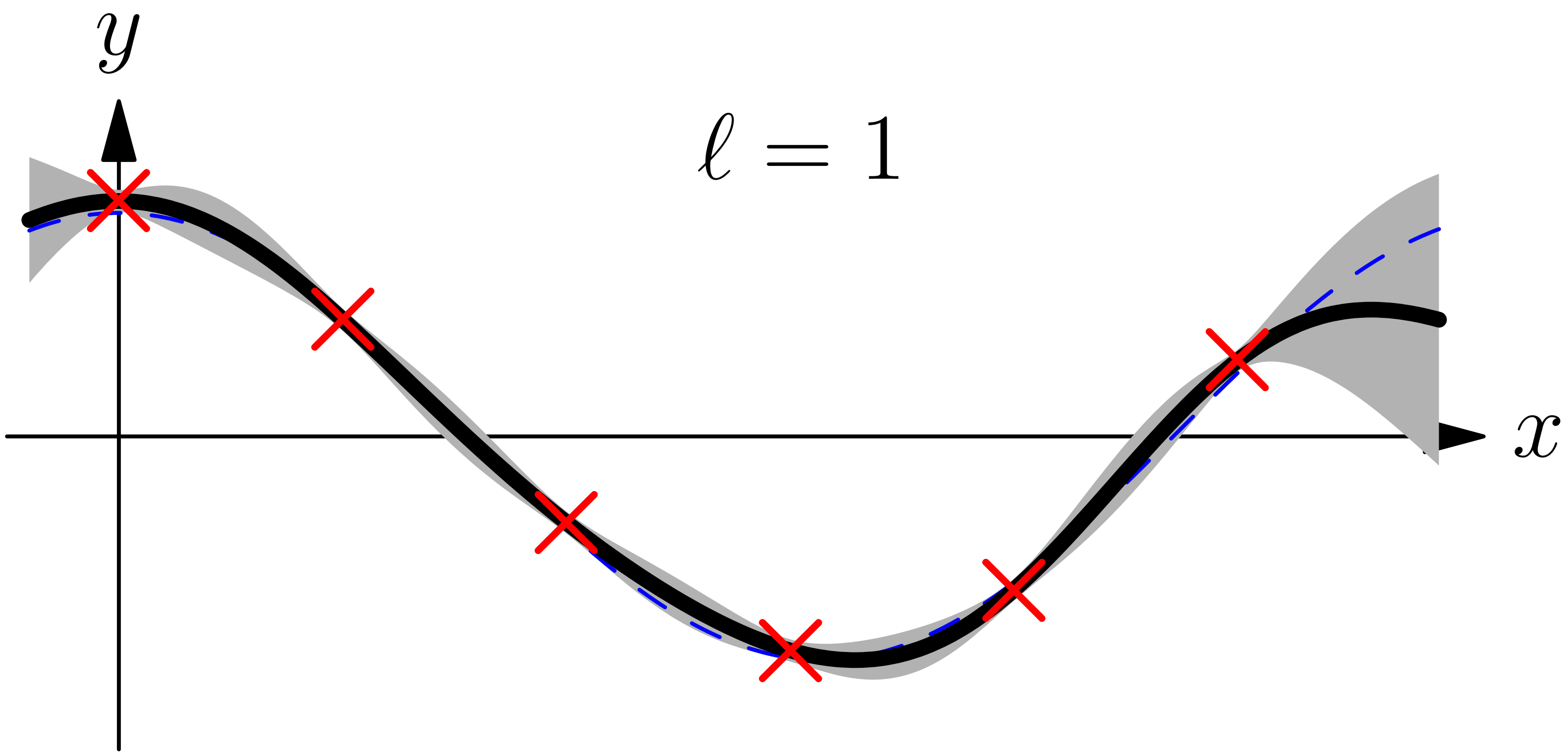
Advanced Machine Learning

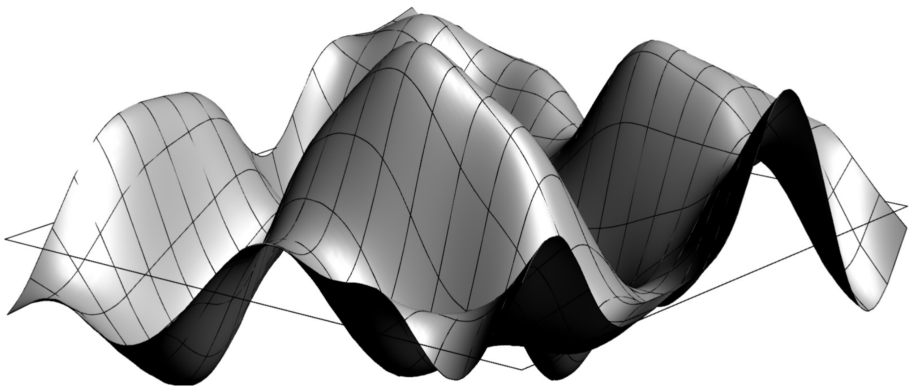
Gaussian Processes



Gaussian Processes, regression

Outline

1. **Introduction**
2. Gaussian Processes
3. Bayesian Inference
4. Hyper-parameters



Gaussian Proccesses

- Gaussian processes (GPs) are a mathematically defined ensemble of functions
- They can be combined with Bayesian inference to give one of the most powerful regression techniques
- Although Bayesian they can be used in a black-box fashion due to the ubiquity of the prior
- Mathematically they are a bit complicated
- In practice they aren't that difficult to use

Gaussian Proccesses

- Gaussian processes (GPs) are a mathematically defined ensemble of functions
- They can be combined with Bayesian inference to give one of the most powerful regression techniques
- Although Bayesian they can be used in a black-box fashion due to the ubiquity of the prior
- Mathematically they are a bit complicated
- In practice they aren't that difficult to use

Gaussian Proccesses

- Gaussian processes (GPs) are a mathematically defined ensemble of functions
- They can be combined with Bayesian inference to give one of the most powerful regression techniques
- Although Bayesian they can be used in a black-box fashion due to the ubiquity of the prior
- Mathematically they are a bit complicated
- In practice they aren't that difficult to use

Gaussian Proccesses

- Gaussian processes (GPs) are a mathematically defined ensemble of functions
- They can be combined with Bayesian inference to give one of the most powerful regression techniques
- Although Bayesian they can be used in a black-box fashion due to the ubiquity of the prior
- Mathematically they are a bit complicated
- In practice they aren't that difficult to use

Gaussian Proccesses

- Gaussian processes (GPs) are a mathematically defined ensemble of functions
- They can be combined with Bayesian inference to give one of the most powerful regression techniques
- Although Bayesian they can be used in a black-box fashion due to the ubiquity of the prior
- Mathematically they are a bit complicated (because Gaussians involve the inverse of matrices which are a real pain to work with)
- In practice they aren't that difficult to use

Gaussian Proccesses

- Gaussian processes (GPs) are a mathematically defined ensemble of functions
- They can be combined with Bayesian inference to give one of the most powerful regression techniques
- Although Bayesian they can be used in a black-box fashion due to the ubiquity of the prior
- Mathematically they are a bit complicated (because Gaussians involve the inverse of matrices which are a real pain to work with)
- In practice they aren't that difficult to use

Regression

- In regression we try to fit a multi-dimensional function to our data
- (You can use Gaussian Processes for classification, e.g. by inferring the probabilities of being in a class, but we ignore this as regression is where GP excel)
- In regression we have some p dimensional feature vectors $\boldsymbol{x}_i$ and some target $y_i \in \mathbb{R}$
- Our task is to fit a function through all the data points

Regression

- In regression we try to fit a multi-dimensional function to our data
- (You can use Gaussian Processes for classification, e.g. by inferring the probabilities of being in a class, but we ignore this as regression is where GP excel)
- In regression we have some p dimensional feature vectors $\boldsymbol{x}_i$ and some target $y_i \in \mathbb{R}$
- Our task is to fit a function through all the data points

Regression

- In regression we try to fit a multi-dimensional function to our data
- (You can use Gaussian Processes for classification, e.g. by inferring the probabilities of being in a class, but we ignore this as regression is where GP excel)
- In regression we have some p dimensional feature vectors $\boldsymbol{x}_i$ and some target $y_i \in \mathbb{R}$
- Our task is to fit a function through all the data points

Regression

- In regression we try to fit a multi-dimensional function to our data
- (You can use Gaussian Processes for classification, e.g. by inferring the probabilities of being in a class, but we ignore this as regression is where GP excel)
- In regression we have some p dimensional feature vectors $\boldsymbol{x}_i$ and some target $y_i \in \mathbb{R}$
- Our task is to fit a function through all the data points

Priors on Functions

- We can think of a solution as a function $f(\boldsymbol{x})$
- We can put a prior probability distribution, $p(f)$, on a function, f, that prefers smooth functions
- We can then compute a posterior probability distribution on functions given the data, $p(f|\mathcal{D})$
- As a likelihood, $p(y_i|f(\boldsymbol{x}_i))$, we use the probability of observing y_i given the true function value is $f(\boldsymbol{x}_i)$
- In general, this would be next to impossible to compute

Priors on Functions

- We can think of a solution as a function $f(\boldsymbol{x})$
- We can put a prior probability distribution, $p(f)$, on a function, f, that prefers smooth functions
- We can then compute a posterior probability distribution on functions given the data, $p(f|\mathcal{D})$
- As a likelihood, $p(y_i|f(\boldsymbol{x}_i))$, we use the probability of observing y_i given the true function value is $f(\boldsymbol{x}_i)$
- In general, this would be next to impossible to compute

Priors on Functions

- We can think of a solution as a function $f(\boldsymbol{x})$
- We can put a prior probability distribution, $p(f)$, on a function, f, that prefers smooth functions
- We can then compute a posterior probability distribution on functions given the data, $p(f|\mathcal{D})$
- As a likelihood, $p(y_i|f(\boldsymbol{x}_i))$, we use the probability of observing y_i given the true function value is $f(\boldsymbol{x}_i)$
- In general, this would be next to impossible to compute

Priors on Functions

- We can think of a solution as a function $f(\boldsymbol{x})$
- We can put a prior probability distribution, $p(f)$, on a function, f, that prefers smooth functions
- We can then compute a posterior probability distribution on functions given the data, $p(f|\mathcal{D})$
- As a likelihood, $p(y_i|f(\boldsymbol{x}_i))$, we use the probability of observing y_i given the true function value is $f(\boldsymbol{x}_i)$
- In general, this would be next to impossible to compute

Priors on Functions

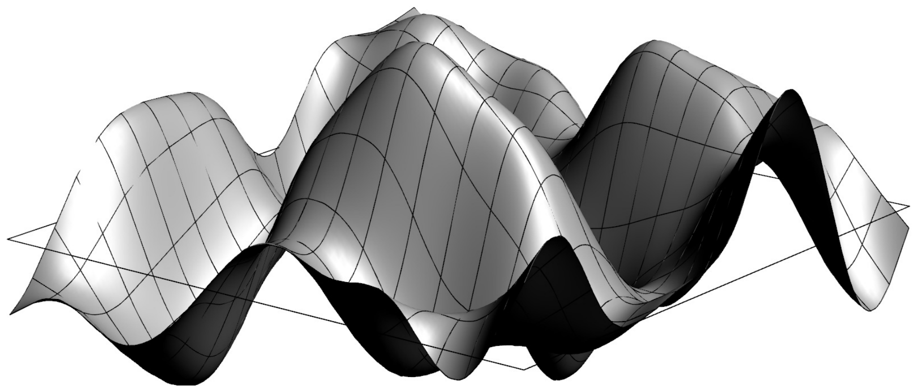
- We can think of a solution as a function $f(\boldsymbol{x})$
- We can put a prior probability distribution, $p(f)$, on a function, f, that prefers smooth functions
- We can then compute a posterior probability distribution on functions given the data, $p(f|\mathcal{D})$
- As a likelihood, $p(y_i|f(\boldsymbol{x}_i))$, we use the probability of observing y_i given the true function value is $f(\boldsymbol{x}_i)$
- In general, this would be next to impossible to compute

Priors on Functions

- We can think of a solution as a function $f(\boldsymbol{x})$
- We can put a prior probability distribution, $p(f)$, on a function, f, that prefers smooth functions
- We can then compute a posterior probability distribution on functions given the data, $p(f|\mathcal{D})$
- As a likelihood, $p(y_i|f(\boldsymbol{x}_i))$, we use the probability of observing y_i given the true function value is $f(\boldsymbol{x}_i)$
- In general, this would be next to impossible to compute, except in the special case where everything is Gaussian (normally) distributed

Outline

1. Introduction
2. **Gaussian Processes**
3. Bayesian Inference
4. Hyper-parameters



Gaussian Processes

- Gaussian Processes are probability distributions over functions
- (Functions can be viewed as vectors in an infinite dimensional vector space)
- In the Gaussian Process, $\mathcal{GP}(m,k)$, the probability of a function, f, is proportional

$$p(f|m,k) \propto \mathrm{e}^{-\frac{1}{2}\int (f(\boldsymbol{x}) - m(\boldsymbol{x}))k^{-1}(\boldsymbol{x},\boldsymbol{y})(f(\boldsymbol{y}) - m(\boldsymbol{y}))\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}}$$

- The function $m(\boldsymbol{x})$ is the mean $\mathbb{E}[f(\boldsymbol{x})]$ (usually taken to be zero in most inference problems)

Gaussian Processes

- Gaussian Processes are probability distributions over functions
- (Functions can be viewed as vectors in an infinite dimensional vector space)
- In the Gaussian Process, $\mathcal{GP}(m,k)$, the probability of a function, f, is proportional

$$p(f|m,k) \propto \mathrm{e}^{-\frac{1}{2}\int (f(\boldsymbol{x}) - m(\boldsymbol{x}))k^{-1}(\boldsymbol{x},\boldsymbol{y})(f(\boldsymbol{y}) - m(\boldsymbol{y}))\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}}$$

- The function $m(\boldsymbol{x})$ is the mean $\mathbb{E}[f(\boldsymbol{x})]$ (usually taken to be zero in most inference problems)

Gaussian Processes

- Gaussian Processes are probability distributions over functions
- (Functions can be viewed as vectors in an infinite dimensional vector space)
- In the Gaussian Process, $\mathcal{GP}(m,k)$, the probability of a function, f, is proportional

$$p(f|m,k) \propto \mathrm{e}^{-\frac{1}{2}\int (f(\boldsymbol{x}) - m(\boldsymbol{x}))k^{-1}(\boldsymbol{x},\boldsymbol{y})(f(\boldsymbol{y}) - m(\boldsymbol{y}))\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}}$$

- The function $m(\boldsymbol{x})$ is the mean $\mathbb{E}[f(\boldsymbol{x})]$ (usually taken to be zero in most inference problems)

Gaussian Processes

- Gaussian Processes are probability distributions over functions
- (Functions can be viewed as vectors in an infinite dimensional vector space)
- In the Gaussian Process, $\mathcal{GP}(m,k)$, the probability of a function, f, is proportional

$$p(f|m,k) \propto \mathrm{e}^{-\frac{1}{2}\int (f(\boldsymbol{x}) - m(\boldsymbol{x}))k^{-1}(\boldsymbol{x},\boldsymbol{y})(f(\boldsymbol{y}) - m(\boldsymbol{y}))\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}}$$

- The function $m(\boldsymbol{x})$ is the mean $\mathbb{E}[f(\boldsymbol{x})]$ (usually taken to be zero in most inference problems)

Meaning of GP

- To understand GP's we can discretise space, $\boldsymbol{x}$, into a lattice of points $\{\boldsymbol{x}_i\}$

- Then (assuming $m(\boldsymbol{x}) = 0$)

$$p(f|m,k) \propto \prod_i \mathrm{e}^{-\frac{f_i^2 k^{-1}(\boldsymbol{x}_i,\boldsymbol{x}_i)}{2} + f_i \sum_j k^{-1}(\boldsymbol{x}_i,\boldsymbol{x}_j) f_j}$$

  where $f_i = f(\boldsymbol{x}_i)$

- We see that the value of the function at each point is normally distributed with a mean that depends on functions at neighbouring points

Meaning of GP

- To understand GP's we can discretise space, $\boldsymbol{x}$, into a lattice of points $\{\boldsymbol{x}_i\}$

- Then (assuming $m(\boldsymbol{x}) = 0$)

$$p(f|m,k) \propto \prod_i \mathrm{e}^{-\frac{f_i^2 k^{-1}(\boldsymbol{x}_i,\boldsymbol{x}_i)}{2} + f_i \sum_j k^{-1}(\boldsymbol{x}_i,\boldsymbol{x}_j) f_j}$$

  where $f_i = f(\boldsymbol{x}_i)$

- We see that the value of the function at each point is normally distributed with a mean that depends on functions at neighbouring points

Meaning of GP

- To understand GP's we can discretise space, $\boldsymbol{x}$, into a lattice of points $\{\boldsymbol{x}_i\}$
- Then (assuming $m(\boldsymbol{x}) = 0$)

$$p(f|m,k) \propto \prod_i \mathrm{e}^{-\frac{f_i^2 k^{-1}(\boldsymbol{x}_i,\boldsymbol{x}_i)}{2} + f_i \sum_j k^{-1}(\boldsymbol{x}_i,\boldsymbol{x}_j) f_j}$$

  where $f_i = f(\boldsymbol{x}_i)$

- We see that the value of the function at each point is normally distributed with a mean that depends on functions at neighbouring points

Covariance function

- $k(\boldsymbol{x}, \boldsymbol{y})$ is a covariance function

$$\mathbb{E}\Big[\big(f(\boldsymbol{x}) - m(\boldsymbol{x})\big)\big(f(\boldsymbol{y}) - m(\boldsymbol{y})\big)\Big] = k(\boldsymbol{x}, \boldsymbol{y})$$

- This is sometimes know as a **kernel**—it must be positive semi-definite

- It is a free "parameter" that the user gets to choose (although we can learn its parameters too)

- If $k(\boldsymbol{x}, \boldsymbol{y})$ is a function of $\boldsymbol{x} - \boldsymbol{y}$ it is **"stationary"**

- If $k(\boldsymbol{x}, \boldsymbol{y})$ is a function of $\|\boldsymbol{x} - \boldsymbol{y}\|$ it is also **"isometric"**

Covariance function

- $k(\boldsymbol{x},\boldsymbol{y})$ is a covariance function

$$\mathbb{E}\big[\big(f(\boldsymbol{x}) - m(\boldsymbol{x})\big)\big(f(\boldsymbol{y}) - m(\boldsymbol{y})\big)\big] = k(\boldsymbol{x},\boldsymbol{y})$$

- This is sometimes know as a **kernel**—it must be positive semi-definite

- It is a free "parameter" that the user gets to choose (although we can learn its parameters too)

- If $k(\boldsymbol{x},\boldsymbol{y})$ is a function of $\boldsymbol{x} - \boldsymbol{y}$ it is **"stationary"**

- If $k(\boldsymbol{x},\boldsymbol{y})$ is a function of $\|\boldsymbol{x} - \boldsymbol{y}\|$ it is also **"isometric"**

Covariance function

- $k(\boldsymbol{x},\boldsymbol{y})$ is a covariance function

$$\mathbb{E}\Big[\big(f(\boldsymbol{x}) - m(\boldsymbol{x})\big)\big(f(\boldsymbol{y}) - m(\boldsymbol{y})\big)\Big] = k(\boldsymbol{x},\boldsymbol{y})$$

- This is sometimes know as a **kernel**—it must be positive semi-definite
- It is a free "parameter" that the user gets to choose (although we can learn its parameters too)
- If $k(\boldsymbol{x},\boldsymbol{y})$ is a function of $\boldsymbol{x} - \boldsymbol{y}$ it is **"stationary"**
- If $k(\boldsymbol{x},\boldsymbol{y})$ is a function of $\|\boldsymbol{x} - \boldsymbol{y}\|$ it is also **"isometric"**

Covariance function

- $k(\boldsymbol{x}, \boldsymbol{y})$ is a covariance function

$$\mathbb{E}\left[\left(f(\boldsymbol{x}) - m(\boldsymbol{x})\right)\left(f(\boldsymbol{y}) - m(\boldsymbol{y})\right)\right] = k(\boldsymbol{x}, \boldsymbol{y})$$

- This is sometimes know as a **kernel**—it must be positive semi-definite (just like in SVMs)
- It is a free "parameter" that the user gets to choose (although we can learn its parameters too)
- If $k(\boldsymbol{x}, \boldsymbol{y})$ is a function of $\boldsymbol{x} - \boldsymbol{y}$ it is **"stationary"**
- If $k(\boldsymbol{x}, \boldsymbol{y})$ is a function of $\|\boldsymbol{x} - \boldsymbol{y}\|$ it is also **"isometric"**

Covariance function

- $k(\boldsymbol{x},\boldsymbol{y})$ is a covariance function

$$\mathbb{E}\Big[\big(f(\boldsymbol{x}) - m(\boldsymbol{x})\big)\big(f(\boldsymbol{y}) - m(\boldsymbol{y})\big)\Big] = k(\boldsymbol{x},\boldsymbol{y})$$

- This is sometimes know as a **kernel**—it must be positive semi-definite (just like in SVMs)

- It is a free "parameter" that the user gets to choose (although we can learn its parameters too)

- If $k(\boldsymbol{x},\boldsymbol{y})$ is a function of $\boldsymbol{x} - \boldsymbol{y}$ it is **"stationary"**

- If $k(\boldsymbol{x},\boldsymbol{y})$ is a function of $\|\boldsymbol{x} - \boldsymbol{y}\|$ it is also **"isometric"**

Covariance function

- $k(\boldsymbol{x},\boldsymbol{y})$ is a covariance function

$$\mathbb{E}\Big[\big(f(\boldsymbol{x}) - m(\boldsymbol{x})\big)\big(f(\boldsymbol{y}) - m(\boldsymbol{y})\big)\Big] = k(\boldsymbol{x},\boldsymbol{y})$$

- This is sometimes know as a **kernel**—it must be positive semi-definite (just like in SVMs)

- It is a free "parameter" that the user gets to choose (although we can learn its parameters too)

- If $k(\boldsymbol{x},\boldsymbol{y})$ is a function of $\boldsymbol{x} - \boldsymbol{y}$ it is **"stationary"**

- If $k(\boldsymbol{x},\boldsymbol{y})$ is a function of $\|\boldsymbol{x} - \boldsymbol{y}\|$ it is also **"isometric"**

Covariance function

- $k(\boldsymbol{x},\boldsymbol{y})$ is a covariance function

$$\mathbb{E}\Big[\big(f(\boldsymbol{x}) - m(\boldsymbol{x})\big)\big(f(\boldsymbol{y}) - m(\boldsymbol{y})\big)\Big] = k(\boldsymbol{x},\boldsymbol{y})$$

- This is sometimes know as a **kernel**—it must be positive semi-definite (just like in SVMs)

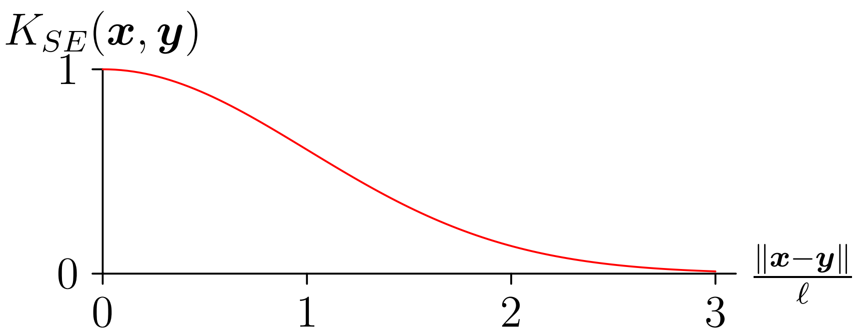
- It is a free "parameter" that the user gets to choose (although we can learn its parameters too)

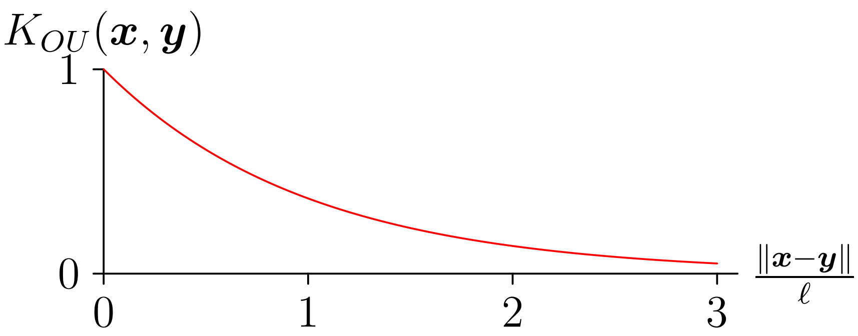
- If $k(\boldsymbol{x},\boldsymbol{y})$ is a function of $\boldsymbol{x} - \boldsymbol{y}$ it is **"stationary"**

- If $k(\boldsymbol{x},\boldsymbol{y})$ is a function of $\|\boldsymbol{x} - \boldsymbol{y}\|$ it is also **"isometric"**

Popular Choices of GP Kernel Function

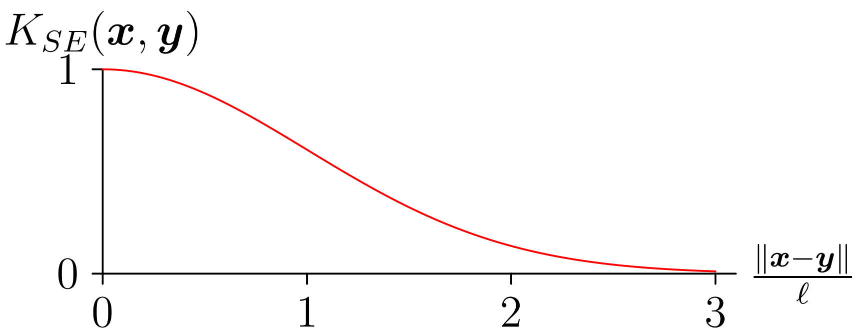
- Constant: $k_{\mathrm{C}}(\boldsymbol{x},\boldsymbol{y}) = C$
- Gaussian noise: $k_{\mathrm{GN}}(\boldsymbol{x},\boldsymbol{y}) = \sigma^2 \delta_{\boldsymbol{x},\boldsymbol{y}}$
- Squared exponential: $k_{\mathrm{SE}}(\boldsymbol{x},\boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\ell^2}\right)$
- Ornstein–Uhlenbeck: $k_{\mathrm{OU}}(\boldsymbol{x},\boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|}{\ell}\right)$

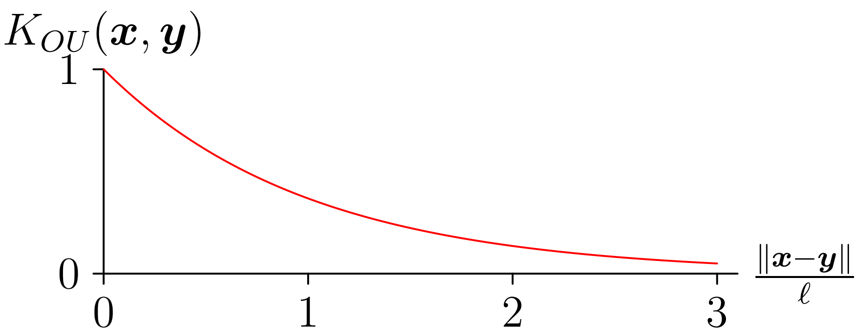




Popular Choices of GP Kernel Function

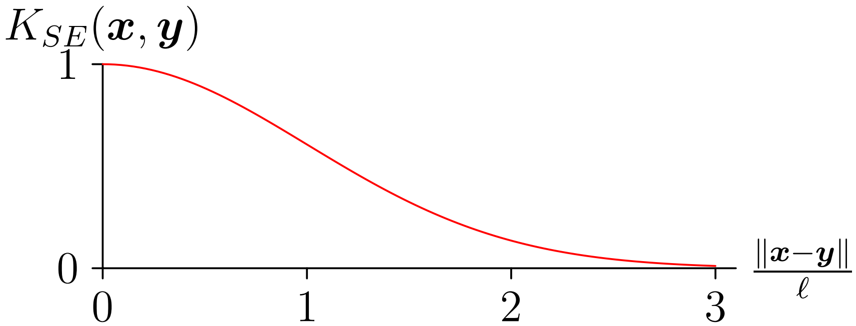
- Constant: $k_{\mathrm{C}}(\boldsymbol{x}, \boldsymbol{y}) = C$
- Gaussian noise: $k_{\mathrm{GN}}(\boldsymbol{x}, \boldsymbol{y}) = \sigma^2 \delta_{\boldsymbol{x}, \boldsymbol{y}}$
- Squared exponential: $k_{\mathrm{SE}}(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{2\ell^2}\right)$
- Ornstein–Uhlenbeck: $k_{\mathrm{OU}}(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|}{\ell}\right)$

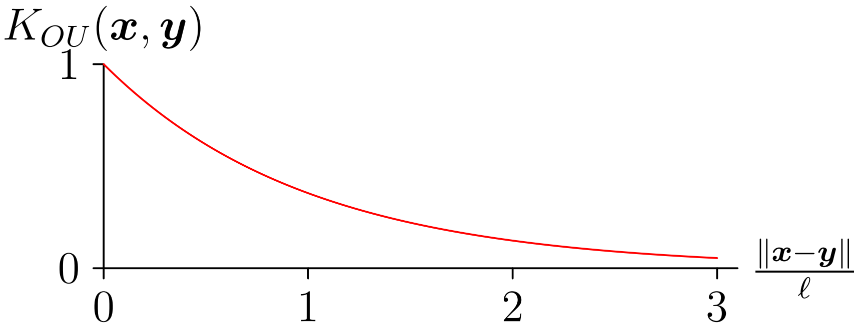




Popular Choices of GP Kernel Function

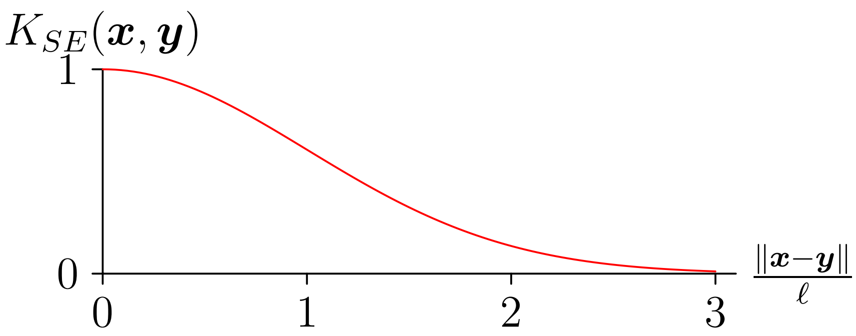
- Constant: $k_{\mathrm{C}}(\boldsymbol{x},\boldsymbol{y}) = C$
- Gaussian noise: $k_{\mathrm{GN}}(\boldsymbol{x},\boldsymbol{y}) = \sigma^2 \delta_{\boldsymbol{x},\boldsymbol{y}}$
- Squared exponential: $k_{\mathrm{SE}}(\boldsymbol{x},\boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\ell^2}\right)$
- Ornstein–Uhlenbeck: $k_{\mathrm{OU}}(\boldsymbol{x},\boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|}{\ell}\right)$

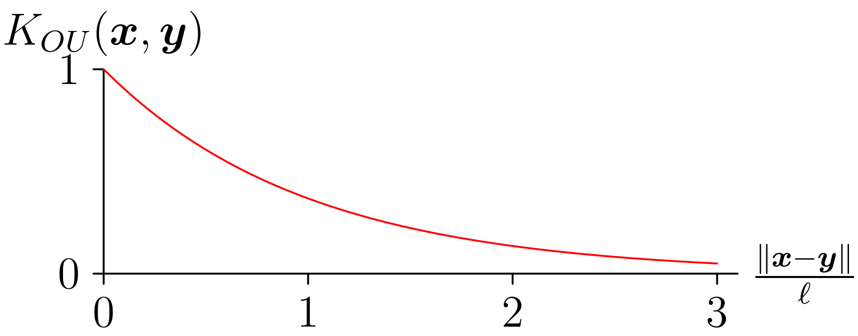




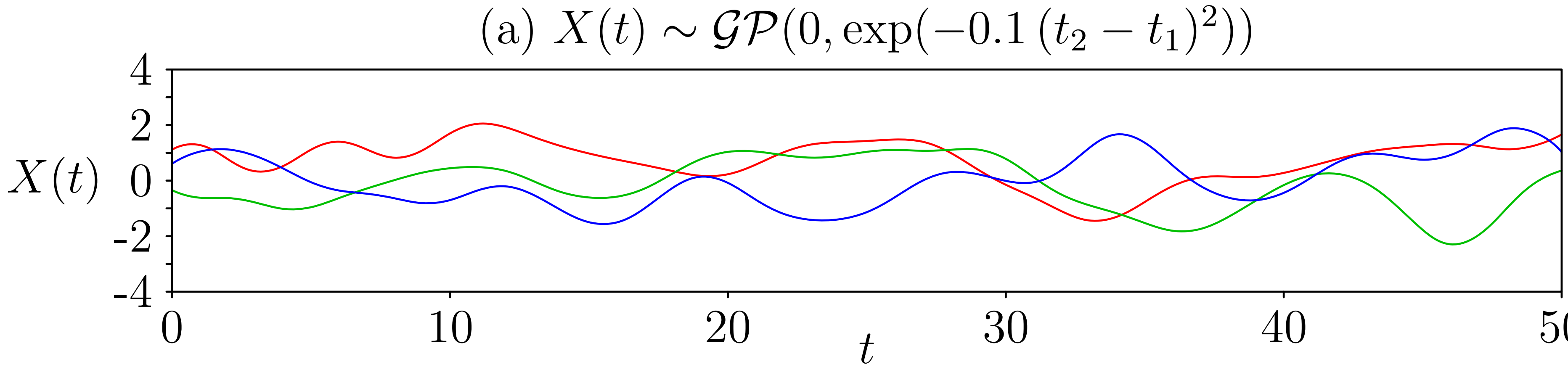
Popular Choices of GP Kernel Function

- Constant: $k_{\mathrm{C}}(\boldsymbol{x},\boldsymbol{y}) = C$
- Gaussian noise: $k_{\mathrm{GN}}(\boldsymbol{x},\boldsymbol{y}) = \sigma^2 \delta_{\boldsymbol{x},\boldsymbol{y}}$
- Squared exponential: $k_{\mathrm{SE}}(\boldsymbol{x},\boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\ell^2}\right)$
- Ornstein–Uhlenbeck: $k_{\mathrm{OU}}(\boldsymbol{x},\boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|}{\ell}\right)$

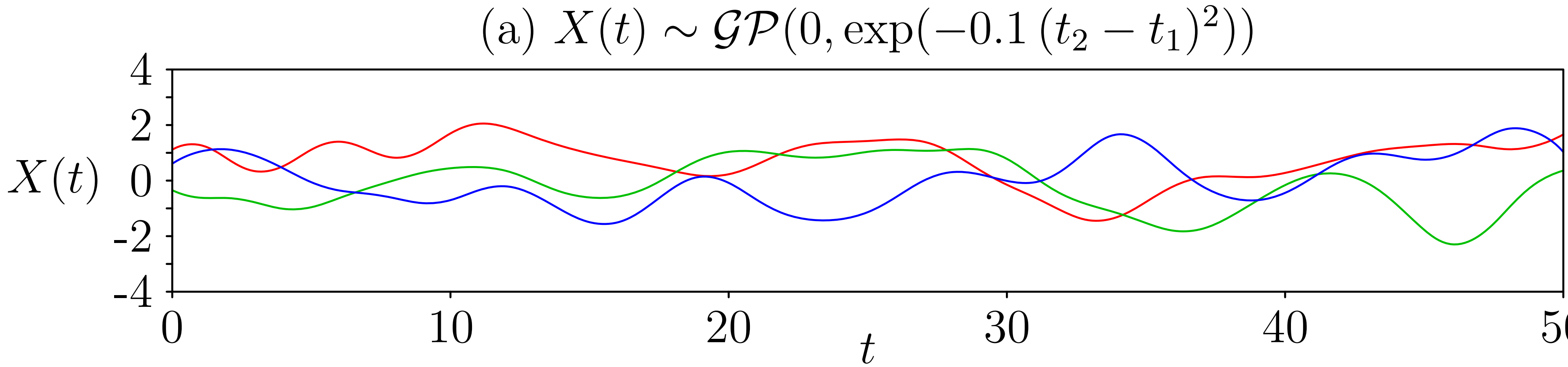


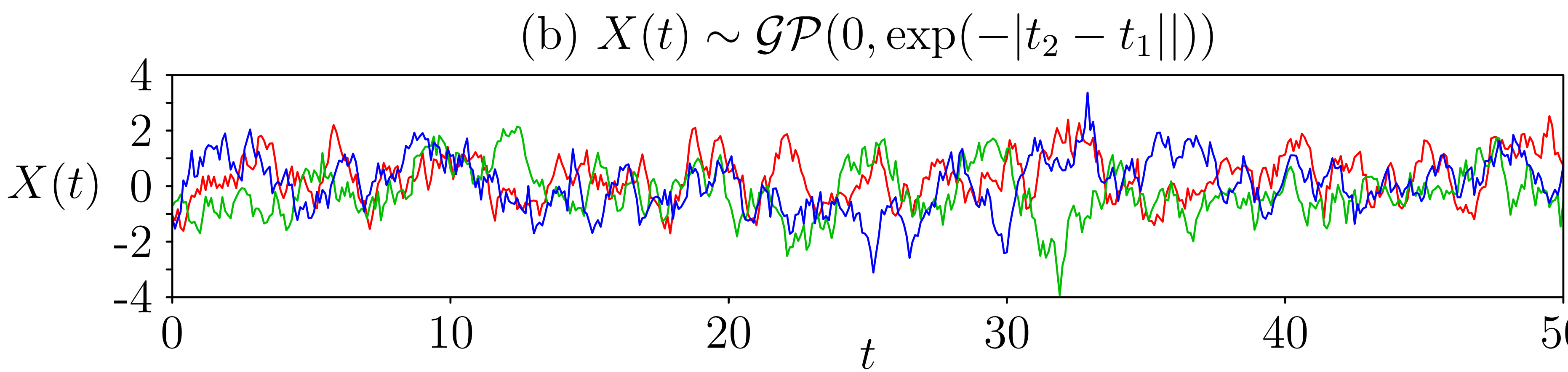


Gaussian Process Worlds

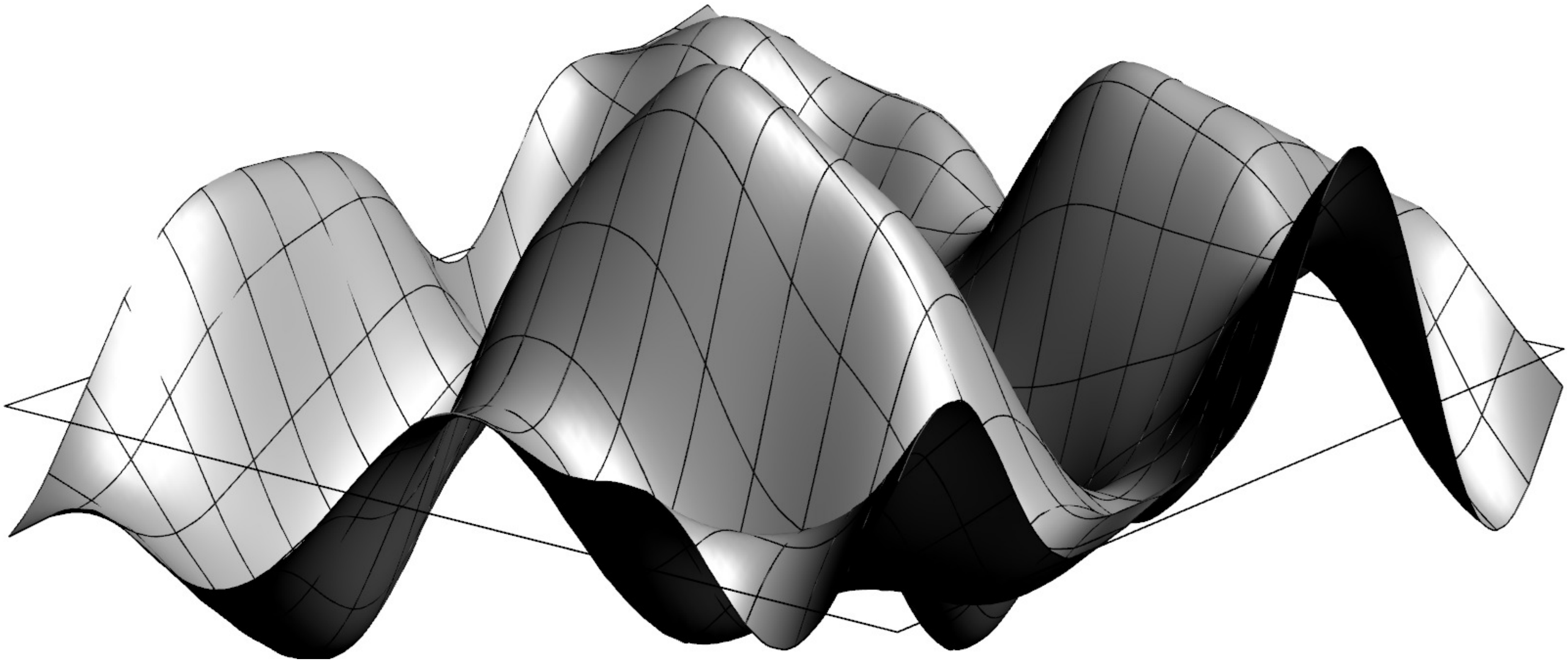


Gaussian Process Worlds



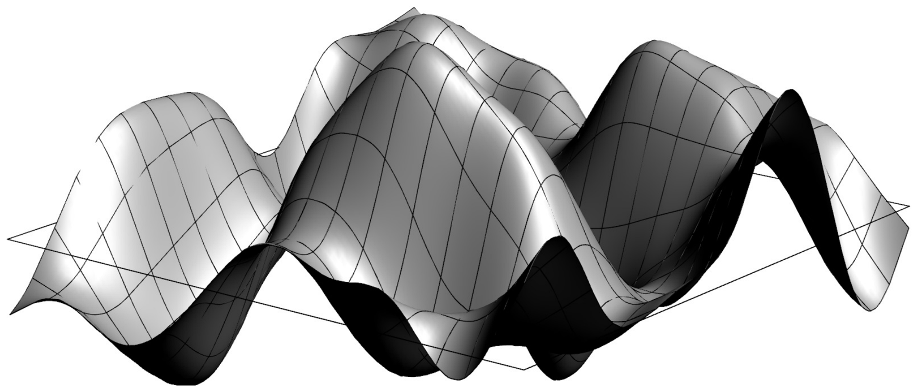


2-D Gaussian Processes



Outline

1. Introduction
2. Gaussian Processes
3. **Bayesian Inference**
4. Hyper-parameters



Observed Gaussian Processes

- Given some data points $\mathcal{D} = \big((\boldsymbol{x}_i, y_i)\big| i = 1, \ldots, m\big)$ the likelihood (assuming Gaussian error are independence of the data point) is given by

$$p(\mathcal{D}|f) = \prod_{i=1}^{m} \mathcal{N}\left(y_i \middle| f(\boldsymbol{x}_i), \sigma^2\right)$$

- Using a Gausssian Process prior we can compute a posterior using Bayes's rule

- The posterior is a Gaussian Process with a shifted mean and variance depending on the data-points

- This direct Bayesian derivation gives the answer involving the inverse matrix of the correlation function, $k^{-1}(\boldsymbol{x}, \boldsymbol{y})$

Observed Gaussian Processes

- Given some data points $\mathcal{D} = \big((\boldsymbol{x}_i, y_i)\big| i = 1, \ldots, m\big)$ the likelihood (assuming Gaussian error are independence of the data point) is given by

$$p(\mathcal{D}|f) = \prod_{i=1}^{m} \mathcal{N}\left(y_i \middle| f(\boldsymbol{x}_i), \sigma^2\right)$$

- Using a Gausssian Process prior we can compute a posterior using Bayes's rule

- The posterior is a Gaussian Process with a shifted mean and variance depending on the data-points

- This direct Bayesian derivation gives the answer involving the inverse matrix of the correlation function, $k^{-1}(\boldsymbol{x}, \boldsymbol{y})$

Observed Gaussian Processes

- Given some data points $\mathcal{D} = \big((\boldsymbol{x}_i, y_i)\big| i = 1, \ldots, m\big)$ the likelihood (assuming Gaussian error are independence of the data point) is given by

$$p(\mathcal{D}|f) = \prod_{i=1}^{m} \mathcal{N}\left(y_i \middle| f(\boldsymbol{x}_i), \sigma^2\right)$$

- Using a Gausssian Process prior we can compute a posterior using Bayes's rule

- The posterior is a Gaussian Process with a shifted mean and variance depending on the data-points

- This direct Bayesian derivation gives the answer involving the inverse matrix of the correlation function, $k^{-1}(\boldsymbol{x}, \boldsymbol{y})$

Observed Gaussian Processes

- Given some data points $\mathcal{D} = \big((\boldsymbol{x}_i, y_i) \big| i = 1, \ldots, m \big)$ the likelihood (assuming Gaussian error are independence of the data point) is given by

$$p(\mathcal{D}|f) = \prod_{i=1}^{m} \mathcal{N}\left(y_i \middle| f(\boldsymbol{x}_i), \sigma^2 \right)$$

- Using a Gausssian Process prior we can compute a posterior using Bayes's rule

- The posterior is a Gaussian Process with a shifted mean and variance depending on the data-points

- This direct Bayesian derivation gives the answer involving the inverse matrix of the correlation function, $k^{-1}(\boldsymbol{x}, \boldsymbol{y})$

Observed Gaussian Processes

- Given some data points $\mathcal{D} = \big((\boldsymbol{x}_i, y_i)\big| i = 1, \ldots, m\big)$ the likelihood (assuming Gaussian error are independence of the data point) is given by

$$p(\mathcal{D}|f) = \prod_{i=1}^{m} \mathcal{N}\left(y_i \middle| f(\boldsymbol{x}_i), \sigma^2\right)$$

- Using a Gausssian Process prior we can compute a posterior using Bayes's rule

- The posterior is a Gaussian Process with a shifted mean and variance depending on the data-points

- This direct Bayesian derivation gives the answer involving the inverse matrix of the correlation function, $k^{-1}(\boldsymbol{x}, \boldsymbol{y})$—this is a pain to work with

Alternative Derivation

- Denoting the target values as a vector $\boldsymbol{y}$ with elements y_i
- Denoting the matrices of covariances between data points as $\mathbf{K}$ with elements $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- Denoting the covariance between the data points and a particular position, $\boldsymbol{x}_*$ as $\boldsymbol{k}_*$ with elements $k(\boldsymbol{x}_i, \boldsymbol{x}_*)$
- Denoting the variance a point $\boldsymbol{x}_*$ as $k_* = k(\boldsymbol{x}_*, \boldsymbol{x}_*)$
- Then the distribution of function values at points at $\boldsymbol{x}_i$ and $\boldsymbol{x}_*$ is

$$p(\boldsymbol{y}, f_*) = \mathcal{N}\left(\begin{pmatrix} \boldsymbol{y} \\ f_* \end{pmatrix} \middle| \mathbf{0}, \begin{pmatrix} \mathbf{K} + \sigma^2 \mathbf{I} & \boldsymbol{k}_* \\ \boldsymbol{k}_*^\mathsf{T} & k_* \end{pmatrix} \right)$$

Alternative Derivation

- Denoting the target values as a vector $\boldsymbol{y}$ with elements y_i
- Denoting the matrices of covariances between data points as $\mathbf{K}$ with elements $k(\boldsymbol{x}_i,\boldsymbol{x}_j)$
- Denoting the covariance between the data points and a particular position, $\boldsymbol{x}_*$ as $\boldsymbol{k}_*$ with elements $k(\boldsymbol{x}_i,\boldsymbol{x}_*)$
- Denoting the variance a point $\boldsymbol{x}_*$ as $k_* = k(\boldsymbol{x}_*,\boldsymbol{x}_*)$
- Then the distribution of function values at points at $\boldsymbol{x}_i$ and $\boldsymbol{x}_*$ is

$$p(\boldsymbol{y}, f_*) = \mathcal{N}\left(\begin{pmatrix} \boldsymbol{y} \\ f_* \end{pmatrix} \middle| \mathbf{0}, \begin{pmatrix} \mathbf{K} + \sigma^2 \mathbf{I} & \boldsymbol{k}_* \\ \boldsymbol{k}_*^\top & k_* \end{pmatrix} \right)$$

Alternative Derivation

- Denoting the target values as a vector $\boldsymbol{y}$ with elements y_i
- Denoting the matrices of covariances between data points as $\mathbf{K}$ with elements $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- Denoting the covariance between the data points and a particular position, $\boldsymbol{x}_*$ as $\boldsymbol{k}_*$ with elements $k(\boldsymbol{x}_i, \boldsymbol{x}_*)$
- Denoting the variance a point $\boldsymbol{x}_*$ as $k_* = k(\boldsymbol{x}_*, \boldsymbol{x}_*)$
- Then the distribution of function values at points at $\boldsymbol{x}_i$ and $\boldsymbol{x}_*$ is

$$p(\boldsymbol{y}, f_*) = \mathcal{N}\left(\begin{pmatrix} \boldsymbol{y} \\ f_* \end{pmatrix} \middle| \mathbf{0}, \begin{pmatrix} \mathbf{K} + \sigma^2 \mathbf{I} & \boldsymbol{k}_* \\ \boldsymbol{k}_*^\top & k_* \end{pmatrix} \right)$$

Alternative Derivation

- Denoting the target values as a vector $\boldsymbol{y}$ with elements y_i
- Denoting the matrices of covariances between data points as $\mathbf{K}$ with elements $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- Denoting the covariance between the data points and a particular position, $\boldsymbol{x}_*$ as $\boldsymbol{k}_*$ with elements $k(\boldsymbol{x}_i, \boldsymbol{x}_*)$
- Denoting the variance a point $\boldsymbol{x}_*$ as $k_* = k(\boldsymbol{x}_*, \boldsymbol{x}_*)$
- Then the distribution of function values at points at $\boldsymbol{x}_i$ and $\boldsymbol{x}_*$ is

$$p(\boldsymbol{y}, f_*) = \mathcal{N}\left(\begin{pmatrix} \boldsymbol{y} \\ f_* \end{pmatrix} \middle| \mathbf{0}, \begin{pmatrix} \mathbf{K} + \sigma^2 \mathbf{I} & \boldsymbol{k}_* \\ \boldsymbol{k}_*^\top & k_* \end{pmatrix} \right)$$

Alternative Derivation

- Denoting the target values as a vector $\boldsymbol{y}$ with elements y_i
- Denoting the matrices of covariances between data points as $\mathbf{K}$ with elements $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- Denoting the covariance between the data points and a particular position, $\boldsymbol{x}_*$ as $\boldsymbol{k}_*$ with elements $k(\boldsymbol{x}_i, \boldsymbol{x}_*)$
- Denoting the variance a point $\boldsymbol{x}_*$ as $k_* = k(\boldsymbol{x}_*, \boldsymbol{x}_*)$
- Then the distribution of function values at points at $\boldsymbol{x}_i$ and $\boldsymbol{x}_*$ is

$$p(\boldsymbol{y}, f_*) = \mathcal{N}\left(\begin{pmatrix} \boldsymbol{y} \\ f_* \end{pmatrix} \middle| \mathbf{0}, \begin{pmatrix} \mathbf{K} + \sigma^2 \mathbf{I} & \boldsymbol{k}_* \\ \boldsymbol{k}_*^\top & k_* \end{pmatrix} \right)$$

Conditional Probability

- To compute the posterior $p(f_*|\boldsymbol{y})$ we use

$$p(f_*|\boldsymbol{y}) = \frac{p(f_*, \boldsymbol{y})}{p(\boldsymbol{y})}$$

- where $p(\boldsymbol{y}) = \int p(f_*, \boldsymbol{y}) \mathrm{d} f_*$

- Because all integrals are Gaussian we can compute the integral to obtain

$$p(f_*|\boldsymbol{y}) = \mathcal{N}\left(f_* \middle| \boldsymbol{k}_*^{\mathsf{T}} (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \boldsymbol{y}, k - \boldsymbol{k}_*^{\mathsf{T}} (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \boldsymbol{k}_* \right)$$

- Looks complicated, but numerically easy to evaluate

Conditional Probability

- To compute the posterior $p(f_*|\boldsymbol{y})$ we use

$$p(f_*|\boldsymbol{y}) = \frac{p(f_*, \boldsymbol{y})}{p(\boldsymbol{y})}$$

- where $p(\boldsymbol{y}) = \int p(f_*, \boldsymbol{y}) \mathrm{d} f_*$

- Because all integrals are Gaussian we can compute the integral to obtain

$$p(f_*|\boldsymbol{y}) = \mathcal{N}\left(f_* \middle| \boldsymbol{k}_*^{\mathsf{T}} (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \boldsymbol{y}, k - \boldsymbol{k}_*^{\mathsf{T}} (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \boldsymbol{k}_* \right)$$

- Looks complicated, but numerically easy to evaluate

Conditional Probability

- To compute the posterior $p(f_*|\boldsymbol{y})$ we use

$$p(f_*|\boldsymbol{y}) = \frac{p(f_*, \boldsymbol{y})}{p(\boldsymbol{y})}$$

- where $p(\boldsymbol{y}) = \int p(f_*, \boldsymbol{y}) \mathrm{d} f_*$

- Because all integrals are Gaussian we can compute the integral to obtain

$$p(f_*|\boldsymbol{y}) = \mathcal{N}\left(f_* \middle| \boldsymbol{k}_*^{\mathsf{T}} (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \boldsymbol{y}, k - \boldsymbol{k}_*^{\mathsf{T}} (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \boldsymbol{k}_* \right)$$

- Looks complicated, but numerically easy to evaluate

Conditional Probability

- To compute the posterior $p(f_*|\boldsymbol{y})$ we use

$$p(f_*|\boldsymbol{y}) = \frac{p(f_*, \boldsymbol{y})}{p(\boldsymbol{y})}$$

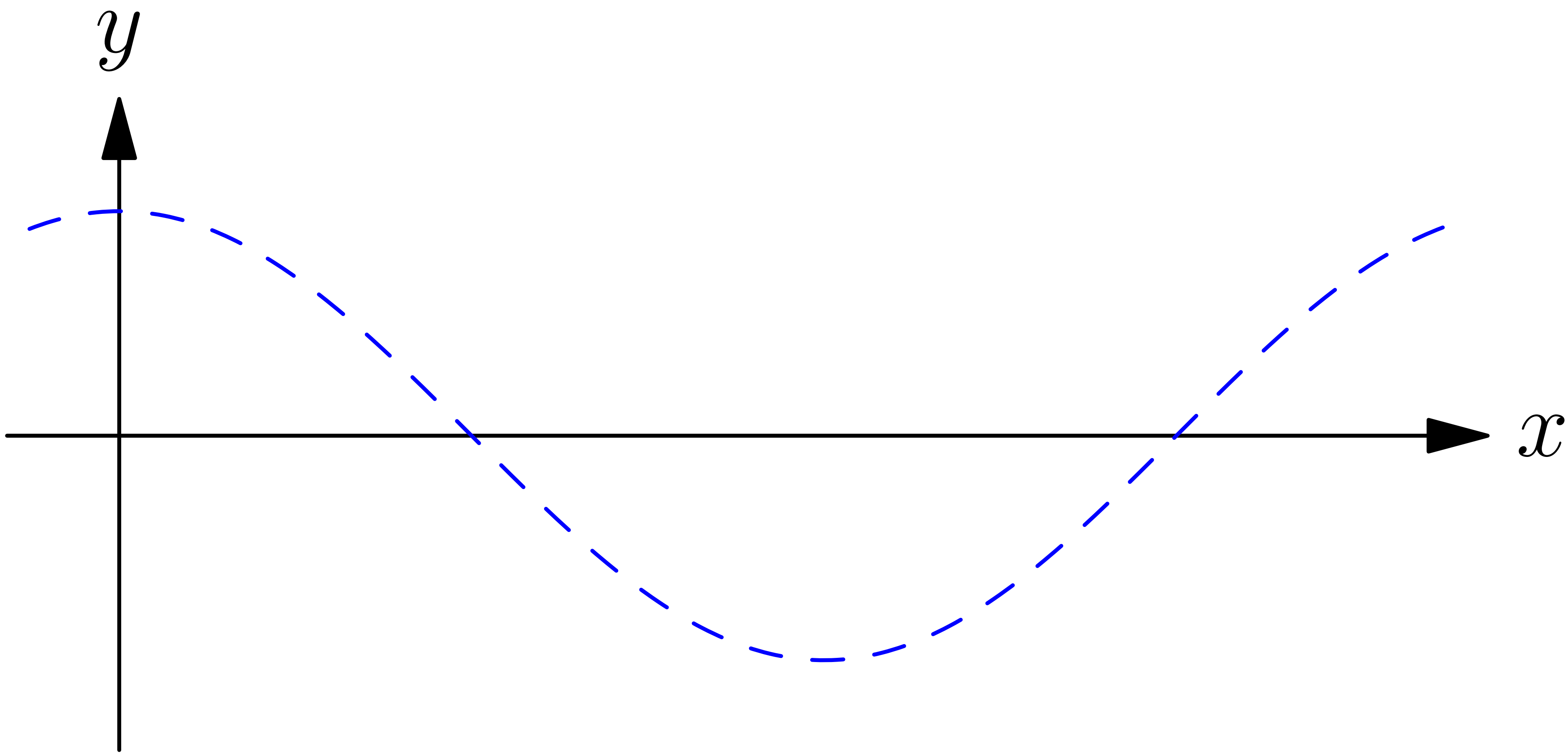
- where $p(\boldsymbol{y}) = \int p(f_*, \boldsymbol{y}) \mathrm{d} f_*$

- Because all integrals are Gaussian we can compute the integral to obtain

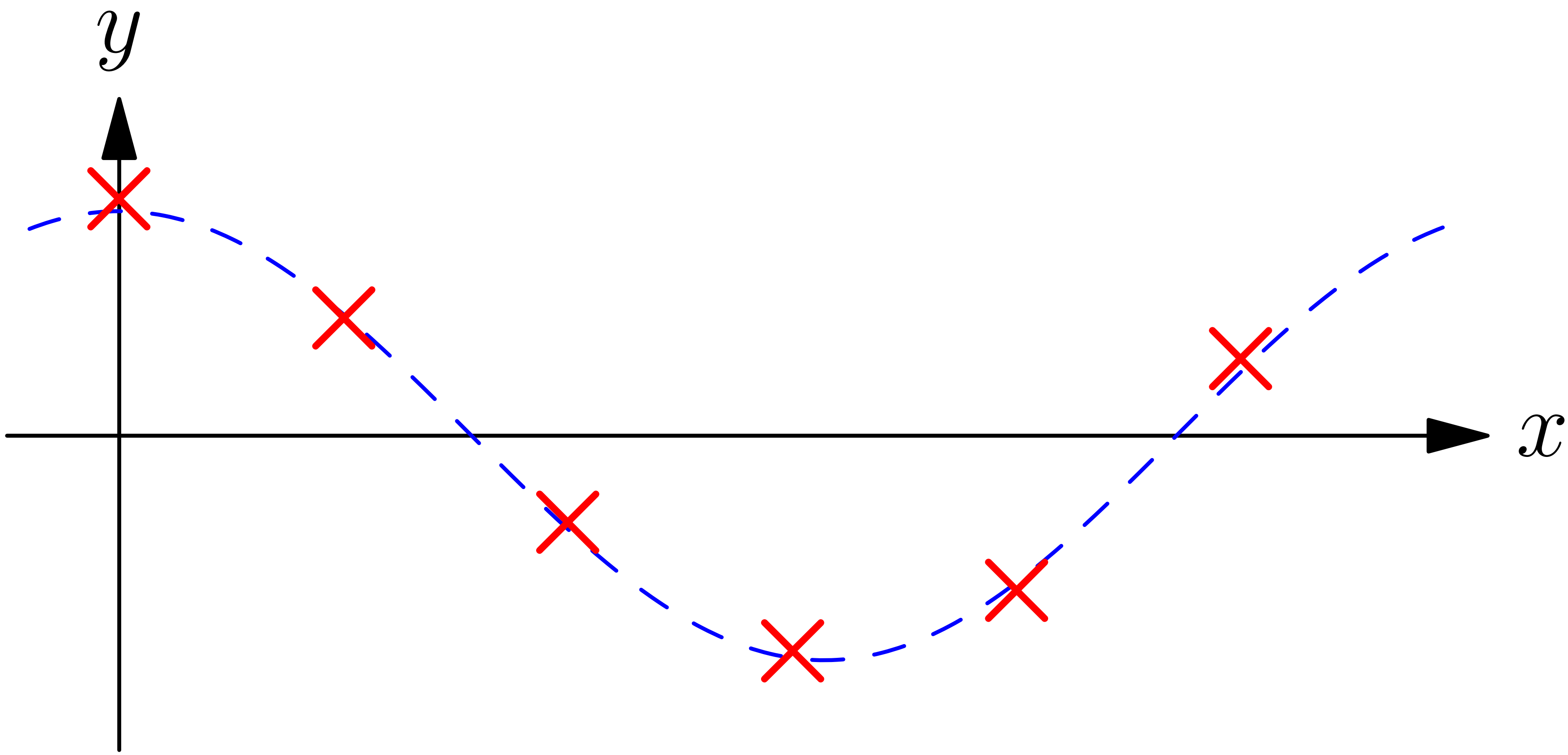
$$p(f_*|\boldsymbol{y}) = \mathcal{N}\left(f_* \middle| \boldsymbol{k}_*^{\mathsf{T}} (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \boldsymbol{y}, k - \boldsymbol{k}_*^{\mathsf{T}} (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \boldsymbol{k}_* \right)$$

- Looks complicated, but numerically easy to evaluate

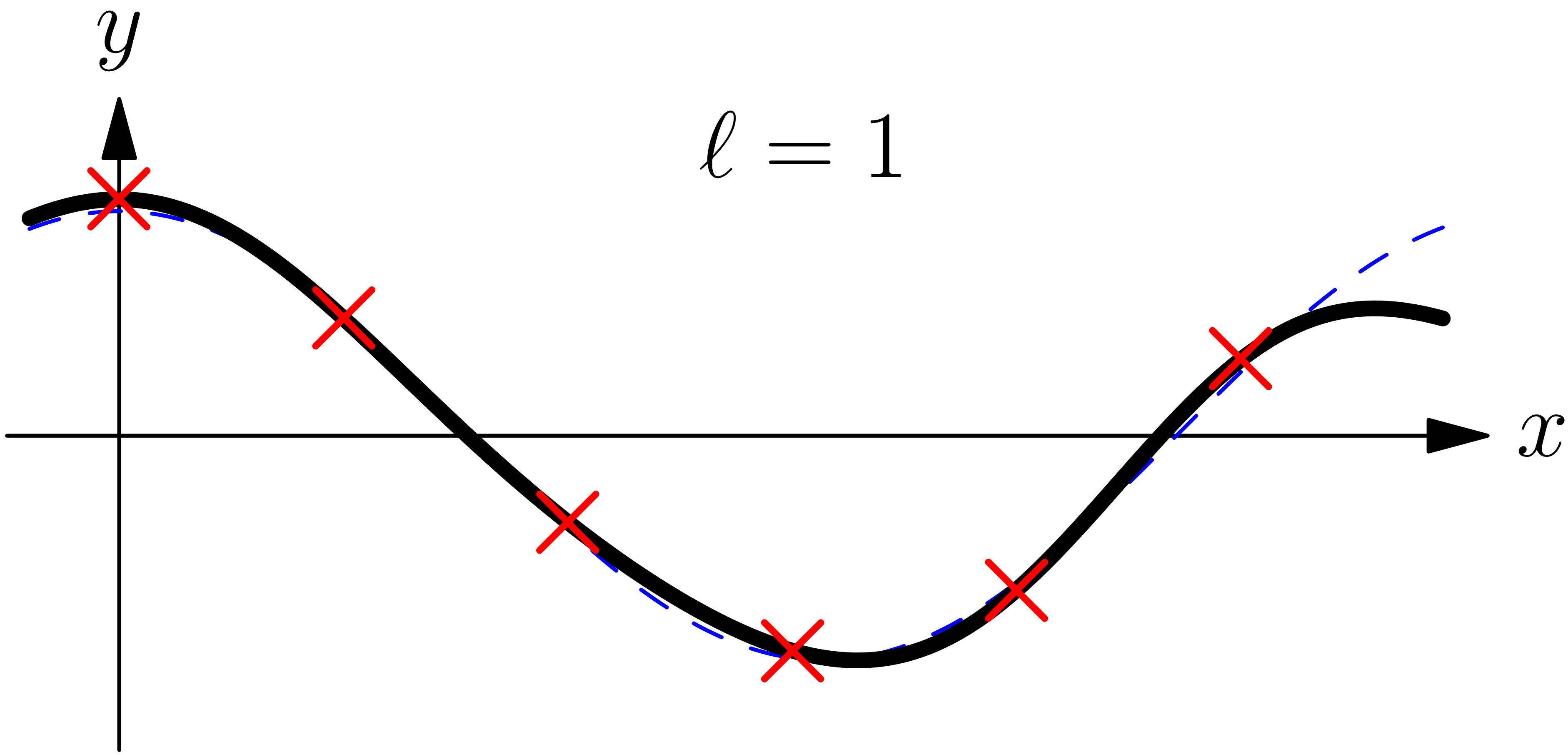
$$K(x,x') = \exp(-(x - x')^2/(2\ell^2))$$



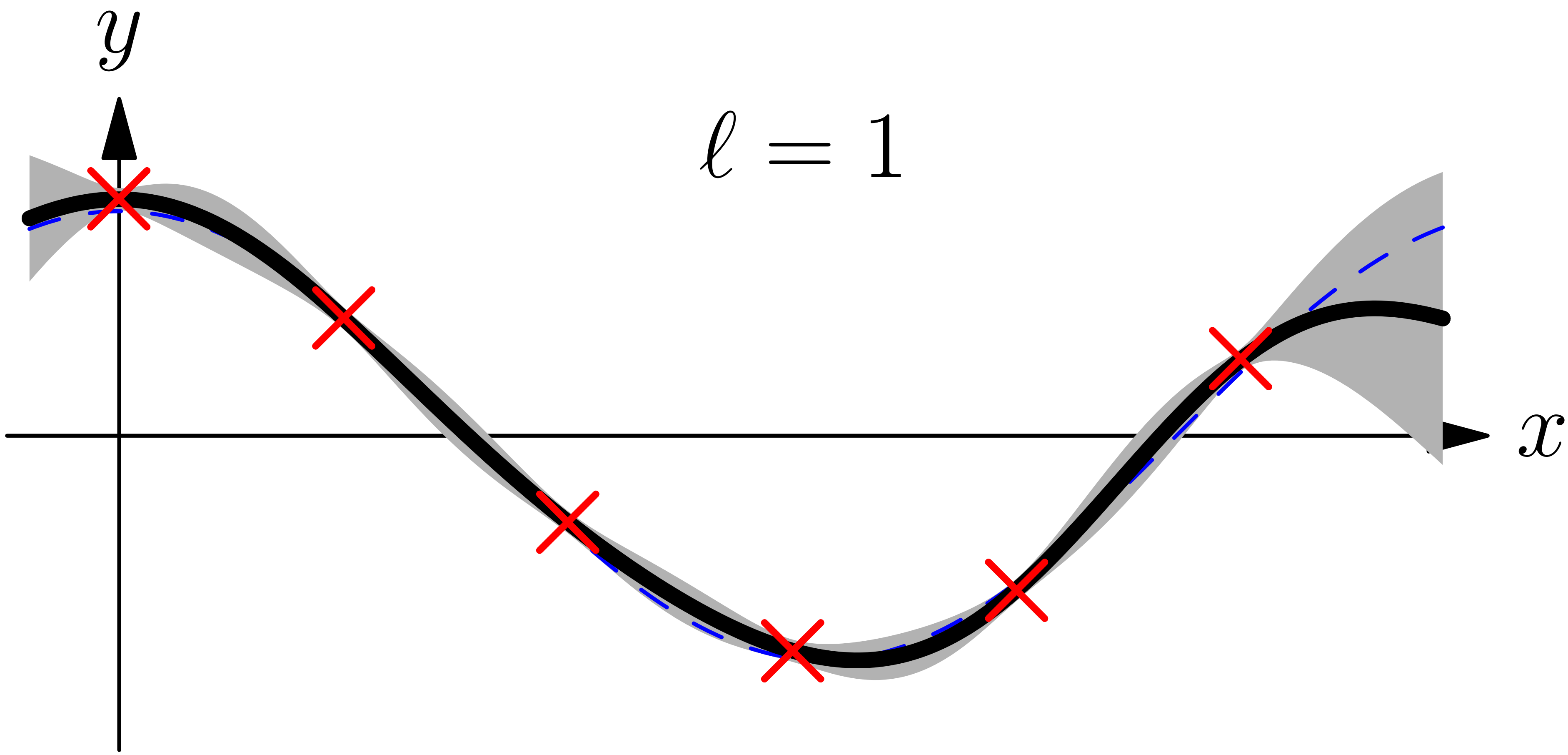
$$K(x,x') = \exp(-(x-x')^2/(2\ell^2))$$



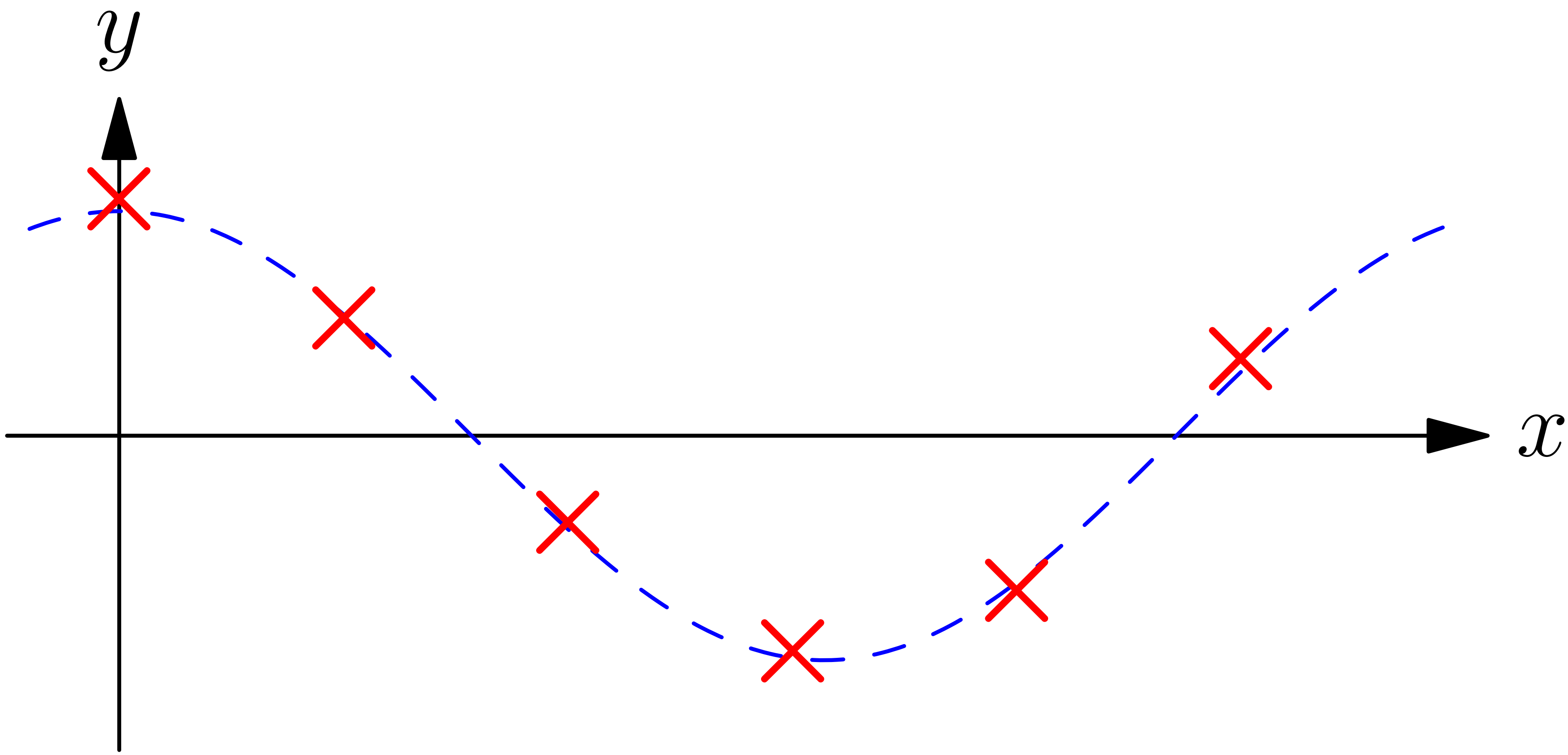
$$K(x,x') = \exp(-(x-x')^2/(2\ell^2))$$



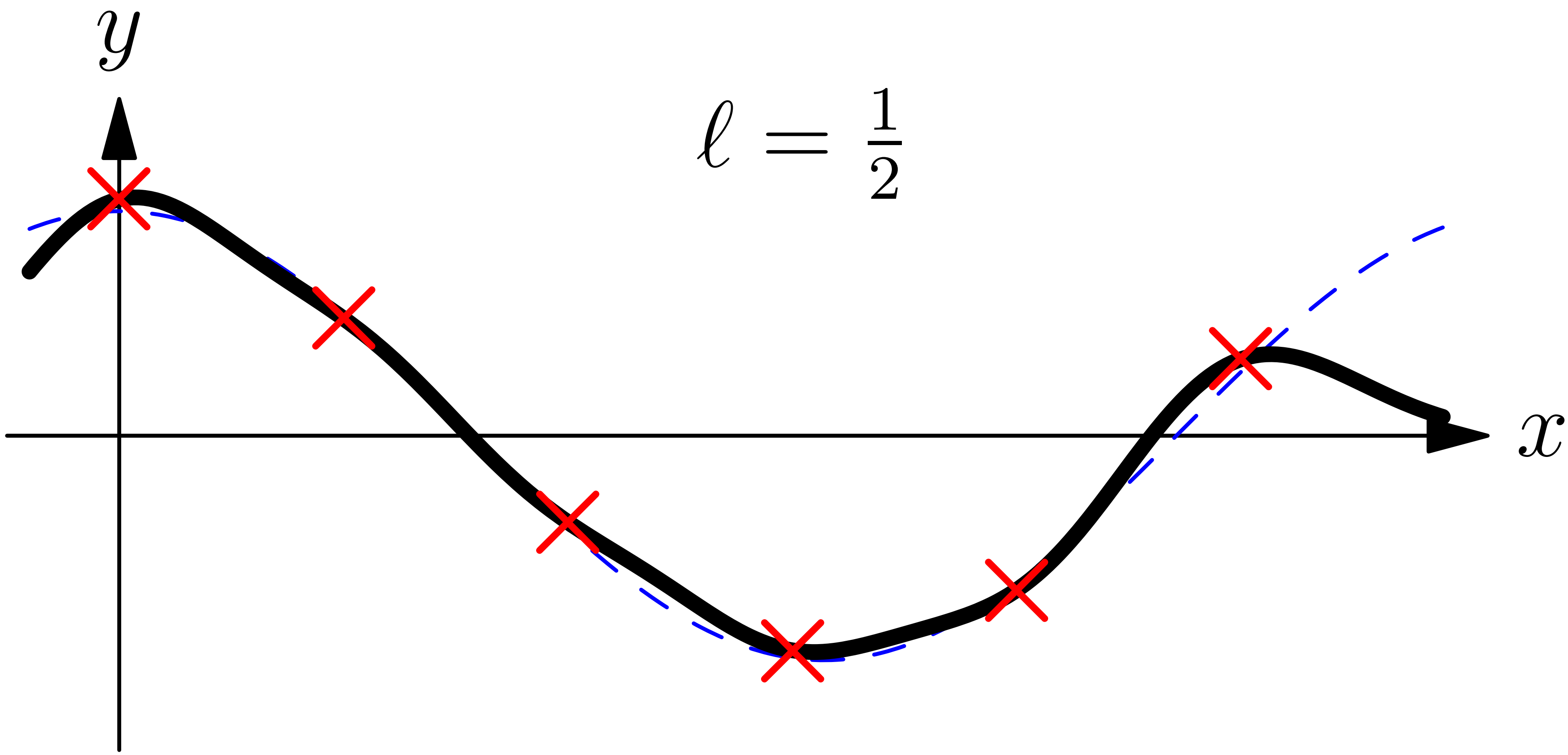
$$K(x,x') = \exp(-(x-x')^2/(2\ell^2))$$



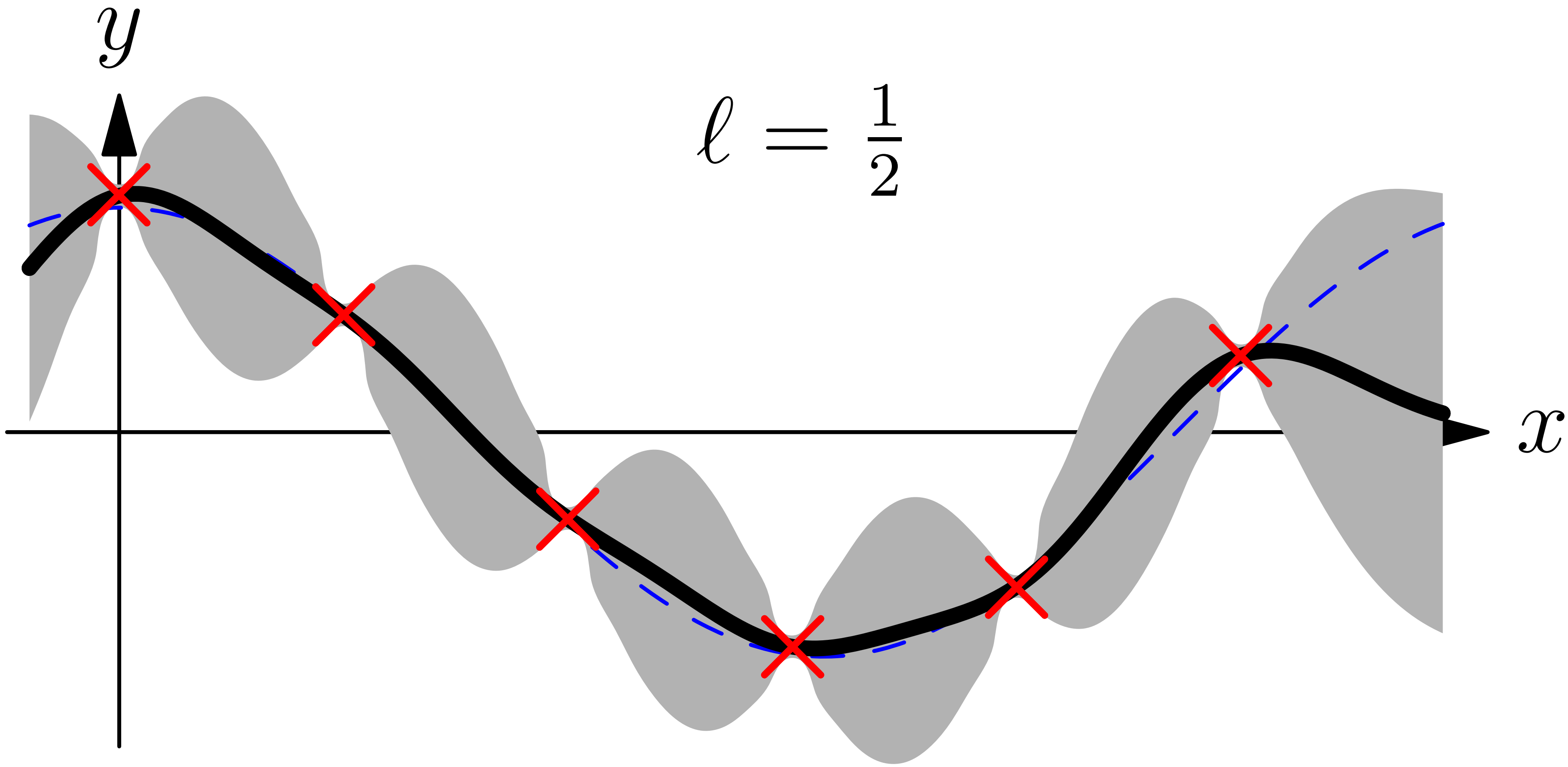
$$K(x,x') = \exp(-(x-x')^2/(2\ell^2))$$



$$K(x,x') = \exp(-(x-x')^2/(2\ell^2))$$



$$K(x,x') = \exp(-(x-x')^2/(2\ell^2))$$



Multi-dimensional Regression

- I've shown a 1-D regression example because it is easy to visualise
- This might be used with a time series
- The much more typical situation in machine learning is for x to have many features so we are doing multi-dimensional regression
- Gaussian process inference were first used in spatial problems where it was known as **krigging**
- It was re-invented by the machine learning community who call it Gaussian Processes (GP)

Multi-dimensional Regression

- I've shown a 1-D regression example because it is easy to visualise
- This might be used with a time series
- The much more typical situation in machine learning is for x to have many features so we are doing multi-dimensional regression
- Gaussian process inference were first used in spatial problems where it was known as **krigging**
- It was re-invented by the machine learning community who call it Gaussian Processes (GP)

Multi-dimensional Regression

- I've shown a 1-D regression example because it is easy to visualise
- This might be used with a time series
- The much more typical situation in machine learning is for x to have many features so we are doing multi-dimensional regression
- Gaussian process inference were first used in spatial problems where it was known as **krigging**
- It was re-invented by the machine learning community who call it Gaussian Processes (GP)

Multi-dimensional Regression

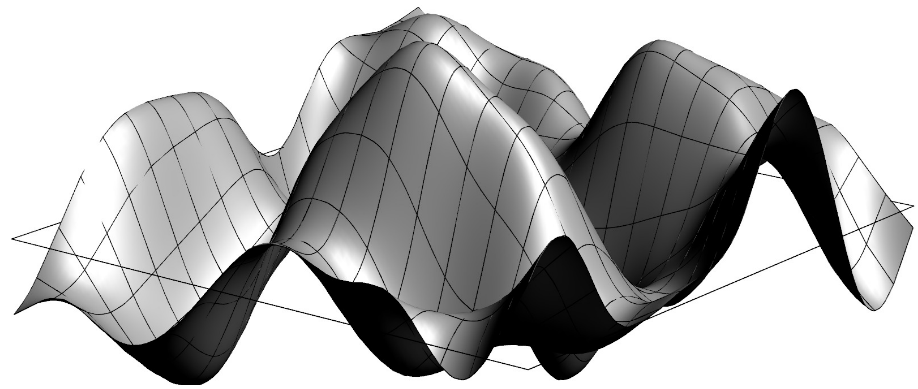
- I've shown a 1-D regression example because it is easy to visualise
- This might be used with a time series
- The much more typical situation in machine learning is for x to have many features so we are doing multi-dimensional regression
- Gaussian process inference were first used in spatial problems where it was known as **kriging**
- It was re-invented by the machine learning community who call it Gaussian Processes (GP)

Multi-dimensional Regression

- I've shown a 1-D regression example because it is easy to visualise
- This might be used with a time series
- The much more typical situation in machine learning is for $\boldsymbol{x}$ to have many features so we are doing multi-dimensional regression
- Gaussian process inference were first used in spatial problems where it was known as **krigging**
- It was re-invented by the machine learning community who call it Gaussian Processes (GP)

Outline

1. Introduction
2. Gaussian Processes
3. Bayesian Inference
4. **Hyper-parameters**



Choosing the Correct Covariance Function

- Choosing the correct covariance function is critical
- Most covariance functions include a continuous **hyper-parameter** (e.g. the correlation length ℓ) that we have to choose correctly
- This is typical of many Bayesian problems were we have some set of hyper-parameters, ϕ, describing the model
- These are different to the normal parameters we learn (e.g. weights $\boldsymbol{w}$ or in GP the functions $f(\boldsymbol{x})$)
- In Bayesian inference we learn the posterior for these normal parameters

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

Choosing the Correct Covariance Function

- Choosing the correct covariance function is critical
- Most covariance functions include a continuous **hyper-parameter** (e.g. the correlation length ℓ) that we have to choose correctly
- This is typical of many Bayesian problems were we have some set of hyper-parameters, ϕ, describing the model
- These are different to the normal parameters we learn (e.g. weights $\boldsymbol{w}$ or in GP the functions $f(\boldsymbol{x})$)
- In Bayesian inference we learn the posterior for these normal parameters

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

Choosing the Correct Covariance Function

- Choosing the correct covariance function is critical
- Most covariance functions include a continuous **hyper-parameter** (e.g. the correlation length ℓ) that we have to choose correctly
- This is typical of many Bayesian problems were we have some set of hyper-parameters, ϕ, describing the model
- These are different to the normal parameters we learn (e.g. weights $\boldsymbol{w}$ or in GP the functions $f(\boldsymbol{x})$)
- In Bayesian inference we learn the posterior for these normal parameters

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

Choosing the Correct Covariance Function

- Choosing the correct covariance function is critical
- Most covariance functions include a continuous **hyper-parameter** (e.g. the correlation length ℓ) that we have to choose correctly
- This is typical of many Bayesian problems were we have some set of hyper-parameters, ϕ, describing the model
- These are different to the normal parameters we learn (e.g. weights $\boldsymbol{w}$ or in GP the functions $f(\boldsymbol{x})$)
- In Bayesian inference we learn the posterior for these normal parameters

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

Choosing the Correct Covariance Function

- Choosing the correct covariance function is critical
- Most covariance functions include a continuous **hyper-parameter** (e.g. the correlation length ℓ) that we have to choose correctly
- This is typical of many Bayesian problems were we have some set of hyper-parameters, $\boldsymbol{\phi}$, describing the model
- These are different to the normal parameters we learn (e.g. weights $\boldsymbol{w}$ or in GP the functions $f(\boldsymbol{x})$)
- In Bayesian inference we learn the posterior for these normal parameters

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

Evidence Framework

- The normalisation factor, $p(\mathcal{D}|\phi)$ is known as the **marginal likelihood** or **evidence**

$$p(\mathcal{D}|\phi) = \int p(\mathcal{D}|f,\phi)\, p(f|\phi)\, \mathrm{d}f$$

- We can perform a Bayesian calculation at a second level by putting a prior on ϕ

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

- From this we can now marginalise out the hyper-parameters

$$p(f|\mathcal{D}) = \int p(f|\mathcal{D},\phi)\, p(\phi|\mathcal{D})\, \mathrm{d}\phi$$

Evidence Framework

- The normalisation factor, $p(\mathcal{D}|\phi)$ is known as the **marginal likelihood** or **evidence**

$$p(\mathcal{D}|\phi) = \int p(\mathcal{D}|f,\phi)\,p(f|\phi)\,\mathrm{d}f$$

- We can perform a Bayesian calculation at a second level by putting a prior on ϕ

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

- From this we can now marginalise out the hyper-parameters

$$p(f|\mathcal{D}) = \int p(f|\mathcal{D},\phi)\,p(\phi|\mathcal{D})\,\mathrm{d}\phi$$

Evidence Framework

- The normalisation factor, $p(\mathcal{D}|\phi)$ is known as the **marginal likelihood** or **evidence**

$$p(\mathcal{D}|\phi) = \int p(\mathcal{D}|f,\phi)p(f|\phi)\,\mathrm{d}f$$

- We can perform a Bayesian calculation at a second level by putting a prior on ϕ

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

- From this we can now marginalise out the hyper-parameters

$$p(f|\mathcal{D}) = \int p(f|\mathcal{D},\phi)p(\phi|\mathcal{D})\,\mathrm{d}\phi$$

Maximum-Likelihood-II

- The integral

$$p(f|\mathcal{D}) = \int p(f|\mathcal{D},\boldsymbol{\phi})\, p(\boldsymbol{\phi}|\mathcal{D})\, \mathrm{d}\boldsymbol{\phi}$$

  usually can't be computed analytically and we have to use Monte Carlo methods (see later lecture)

- An alternative is to use the most likely hyper-parameter

- We can find this by using gradient search of $p(\mathcal{D}|\boldsymbol{\phi})$

- This is sometimes referred to as ML-II

- Normally even this can be difficult, but for GP its not too difficult

Maximum-Likelihood-II

- The integral

$$p(f|\mathcal{D}) = \int p(f|\mathcal{D}, \boldsymbol{\phi}) p(\boldsymbol{\phi}|\mathcal{D}) \, \mathrm{d}\boldsymbol{\phi}$$

  usually can't be computed analytically and we have to use Monte Carlo methods (see later lecture)

- An alternative is to use the most likely hyper-parameter

- We can find this by using gradient search of $p(\mathcal{D}|\boldsymbol{\phi})$

- This is sometimes referred to as ML-II

- Normally even this can be difficult, but for GP its not too difficult

Maximum-Likelihood-II

- The integral

$$p(f|\mathcal{D}) = \int p(f|\mathcal{D},\boldsymbol{\phi})p(\boldsymbol{\phi}|\mathcal{D})\,\mathrm{d}\boldsymbol{\phi}$$

  usually can't be computed analytically and we have to use Monte Carlo methods (see later lecture)

- An alternative is to use the most likely hyper-parameter
- We can find this by using gradient search of $p(\mathcal{D}|\boldsymbol{\phi})$
- This is sometimes referred to as ML-II
- Normally even this can be difficult, but for GP its not too difficult

Maximum-Likelihood-II

- The integral

$$p(f|\mathcal{D}) = \int p(f|\mathcal{D},\boldsymbol{\phi})\,p(\boldsymbol{\phi}|\mathcal{D})\,\mathrm{d}\boldsymbol{\phi}$$

  usually can't be computed analytically and we have to use Monte Carlo methods (see later lecture)

- An alternative is to use the most likely hyper-parameter
- We can find this by using gradient search of $p(\mathcal{D}|\boldsymbol{\phi})$
- This is sometimes referred to as ML-II
- Normally even this can be difficult, but for GP its not too difficult

Maximum-Likelihood-II

- The integral

$$p(f|\mathcal{D}) = \int p(f|\mathcal{D},\boldsymbol{\phi})\, p(\boldsymbol{\phi}|\mathcal{D})\, \mathrm{d}\boldsymbol{\phi}$$

  usually can't be computed analytically and we have to use Monte Carlo methods (see later lecture)

- An alternative is to use the most likely hyper-parameter

- We can find this by using gradient search of $p(\mathcal{D}|\boldsymbol{\phi})$

- This is sometimes referred to as ML-II

- Normally even this can be difficult, but for GP its not too difficult

Evidence for GP

- For GP the (log)-evidence can be computed in closed form

$$\log(p(\mathcal{D}|\boldsymbol{\phi})) = -\frac{1}{2}\boldsymbol{y}^{\mathsf{T}}(\mathbf{K}+\sigma^2\mathbf{I})\boldsymbol{y} - \frac{1}{2}\log\bigl(|\mathbf{K}+\sigma^2\mathbf{I}|\bigr) - \frac{m}{2}\log(2\pi)$$

  - ⋆ First term measures goodness of fit
  - ⋆ Second term measure complexity of model
  - ⋆ Last term is a common normalisation constant

- Can efficiently compute derivatives and find best parameters

Evidence for GP

- For GP the (log)-evidence can be computed in closed form

$$\log(p(\mathcal{D}|\phi)) = -\frac{1}{2}\boldsymbol{y}^{\mathsf{T}}(\mathbf{K}+\sigma^2\mathbf{I})\boldsymbol{y} - \frac{1}{2}\log\big(|\mathbf{K}+\sigma^2\mathbf{I}|\big) - \frac{m}{2}\log(2\pi)$$

  - ⋆ First term measures goodness of fit
  - ⋆ Second term measure complexity of model
  - ⋆ Last term is a common normalisation constant

- Can efficiently compute derivatives and find best parameters

Evidence for GP

- For GP the (log)-evidence can be computed in closed form

$$\log(p(\mathcal{D}|\phi)) = -\frac{1}{2}\boldsymbol{y}^{\mathsf{T}}(\mathbf{K}+\sigma^2\mathbf{I})\boldsymbol{y} - \frac{1}{2}\log\left(|\mathbf{K}+\sigma^2\mathbf{I}|\right) - \frac{m}{2}\log(2\pi)$$

  - ⋆ First term measures goodness of fit
  - ⋆ Second term measure complexity of model
  - ⋆ Last term is a common normalisation constant

- Can efficiently compute derivatives and find best parameters

Evidence for GP

- For GP the (log)-evidence can be computed in closed form

$$\log(p(\mathcal{D}|\boldsymbol{\phi})) = -\frac{1}{2}\boldsymbol{y}^{\mathsf{T}}(\mathbf{K}+\sigma^2\mathbf{I})\boldsymbol{y} - \frac{1}{2}\log\big(|\mathbf{K}+\sigma^2\mathbf{I}|\big) - \frac{m}{2}\log(2\pi)$$

  - ⋆ First term measures goodness of fit
  - ⋆ Second term measure complexity of model
  - ⋆ Last term is a common normalisation constant

- Can efficiently compute derivatives and find best parameters

Evidence for GP

- For GP the (log)-evidence can be computed in closed form

$$\log(p(\mathcal{D}|\phi)) = -\frac{1}{2}\boldsymbol{y}^{\mathsf{T}}(\mathbf{K}+\sigma^2\mathbf{I})\boldsymbol{y} - \frac{1}{2}\log\big(|\mathbf{K}+\sigma^2\mathbf{I}|\big) - \frac{m}{2}\log(2\pi)$$

  - ⋆ First term measures goodness of fit
  - ⋆ Second term measure complexity of model
  - ⋆ Last term is a common normalisation constant

- Can efficiently compute derivatives and find best parameters

Evidence for GP

- For GP the (log)-evidence can be computed in closed form

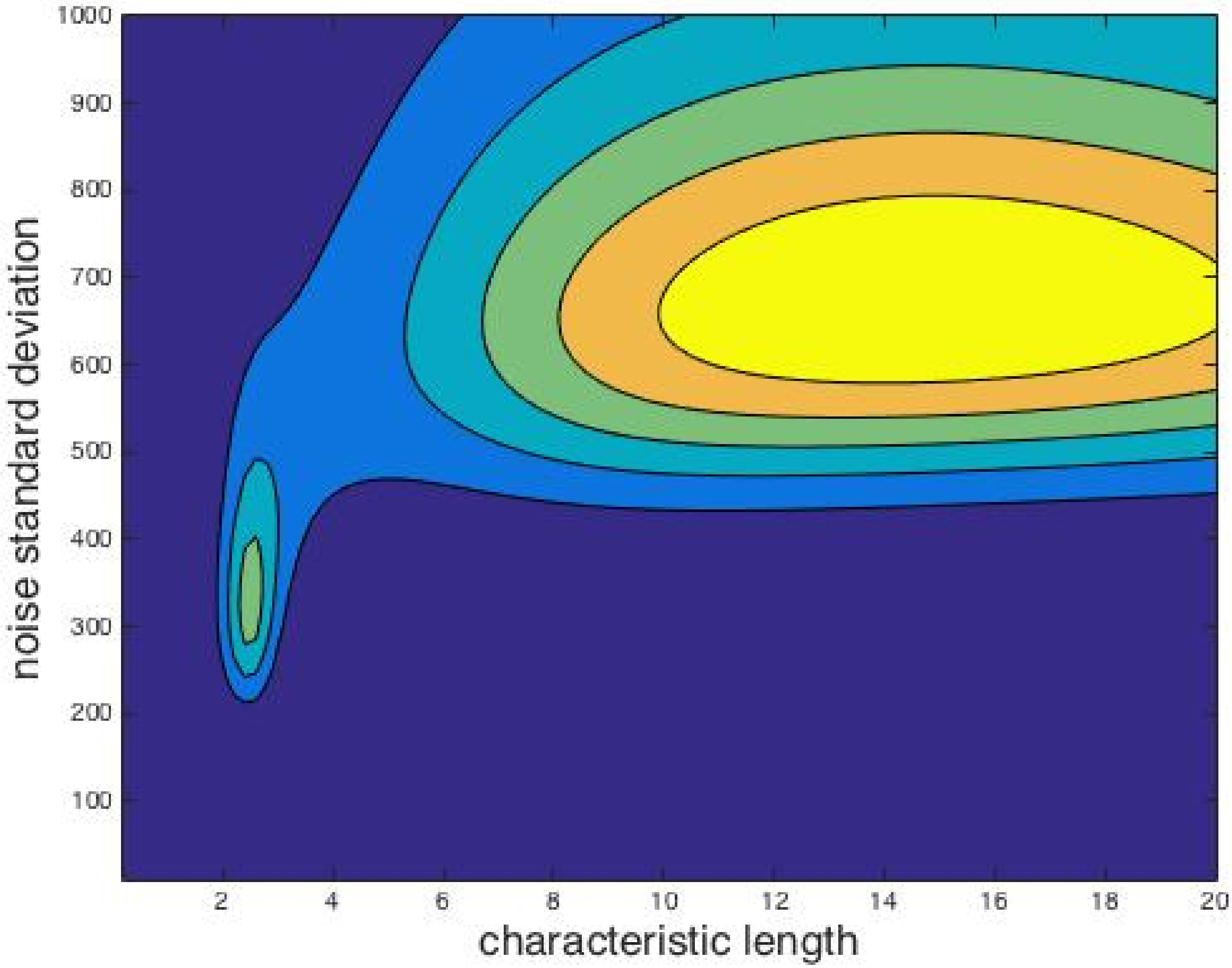
$$\log(p(\mathcal{D}|\boldsymbol{\phi})) = -\frac{1}{2}\boldsymbol{y}^{\mathsf{T}}(\mathbf{K}+\sigma^2\mathbf{I})\boldsymbol{y} - \frac{1}{2}\log\big(|\mathbf{K}+\sigma^2\mathbf{I}|\big) - \frac{m}{2}\log(2\pi)$$

  - ⋆ First term measures goodness of fit
  - ⋆ Second term measure complexity of model
  - ⋆ Last term is a common normalisation constant

- Can efficiently compute derivatives and find best parameters

- Could overfit!

Example (slightly pathological)



Conclusions

- Gaussian processes are very powerful for regression (and classification?)
- Because all calculations involve Gaussian integrals we can compute everything in closed form
- (Actually its a pain to do the mathematics because you end up working with inverse of matrices)
- Fairly generic (black-box) technique because the prior captures many continuity constraints
- We can use the evidence framework (probability of data) to do model selection and hyper-parameter optimisations

Conclusions

- Gaussian processes are very powerful for regression (and classification?)
- Because all calculations involve Gaussian integrals we can compute everything in closed form
- (Actually its a pain to do the mathematics because you end up working with inverse of matrices)
- Fairly generic (black-box) technique because the prior captures many continuity constraints
- We can use the evidence framework (probability of data) to do model selection and hyper-parameter optimisations

Conclusions

- Gaussian processes are very powerful for regression (and classification?)
- Because all calculations involve Gaussian integrals we can compute everything in closed form
- (Actually its a pain to do the mathematics because you end up working with inverse of matrices)
- Fairly generic (black-box) technique because the prior captures many continuity constraints
- We can use the evidence framework (probability of data) to do model selection and hyper-parameter optimisations

Conclusions

- Gaussian processes are very powerful for regression (and classification?)
- Because all calculations involve Gaussian integrals we can compute everything in closed form
- (Actually its a pain to do the mathematics because you end up working with inverse of matrices)
- Fairly generic (black-box) technique because the prior captures many continuity constraints
- We can use the evidence framework (probability of data) to do model selection and hyper-parameter optimisations

Conclusions

- Gaussian processes are very powerful for regression (and classification?)
- Because all calculations involve Gaussian integrals we can compute everything in closed form
- (Actually its a pain to do the mathematics because you end up working with inverse of matrices)
- Fairly generic (black-box) technique because the prior captures many continuity constraints
- We can use the evidence framework (probability of data) to do model selection and hyper-parameter optimisations