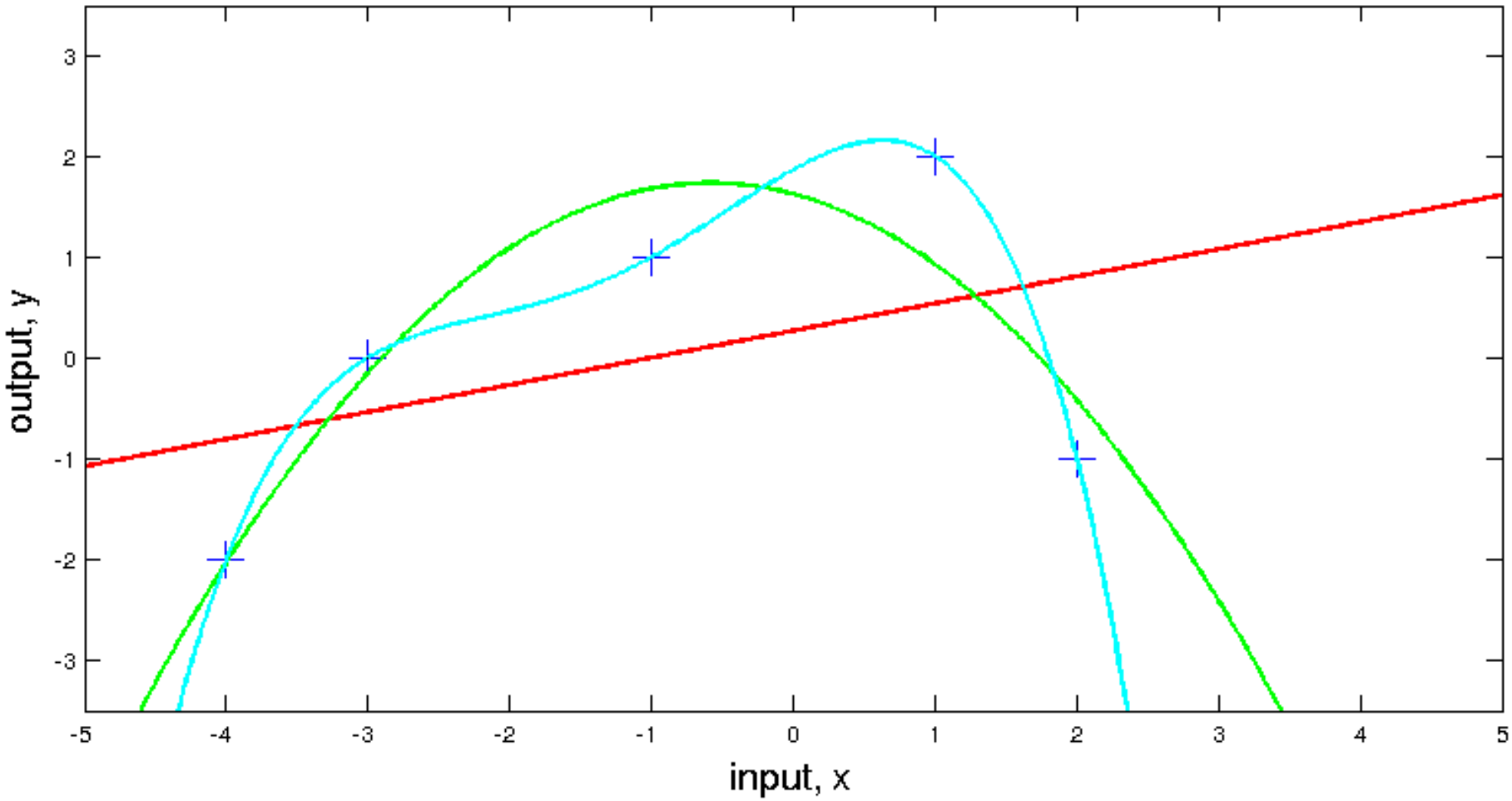
Task: Model f in a Bayesian manner



From Formula 1 to Functions

$$y = f(x) \tag{1}$$

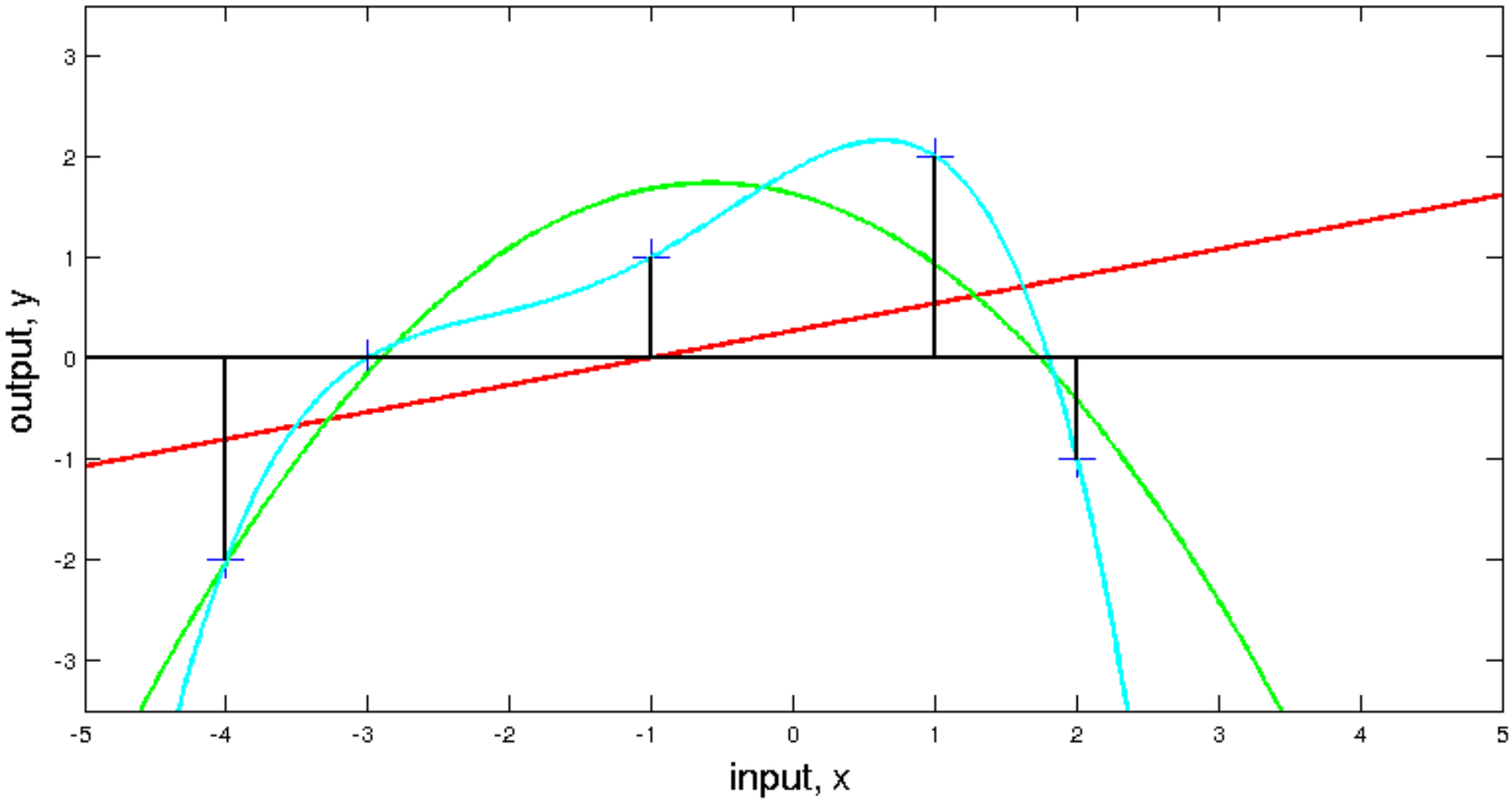
Task: Model f in a Bayesian manner



From Formula 1 to Functions

$$y = f(x) \tag{1}$$

Task: Model f in a Bayesian manner



Nonlinear Models

Parametric nonlinear models,

- Polynomial, $f(x) = \sum_{i=0}^{N} w_i x^i$
- Radial Basis Function, $f(x) = \sum_{i=1}^{N} w_i \phi(x - c_i)$
- Fourier, $f(x) = \sum_{i=1}^{N} a_i \sin(w_i x - c_i)$

Problems,

- Either have to fix N or put a prior on it
- Can have a very large number of parameters to estimate
- Inference can be difficult
- Unclear how to choose priors

Nonparametric Models

Allow you to do inference on function values directly

Nonparametric Models

Allow you to do inference on function values directly

Parameterised directly by the training data

Nonparametric Models

Allow you to do inference on function values directly

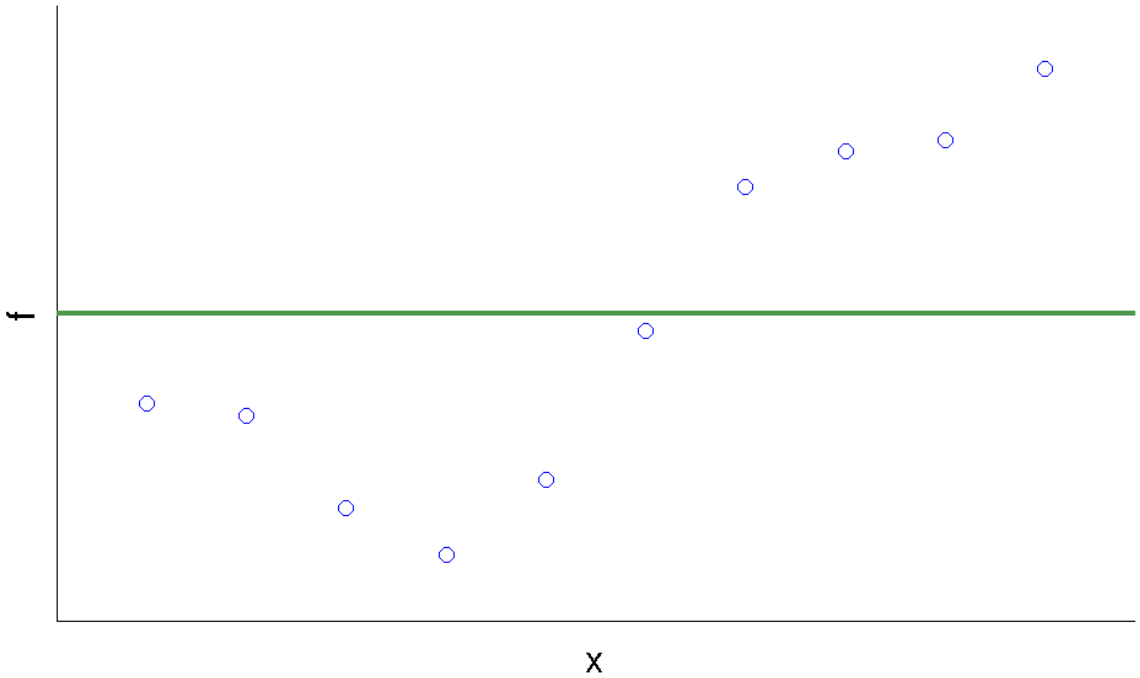
Parameterised directly by the training data

- Very flexible class of models with limited prior assumptions
- Infinite number of features/basis functions
- Finite, often very small, number of parameters left to deal with
- Intuitive, high level priors: smoothness, periodicity

Gaussian Processes

Prior distribution over $f(x)$ is Gaussian

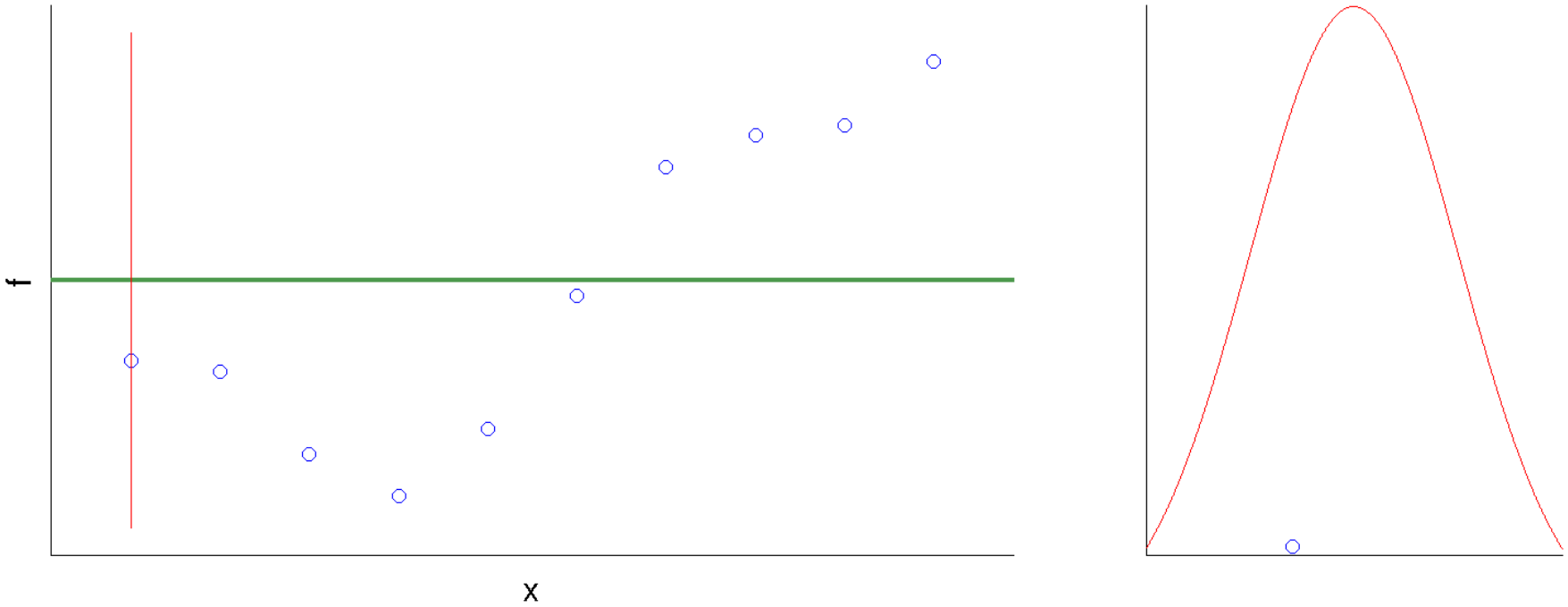
$$\begin{aligned} f &\sim \mathcal{GP}(m,\ k) \\ \Rightarrow\ P(f(x)) &= \mathcal{N}(m(x),\ k(x)) \end{aligned} \tag{2}$$



Gaussian Processes

Prior distribution over $f(x)$ is Gaussian

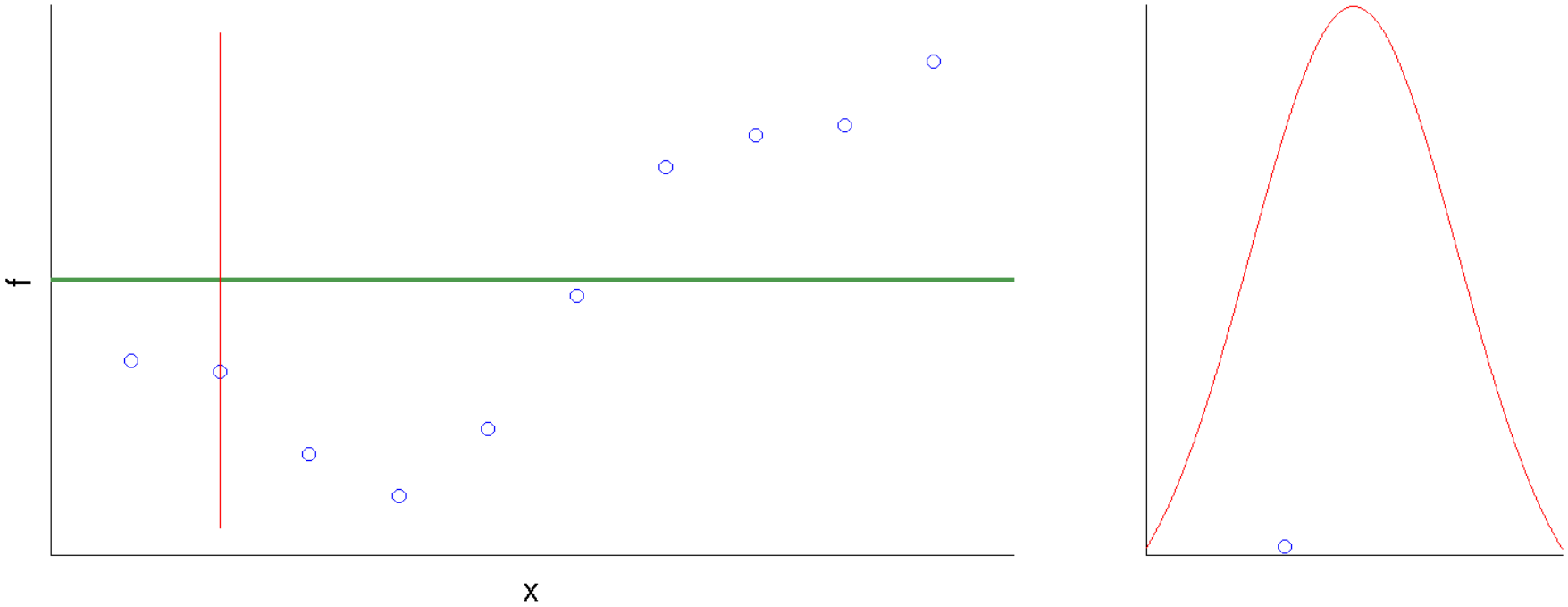
$$\begin{aligned} f &\sim \mathcal{GP}(m,\ k) \\ \Rightarrow\ P(f(x)) &= \mathcal{N}(m(x),\ k(x)) \end{aligned} \tag{2}$$



Gaussian Processes

Prior distribution over $f(x)$ is Gaussian

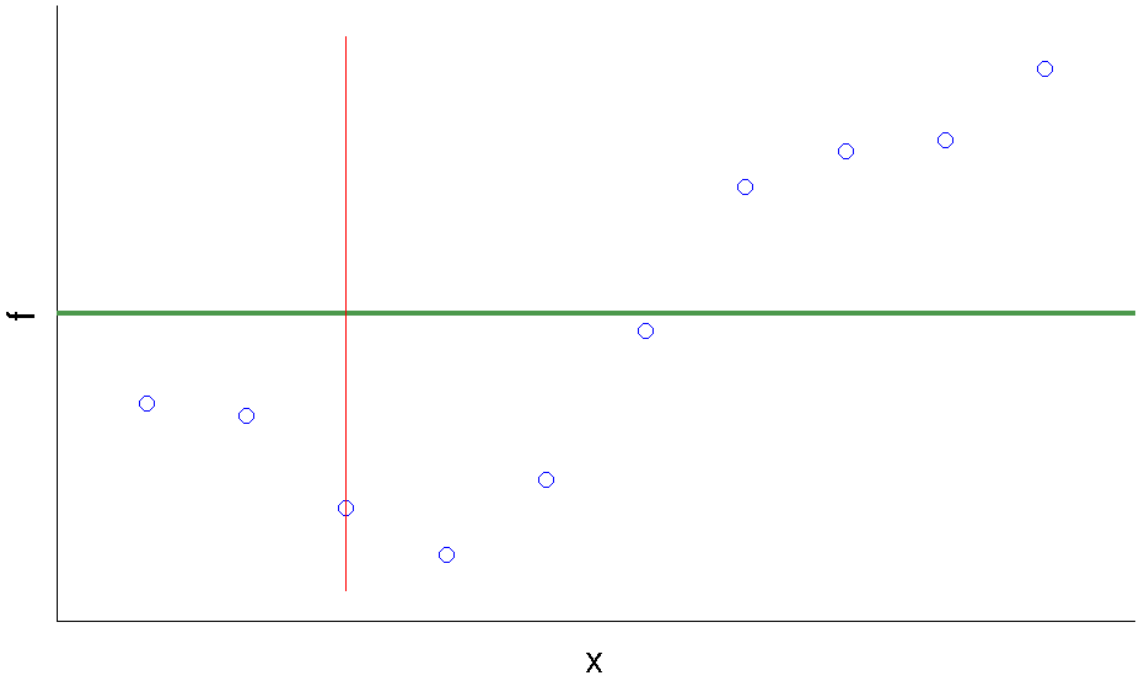
$$\begin{aligned} f &\sim \mathcal{GP}(m,\ k) \\ \Rightarrow\ P(f(x)) &= \mathcal{N}(m(x),\ k(x)) \end{aligned} \tag{2}$$

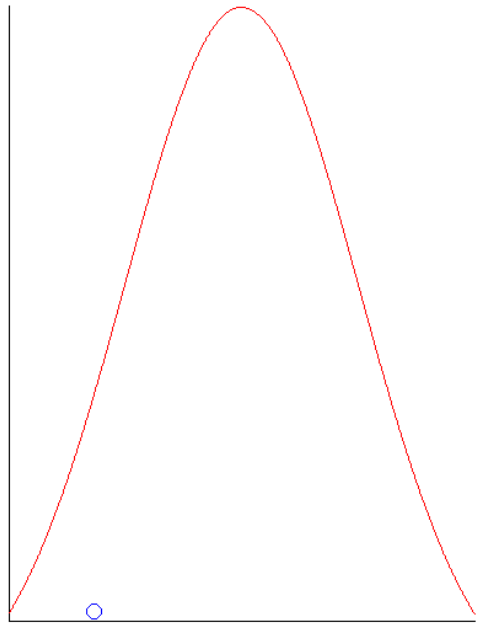


Gaussian Processes

Prior distribution over $f(x)$ is Gaussian

$$\begin{aligned} f &\sim \mathcal{GP}(m,\ k) \\ \Rightarrow\ P(f(x)) &= \mathcal{N}(m(x),\ k(x)) \end{aligned} \tag{2}$$

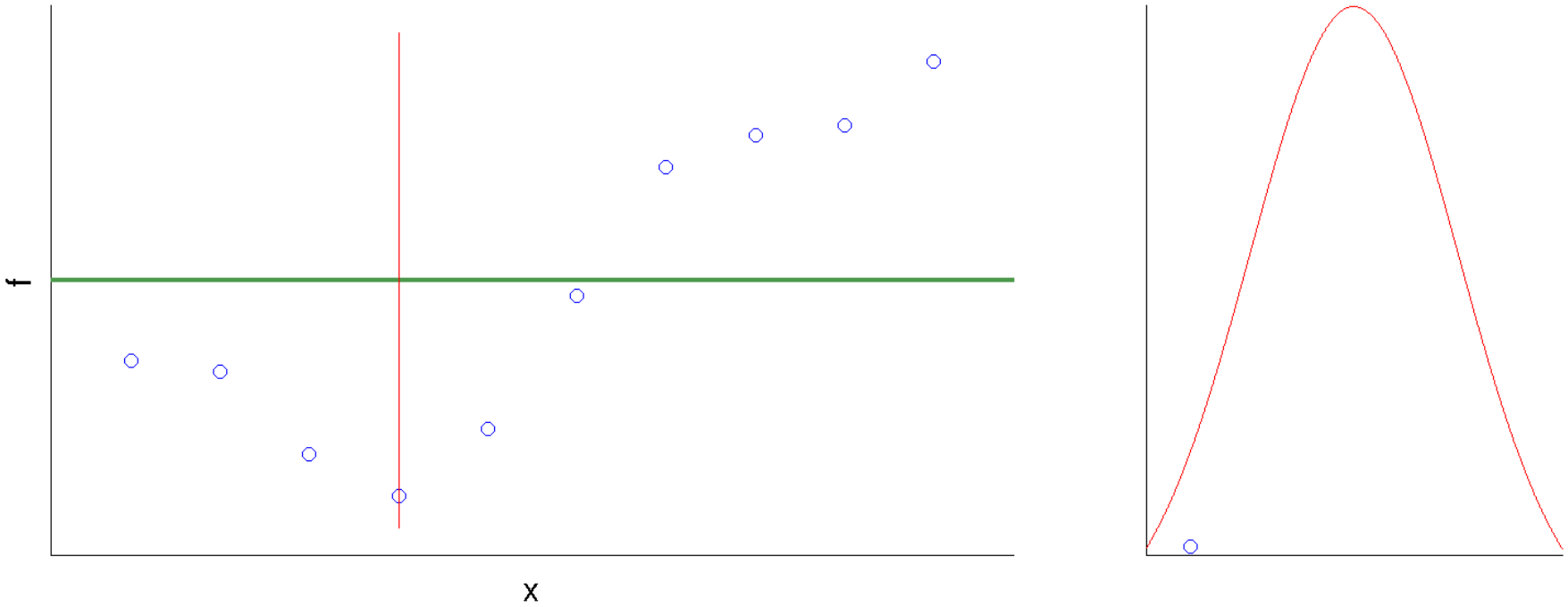




Gaussian Processes

Prior distribution over $f(x)$ is Gaussian

$$\begin{aligned} f &\sim \mathcal{GP}(m,\ k) \\ \Rightarrow\ P(f(x)) &= \mathcal{N}(m(x),\ k(x)) \end{aligned} \tag{2}$$

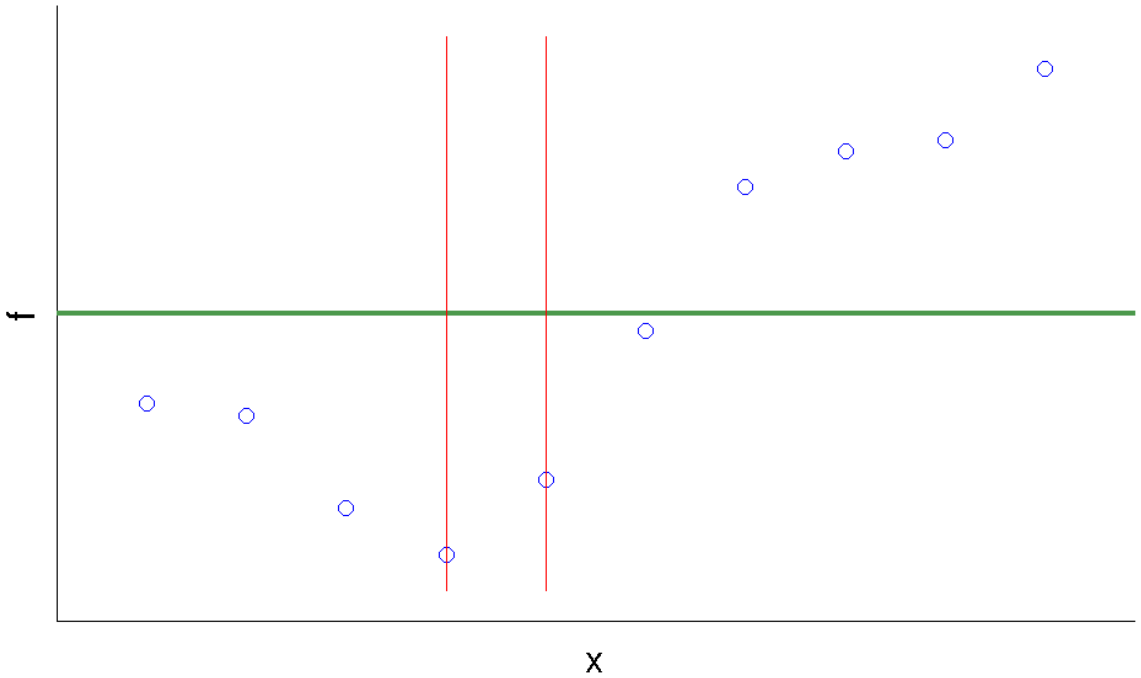


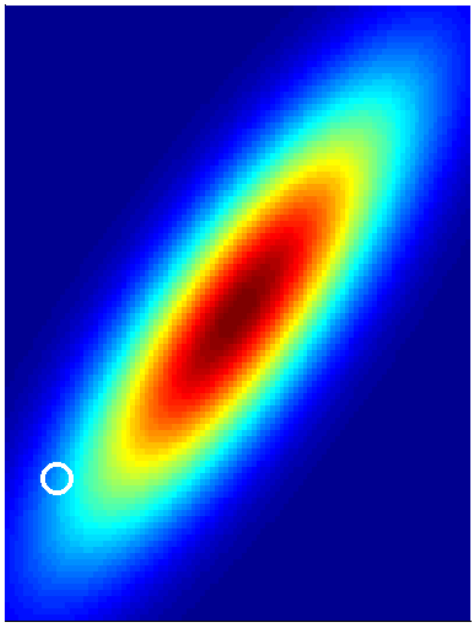
Gaussian Processes

Prior distribution over $f(x)$ is Gaussian

$$f \sim \mathcal{GP}(m,\ k) \tag{2}$$

$$\Rightarrow\ P(f(x_1),\ f(x_2)) = \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ m(x_2) \end{bmatrix}, \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) \\ k(x_2,x_1) & k(x_2,x_2) \end{bmatrix}\right)$$



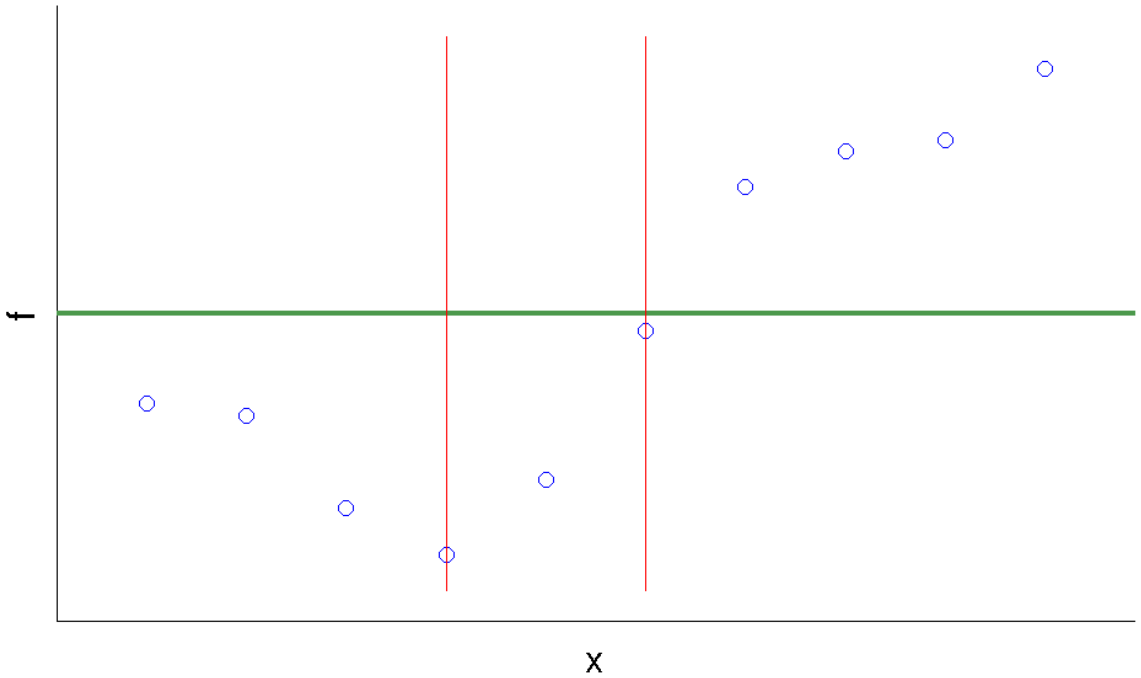


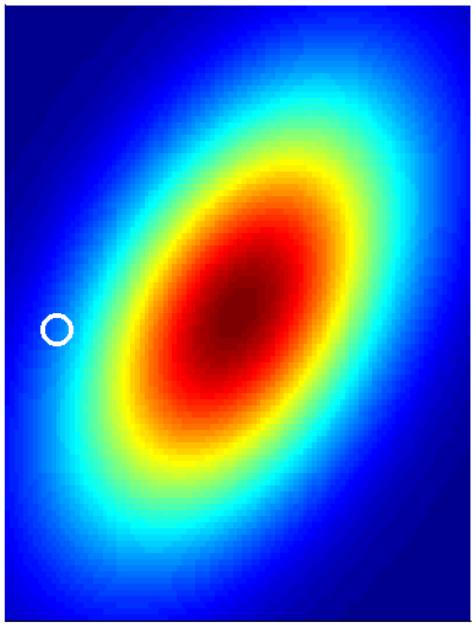
Gaussian Processes

Prior distribution over $f(x)$ is Gaussian

$$f \sim \mathcal{GP}(m,\ k) \tag{2}$$

$$\Rightarrow\ P(f(x_1),\ f(x_2)) = \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ m(x_2) \end{bmatrix}, \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) \\ k(x_2,x_1) & k(x_2,x_2) \end{bmatrix}\right)$$



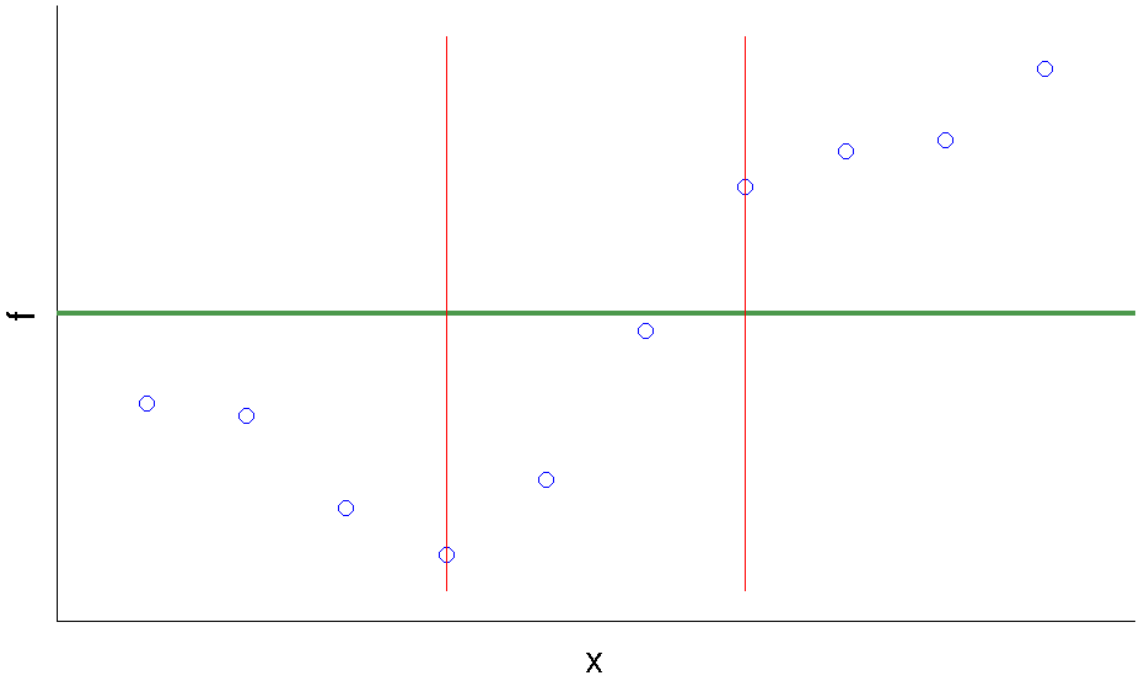


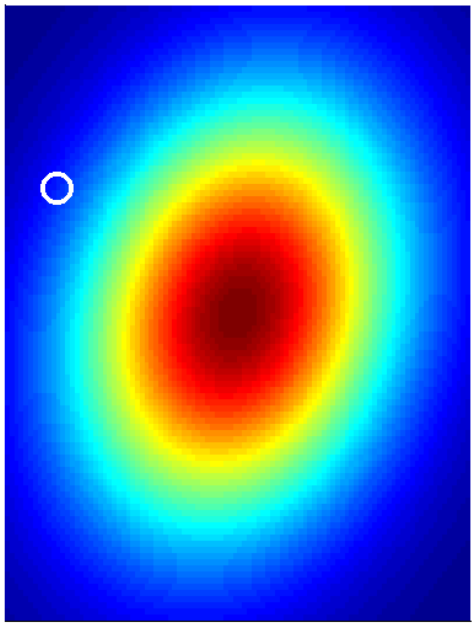
Gaussian Processes

Prior distribution over $f(x)$ is Gaussian

$$f \sim \mathcal{GP}(m,\ k) \tag{2}$$

$$\Rightarrow\ P(f(x_1),\ f(x_2)) = \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ m(x_2) \end{bmatrix}, \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) \\ k(x_2,x_1) & k(x_2,x_2) \end{bmatrix}\right)$$



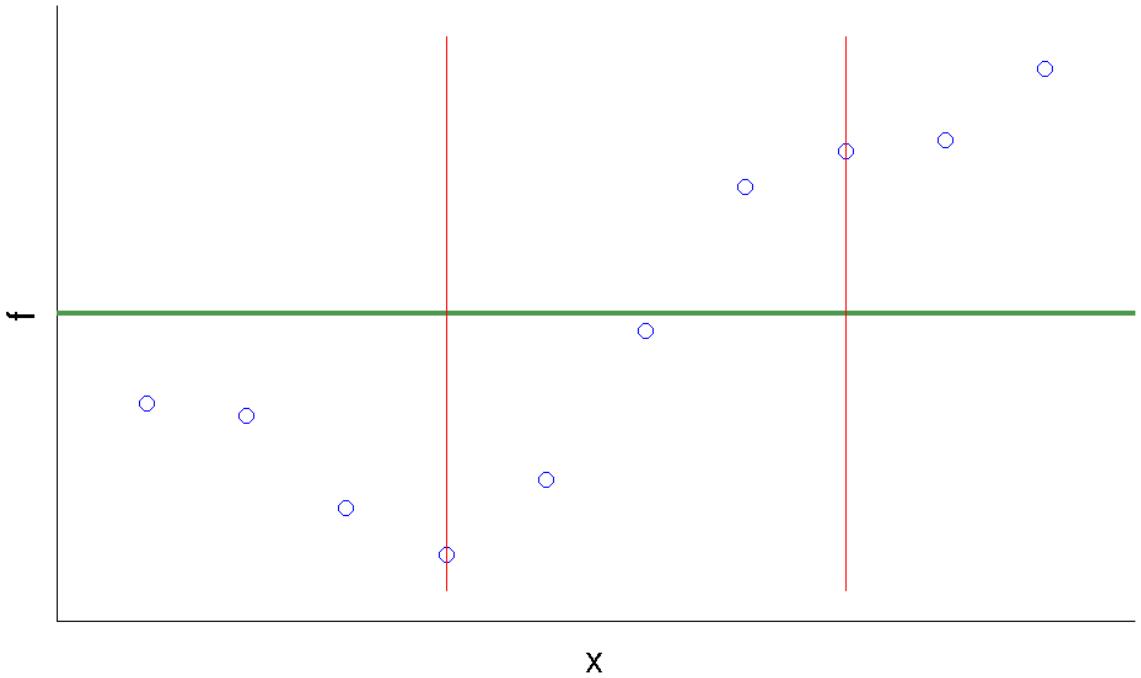


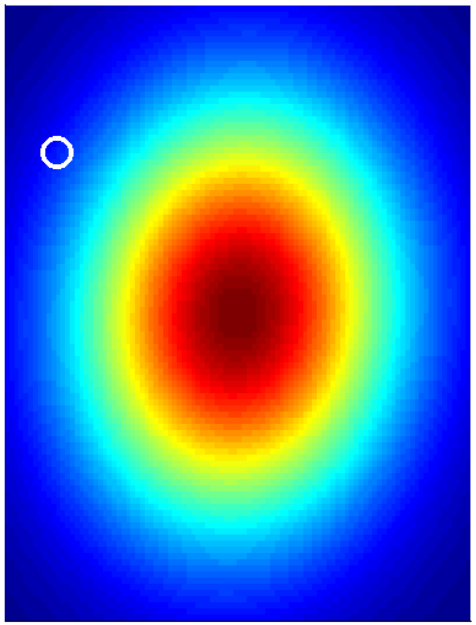
Gaussian Processes

Prior distribution over $f(x)$ is Gaussian

$$f \sim \mathcal{GP}(m,\ k) \tag{2}$$

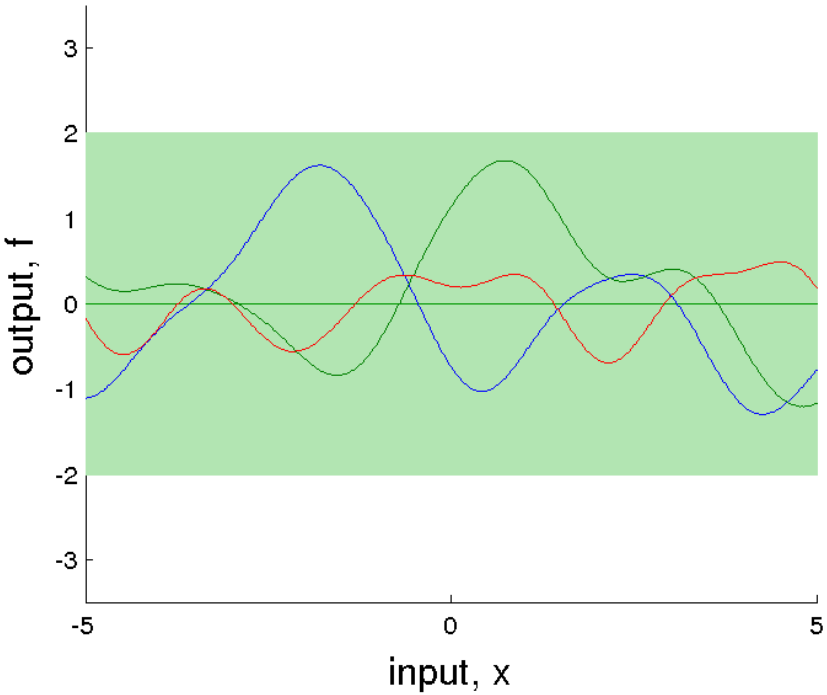
$$\Rightarrow\ P(f(x_1),\ f(x_2)) = \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ m(x_2) \end{bmatrix}, \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) \\ k(x_2,x_1) & k(x_2,x_2) \end{bmatrix}\right)$$





Gaussian Processes

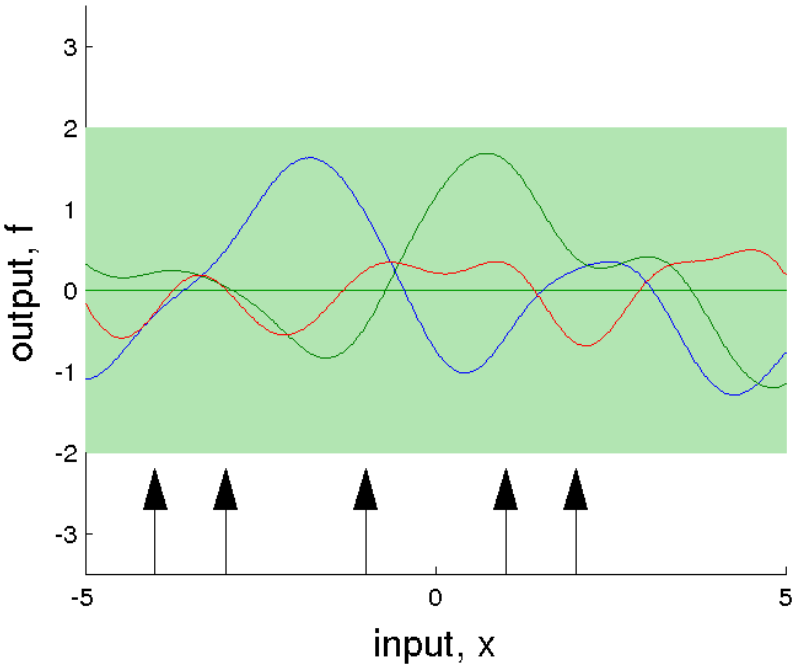
$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x) \\ m(x_*) \end{bmatrix}, \begin{bmatrix} k(x,x) & k(x,x_*) \\ k(x_*,x) & k(x_*,x_*) \end{bmatrix}\right) \tag{3}$$



Prior, $P(f, f^*)$

Gaussian Processes

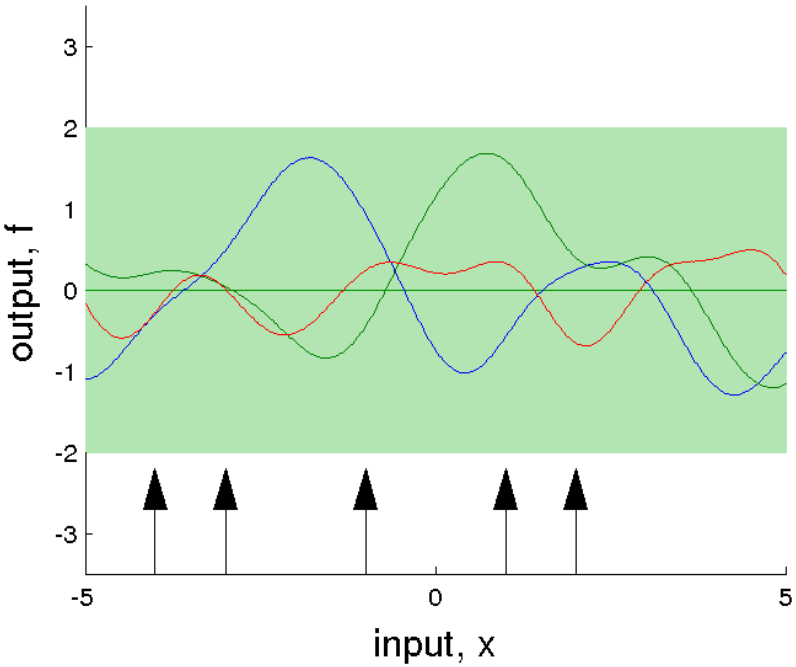
$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x) \\ m(x_*) \end{bmatrix}, \begin{bmatrix} k(x,x) & k(x,x_*) \\ k(x_*,x) & k(x_*,x_*) \end{bmatrix} \right) \tag{3}$$



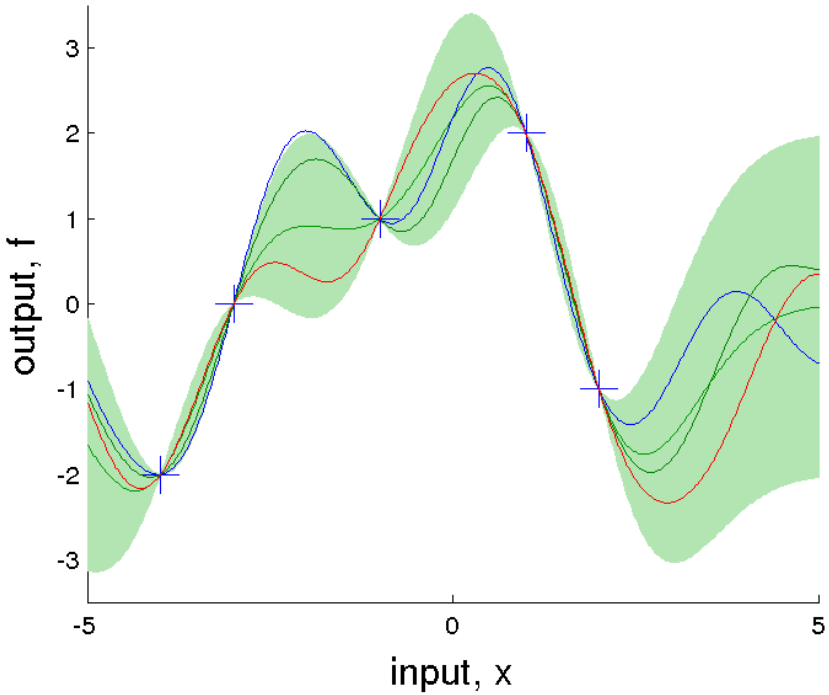
Prior, $P(f, f^*)$

Gaussian Processes

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x) \\ m(x_*) \end{bmatrix}, \begin{bmatrix} k(x,x) & k(x,x_*) \\ k(x_*,x) & k(x_*,x_*) \end{bmatrix} \right) \tag{3}$$



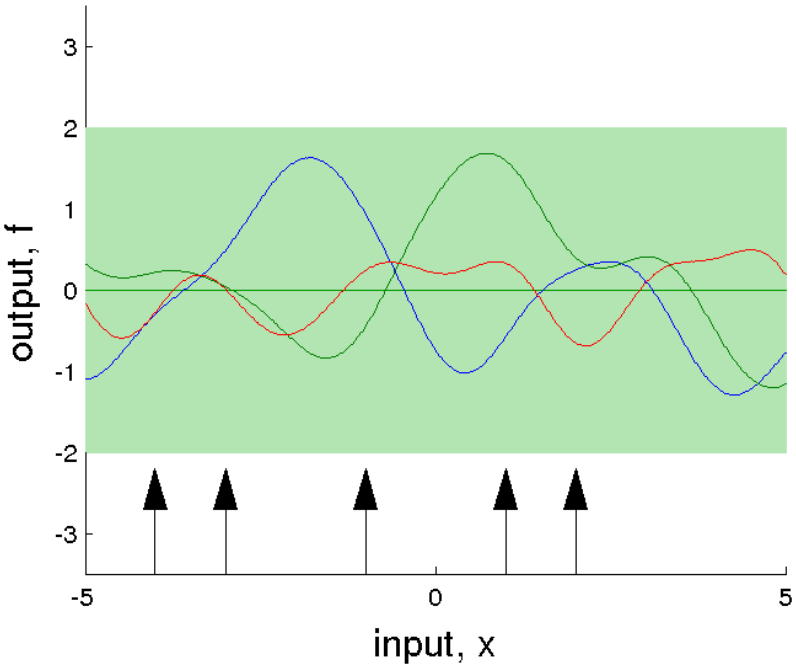
Prior, $P(f, f^*)$



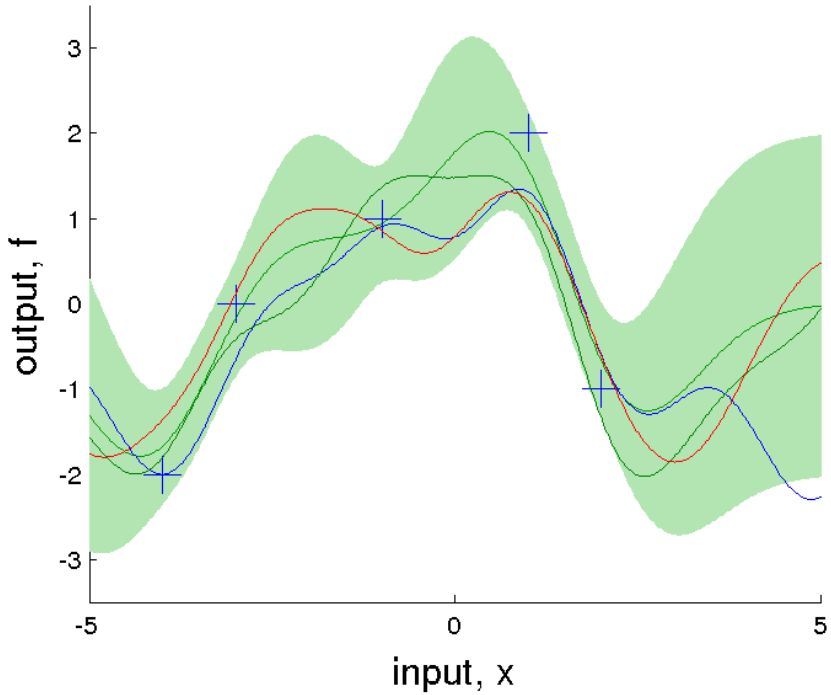
Posterior, $P(f^* \mid f)$

Gaussian Processes

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x) \\ m(x_*) \end{bmatrix}, \begin{bmatrix} k(x,x)+\sigma_n^2 I & k(x,x_*) \\ k(x_*,x) & k(x_*,x_*) \end{bmatrix} \right) \quad (4)$$



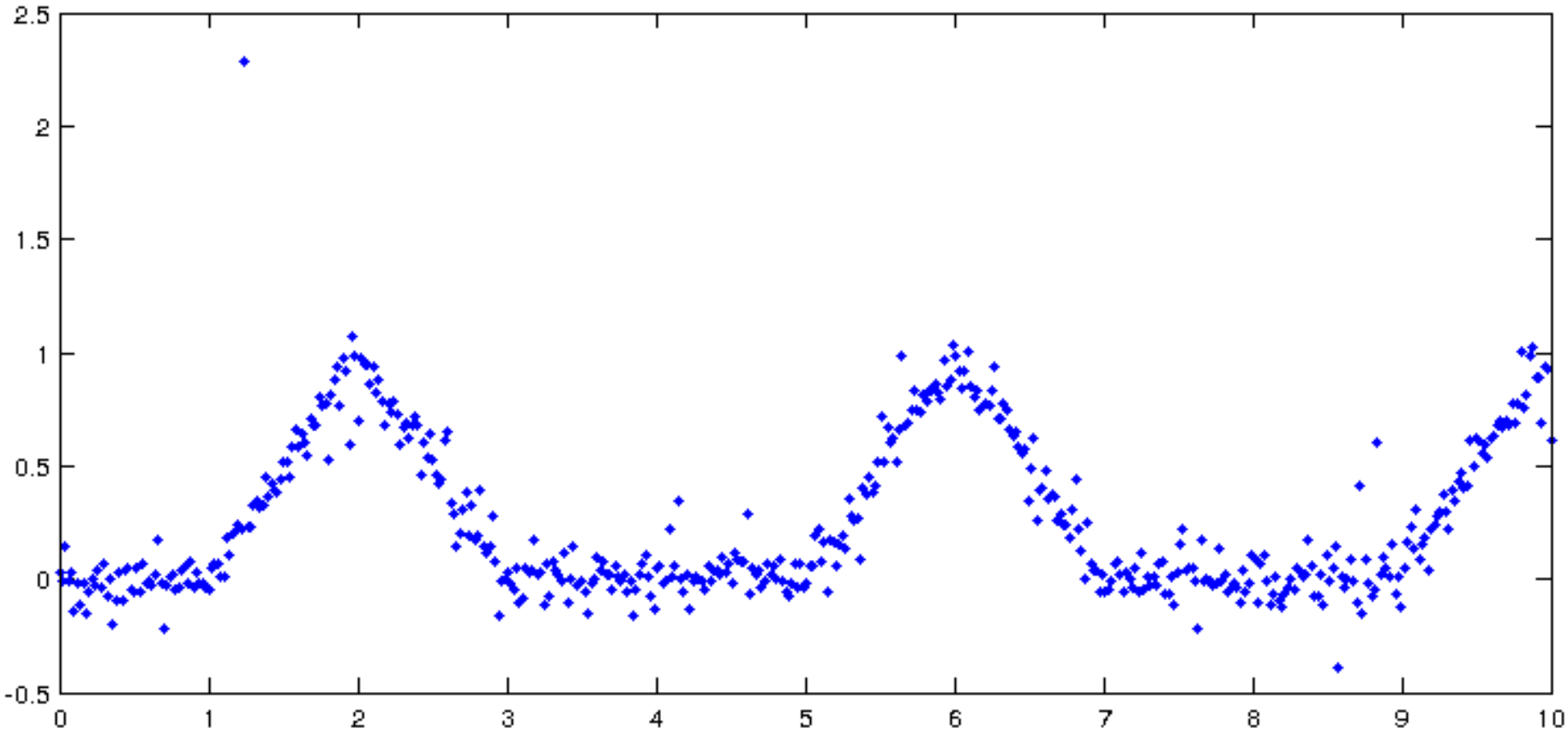
Prior, $P(f, f^*)$



Posterior, $P(f^* \mid y)$

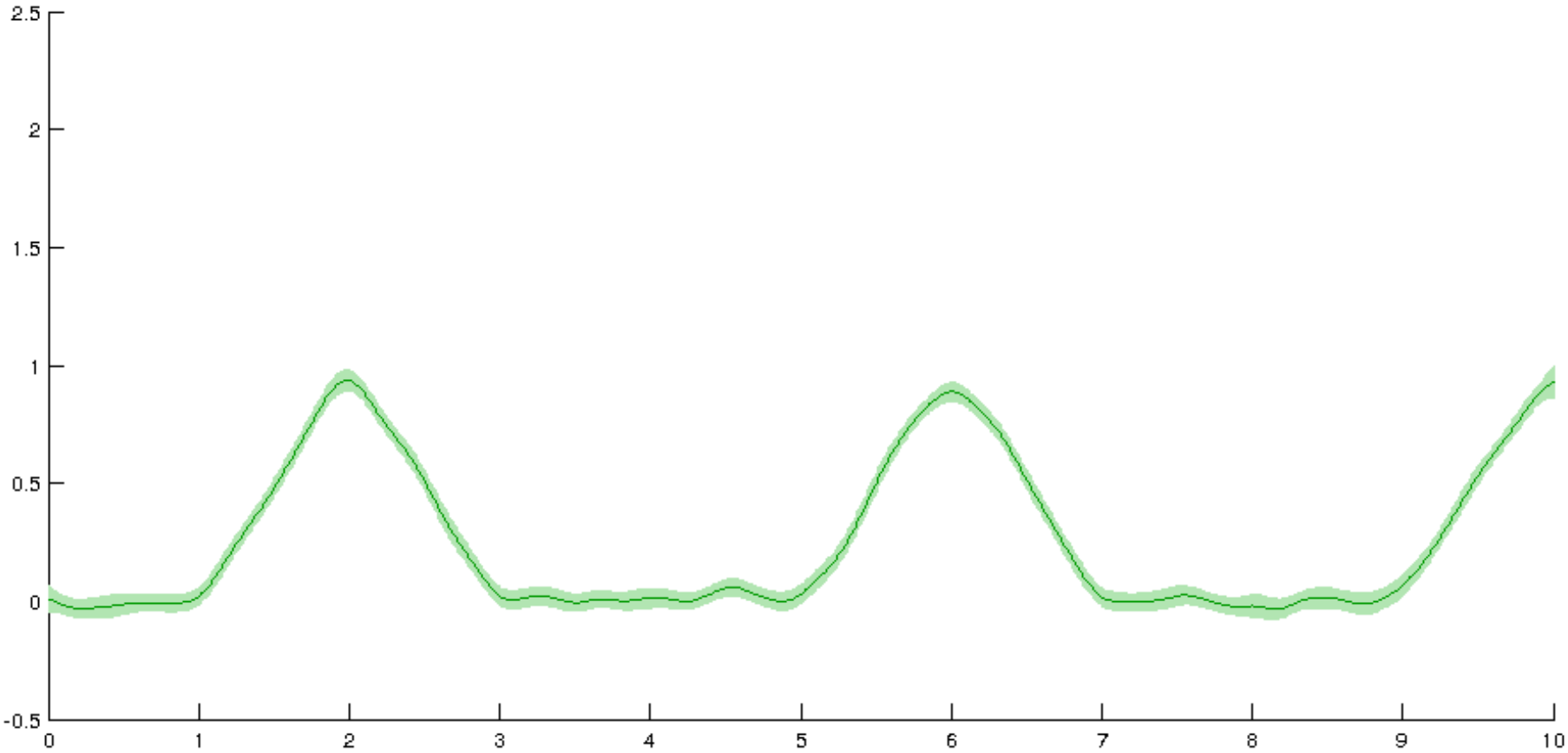
Gaussian Processes

- But my system doesn't have Gaussian noise...
- ...and my system isn't smooth



Gaussian Processes

- But my system doesn't have Gaussian noise...
- ...and my system isn't smooth



GP-NARX Models

Use a GP to model the mapping,

$$y_t = f(y_{t-1}, \ldots, y_{t-k}) + \epsilon \tag{5}$$

$$f \sim \mathcal{GP}(m,\ k) \tag{6}$$

GP-NARX Models

Use a GP to model the mapping,

$$y_t = f(y_{t-1}, \ldots, y_{t-k}) + \epsilon \tag{5}$$

$$f \sim \mathcal{GP}(m, k) \tag{6}$$

If ϵ is Gaussian,

$$p(f^* | x^*, X, y) = \mathcal{N}(\mu, \Sigma) \tag{7}$$

GP-NARX Models

Use a GP to model the mapping,

$$y_t = f(y_{t-1}, \ldots, y_{t-k}) + \epsilon \tag{5}$$
$$f \sim \mathcal{GP}(m, k) \tag{6}$$

If ϵ is Gaussian,

$$p(f^* | x^*, X, y) = \mathcal{N}(\mu, \Sigma) \tag{7}$$

$$\mu = k(x^*, X) \left[K(X, X) + \sigma_\epsilon^2 I\right]^{-1} y \tag{8}$$
$$\Sigma = k(x^*, x^*) - k(x^*, X) \left[K(X, X) + \sigma_\epsilon^2 I\right]^{-1} k(X, x^*) \tag{9}$$

GP-NARX Models

Use a GP to model the mapping,

$$y_t = f(y_{t-1}, \ldots, y_{t-k}) + \epsilon \tag{10}$$

$$f \sim \mathcal{GP}(m,\ k) \tag{11}$$

Problem: noise in the inputs, "errors in the variables"

GP-NARX Models

Use a GP to model the mapping,

$$y_t = f(y_{t-1}, \ldots, y_{t-k}) + \epsilon \tag{10}$$
$$f \sim \mathcal{GP}(m, k) \tag{11}$$

Problem: noise in the inputs, "errors in the variables"

- Noisy Input GP, [McHutchon & Rasmussen, NIPS 2011]

GP-NARX Models

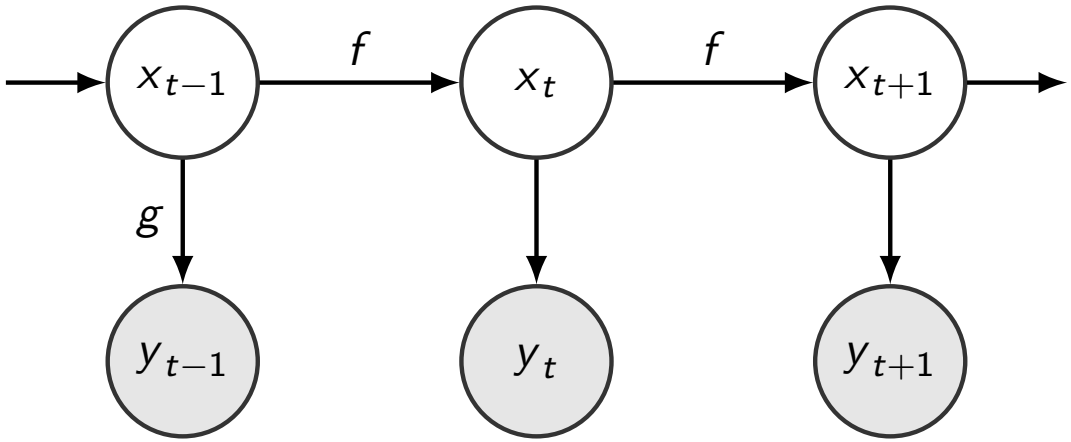
Use a GP to model the mapping,

$$y_t = f(y_{t-1}, \ldots, y_{t-k}) + \epsilon \tag{10}$$
$$f \sim \mathcal{GP}(m, k) \tag{11}$$

Problem: noise in the inputs, "errors in the variables"

- Noisy Input GP, [McHutchon & Rasmussen, NIPS 2011]
- GP-FNARX, [Frigola & Rasmussen, CDC 2013]

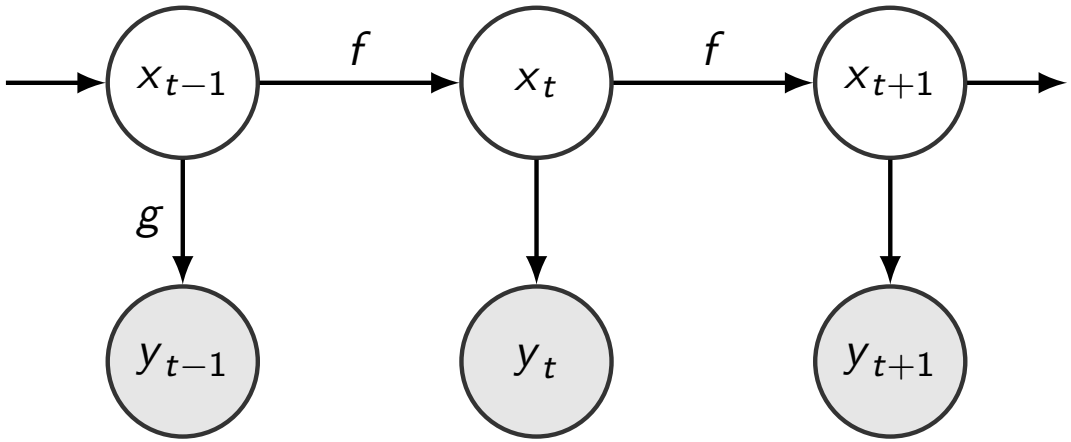
GP Latent Variable Models



GP-LVM [Lawrence, NIPS 2004], no latent state dynamics

- GP Dynamical Model, [Wang *et al.*, NIPS 2005 & PAMI 2008]
- Variational GP Dynamical System, [Damianou *et al.*, NIPS 2011]
- BayesFilter [Ko & Fox, 2009 & 2011]

GP Latent Variable Models



$$x_t = f(x_{t-1}) + \epsilon \tag{12}$$

$$y_t = g(x_t) + \nu \tag{13}$$

- $f \sim \mathcal{GP}(m, k)$

Analytic Approach

$$x_t = f(x_{t-1}) + \epsilon \tag{12}$$
$$y_t = g(x_t) + \nu \tag{13}$$

- $f \sim \mathcal{GP}(m,\ k)$
- $g = C$ or $g \sim \mathcal{GP}(m_g,\ k_g)$
- $\epsilon \sim \mathcal{N}(0, \Sigma_\epsilon)$ and $\nu \sim \mathcal{N}(0, \Sigma_\nu)$

Analytic Approach

$$x_t = f(x_{t-1}) + \epsilon \tag{12}$$
$$y_t = g(x_t) + \nu \tag{13}$$

- $f \sim \mathcal{GP}(m,\ k)$
- $g = C$ or $g \sim \mathcal{GP}(m_g,\ k_g)$
- $\epsilon \sim \mathcal{N}(0, \Sigma_\epsilon)$ and $\nu \sim \mathcal{N}(0, \Sigma_\nu)$
- Introduce a pseudo-training set $\{\tilde{x},\ \tilde{f}\}$

$$\theta = \{\tilde{x},\ \tilde{f},\ \theta_m,\ \theta_k,\ C,\ \Sigma_\epsilon,\ \Sigma_\nu\} \tag{14}$$

Analytic Approach

$$x_t = f(x_{t-1}) + \epsilon \tag{12}$$
$$y_t = g(x_t) + \nu \tag{13}$$

- $f \sim \mathcal{GP}(m,\ k)$
- $g = C$ or $g \sim \mathcal{GP}(m_g,\ k_g)$
- $\epsilon \sim \mathcal{N}(0, \Sigma_\epsilon)$ and $\nu \sim \mathcal{N}(0, \Sigma_\nu)$
- Introduce a pseudo-training set $\{\tilde{x},\ \tilde{f}\}$

$$\theta = \{\tilde{x},\ \tilde{f},\ \theta_m,\ \theta_k,\ C,\ \Sigma_\epsilon,\ \Sigma_\nu\} \tag{14}$$

- Treat f in a Bayesian way, and θ in a maximum likelihood manner

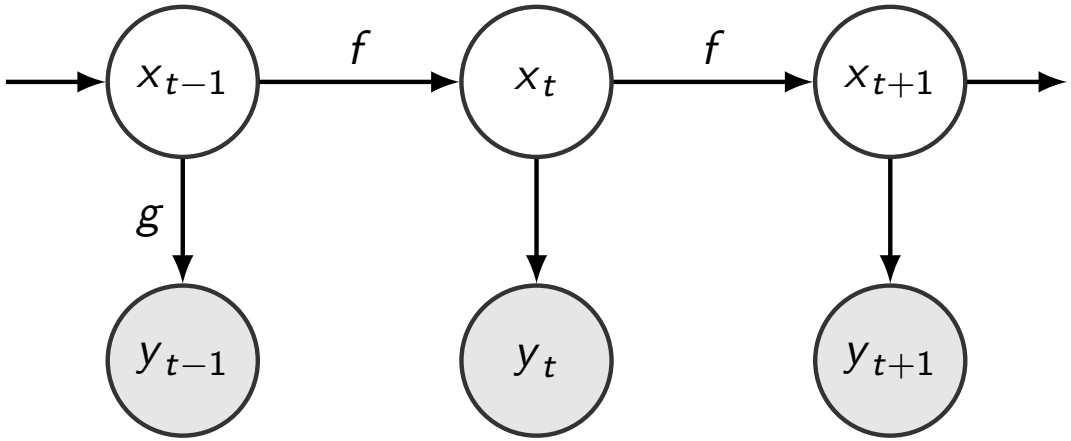
Marginal Likelihood

Probability of the data under the model averaging over our belief about the latent variables and the function f

$$P(y_1, \ldots, y_T \mid \theta) = P(y_1 \mid \theta)P(y_2 \mid y_1, \theta)P(y_3 \mid y_2, y_1, \theta)\ldots \quad (15)$$

$$= P(y_1 \mid \theta)\prod_{t=2}^{T} P(y_t \mid y_{t-1}, \ldots, y_1, \theta) \quad (16)$$

Marginal Likelihood



$$P(y_t \mid y_{t-1}, \ldots, y_1, \theta) = \int \int \underbrace{P(x_{t-1} \mid y_{t-1}, \ldots, y_1)}_{\text{filtered state at t-1}} \underbrace{P(x_t \mid x_{t-1})}_{\text{transition}} \underbrace{P(y_t \mid x_t)}_{\text{observation}} dx_t dx_{t-1} \quad (17)$$

Direct Optimisation

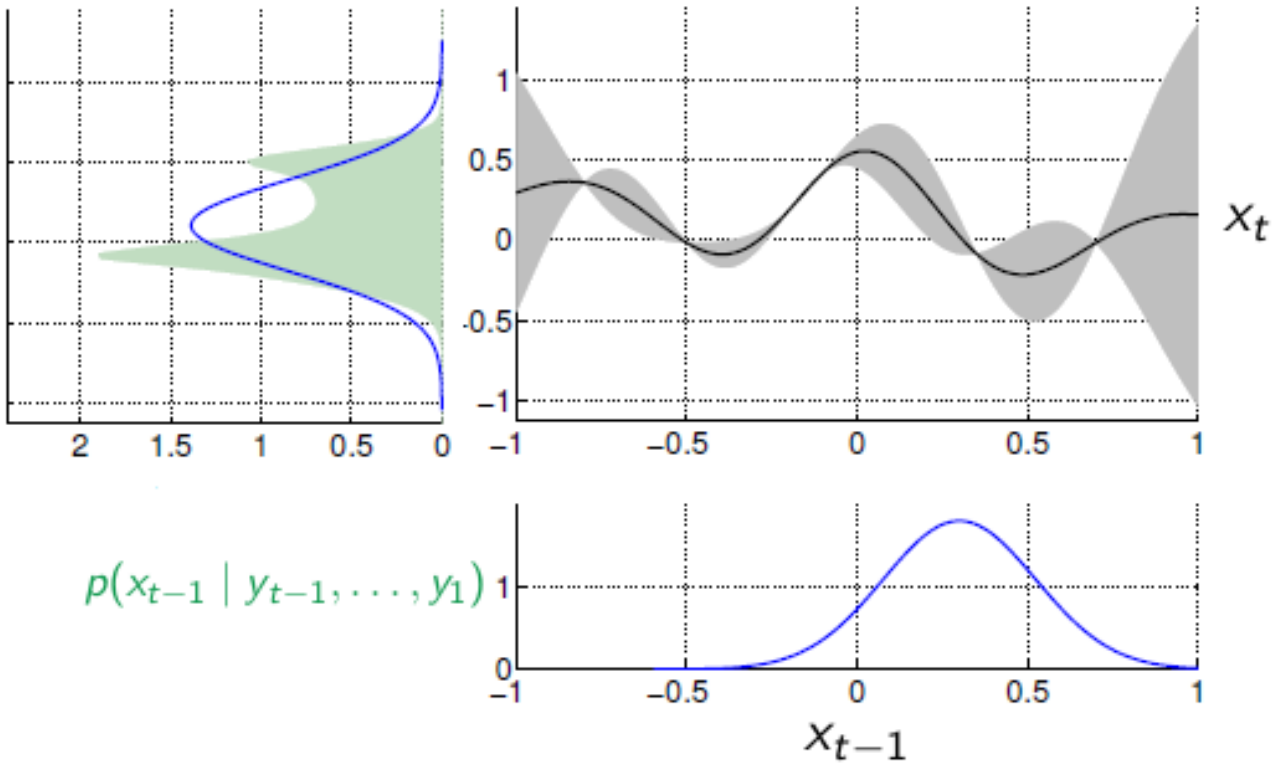
- Project filtered state distribution to a Gaussian
- Method 1:

$$\int P(y_t \mid x_t) \int P(x_{t-1} \mid y_{t-1}, \ldots, y_1)\, P(x_t \mid x_{t-1})\, dx_{t-1}\, dx_t$$

Direct Optimisation

- Project filtered state distribution to a Gaussian
- Method 1:

$$\int P(y_t \mid x_t) \int P(x_{t-1} \mid y_{t-1}, \ldots, y_1)\, P(x_t \mid x_{t-1})\, dx_{t-1}\, dx_t$$



Direct Optimisation

- Project filtered state distribution to a Gaussian
- Method 1: Project $P(x_t \mid y_{t-1}, \ldots, y_1)$ to a Gaussian

$$\int P(y_t \mid x_t) \int P(x_{t-1} \mid y_{t-1}, \ldots, y_1)\, P(x_t \mid x_{t-1})\, dx_{t-1}\, dx_t$$

- Method 2:

$$\int P(x_{t-1} \mid y_{t-1}, \ldots, y_1) \int P(x_t \mid x_{t-1})\, P(y_t \mid x_t)\, dx_t\, dx_{t-1}$$

Direct Optimisation

- Project filtered state distribution to a Gaussian
- Method 1: Project $P(x_t \mid y_{t-1}, \ldots, y_1)$ to a Gaussian

$$\int P(y_t \mid x_t) \int P(x_{t-1} \mid y_{t-1}, \ldots, y_1)\, P(x_t \mid x_{t-1})\, dx_{t-1}\, dx_t$$

- Method 2: Simple sampling

$$\int P(x_{t-1} \mid y_{t-1}, \ldots, y_1) \int P(x_t \mid x_{t-1})\, P(y_t \mid x_t)\, dx_t\, dx_{t-1}$$

- Similar steps to compute filter distribution at time t, $P(x_t \mid y_t, \ldots, y_1)$

Direct Optimisation

for t = 1:T

compute

$P(x_t \mid y_t, \ldots, y_1, \theta)$

$\log P(y_t \mid y_{t-1}, \ldots, y_1, \theta)$

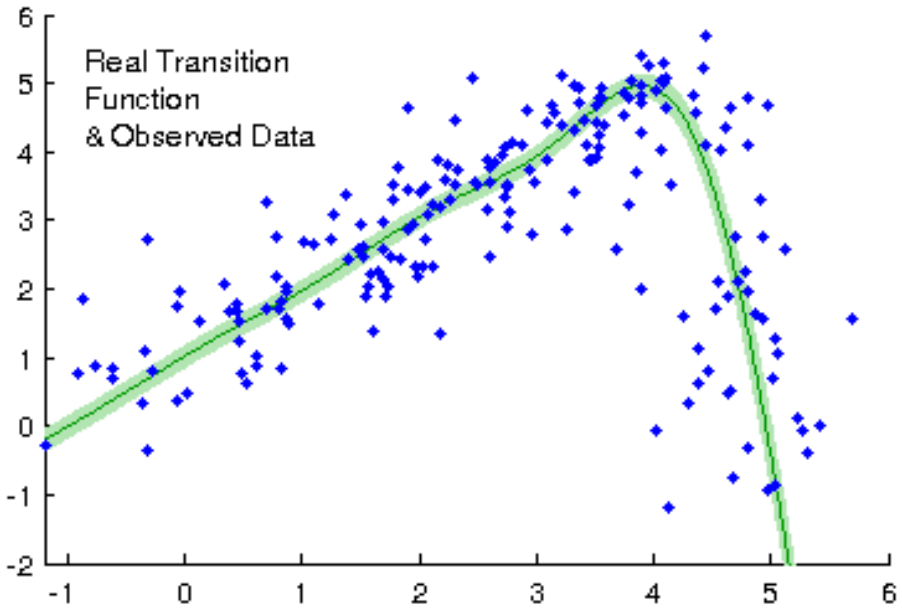
derivatives w.r.t. θ

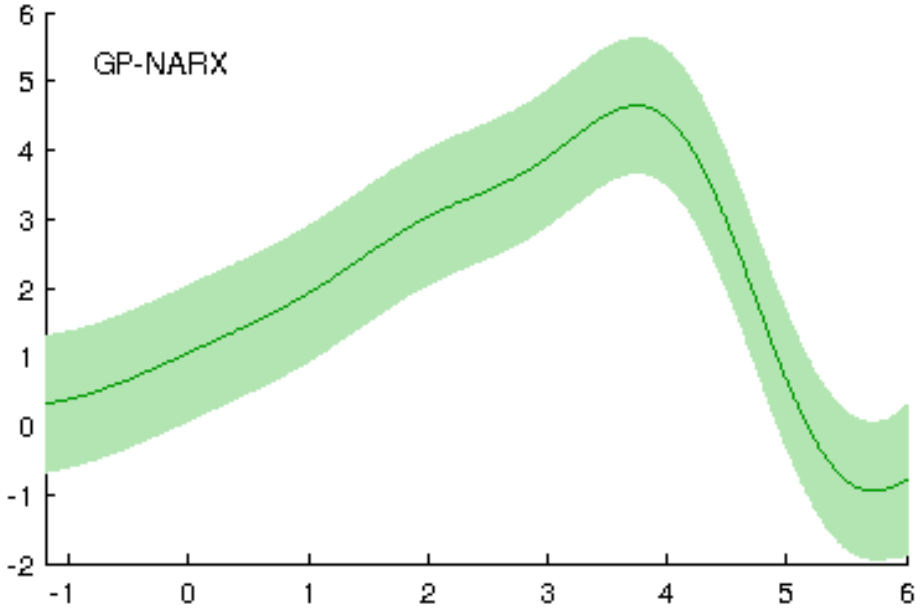
end

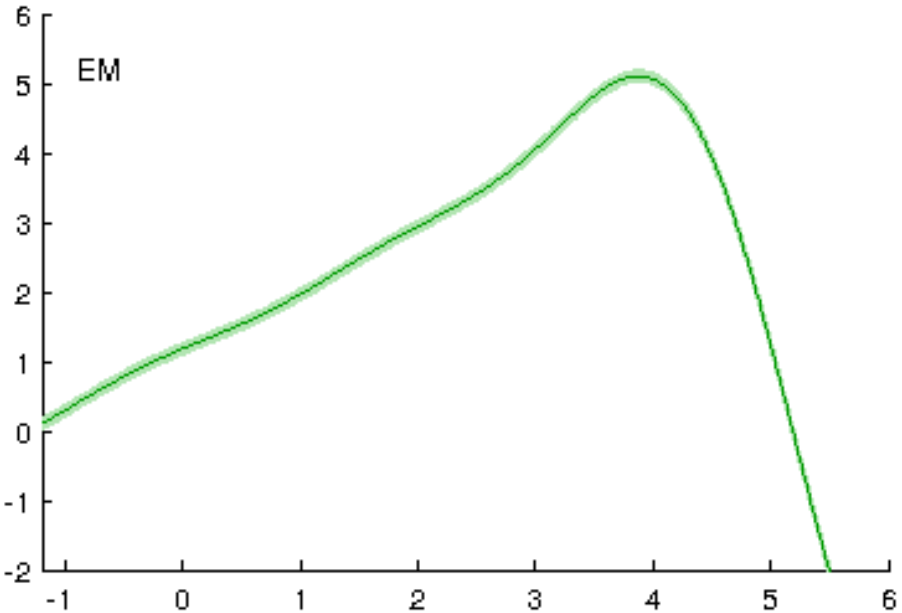
$$\log P(y_1, \ldots, y_T \mid \theta) = \sum \log P(y_t \mid y_{t-1}, \ldots, y_1, \theta)$$

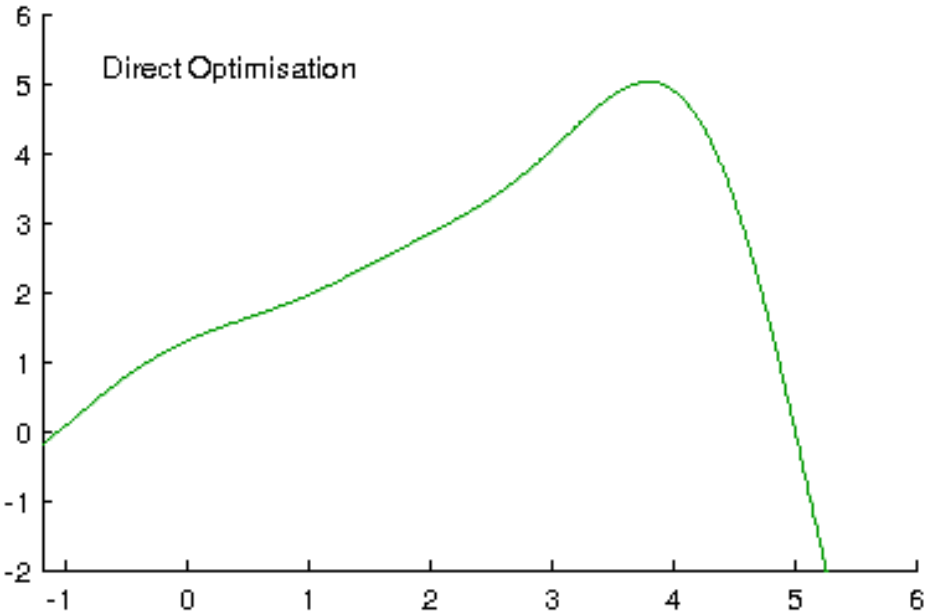
optimise w.r.t. θ

Example: 1D System

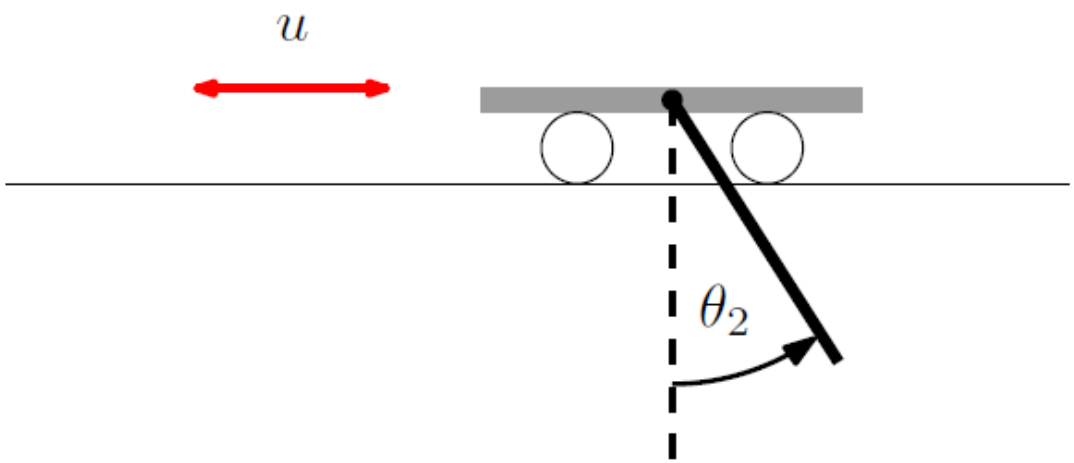






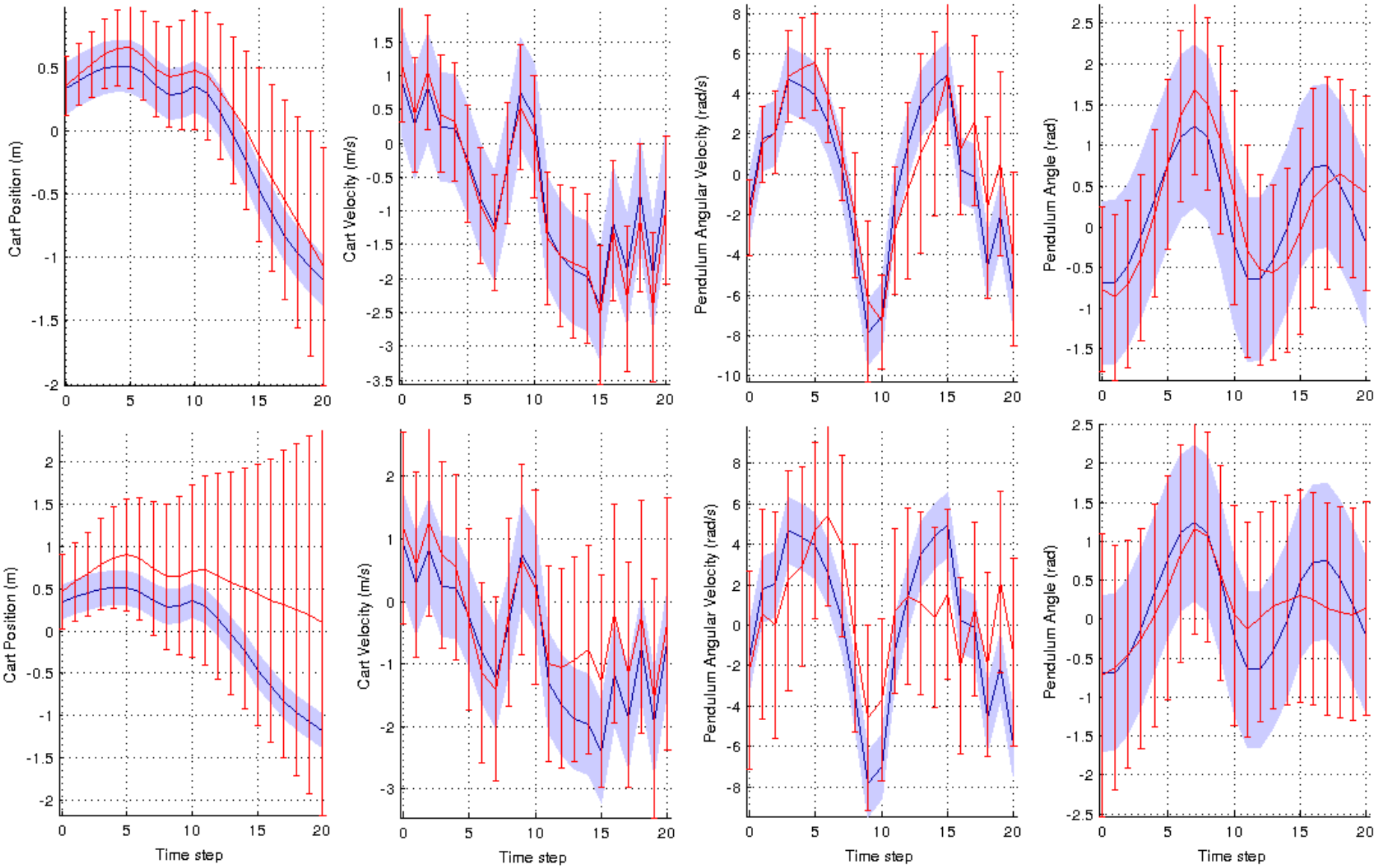


Example: Cart and Pole System



- 4 state variables
- 1 control - driven by white noise
- find distribution over future trajectory given feed-forward controls

Example: Cart and Pole System



System Identification with GP-SSMs using particle Markov Chain Monte Carlo (PMCMC) [1]

Nonlinear state-space model:

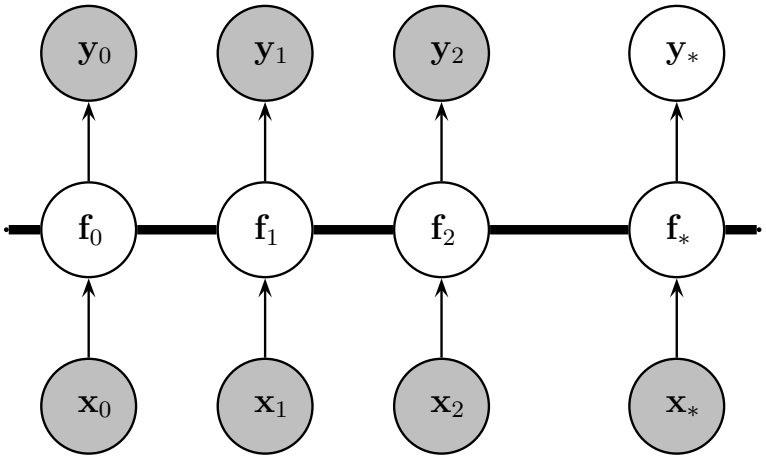
$$x_{t+1} = f(x_t) + \mathbf{v}_t,$$
$$\mathbf{y}_t = g(x_t) + \mathbf{e}_t.$$

Generative model for Gaussian process state-space models (GP-SSMs):

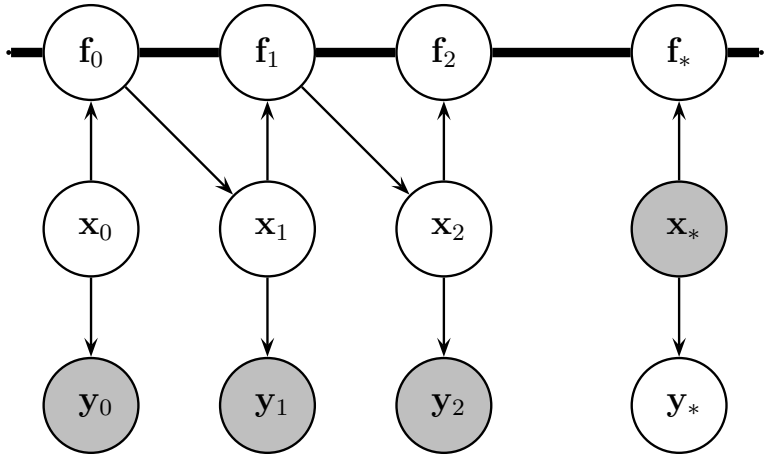
$$f(x_t) \sim \mathcal{GP}\big(m_{\theta_x}(x_t), k_{\theta_x}(x_t, x_t')\big),$$
$$x_{t+1} \mid \mathbf{f}_t \sim \mathcal{N}(x_{t+1} \mid \mathbf{f}_t, \mathbf{Q}),$$
$$\mathbf{y}_t \mid x_t \sim p(\mathbf{y}_t \mid x_t, \theta_{\mathbf{y}}).$$

[1] joint work with Fredrik Lindsten, Thomas B. Schön and Carl Rasmussen, NIPS 2013

Graphical models



Gaussian process regression, $\mathbf{y} = f(x) + \epsilon$



GP-SSM

Computing the smoothing distribution is system identification!

This is a very particular property of this model!!!

Challenge: find a good representation of

$$p(x_{0:T} \mid \theta, \mathbf{y}_{0:T}).$$

Sampling the states given the data (smoothing)

Posterior is proportional to likelihood times prior

$$p(\mathbf{x}_{0:T} \mid \mathbf{y}_{0:T}, \theta) \propto p(\mathbf{y}_{0:T} \mid \mathbf{x}_{0:T}, \theta)\ p(\mathbf{x}_{0:T} \mid \theta).$$

We use Particle Gibbs with Ancestor Sampling (PG-AS).

(cf. Thomas Schön's tutorial)

Marginalising the latent function

To do inference we need the joint distribution over latent variables

$$p(\mathbf{x}_{0:T}, \mathbf{f}_{0:T} \mid \theta)$$

But $\mathbf{x}_{0:T}$ and $\mathbf{f}_{0:T}$ are tightly coupled. We can marginalise the latent function analytically

$$p(x_{1:T} \mid \theta, x_0) = \prod_{t=1}^{T} p(x_t \mid \theta, x_{0:t-1}) = \prod_{t=1}^{T} \mathcal{N}\big(x_t \mid \mu_t(x_{0:t-1}), \Sigma_t(x_{0:t-1})\big).$$

where μ_t and Σ_t have analogous expressions to those in GP regression.

What about the hyper-parameters?

We have shown how PMCMC can give us samples from

$$p(x_{0:T} \mid \theta, \mathbf{y}_{0:T}).$$

We can alternate this with sampling from

$$p(\theta \mid x_{0:T}, \mathbf{y}_{0:T})$$

using slice sampling.

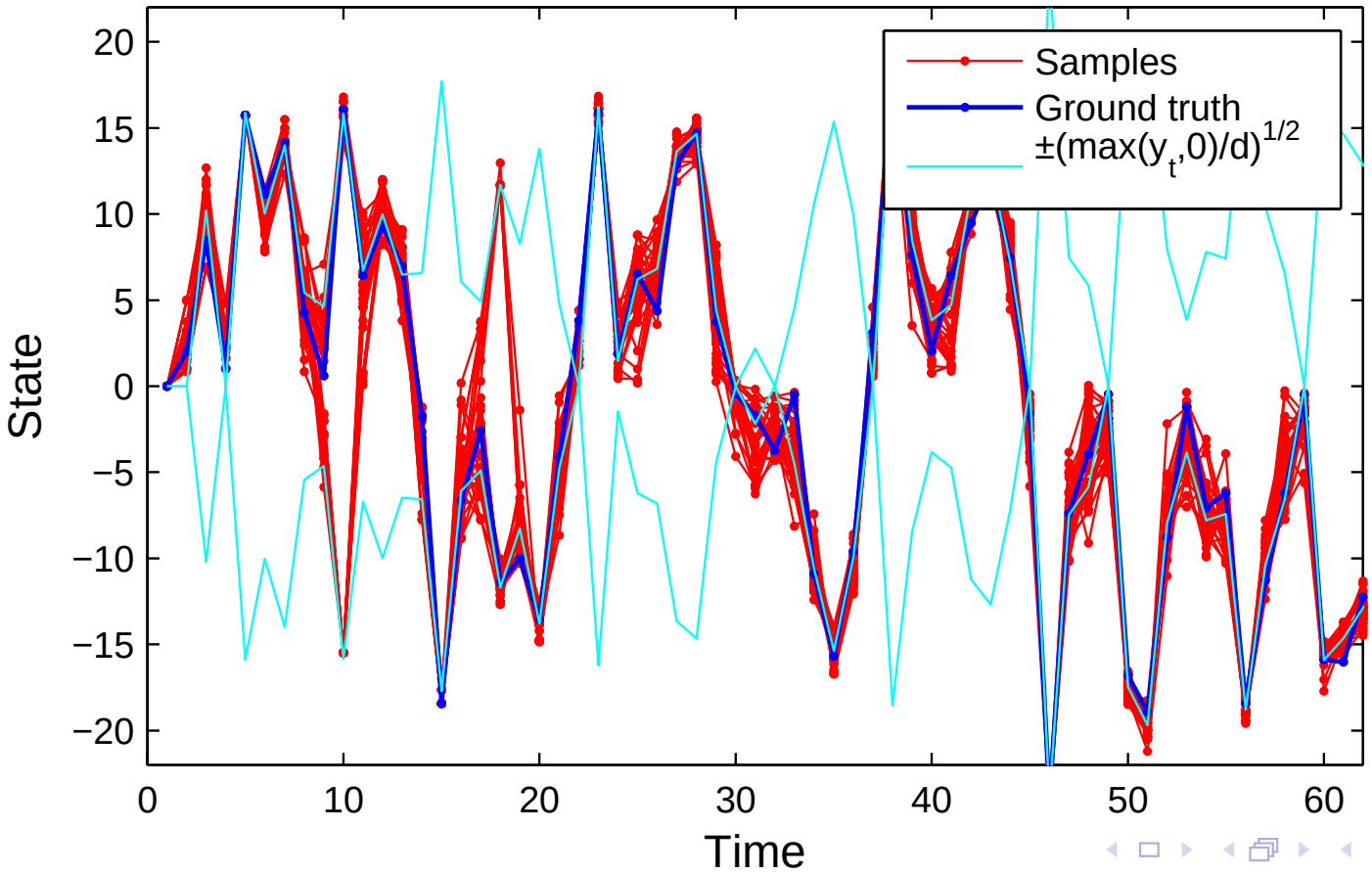
This is a (blocked) Gibbs sampler which ultimately targets

$$p(x_{0:T}, \theta \mid \mathbf{y}_{0:T}).$$

Experiments: 1-dim Benchmark System

$$x_{t+1} = ax_t + b\frac{x_t}{1+x_t^2} + cu_t + v_t, \quad v_t \sim \mathcal{N}(0, q),$$

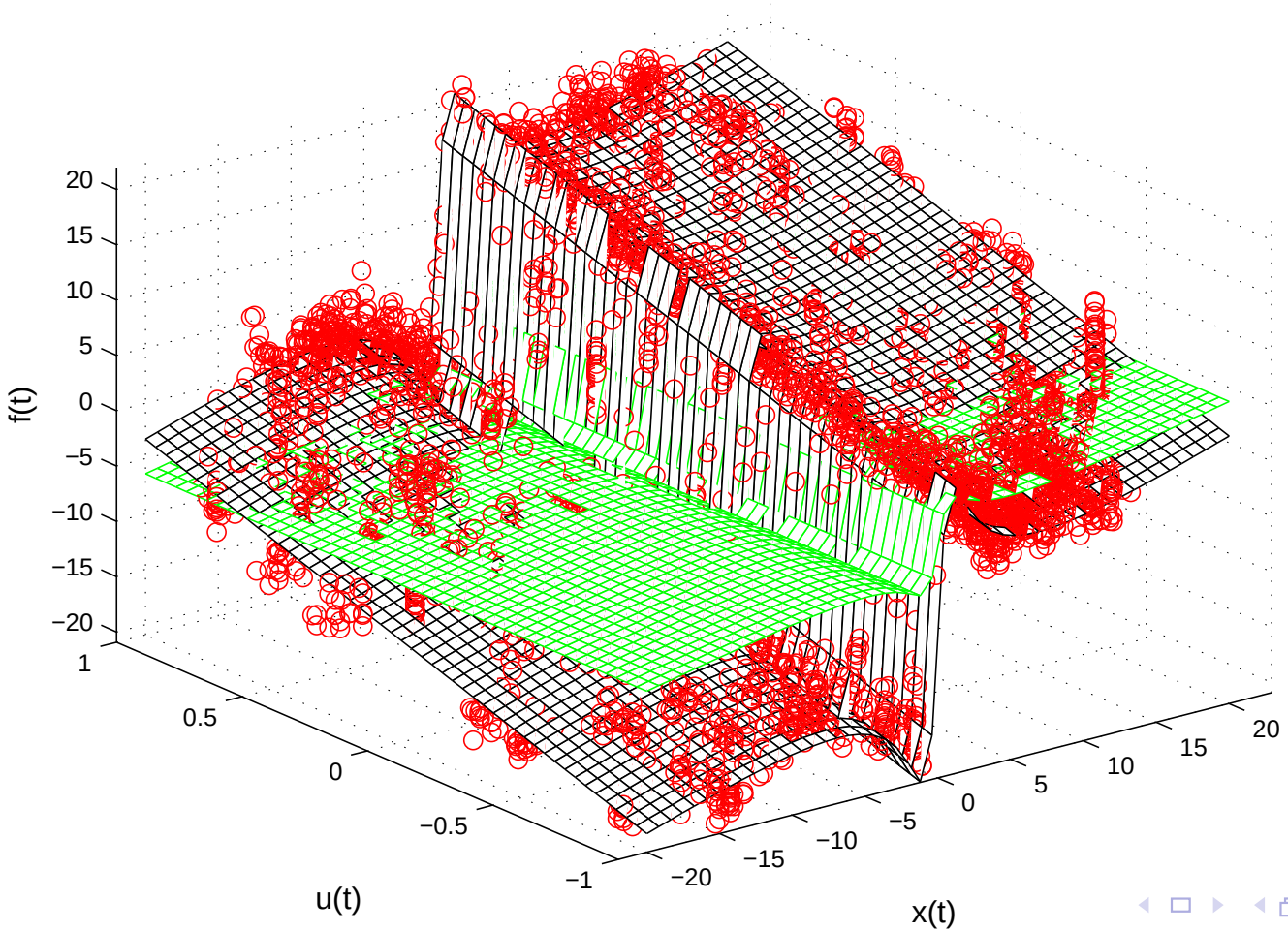
$$y_t = dx_t^2 + e_t, \quad e_t \sim \mathcal{N}(0, r).$$



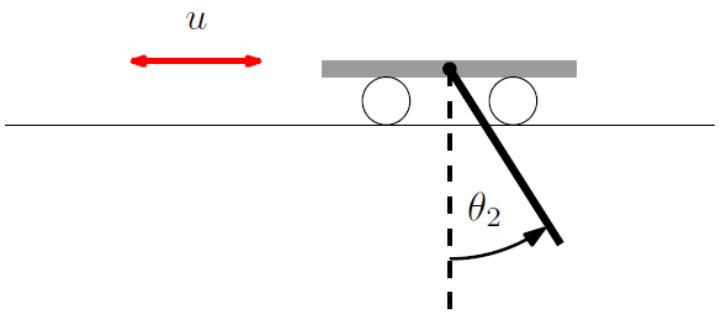
Experiments: 1-dim Benchmark System

$$x_{t+1} = ax_t + b\frac{x_t}{1+x_t^2} + cu_t + v_t, \quad v_t \sim \mathcal{N}(0, q),$$

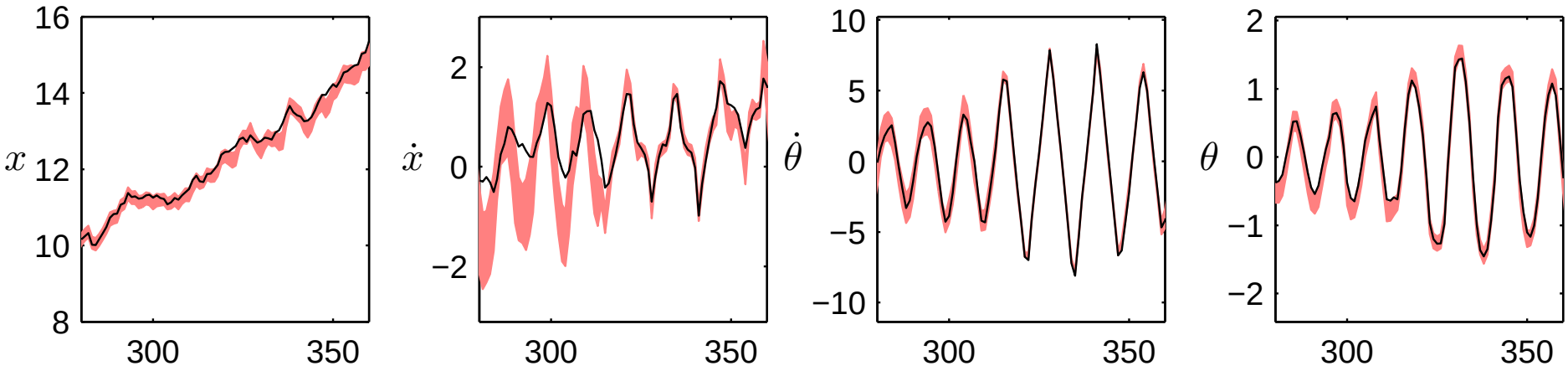
$$y_t = dx_t^2 + e_t, \quad e_t \sim \mathcal{N}(0, r).$$



Experiments: 4-dim Cart and Pole System



One step ahead predictions with error bars



Take-home Messages

Bayesian inference:

- Quantifies uncertainty.
- Uses this uncertainty to make predictions/decisions.

Bayesian nonparametrics:

- Highly flexible - doesn't exclude complex explanations a priori.