

# **STA414/2104**

## **Weeks 10-11: Gaussian Processes**

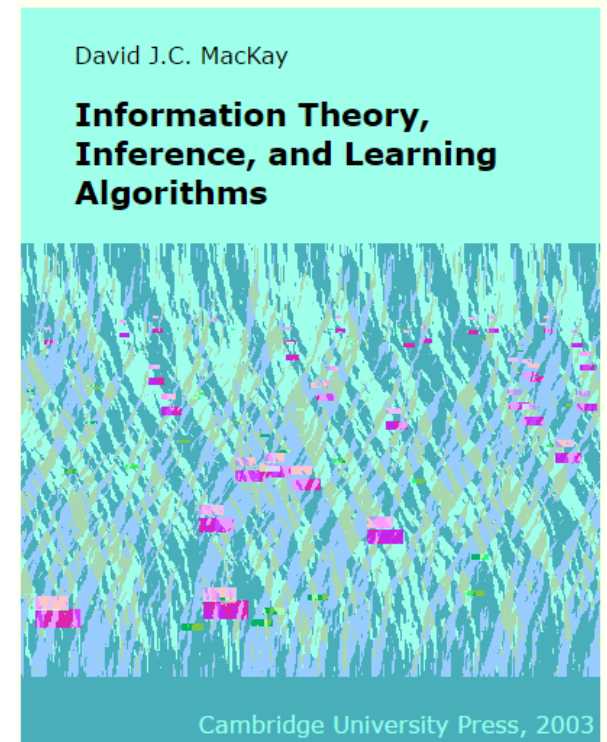
Department of Statistics  
[www.utstat.utoronto.ca](http://www.utstat.utoronto.ca)

Delivered by Mark Ebden, 19-26 March 2018  
with thanks to Russ Salakhutdinov

# Examples of extra perspectives

- Murphy 15.1 – 15.2
- Bishop 6.4
- MacKay chapter 45
- Gentle introduction:

<https://arxiv.org/abs/1505.02965>



# Outline

- Kernel functions
- Introduction to GPs
- A closer look at covariance matrices
- The theory of GPs
- Applications



# Linear Regression Revisited

- Consider the following linear model, defined in terms of a linear combination of  $M$  fixed basis functions:

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \phi(\mathbf{x})$$

- We place a Gaussian prior over model parameters:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w} | \mathbf{0}, \alpha^{-1} \mathbf{I})$$

- For any given fixed value of  $\mathbf{w}$ , we have a corresponding linear function. A probability distribution over  $\mathbf{w}$  defines a probability distribution over functions
- Given a dataset  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ , we will denote the values of the function as  $\mathbf{f} = [f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_N)]^T$ .

- Hence:

$$\mathbf{f} = \Phi \mathbf{w}.$$

$N \times M$                        $M \times 1$

design matrix                      vector of model parameters

# Linear Regression Revisited

$$\mathbf{f} = \Phi \mathbf{w}, \quad p(\mathbf{w}) = \mathcal{N}(\mathbf{w} | \mathbf{0}, \alpha^{-1} \mathbf{I})$$

- Observe that  $\mathbf{f}$  is a linear combination of Gaussian random variables, and hence is **itself Gaussian**:

$$\mathbb{E}[\mathbf{f}] = \Phi \mathbb{E}[\mathbf{w}] = \mathbf{0}$$

$$\text{cov}(\mathbf{f}) = \mathbb{E}[\mathbf{f} \mathbf{f}^T] = \Phi \mathbb{E}[\mathbf{w} \mathbf{w}^T] \Phi^T = \frac{1}{\alpha} \Phi \Phi^T = \mathbf{K}$$

Here,  $\mathbf{K}$  is an example of a **Gram matrix**, with elements:

$$\mathbf{K}_{nm} = k(\mathbf{x}_n, \mathbf{x}_m) = \frac{1}{\alpha} \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m),$$

where  $k(\mathbf{x}, \mathbf{x}')$  is the **kernel function**

- This model provides a **particular example of a Gaussian process**

# Recall from Lecture 4, Bayesian linear regression...

## Predictive Distribution

- We can make predictions for a new input vector  $\mathbf{x}$  by integrating over the posterior distribution:

$$\begin{aligned} p(t|\mathbf{t}, \mathbf{x}, \mathbf{X}, \alpha, \beta) &= \int p(t|\mathbf{x}, \mathbf{w}, \beta) p(\mathbf{w}|\mathbf{t}, \mathbf{X}, \alpha, \beta) d\mathbf{w} \\ &= \mathcal{N}(t|\mathbf{m}_N^T \phi(\mathbf{x}), \sigma_N^2(\mathbf{x})), \end{aligned}$$

where

$$\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \phi(\mathbf{x})^T \mathbf{S}_N \phi(\mathbf{x}).$$

Noise in the  
target values

Uncertainty  
associated with  
parameter values.

$$\begin{aligned} \mathbf{m}_N &= \beta \mathbf{S}_N \Phi^T \mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha \mathbf{I} + \beta \Phi^T \Phi. \end{aligned}$$

- As  $N \rightarrow \infty$ :
  - The second term goes to zero
  - The variance of the predictive distribution arises only from the additive noise governed by parameter  $\beta$

# Equivalent Kernel

- The predictive mean can be written as:

$$\begin{aligned}y(\mathbf{x}, \mathbf{m}_N) &= \mathbf{m}_N^T \phi(\mathbf{x}) = \beta \phi(\mathbf{x})^T \mathbf{S}_N \Phi^T \mathbf{t} \\&= \sum_{n=1}^N \underbrace{\beta \phi(\mathbf{x})^T \mathbf{S}_N \phi(\mathbf{x}_n)}_{k(\mathbf{x}, \mathbf{x}_n)} t_n \\&= \sum_{n=1}^N k(\mathbf{x}, \mathbf{x}_n) t_n.\end{aligned}$$

$$\begin{aligned}\mathbf{m}_N &= \beta \mathbf{S}_N \Phi^T \mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha \mathbf{I} + \beta \Phi^T \Phi.\end{aligned}$$

Equivalent kernel  
or smoother  
matrix.

- The mean of the predictive distribution at a time  $\mathbf{x}$  can be written as a linear combination of the training set target values.
- Such regression functions are called linear smoothers.

# Outline

- Kernel functions
- **Introduction to GPs**
- A closer look at covariance matrices
- The theory of GPs
- Applications





# Gaussian Processes

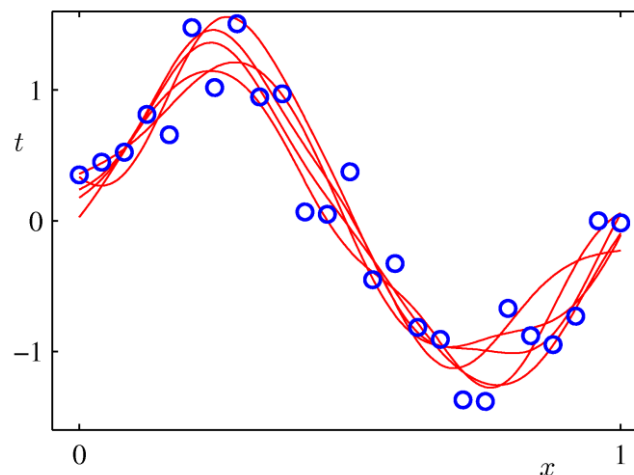
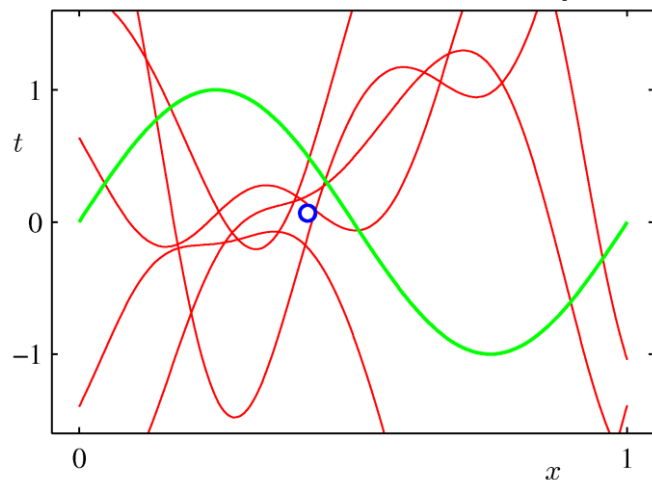
- So far, we have considered **linear regression models** of the form:

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$$

where  $\mathbf{w}$  is a vector of parameters and  $\boldsymbol{\phi}(\mathbf{x})$  is a vector of fixed nonlinear basis functions

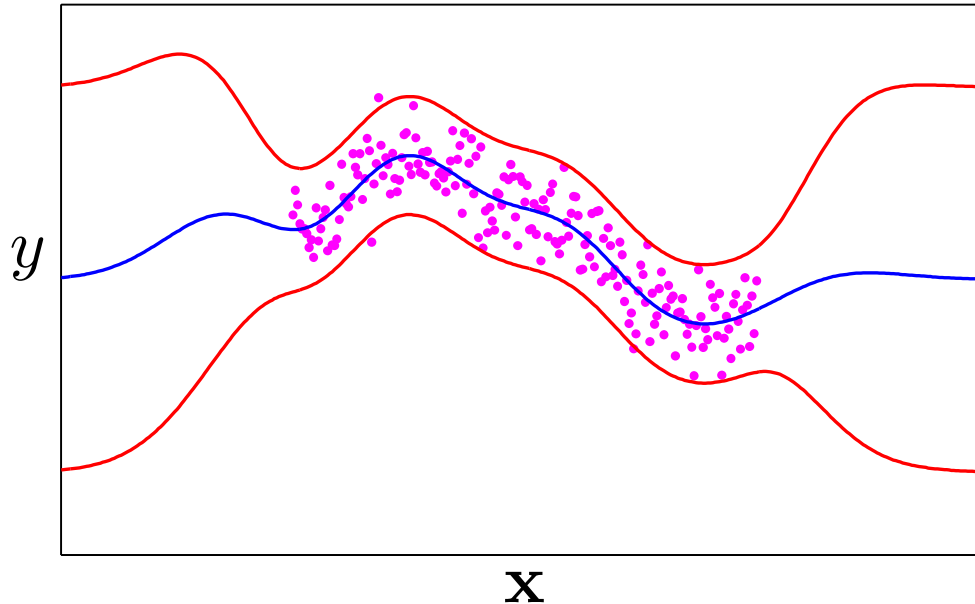
- A prior distribution over  $\mathbf{w}$  induces a **prior distribution over functions**  $f(\mathbf{x}, \mathbf{w})$
- Given a training dataset, we compute the posterior distribution over  $\mathbf{w}$ , which induces a **posterior distribution** over functions  $f(\mathbf{x}, \mathbf{w})$

Samples from the posterior



# Gaussian Processes

- You want to learn a function  $f$  with error bars from data  $\mathcal{D}$



- A Gaussian process defines a distribution over functions  $p(f)$  which can be used for Bayesian regression:

$$p(f|\mathcal{D}) = \frac{p(f)p(\mathcal{D}|f)}{p(\mathcal{D})}$$

# Gaussian Processes

- In the Gaussian process viewpoint, we define a **prior probability distribution over functions directly**
- May seem difficult: How can we define a distribution over the uncountably infinite space of functions?
- **Insight**: for a finite training set, we only need to consider the values of the functions at a discrete set of input values  $x_n$
- Hence in practice, we work in a **finite space**
- Many related models: In the geostatistics literature, GP regression is known as kriging. See also the book on GPs by Rasmussen & Williams (2006)

# Gaussian Process

- A Gaussian process (GP) is a **random function**  $\mathbf{f}$  such that **for any finite set** of input points  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ ,

$$\begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_N) \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} m(\mathbf{x}_1) \\ \vdots \\ m(\mathbf{x}_N) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & \cdots & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \right)$$

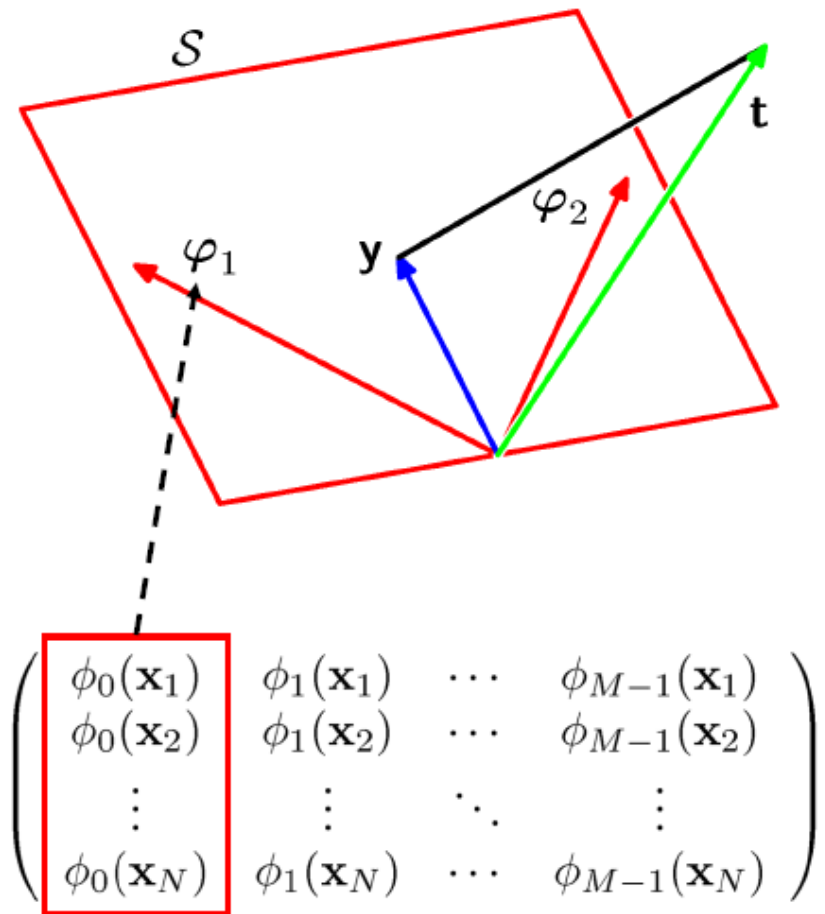
where the parameters are the **mean function**  $m(\mathbf{x})$  and **covariance kernel**  $k(\mathbf{x}, \mathbf{x}')$

- Note that a **random function is a stochastic process**. It is a collection of random variables  $\{f(\mathbf{x})\}$ , one for each possible value  $\mathbf{x}$  (see Rasmussen and Williams, 2006)
- Key point about Gaussian Processes:** Given a dataset  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ , the distribution over  $[f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_N)]$  is completely specified by the second-order statistics: the mean and covariance

# Recall the curiosity from Lecture 3...

## Geometry of Least Squares

- Consider an  $N$ -dimensional space, so that  $\mathbf{t} = [t_1, t_2, \dots, t_N]^T$  is a vector in that space.
- Each basis function  $\phi_j(\mathbf{x}_n)$ , evaluated at the  $N$  data points, can be represented as a vector in the same space.
- If  $M < N$  then the  $M$  basis functions,  $\phi_j(\mathbf{x}_n)$ , will span a linear subspace  $S$  of dimensionality  $M$ .
- Define:  $\mathbf{y} = \Phi \mathbf{w}_{\text{ML}}$ .
- The sum-of-squares error is equal to the squared Euclidean distance between  $\mathbf{y}$  and  $\mathbf{t}$  (up to a factor of  $1/2$ ).



# Outline

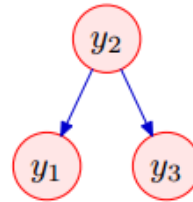
- Kernel functions
- Introduction to GPs
- **A closer look at covariance matrices**
- The theory of GPs
- Applications



# The meaning of the elements in a covariance matrix for a multivariate normal distribution

- A zero in a covariance matrix  $\mathbf{K}$  at location  $i,j$  means:  
 $i$  and  $j$  are eligible for independence. Recall  $i \perp j \Rightarrow \text{cov}(i,j) = 0$
- A positive off-diagonal element in  $\mathbf{K}$  means:  
There is a positive correlation between  $i$  and  $j$   
*And: negative  $\rightarrow$  negative*
- A zero in an inverse covariance matrix  $\mathbf{K}^{-1}$  at location  $i,j$  means:  
Conditional on all other variables,  $i$  and  $j$  are independent
- A positive off-diagonal element in  $\mathbf{K}^{-1}$  means:  
Conditional on all other variables, there is a negative corr.'In between  $i$  and  $j$   
*And vice versa: negative  $\rightarrow$  positive*
- The off-diagonal entries in  $\mathbf{K}$  tell us how the mean of *the conditional distribution of one variable given the others* depends on the others
- Leaving out a variable (removing a row and column) leaves the rest of  $\mathbf{K}$  unchanged but usually changes the rest of  $\mathbf{K}^{-1}$

# Quiz on the meaning of **K**



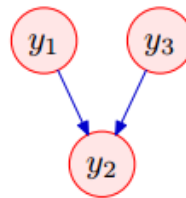
1. Assuming that the variables  $y_1, y_2, y_3$  in this belief network have a joint Gaussian distribution, which of the following matrices *could be* the covariance matrix?

A or B

$y_1$  and  $y_3$  vary together so can't be zero at position 1,3 and 3,1

A	B	C	D
$\begin{bmatrix} 9 & 3 & 1 \\ 3 & 9 & 3 \\ 1 & 3 & 9 \end{bmatrix}$	$\begin{bmatrix} 8 & -3 & 1 \\ -3 & 9 & -3 \\ 1 & -3 & 8 \end{bmatrix}$	$\begin{bmatrix} 9 & 3 & 0 \\ 3 & 9 & 3 \\ 0 & 3 & 9 \end{bmatrix}$	$\begin{bmatrix} 9 & -3 & 0 \\ -3 & 10 & -3 \\ 0 & -3 & 9 \end{bmatrix}$

2. Which of the matrices could be the *inverse* covariance matrix?



C or D

since  $y_1$  and  $y_3$  coparents of  $y_2$   
 $y_1$  and  $y_3$  are independent

3. Which of the matrices could be the covariance matrix of the second graphical model?

4. Which of the matrices could be the inverse covariance matrix of the second graphical model?



# Quiz on the meaning of $\mathbf{K}$

5. Let three variables  $y_1, y_2, y_3$  have covariance matrix  $\mathbf{K}_{(3)}$ , and inverse covariance matrix  $\mathbf{K}_{(3)}^{-1}$ .

$$\mathbf{K}_{(3)} = \begin{bmatrix} 1 & .5 & 0 \\ .5 & 1 & .5 \\ 0 & .5 & 1 \end{bmatrix} \quad \mathbf{K}_{(3)}^{-1} = \begin{bmatrix} 1.5 & -1 & .5 \\ -1 & 2 & -1 \\ .5 & -1 & 1.5 \end{bmatrix}$$

Now focus on the variables  $y_1$  and  $y_2$ . Which statements about *their* covariance matrix  $\mathbf{K}_{(2)}$  and inverse covariance matrix  $\mathbf{K}_{(2)}^{-1}$  are true?

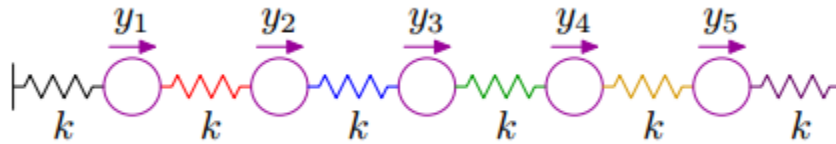
$$\begin{array}{cc} \text{(A)} & \text{(B)} \\ \mathbf{K}_{(2)} = \begin{bmatrix} 1 & .5 \\ .5 & 1 \end{bmatrix} & \mathbf{K}_{(2)}^{-1} = \begin{bmatrix} 1.5 & -1 \\ -1 & 2 \end{bmatrix} \end{array}$$

true

false

# Quiz on the meaning of **K**

- Example:



- Which is the covariance matrix, and which the inverse covariance?

$$\frac{k}{T} \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

$$\frac{T}{k} \begin{bmatrix} 0.83 & 0.67 & 0.50 & 0.33 & 0.17 \\ 0.67 & 1.33 & 1.00 & 0.67 & 0.33 \\ 0.50 & 1.00 & 1.50 & 1.00 & 0.50 \\ 0.33 & 0.67 & 1.00 & 1.33 & 0.67 \\ 0.17 & 0.33 & 0.50 & 0.67 & 0.83 \end{bmatrix}$$

inverse covariance

# Outline

- Kernel functions
- Introduction to GPs
- A closer look at covariance matrices
- **The theory of GPs**
- Applications



# Gaussian Process

- In many applications, we will have no prior knowledge about the mean function  $f(\mathbf{x})$ . By symmetry, **we take it to be zero**
- The specification of a Gaussian Process is then completed by **specifying the covariance function**, evaluated at any two input points  $\mathbf{x}_n$  and  $\mathbf{x}_m$ :

$$\mathbb{E}[f(\mathbf{x}_n)f(\mathbf{x}_m)] = k(\mathbf{x}_n, \mathbf{x}_m).$$

- One commonly used covariance function is the squared exponential:

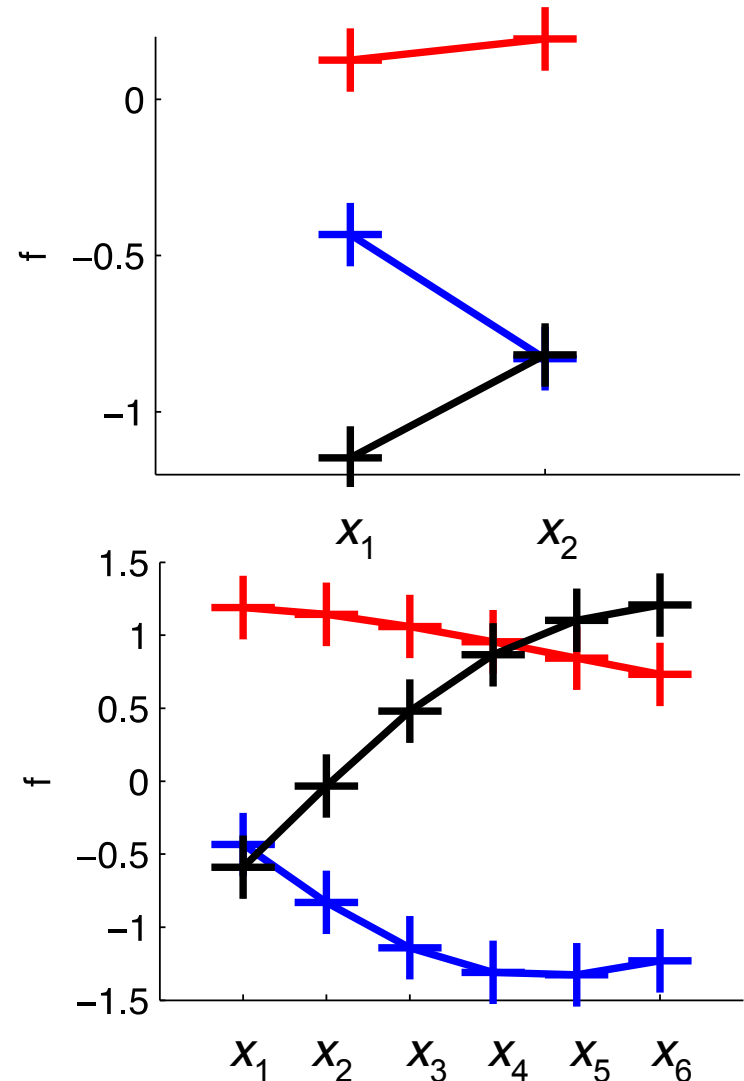
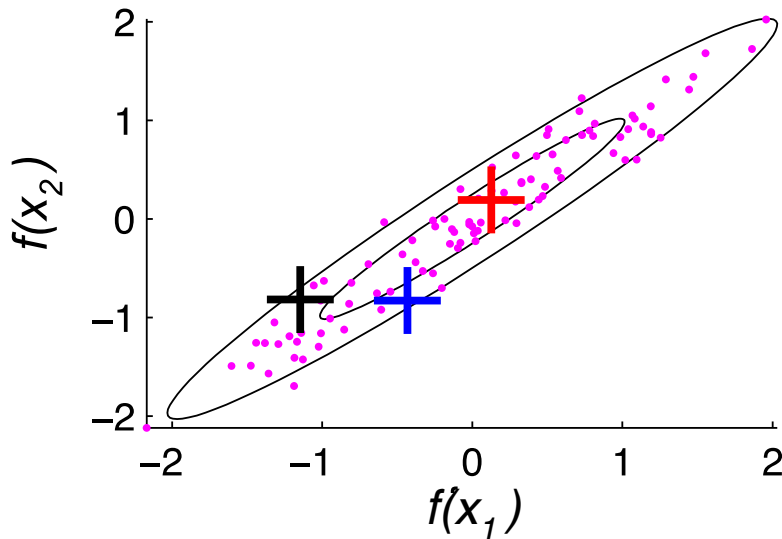
$$k(\mathbf{x}_n, \mathbf{x}_m) = \exp\left(-\frac{\theta}{2} \|\mathbf{x}_n - \mathbf{x}_m\|^2\right)$$

kernel of gaussian distribution

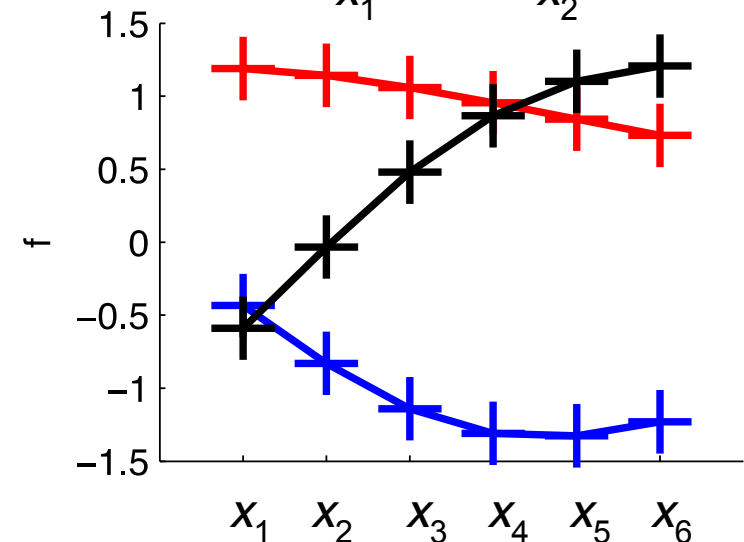
- The covariance function (kernel function) is typically chosen to express the property that, for **inputs  $\mathbf{x}_n$  and  $\mathbf{x}_m$  that are similar**, the corresponding values  $f(\mathbf{x}_n)$  and  $f(\mathbf{x}_m)$  will **be more strongly correlated** than for dissimilar points

# Visualizing Draws from GPs

- Visualizing draws from 2-D Gaussian:

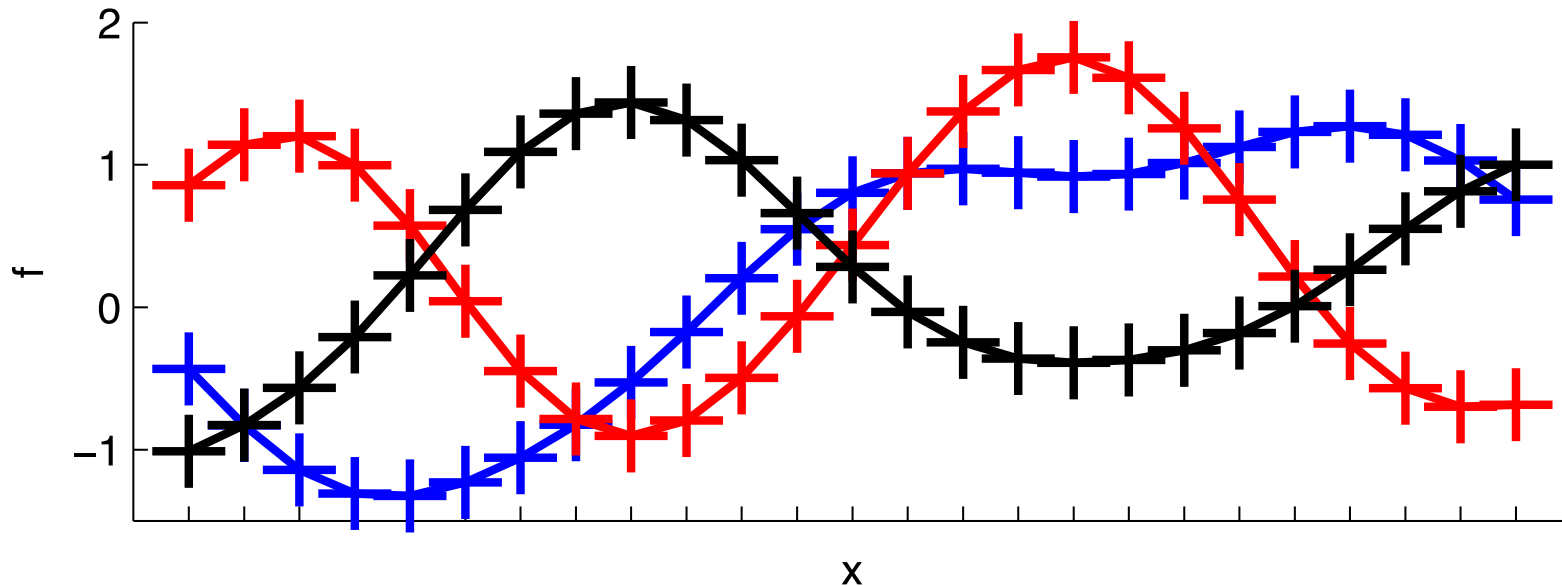


- Three draws from a 6-D Gaussian:



# Visualizing Draws from GPs

- Three draws from 25-D Gaussian

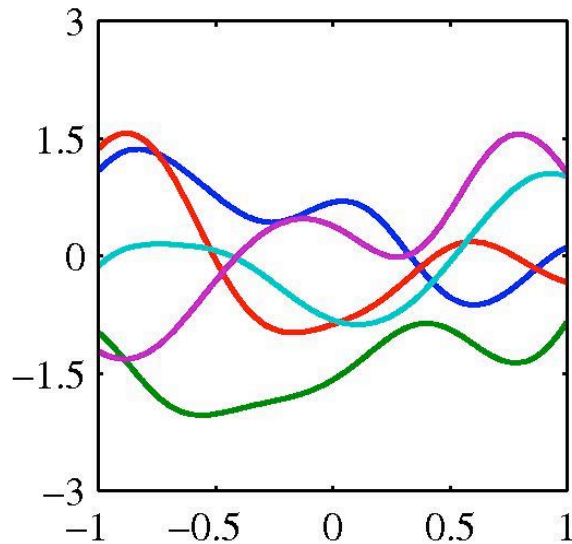


- To generate these, the mean was set to a 25-vector of zeros
- The covariance was set using a covariance function:  $\Sigma_{nm} = k(\mathbf{x}_n, \mathbf{x}_m)$ .
- The x's are the ticks on the axis

We can visualize draws from a GP as iteratively sampling  $f(x_n) \mid f(x_1), \dots, f(x_{n-1})$  on a sequence of input points  $x_1, x_2, \dots, x_n$ .

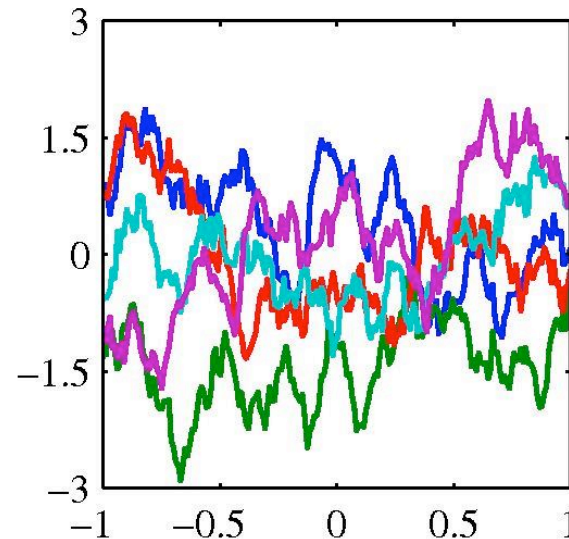
# Samples from GPs

Squared-exponential kernel



$$k(x_n, x_m) = \exp\left(-\frac{\theta}{2}(x_n - x_m)^2\right)$$

Exponential kernel



$$k(x_n, x_m) = \exp(-\theta|x_n - x_m|)$$

- Ornstein-Uhlenbeck process that describes Brownian motion

# GPs for Regression

- We need to account for **noise on the observed target values**:

$$t_n = f_n + \epsilon_n,$$

where  $f_n = f(\mathbf{x}_n)$ , and  $\epsilon_n$  is an **independent random noise variable**. We will assume Gaussian noise:

$$p(t_n|f_n) = \mathcal{N}(t_n|f_n, \beta^{-1}).$$

- Given a dataset  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ , and corresponding target values  $\mathbf{t} = \{t_1, t_2, \dots, t_N\}$ , the **conditional** takes the form:

$$p(\mathbf{t}|\mathbf{f}) = \mathcal{N}(\mathbf{t}|\mathbf{f}, \beta^{-1}\mathbf{I}_N).$$

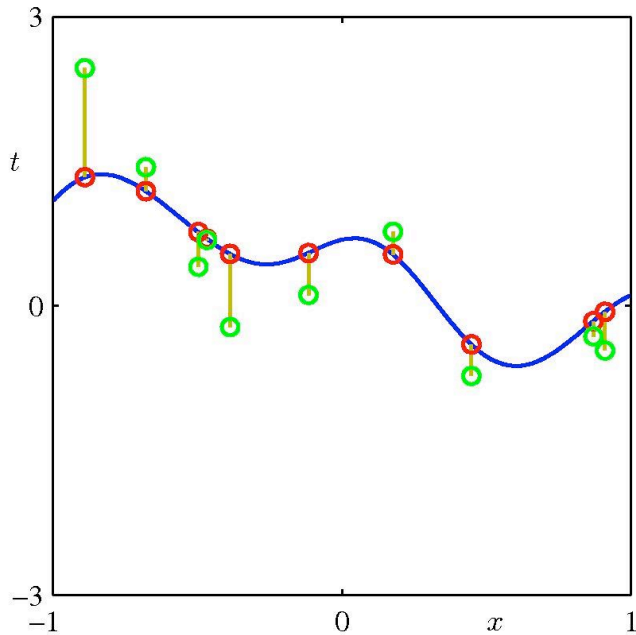
- From the definitions of a Gaussian process, **the marginal distribution**  $p(\mathbf{f})$  is given by the Gaussian:

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}).$$



# Illustration

- Here we sample targets  $\{t_n\}$  from a Gaussian process



- The blue curve shows a sample from a GP prior:

$$f \sim \mathcal{GP}$$

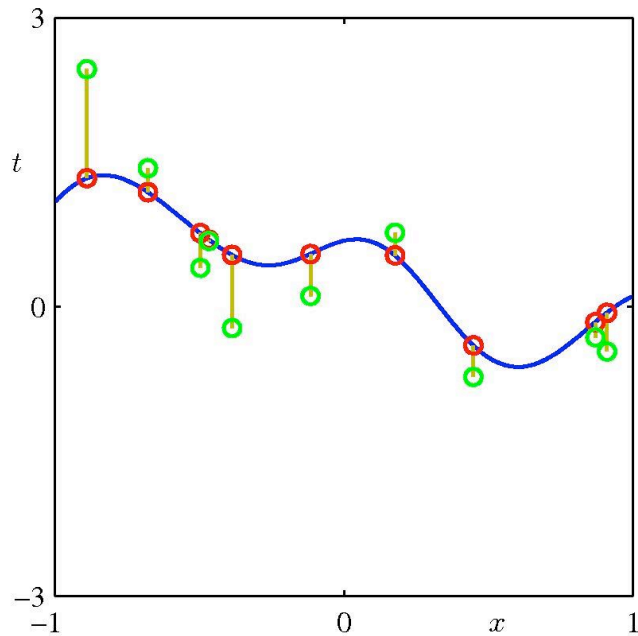
- The red points show the values of  $f_n$ , obtained by evaluating the function at a set of input values  $\{x_n\}$

- The green points show the corresponding values of  $\{t_n\}$ :

$$p(t_n | f_n) = \mathcal{N}(t_n | f_n, \beta^{-1}).$$

# Marginal Distribution

- The **marginal distribution**  $p(\mathbf{t})$ , conditioned on the set of inputs  $\mathbf{X}$ , can be obtained by integrating over  $\mathbf{f}$ :



$$p(\mathbf{t}) = \int p(\mathbf{t}|\mathbf{f})p(\mathbf{f})d\mathbf{f} = \mathcal{N}(\mathbf{t}|\mathbf{0}, \mathbf{C}),$$

where the covariance matrix is given by:

$$C(\mathbf{x}_n, \mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m) + \beta^{-1}\delta_{nm}.$$

- The **two Gaussian sources of randomness**, one associated with  $f(\mathbf{x})$  and the other with noise, are independent, and so their covariances add

# Covariance Function

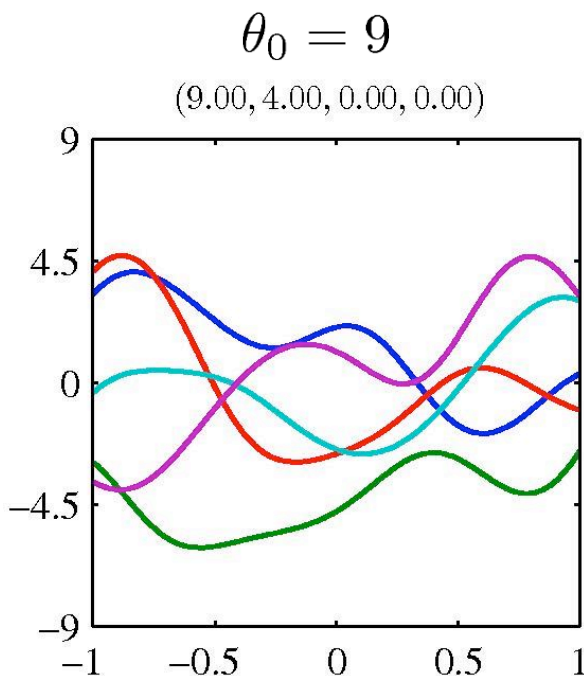
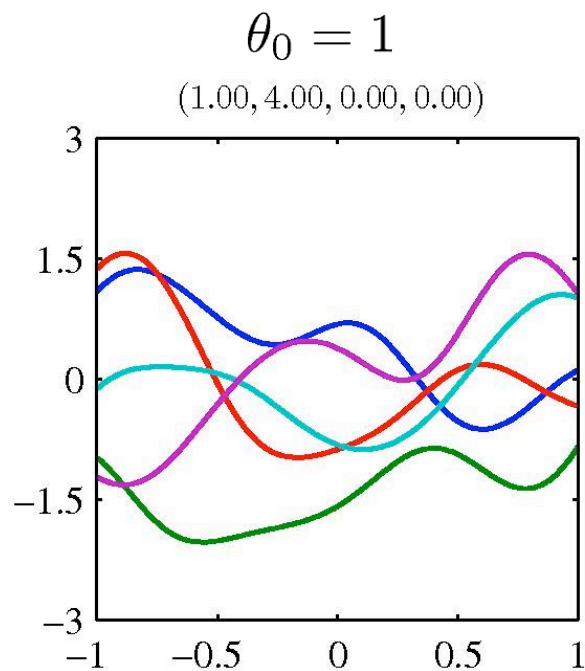
- One widely used covariance (kernel) function for GP regression is given by the squared-exponential plus constant and linear terms:

positive semidefinite kernel

$$k(\mathbf{x}_n, \mathbf{x}_m) = \theta_0 \exp\left(-\frac{\theta_1}{2} \|\mathbf{x}_n - \mathbf{x}_m\|^2\right) + \theta_2 + \theta_3 \mathbf{x}_n^T \mathbf{x}_m$$

sum of valid kernels is a valid kernel

- Note that the last term corresponds to a parametric model that is a linear function of the input variables



standard  
deviation  
gone up by a  
factor of 3

# Covariance Function

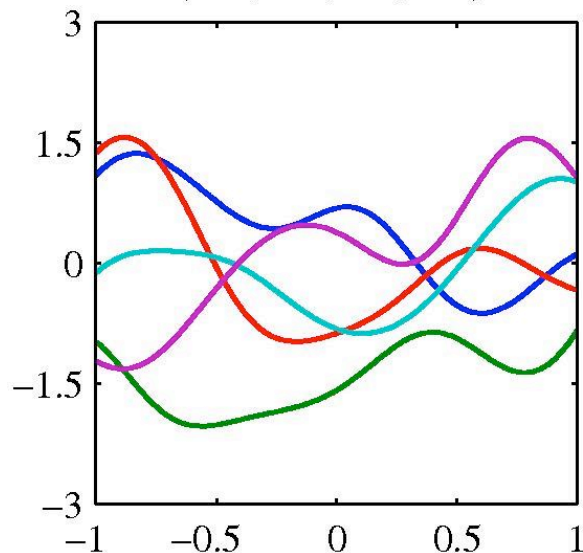
- One widely used covariance (kernel) function for GP regression is given by the **squared-exponential** plus constant and linear terms:

$$k(\mathbf{x}_n, \mathbf{x}_m) = \theta_0 \exp\left(-\frac{\theta_1}{2} \|\mathbf{x}_n - \mathbf{x}_m\|^2\right) + \theta_2 + \theta_3 \mathbf{x}_n^T \mathbf{x}_m$$

- Note that the last term corresponds to a parametric model that is a **linear function of the input variables**

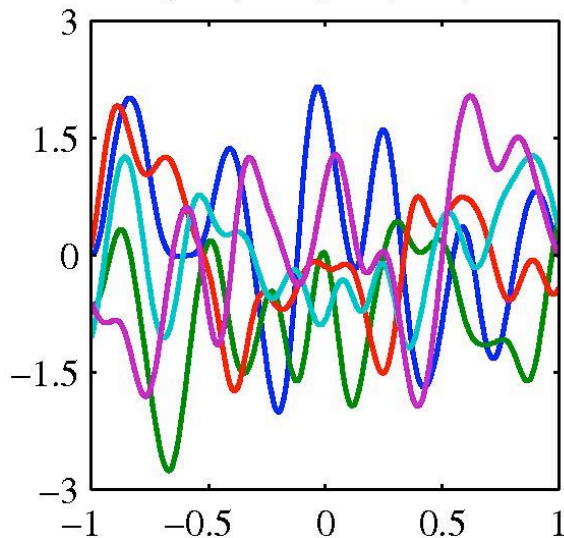
$$\theta_1 = 1$$

(1.00, 4.00, 0.00, 0.00)



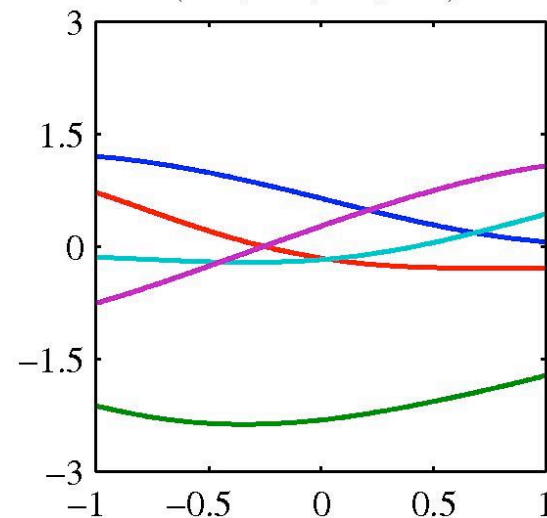
$$\theta_1 = 64$$

(1.00, 64.00, 0.00, 0.00)



$$\theta_1 = 0.25$$

(1.00, 0.25, 0.00, 0.00)



# Covariance Function

- One widely used covariance (kernel) function for GP regression is given by the squared-exponential plus constant and linear terms:

$$k(\mathbf{x}_n, \mathbf{x}_m) = \theta_0 \exp\left(-\frac{\theta_1}{2} \|\mathbf{x}_n - \mathbf{x}_m\|^2\right) + \theta_2 + \theta_3 \mathbf{x}_n^T \mathbf{x}_m$$

corresponds back to linear regression

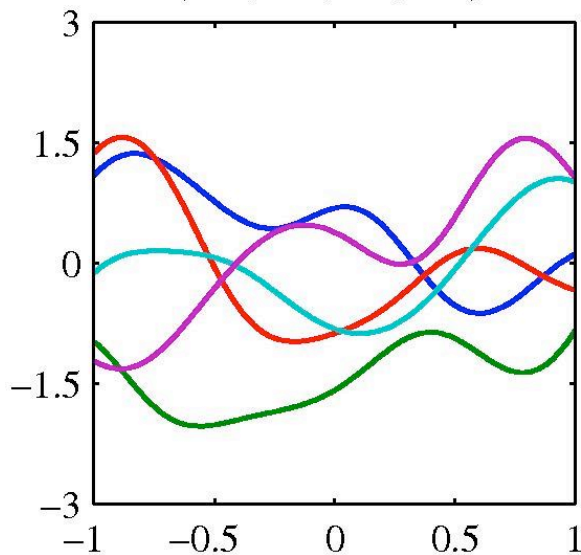
- Note that the last term corresponds to a parametric model that is a linear function of the input variables

variance vary w.r.t. x,  
curves a mostly linear (hence the name)

variance is larger regardless of value of x

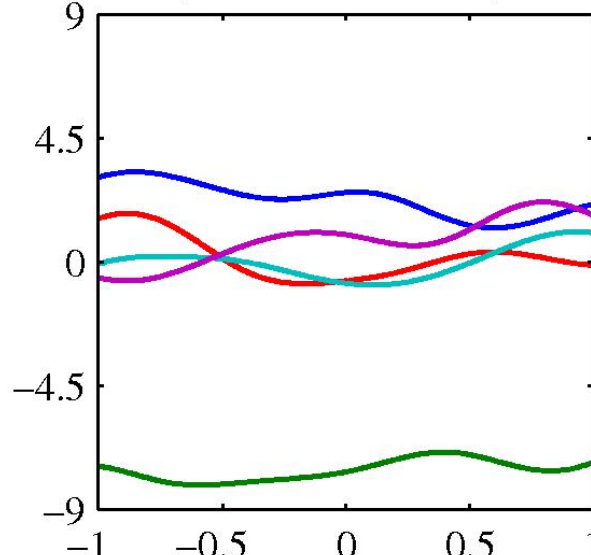
$$\theta_2 = 0, \theta_3 = 0$$

(1.00, 4.00, 0.00, 0.00)



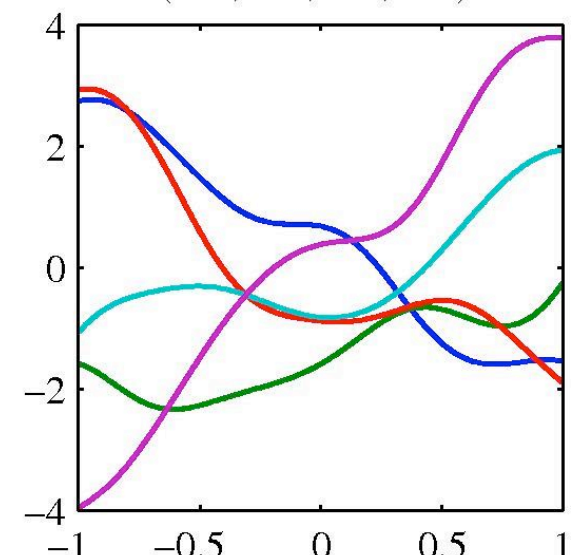
$$\theta_2 = 10, \theta_3 = 0$$

(1.00, 4.00, 10.00, 0.00)



$$\theta_2 = 0, \theta_3 = 5$$

(1.00, 4.00, 0.00, 5.00)



# Prediction

- Suppose we are given a dataset,  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ , with target values  $\mathbf{t} = \{t_1, t_2, \dots, t_N\}$ .
- Our goal is predict  $t_{N+1}$  for a new input vector  $\mathbf{x}_{N+1}$
- Note that the joint distribution over  $\mathbf{t}$  and  $t_{N+1}$  is given by:

$$P\left(\begin{bmatrix} \mathbf{t} \\ t_{N+1} \end{bmatrix}\right) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{C}_N & \mathbf{k} \\ \mathbf{k}^T & c \end{bmatrix}\right)$$

where  $C_N$  is the  $N \times N$  matrix with elements:

$$C_N(\mathbf{x}_n, \mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m) + \beta^{-1} \delta_{nm}.$$

covariance for the noise

$c$  is the scalar:

$$c = k(\mathbf{x}_{N+1}, \mathbf{x}_{N+1}) + \beta^{-1}$$

and  $\mathbf{k}$  is the  $N \times 1$  vector with elements  $k(\mathbf{x}_n, \mathbf{x}_{N+1})$

# Prediction

- Suppose we are given a dataset,  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ , with target values  $\mathbf{t} = \{t_1, t_2, \dots, t_N\}$ .
- Our goal is predict  $t_{N+1}$  for a new input vector  $\mathbf{x}_{N+1}$
- Note that the joint distribution over  $\mathbf{t}$  and  $t_{N+1}$  is given by:

$$P\left(\begin{bmatrix} \mathbf{t} \\ t_{N+1} \end{bmatrix}\right) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{C}_N & \mathbf{k} \\ \mathbf{k}^T & c \end{bmatrix}\right)$$

- Hence the conditional distribution is Gaussian:

$$P(t_{N+1}|\mathbf{t}) = \mathcal{N}(m(\mathbf{x}_{N+1}), \sigma^2(\mathbf{x}_{N+1}))$$

with

$$m(\mathbf{x}_{N+1}) = \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t}$$

$$\sigma^2(\mathbf{x}_{N+1}) = c - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k}$$

derivation..

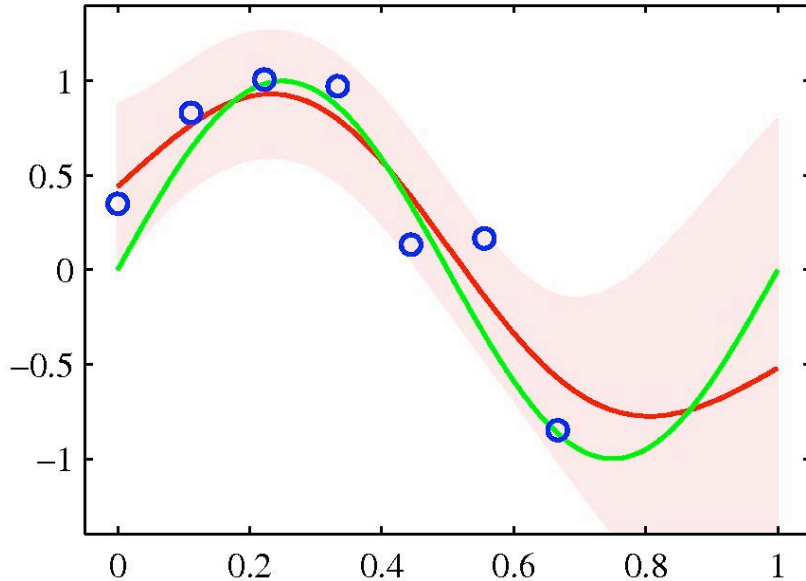
directly from conditional distribution formula for gaussians

Key results that define  
GP regression

Positive: hence the  
reduction in uncertainty

# Illustration

- GP regression applied to the sinusoidal data set



- The green curve shows the true function
- The blue data points are samples from the true function plus some additive Gaussian noise
- The red curve shows the mean of the GP predictive distribution, with shaded region corresponding to  $\pm 2$  standard deviations

- There is a restriction on the kernel function. The covariance matrix:

$$C(\mathbf{x}_n, \mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m) + \beta^{-1} \delta_{nm}.$$

must be positive definite



# Mean of Predictive Distribution

- Note that the **mean of the predictive distribution**

$$m(\mathbf{x}_{N+1}) = \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t}$$

can be written as a function of  $\mathbf{x}_{N+1}$ :

Linear combination

$$m(\mathbf{x}_{N+1}) = \sum_{n=1}^N a_n k(\mathbf{x}_n, \mathbf{x}_{N+1})$$

$a_n$  is the  $n^{\text{th}}$  component of  $\mathbf{C}_N^{-1} \mathbf{t}$

- Also, note that **the mean and variance** of the predictive distribution both depend on  $\mathbf{x}_{N+1}$ .

$$m(\mathbf{x}_{N+1}) = \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t}$$

$$\sigma^2(\mathbf{x}_{N+1}) = c - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k}$$

**Remember:**  $\mathbf{k}$  is the  $N \times 1$  vector with elements  $k(\mathbf{x}_n, \mathbf{x}_{N+1})$ .

give me another point that is similar to what the training data are doing

# Computational Complexity

- The central computation in using GPs will involve the inversion of an  $N \times N$  matrix  $\mathbf{C}_N$ , which is of order  $O(N^3)$ :

$$m(\mathbf{x}_{N+1}) = \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t}$$
$$\sigma^2(\mathbf{x}_{N+1}) = c - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k}$$

- By contrast, in the basis function model, we have to invert a matrix  $\mathbf{S}_N$  of size  $M \times M$  (where  $M$  is the number of basis functions).
- If the number of  $M$  basis functions is smaller than the number of  $N$  data points, then it will be computationally more efficient to work in the basis function framework (see the first few slides)
- The advantage of GPs is that we can consider covariance functions that can only be expressed in terms of an infinite number of basis functions.

# Learning the Hyperparameters

- The predictions of a GP regression model will depend on the **choice of the covariance function**.
- Instead of fixing the covariance function, we may prefer to use a parametric family of functions and **infer the parameter values from data**.
- These parameters may govern the length scale of the correlations or the precision of the noise model and **correspond to the hyperparameters in a standard parametric model**.

hyperparameters

The diagram shows the word "hyperparameters" at the top. Four blue arrows point from it to the parameters in the equation below:  $\theta_0$ ,  $\theta_1$ ,  $\theta_2$ , and  $\theta_3$ .


$$k(\mathbf{x}_n, \mathbf{x}_m) = \theta_0 \exp \left( -\frac{\theta_1}{2} \|\mathbf{x}_n - \mathbf{x}_m\|^2 \right) + \theta_2 + \theta_3 \mathbf{x}_n^T \mathbf{x}_m$$

- How can we infer the values of these parameters?

# Learning the Hyperparameters

- We can compute the **marginal likelihood function**:

$$p(\mathbf{t}|\theta) = \int p(\mathbf{t}|\mathbf{f}, \theta) p(\mathbf{f}|\theta) d\mathbf{f} = \mathcal{N}(\mathbf{t}|\mathbf{0}, \mathbf{C}_N),$$

 Hyperparameters of the GP model

- One option is to **maximize the log of the marginal likelihood** with respect to  $\theta$

$$\ln p(\mathbf{t}|\theta) = -\frac{1}{2} \ln |\mathbf{C}_N| - \frac{1}{2} \mathbf{t}^T \mathbf{C}_N^{-1} \mathbf{t} - \frac{N}{2} \ln(2\pi).$$

- This corresponds to the **type II maximum likelihood**, or **empirical Bayes**
- The maximization can be performed using **gradient-based optimization techniques**, such as conjugate gradients. The gradients take the form:

$$\frac{\partial}{\partial \theta_i} \ln p(\mathbf{t}|\theta) = -\frac{1}{2} \text{Tr} \left( \mathbf{C}_N^{-1} \frac{\partial \mathbf{C}_N}{\partial \theta_i} \right) + \frac{1}{2} \mathbf{t}^T \mathbf{C}_N^{-1} \frac{\partial \mathbf{C}_N}{\partial \theta_i} \mathbf{C}_N^{-1} \mathbf{t}.$$

# Learning the Hyperparameters

$$\ln p(\mathbf{t}|\theta) = -\frac{1}{2} \ln |\mathbf{C}_N| - \frac{1}{2} \mathbf{t}^T \mathbf{C}_N^{-1} \mathbf{t} - \frac{N}{2} \ln(2\pi).$$

- Because  $\ln p(\mathbf{t} | \theta)$  will be a **nonconvex function**, it will have **multiple maxima**.
- In the **fully Bayesian approach**, we can introduce a prior  $p(\theta)$  and infer the posterior  $p(\theta | \mathbf{t})$ .
- In general, the posterior will not have a closed form solution, so we must resort to approximations (typically MCMC).
- **Noise**: We have assumed that the additive noise, governed by  $\beta$ , **is constant**.

$$p(t_n | f_n) = \mathcal{N}(t_n | f_n, \beta^{-1}).$$

For some models, known as **heteroscedastic**, the noise variance itself will depend on  $\mathbf{x}$  – e.g. by introducing another GP that will model  $\log \beta(\mathbf{x})$

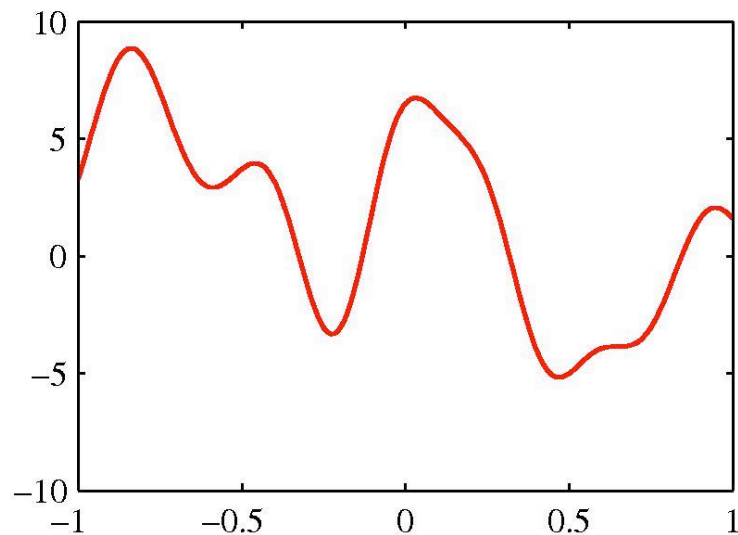
# Classification with GPs

- Consider a two-class problem with targets  $t \in \{0,1\}$
- Define a Gaussian process over a function  $f(\mathbf{x})$
- Transform the function using a sigmoid function:

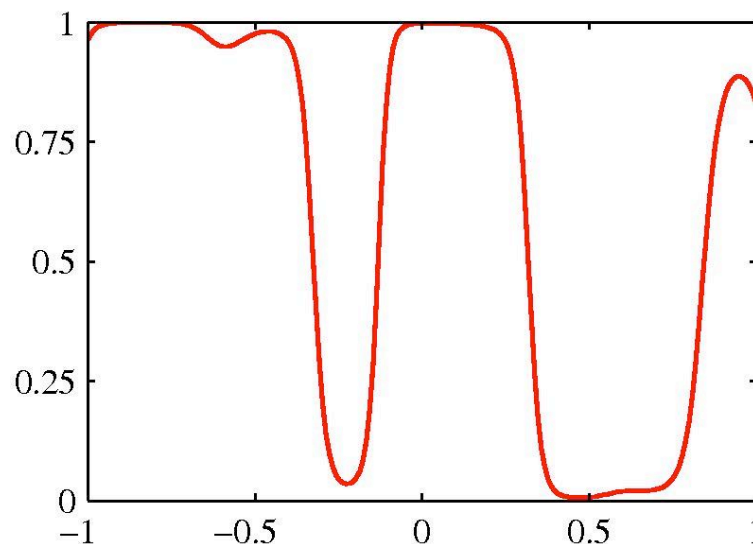
$$y(\mathbf{x}) = \sigma(f(\mathbf{x})) = \frac{1}{1 + \exp(-f(\mathbf{x}))}$$

- Hence  $y(\mathbf{x}) \in (0,1)$

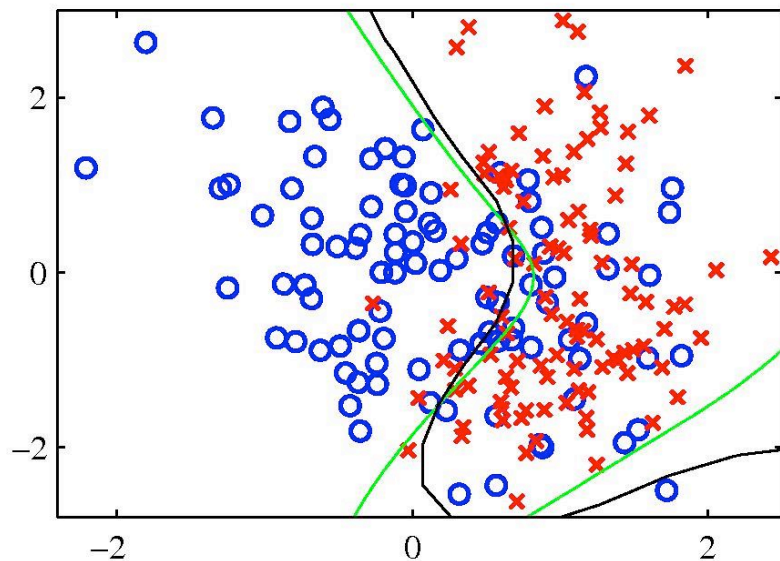
$f \sim \mathcal{GP}$



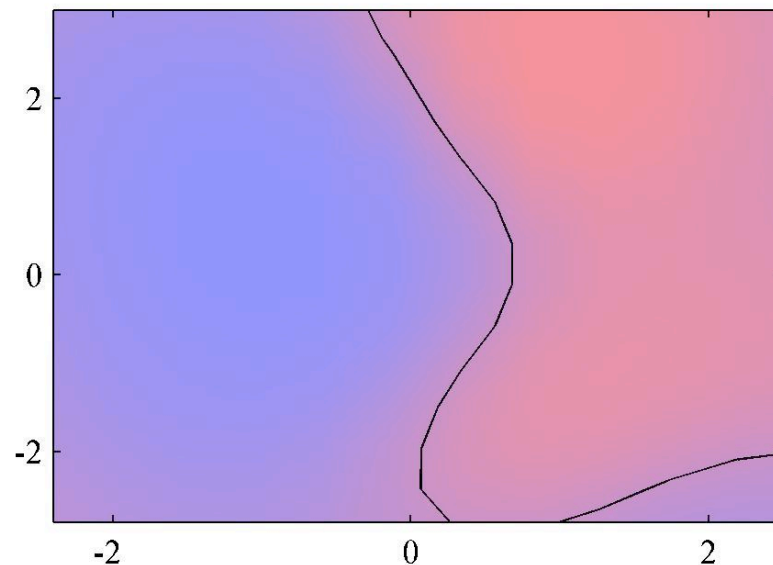
Transformed sample using  
sigmoid function



# Classification Results



Optimal decision boundary from the true distribution (green) and the decision boundary from GP classifier (black)



Predictive posterior probability together with GP decision boundary

# Gaussian Process Latent Variable Model

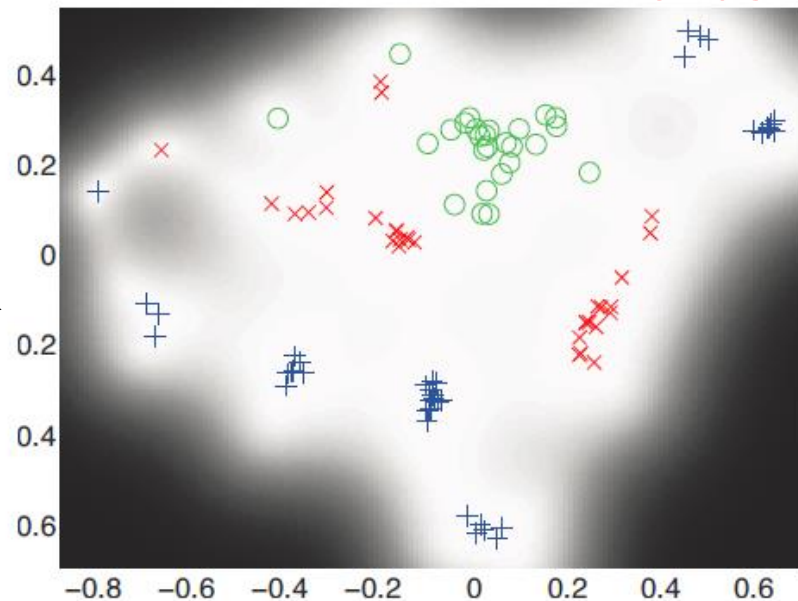
- Suppose you would like to reduce the dimensionality of  $n$  data points
- We have discussed so far PCA and neural-network autoencoders, but GPs can help here as well:
  - Assume that for the  $i$ th dimension of your dataset, the  $n$  elements are a sample from a Gaussian process based in a low-dimensional space
  - You can use the same GP for each dimension



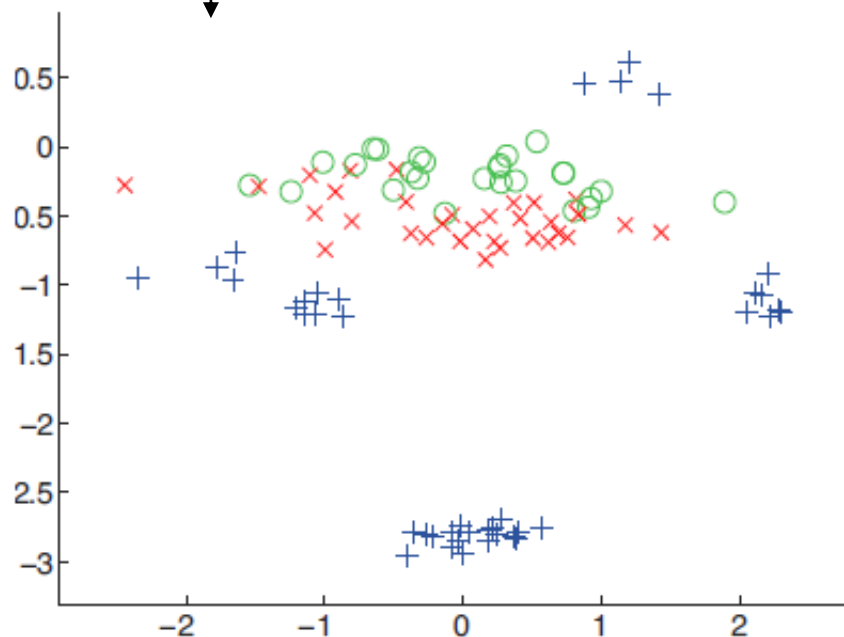
# Dimensionality reduction on the oil-flow dataset (Bishop p 678)

Bonus

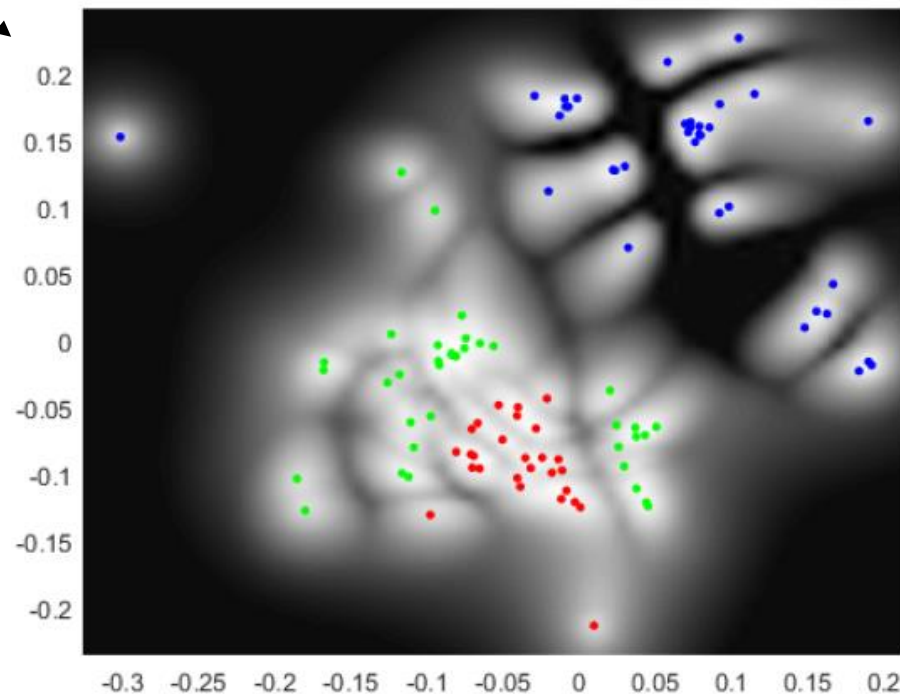
GP-LVM (Lawrence)



PCA



GP-LVM (Ebden)



# Applications of Gaussian Processes

- Regression
- Classification
- Unsupervised learning models
- Integration
- Global optimization
- More (see Chapter 9 of Rasmussen and Williams, 2006)

