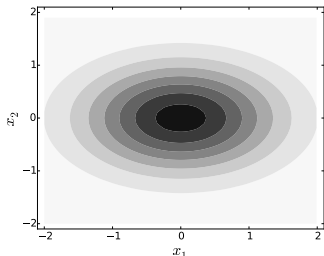
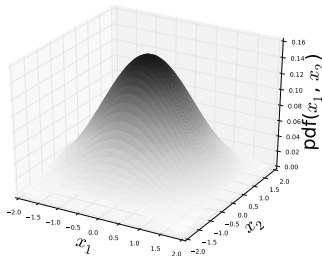


# A Gentle Introduction to Gaussian Processes

John Joseph Valletta

University of Exeter, Penryn Campus, UK

Internal Maths Seminar: 20th October 2015



- Motivation
- The Gaussian Distribution
- Gaussian Process
- Gaussian Process Regression - A Toy Example
- Gaussian Process Regression - CO<sub>2</sub> Concentrations
- Modelling Gene Expression Time-Series

- Motivation
- The Gaussian Distribution
- Gaussian Process
- Gaussian Process Regression - A Toy Example
- Gaussian Process Regression - CO<sub>2</sub> Concentrations
- Modelling Gene Expression Time-Series

- Motivation
- The Gaussian Distribution
- Gaussian Process
- Gaussian Process Regression - A Toy Example
- Gaussian Process Regression - CO<sub>2</sub> Concentrations
- Modelling Gene Expression Time-Series

- Motivation
- The Gaussian Distribution
- Gaussian Process
- Gaussian Process Regression - A Toy Example
- Gaussian Process Regression - CO<sub>2</sub> Concentrations
- Modelling Gene Expression Time-Series

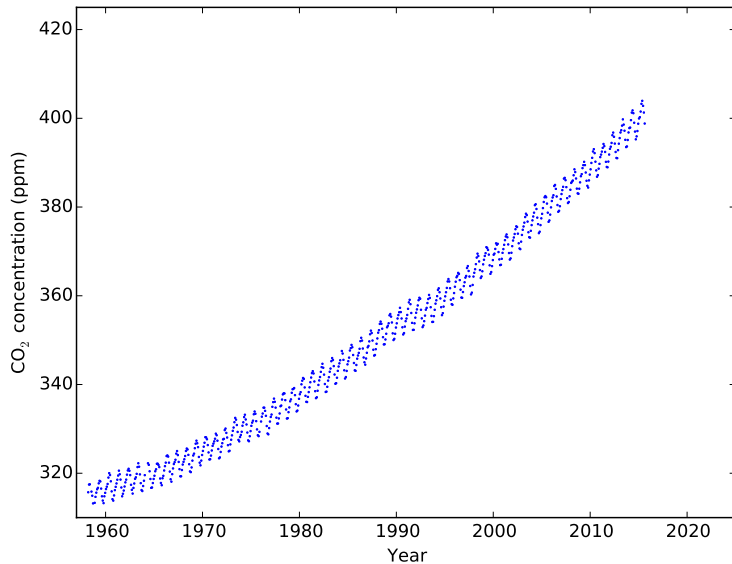
- Motivation
- The Gaussian Distribution
- Gaussian Process
- Gaussian Process Regression - A Toy Example
- Gaussian Process Regression - CO<sub>2</sub> Concentrations
- Modelling Gene Expression Time-Series

- Motivation
- The Gaussian Distribution
- Gaussian Process
- Gaussian Process Regression - A Toy Example
- Gaussian Process Regression - CO<sub>2</sub> Concentrations
- Modelling Gene Expression Time-Series

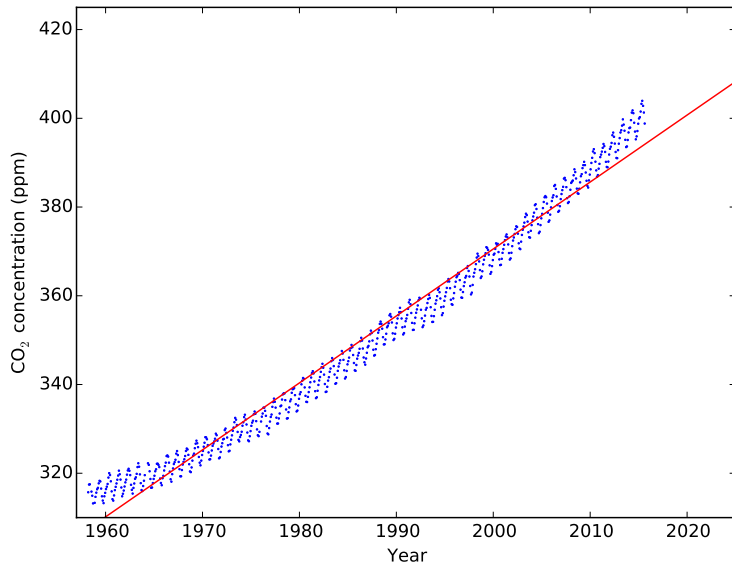
- Motivation
- The Gaussian Distribution
- Gaussian Process
- Gaussian Process Regression - A Toy Example
- Gaussian Process Regression - CO<sub>2</sub> Concentrations
- Modelling Gene Expression Time-Series



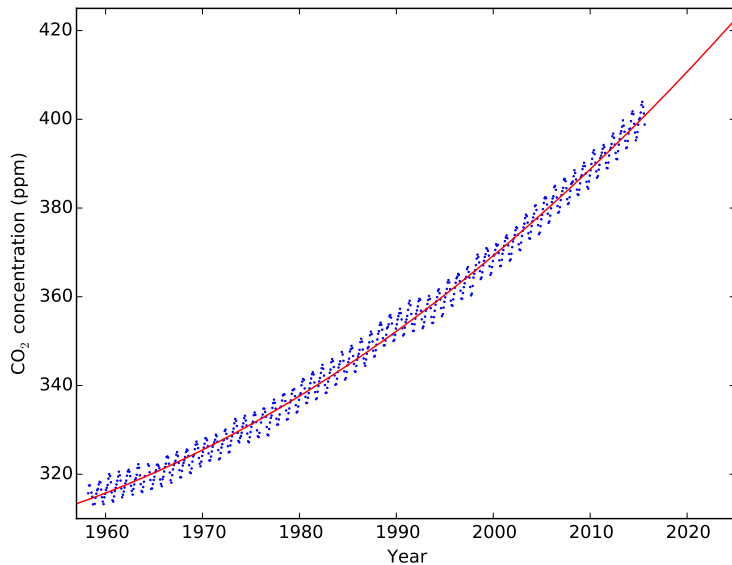
# Motivation



# Motivation



# Motivation

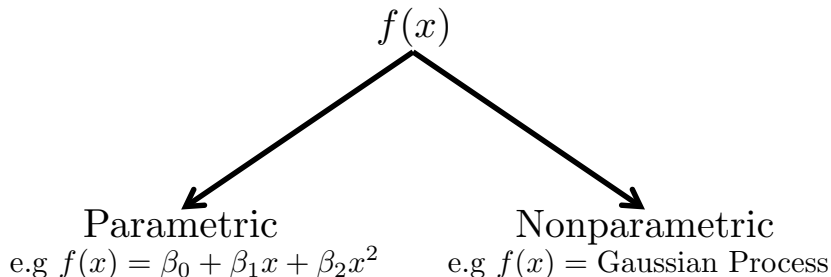


# The Data Modelling Task

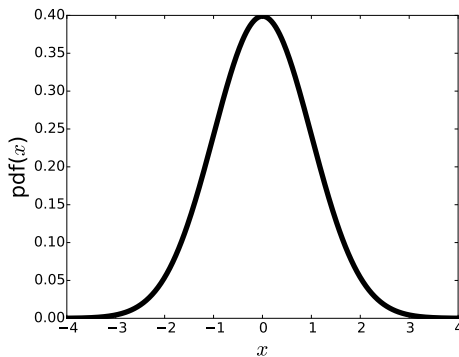
- **data:**  $\mathbf{x} = \{x_1, \dots, x_N\}$ ,  $\mathbf{y} = \{y_1, \dots, y_N\}$
- **model:**  $y = f(x) + \epsilon$
- **predictions:**  $y^* = f(x^*)$

# The Data Modelling Task

- **data:**  $\mathbf{x} = \{x_1, \dots, x_N\}$ ,  $\mathbf{y} = \{y_1, \dots, y_N\}$
- **model:**  $y = f(x) + \epsilon$
- **predictions:**  $y^* = f(x^*)$

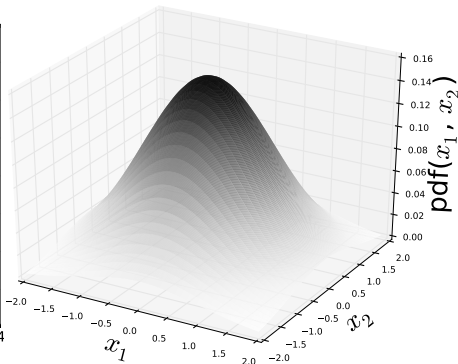


# The Gaussian Distribution



$$\mathcal{N}(\mu, \sigma^2)$$

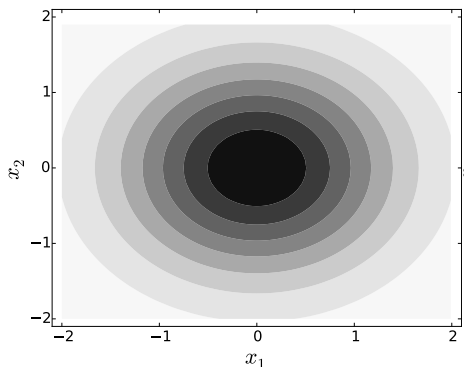
A draw from this distribution is a 1D vector  
e.g  $x = [0.2]$



$$\mathcal{N}(\mu, \Sigma)$$

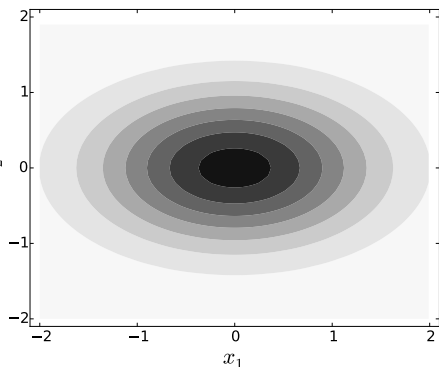
A draw from this distribution is a 2D vector  
e.g  $\mathbf{x} = \begin{bmatrix} 0.3 \\ -0.4 \end{bmatrix}$

# The Covariance Matrix



**Isotropic**

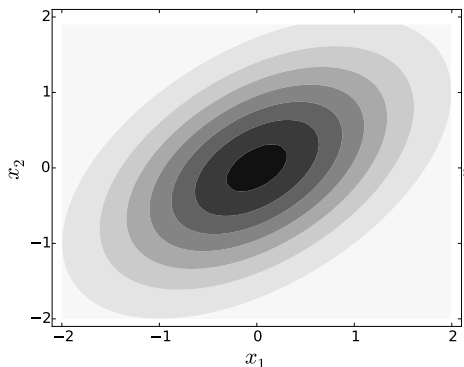
$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



**Diagonal**

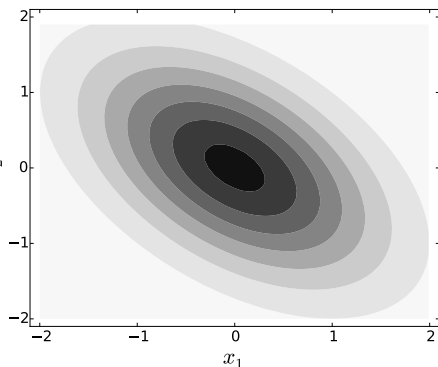
$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix}$$

# The Covariance Matrix



**General Form**

$$\Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$



**General Form**

$$\Sigma = \begin{pmatrix} 1 & -0.5 \\ -0.5 & 1 \end{pmatrix}$$



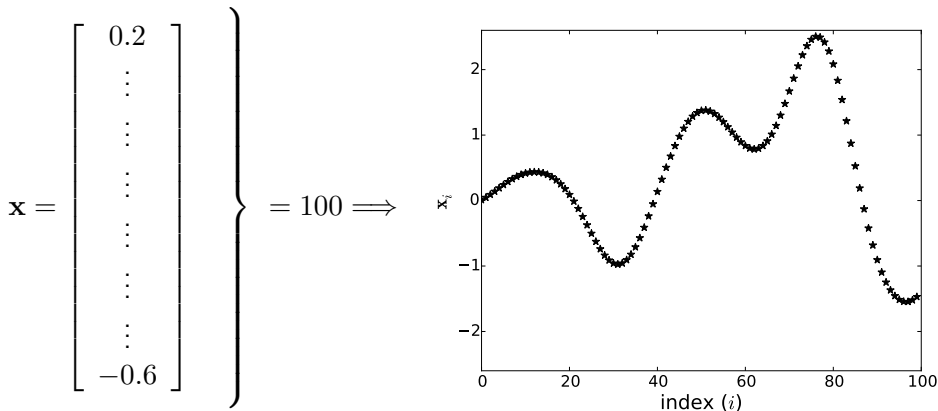
# Sampling from a Multivariate Gaussian

What does a *single* sample from a 100 dimensional Gaussian look like?

$$\mathbf{x} = \left[ \begin{array}{c} 0.2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ -0.6 \end{array} \right] \Bigg\} = 100 \Rightarrow$$

# Sampling from a Multivariate Gaussian

What does a *single* sample from a 100 dimensional Gaussian look like?



# Gaussian Process in a Nutshell

**Recall:** What we are after is  $y = f(x)$

**Trick:** Think about a function as an infinitely-long vector

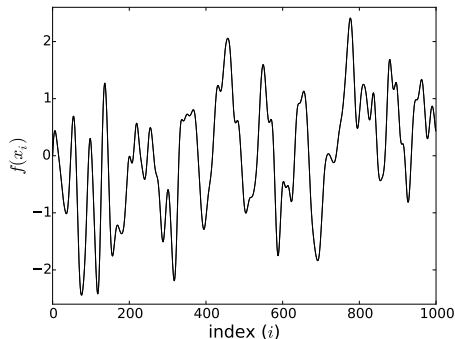
$$f(x) = \left[ \begin{array}{c} f(x_1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f(x_\infty) \end{array} \right] = \infty \Rightarrow$$

# Gaussian Process in a Nutshell

**Recall:** What we are after is  $y = f(x)$

**Trick:** Think about a function as an infinitely-long vector

$$f(x) = \left[ \begin{array}{c} f(x_1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f(x_\infty) \end{array} \right] \Bigg\} = \infty \Rightarrow$$

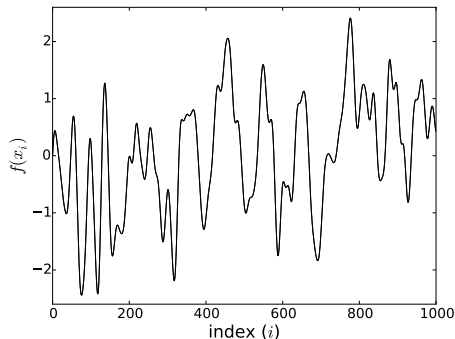


# Gaussian Process in a Nutshell

**Recall:** What we are after is  $y = f(x)$

**Trick:** Think about a function as an infinitely-long vector

$$f(x) = \left[ \begin{array}{c} f(x_1) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f(x_\infty) \end{array} \right] = \infty \Rightarrow$$



**Computational Madness:** Ask only for the properties of the function at a *finite* number of points

# Gaussian Process in a Nutshell

## 3 dimensional Gaussian

Mean vector

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix}$$

Covariance matrix

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_1\sigma_2 & \sigma_1\sigma_3 \\ \sigma_2\sigma_1 & \sigma_2^2 & \sigma_2\sigma_3 \\ \sigma_3\sigma_1 & \sigma_3\sigma_2 & \sigma_3^2 \end{pmatrix}$$

## $\infty$ dimensional Gaussian

Mean *function*

$$\mu = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_\infty \end{pmatrix} = m(\mathbf{x})$$

Covariance *function*

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_1\sigma_2 & \dots & \sigma_1\sigma_\infty \\ \sigma_2\sigma_1 & \sigma_2^2 & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \dots & \sigma_\infty^2 \end{pmatrix} = k(\mathbf{x}, \mathbf{x}')$$

## Definition

A Gaussian Process (GP) is an infinite collection of random variables, any finite number of which have a joint normal distribution.

Essentially an infinite dimension multivariate Gaussian distribution, characterised by a mean function  $m(\mathbf{x})$  and a covariance function  $k(\mathbf{x}, \mathbf{x}')$

## Rationale

Instead of inferring the parameters of a fixed model structure  $(\beta_0, \beta_1, \dots)$ , with GPs we model the *correlation* between inputs. That is, inputs  $(\mathbf{x})$  that are close/similar to each other are likely to give rise to a similar output  $(f(\mathbf{x}))$

$$f(\mathbf{x}) \sim \text{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$$

# Covariance Function

- Vital ingredient in Gaussian Process<sup>1</sup>
  - Encodes our assumptions about the function we wish to model (smooth, stationary, etc.)
  - Quantifies the *similarity* between two data points; crucial for predicting a test point  $\mathbf{x}^*$
  - Needs to satisfy a set of mathematical conditions (beyond the scope of this intro)
  - A very popular choice is the Squared Exponential:  
(also known as RBF, Gaussian and Exponentiated Quadratic Kernel Function)

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp \left( -\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2} \right)$$

e.g if we set  $\alpha = 1$  and  $l = 1$  then:

$$k(0, 0) = e^0 = 1, \quad k(0, 1) = e^{-\frac{1}{2}} = 0.6, \quad k(0, 2) = e^{-2} = 0.14$$

---

<sup>1</sup>without much loss of generality we can assume that  $m(\mathbf{x}) \equiv 0$



# Covariance Function

- Vital ingredient in Gaussian Process<sup>1</sup>
- Encodes our assumptions about the function we wish to model (smooth, stationary, etc.)
- Quantifies the *similarity* between two data points; crucial for predicting a test point  $\mathbf{x}^*$
- Needs to satisfy a set of mathematical conditions (beyond the scope of this intro)
- A very popular choice is the Squared Exponential:  
(also known as RBF, Gaussian and Exponentiated Quadratic Kernel Function)

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp \left( -\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2} \right)$$

e.g if we set  $\alpha = 1$  and  $l = 1$  then:

$$k(0, 0) = e^0 = 1, \quad k(0, 1) = e^{-\frac{1}{2}} = 0.6, \quad k(0, 2) = e^{-2} = 0.14$$

---

<sup>1</sup>without much loss of generality we can assume that  $m(\mathbf{x}) \equiv 0$

# Covariance Function

- Vital ingredient in Gaussian Process<sup>1</sup>
- Encodes our assumptions about the function we wish to model (smooth, stationary, etc.)
- Quantifies the *similarity* between two data points; crucial for predicting a test point  $\mathbf{x}^*$
- Needs to satisfy a set of mathematical conditions (beyond the scope of this intro)
- A very popular choice is the Squared Exponential:  
(also known as RBF, Gaussian and Exponentiated Quadratic Kernel Function)

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2}\right)$$

e.g if we set  $\alpha = 1$  and  $l = 1$  then:

$$k(0, 0) = e^0 = 1, \quad k(0, 1) = e^{-\frac{1}{2}} = 0.6, \quad k(0, 2) = e^{-2} = 0.14$$

---

<sup>1</sup>without much loss of generality we can assume that  $m(\mathbf{x}) \equiv 0$

# Covariance Function

- Vital ingredient in Gaussian Process<sup>1</sup>
- Encodes our assumptions about the function we wish to model (smooth, stationary, etc.)
- Quantifies the *similarity* between two data points; crucial for predicting a test point  $\mathbf{x}^*$
- Needs to satisfy a set of mathematical conditions (beyond the scope of this intro)
- A very popular choice is the Squared Exponential:  
(also known as RBF, Gaussian and Exponentiated Quadratic Kernel Function)

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp \left( -\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2} \right)$$

e.g if we set  $\alpha = 1$  and  $l = 1$  then:

$$k(0, 0) = e^0 = 1, \quad k(0, 1) = e^{-\frac{1}{2}} = 0.6, \quad k(0, 2) = e^{-2} = 0.14$$

---

<sup>1</sup>without much loss of generality we can assume that  $m(\mathbf{x}) \equiv 0$

# Covariance Function

- Vital ingredient in Gaussian Process<sup>1</sup>
- Encodes our assumptions about the function we wish to model (smooth, stationary, etc.)
- Quantifies the *similarity* between two data points; crucial for predicting a test point  $\mathbf{x}^*$
- Needs to satisfy a set of mathematical conditions (beyond the scope of this intro)
- A very popular choice is the Squared Exponential:  
(also known as RBF, Gaussian and Exponentiated Quadratic Kernel Function)

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp \left( -\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2} \right)$$

e.g if we set  $\alpha = 1$  and  $l = 1$  then:

$$k(0, 0) = e^0 = 1, \quad k(0, 1) = e^{-\frac{1}{2}} = 0.6, \quad k(0, 2) = e^{-2} = 0.14$$

---

<sup>1</sup>without much loss of generality we can assume that  $m(\mathbf{x}) \equiv 0$

# Gaussian Process Regression

- Shift the problem of inferring model parameters to choosing covariance function and its (hyper)parameters
- Choose  $m(\mathbf{x})$  and  $k(\mathbf{x}, \mathbf{x}')$  that reflect some prior belief
- This defines a *prior* on the function class itself; it is a prior on a *function* and not parameters of some fixed model structure
- Under a Bayesian framework this prior is “reshaped” by the observed data to obtain a posterior distribution on the *function*

# Gaussian Process Regression

- Shift the problem of inferring model parameters to choosing covariance function and its (hyper)parameters
- Choose  $m(\mathbf{x})$  and  $k(\mathbf{x}, \mathbf{x}')$  that reflect some prior belief
- This defines a *prior* on the function class itself; it is a prior on a *function* and not parameters of some fixed model structure
- Under a Bayesian framework this prior is “reshaped” by the observed data to obtain a posterior distribution on the *function*

# Gaussian Process Regression

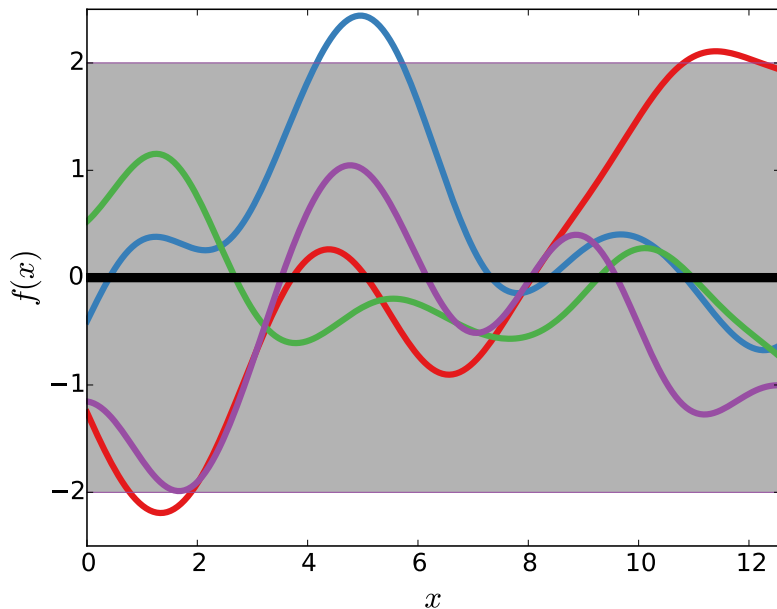
- Shift the problem of inferring model parameters to choosing covariance function and its (hyper)parameters
- Choose  $m(\mathbf{x})$  and  $k(\mathbf{x}, \mathbf{x}')$  that reflect some prior belief
- This defines a *prior* on the function class itself; it is a prior on a *function* and not parameters of some fixed model structure
- Under a Bayesian framework this prior is “reshaped” by the observed data to obtain a posterior distribution on the *function*

# Gaussian Process Regression

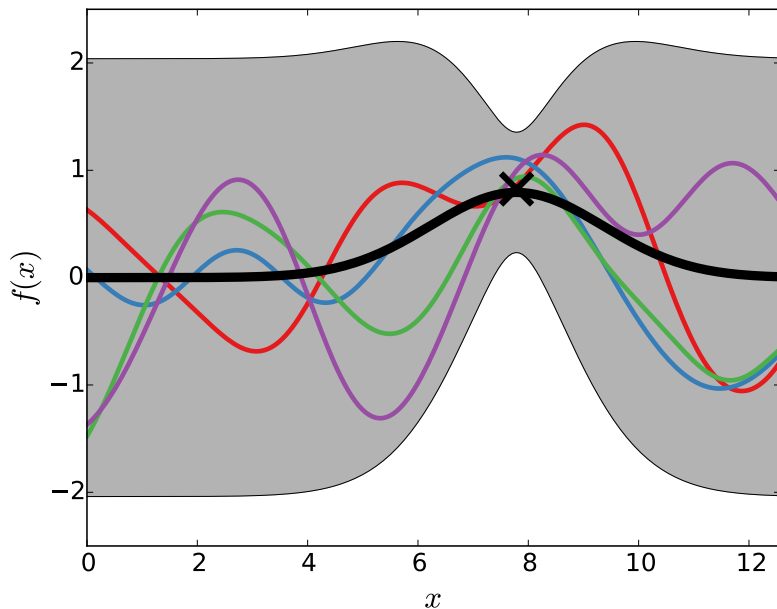
- Shift the problem of inferring model parameters to choosing covariance function and its (hyper)parameters
- Choose  $m(\mathbf{x})$  and  $k(\mathbf{x}, \mathbf{x}')$  that reflect some prior belief
- This defines a *prior* on the function class itself; it is a prior on a *function* and not parameters of some fixed model structure
- Under a Bayesian framework this prior is “reshaped” by the observed data to obtain a posterior distribution on the *function*



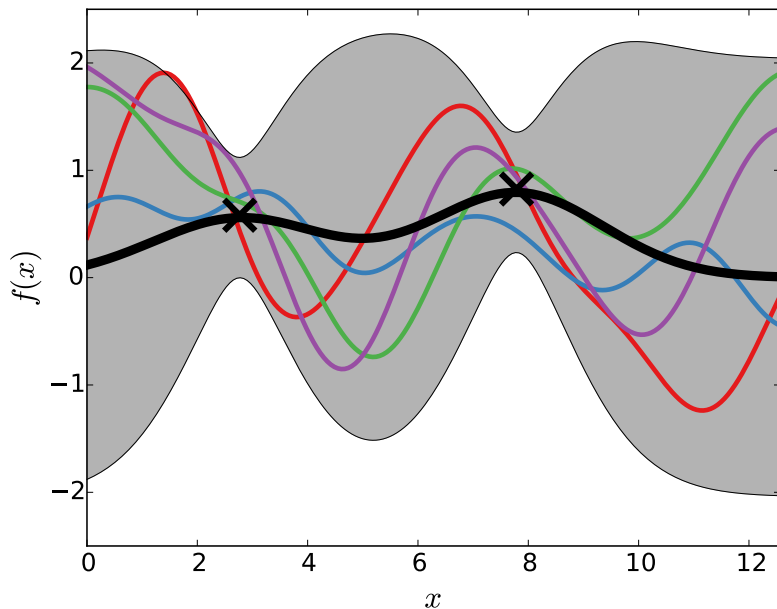
# Gaussian Process Regression - A Toy Example



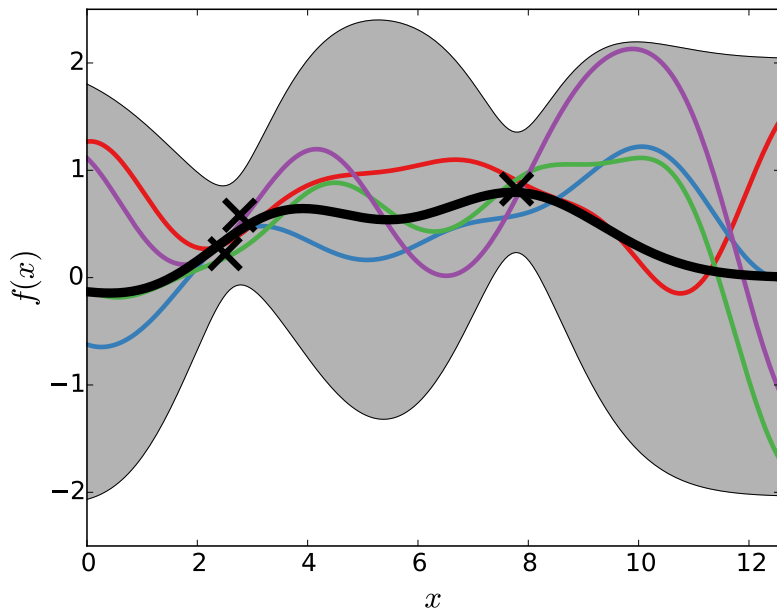
# Gaussian Process Regression - A Toy Example



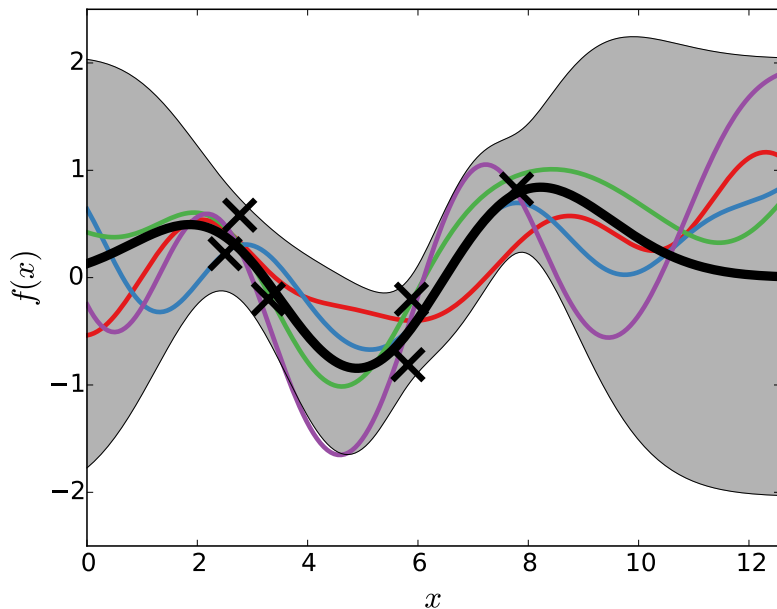
# Gaussian Process Regression - A Toy Example



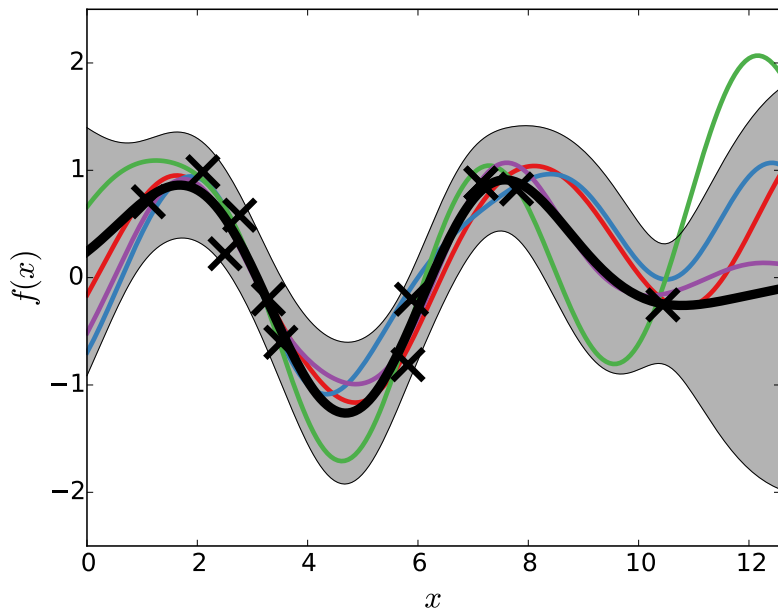
# Gaussian Process Regression - A Toy Example



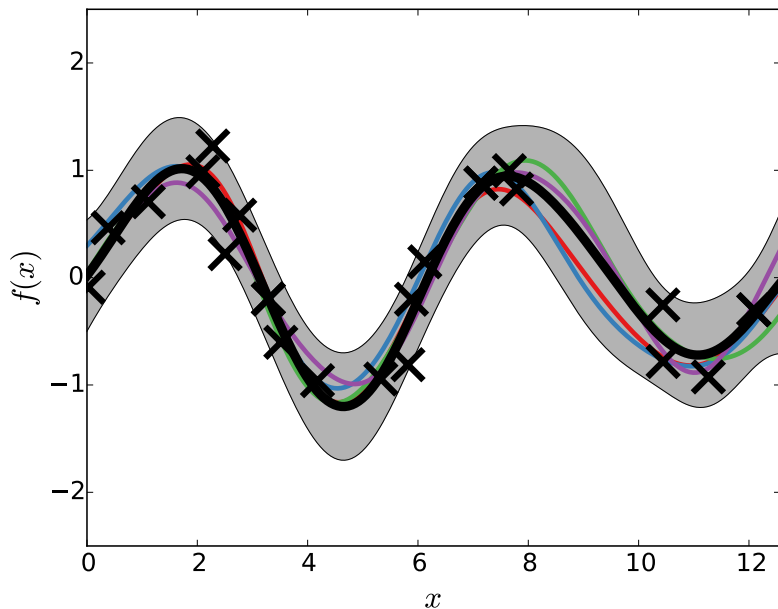
# Gaussian Process Regression - A Toy Example



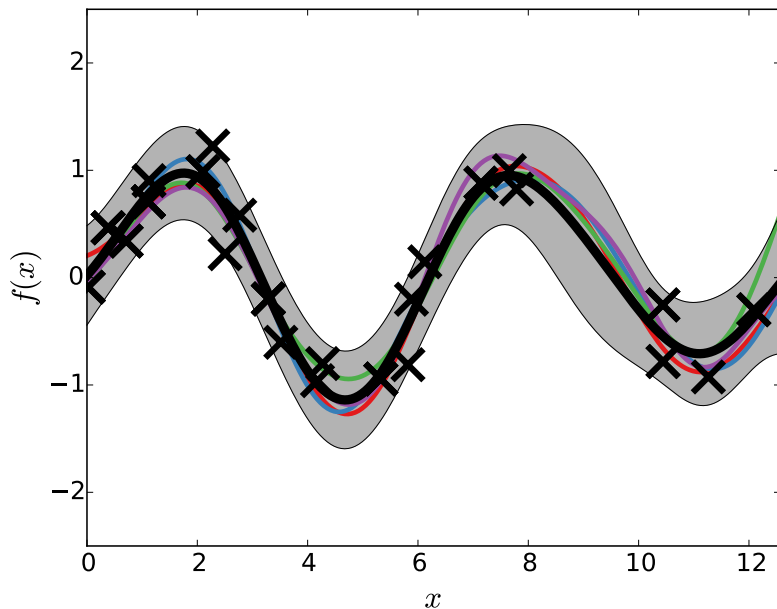
# Gaussian Process Regression - A Toy Example



# Gaussian Process Regression - A Toy Example



# Gaussian Process Regression - A Toy Example





# Misspecifying the Covariance Function

- The Squared Exponential covariance function was used in the previous example:

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp \left( \frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2} \right)$$

- Choosing a covariance functions is akin to choosing a model structure; it dictates the class of functions that can be represented by the Gaussian Process<sup>2</sup>
- Misspecifying the covariance function and/or its (hyper)parameters has a detrimental effect on the model fit
- For the squared exponential case the lengthscale  $l$  dictates how much the function is allowed to bend

---

<sup>2</sup>this class of functions however is typically wider than that offered by a parametric model

# Misspecifying the Covariance Function

- The Squared Exponential covariance function was used in the previous example:

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp \left( -\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2} \right)$$

- Choosing a covariance functions is akin to choosing a model structure; it dictates the class of functions that can be represented by the Gaussian Process<sup>2</sup>
- Misspecifying the covariance function and/or its (hyper)parameters has a detrimental effect on the model fit
- For the squared exponential case the lengthscale  $l$  dictates how much the function is allowed to bend

---

<sup>2</sup>this class of functions however is typically wider than that offered by a parametric model

# Misspecifying the Covariance Function

- The Squared Exponential covariance function was used in the previous example:

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp \left( \frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2} \right)$$

- Choosing a covariance functions is akin to choosing a model structure; it dictates the class of functions that can be represented by the Gaussian Process<sup>2</sup>
- Misspecifying the covariance function and/or its (hyper)parameters has a detrimental effect on the model fit
- For the squared exponential case the lengthscale  $l$  dictates how much the function is allowed to bend

---

<sup>2</sup>this class of functions however is typically wider than that offered by a parametric model

# Misspecifying the Covariance Function

- The Squared Exponential covariance function was used in the previous example:

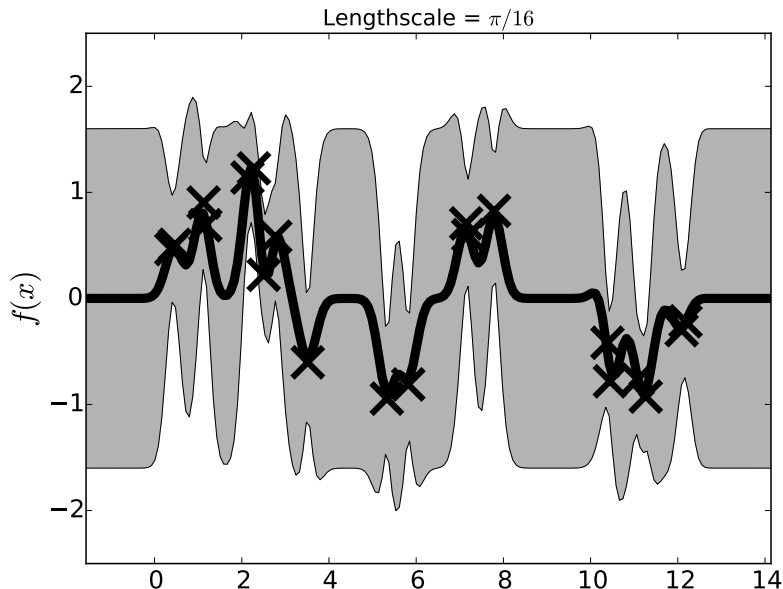
$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp \left( -\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2} \right)$$

- Choosing a covariance functions is akin to choosing a model structure; it dictates the class of functions that can be represented by the Gaussian Process<sup>2</sup>
- Misspecifying the covariance function and/or its (hyper)parameters has a detrimental effect on the model fit
- For the squared exponential case the lengthscale  $l$  dictates how much the function is allowed to bend

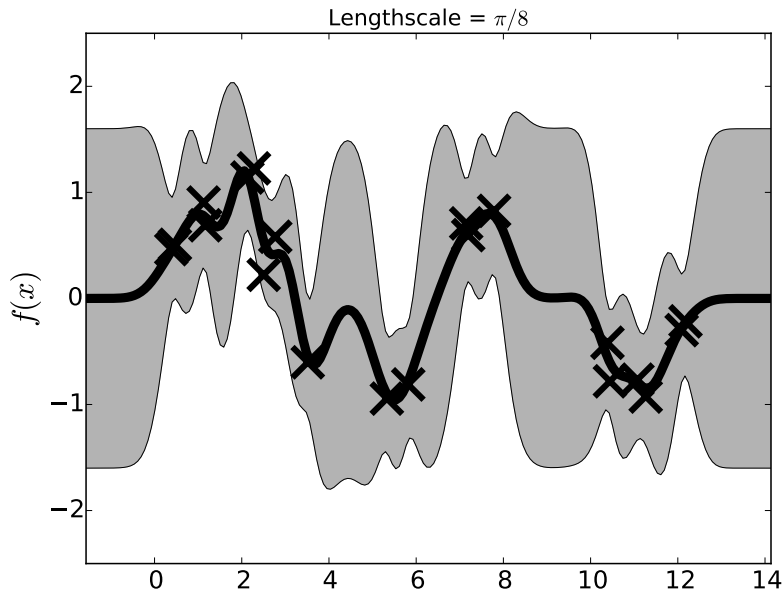
---

<sup>2</sup>this class of functions however is typically wider than that offered by a parametric model

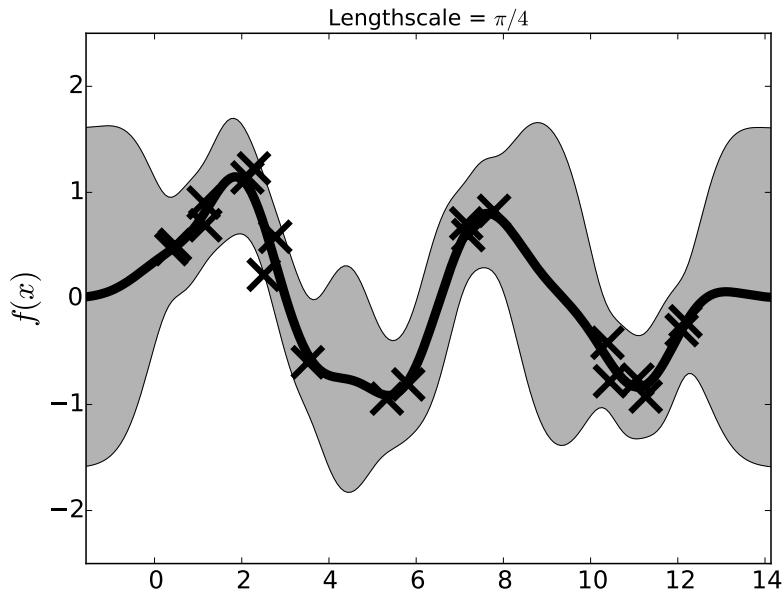
# The Effect of Lengthscale on Model Fit



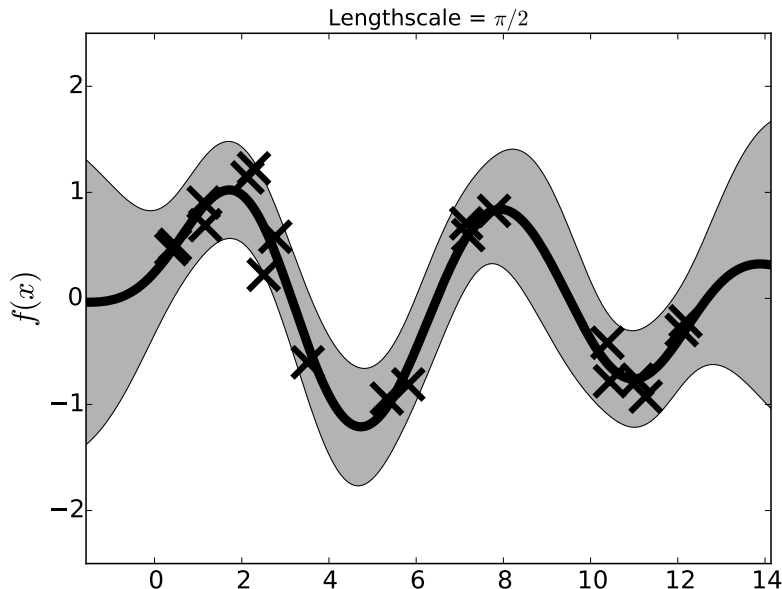
# The Effect of Lengthscale on Model Fit



# The Effect of Lengthscale on Model Fit

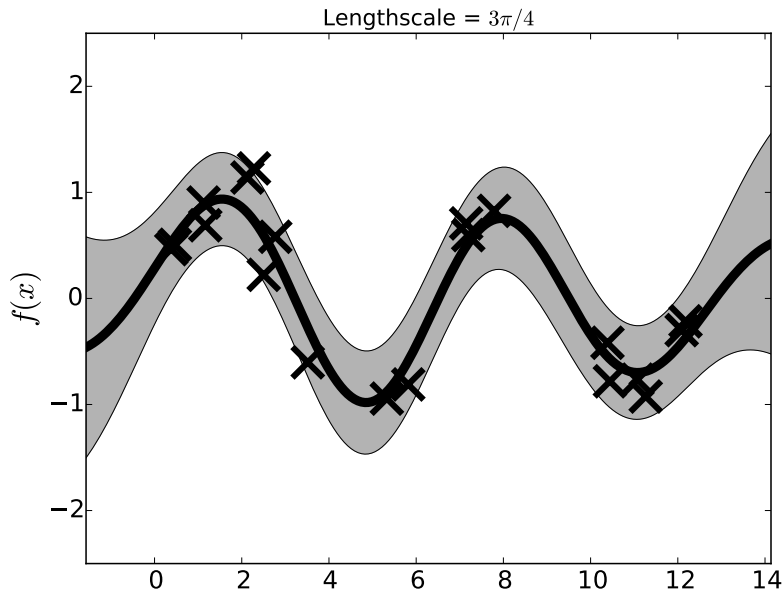


# The Effect of Lengthscale on Model Fit

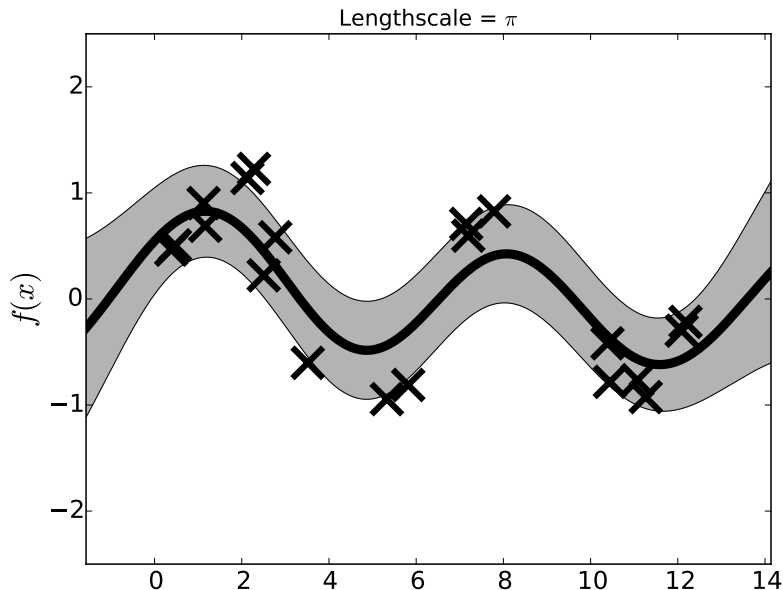




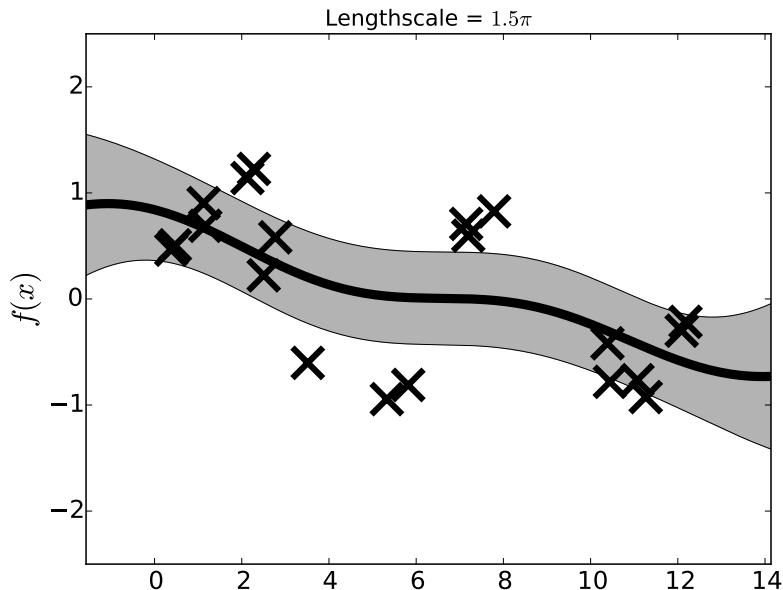
# The Effect of Lengthscale on Model Fit



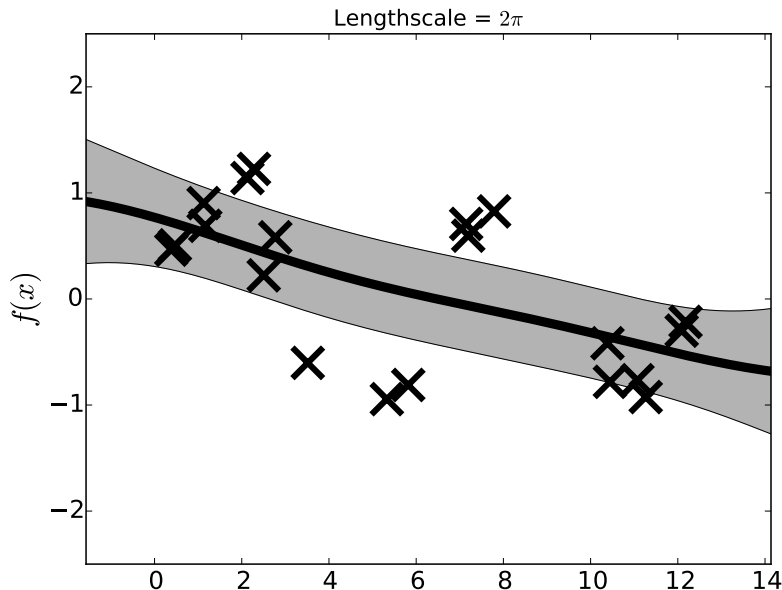
# The Effect of Lengthscale on Model Fit



# The Effect of Lengthscale on Model Fit



# The Effect of Lengthscale on Model Fit



# Learning the Covariance Parameters

- Let us define  $\theta$  as a vector containing all (hyper)parameters  
e.g  $\theta = \{\alpha, l, \sigma_n^2\}$

- We choose  $\theta$  that maximises the log marginal likelihood, that is:

$$\ln p(y|\mathbf{x}, \theta) = -\frac{1}{2}y^T(k(\mathbf{x}, \mathbf{x}') + \sigma_n^2 I)^{-1}y - \frac{1}{2} \ln |k(\mathbf{x}, \mathbf{x}') + \sigma_n^2 I| - \frac{n}{2} \ln 2\pi$$

- Cannot guarantee a global optimum, try different initial conditions
- Constraint (hyper)parameters to some sensible limits
- **Note:** Choosing the right covariance function (e.g Squared Exponential, Matérn, Rational Quadratic, etc.) is not always easy and should be treated akin to a model selection problem

# Learning the Covariance Parameters

- Let us define  $\theta$  as a vector containing all (hyper)parameters  
e.g  $\theta = \{\alpha, l, \sigma_n^2\}$
- We choose  $\theta$  that maximises the log marginal likelihood, that is:

$$\ln p(y|\mathbf{x}, \theta) = -\frac{1}{2}y^T(k(\mathbf{x}, \mathbf{x}') + \sigma_n^2 I)^{-1}y - \frac{1}{2} \ln |k(\mathbf{x}, \mathbf{x}') + \sigma_n^2 I| - \frac{n}{2} \ln 2\pi$$

- Cannot guarantee a global optimum, try different initial conditions
- Constraint (hyper)parameters to some sensible limits
- **Note:** Choosing the right covariance function (e.g Squared Exponential, Matérn, Rational Quadratic, etc.) is not always easy and should be treated akin to a model selection problem

# Learning the Covariance Parameters

- Let us define  $\theta$  as a vector containing all (hyper)parameters  
e.g  $\theta = \{\alpha, l, \sigma_n^2\}$
- We choose  $\theta$  that maximises the log marginal likelihood, that is:

$$\ln p(y|\mathbf{x}, \theta) = -\frac{1}{2}y^T(k(\mathbf{x}, \mathbf{x}') + \sigma_n^2 I)^{-1}y - \frac{1}{2} \ln |k(\mathbf{x}, \mathbf{x}') + \sigma_n^2 I| - \frac{n}{2} \ln 2\pi$$

- Cannot guarantee a global optimum, try different initial conditions
- Constraint (hyper)parameters to some sensible limits
- **Note:** Choosing the right covariance function (e.g Squared Exponential, Matérn, Rational Quadratic, etc.) is not always easy and should be treated akin to a model selection problem

# Learning the Covariance Parameters

- Let us define  $\theta$  as a vector containing all (hyper)parameters  
e.g  $\theta = \{\alpha, l, \sigma_n^2\}$
- We choose  $\theta$  that maximises the log marginal likelihood, that is:

$$\ln p(y|\mathbf{x}, \theta) = -\frac{1}{2}y^T(k(\mathbf{x}, \mathbf{x}') + \sigma_n^2 I)^{-1}y - \frac{1}{2} \ln |k(\mathbf{x}, \mathbf{x}') + \sigma_n^2 I| - \frac{n}{2} \ln 2\pi$$

- Cannot guarantee a global optimum, try different initial conditions
- Constraint (hyper)parameters to some sensible limits
- **Note:** Choosing the right covariance function (e.g Squared Exponential, Matérn, Rational Quadratic, etc.) is not always easy and should be treated akin to a model selection problem



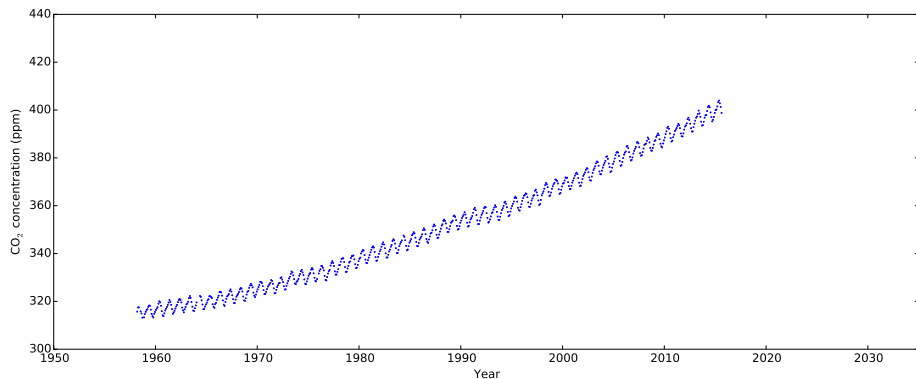
# Learning the Covariance Parameters

- Let us define  $\theta$  as a vector containing all (hyper)parameters  
e.g  $\theta = \{\alpha, l, \sigma_n^2\}$
- We choose  $\theta$  that maximises the log marginal likelihood, that is:

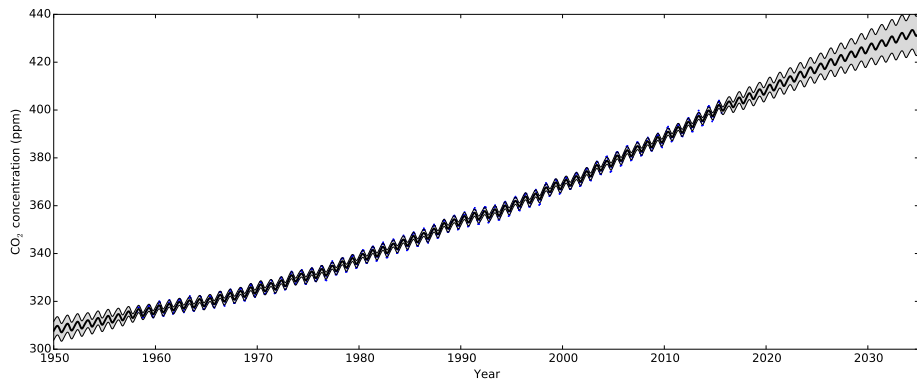
$$\ln p(y|\mathbf{x}, \theta) = -\frac{1}{2}y^T(k(\mathbf{x}, \mathbf{x}') + \sigma_n^2 I)^{-1}y - \frac{1}{2} \ln |k(\mathbf{x}, \mathbf{x}') + \sigma_n^2 I| - \frac{n}{2} \ln 2\pi$$

- Cannot guarantee a global optimum, try different initial conditions
- Constraint (hyper)parameters to some sensible limits
- **Note:** Choosing the right covariance function (e.g Squared Exponential, Matérn, Rational Quadratic, etc.) is not always easy and should be treated akin to a model selection problem

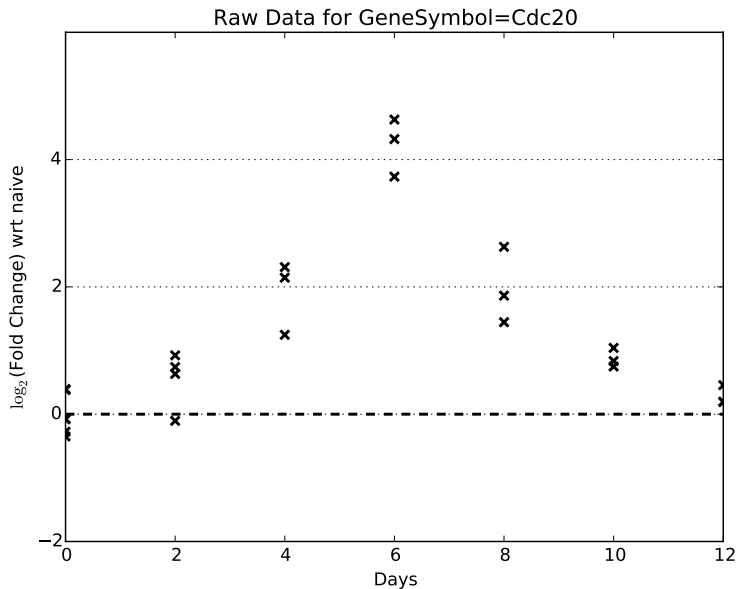
# Modelling CO<sub>2</sub> Concentrations



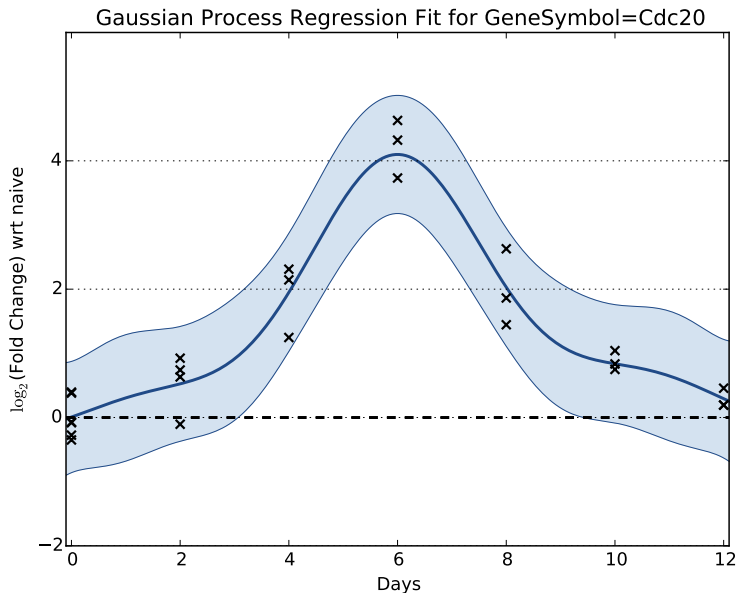
# Modelling CO<sub>2</sub> Concentrations



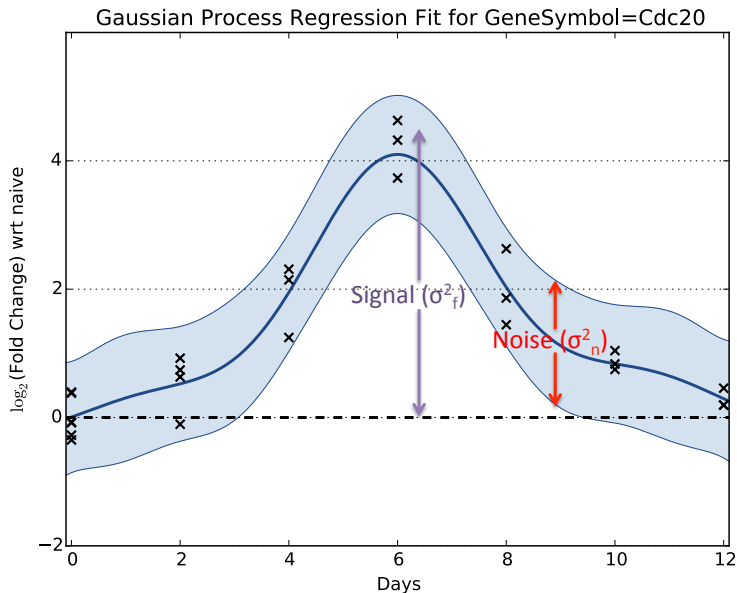
# Modelling Gene Expression Time-Series



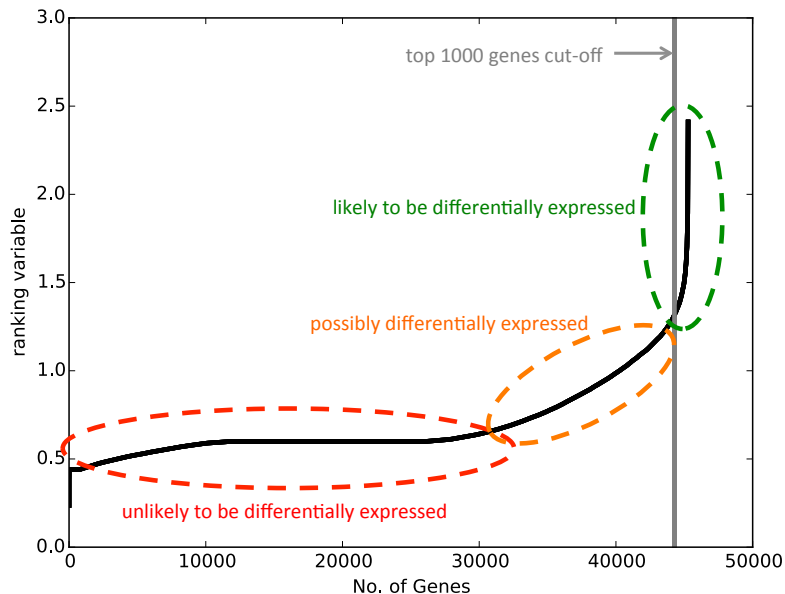
# Modelling Gene Expression Time-Series



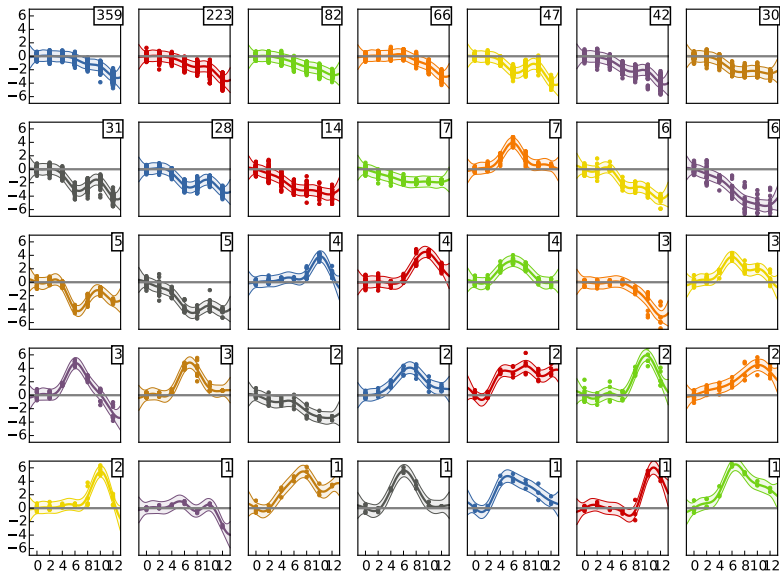
# Modelling Gene Expression Time-Series



# Ranking Gene Expression Time-Series



# Clustering Gene Expression Time-Series

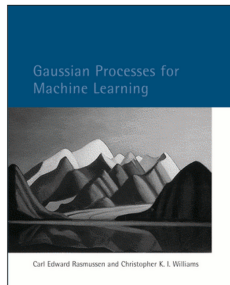




- Rasmussen and Williams book

## **Gaussian Processes for Machine Learning**

Carl Edward Rasmussen and Christopher K. I. Williams  
The MIT Press, 2006. ISBN 0-262-18253-X.



- <http://www.gaussianprocess.org/>
- The GPy Python Module by Neil Lawrence's Sheffield Group:  
<https://github.com/SheffieldML/GPy>