M5MS10
Machine Learning

Spring 2018

Lecture 6

Dr Ben Calderhead

b.calderhead@imperial.ac.uk

M5MS10
Machine Learning

Spring 2018

Lecture 6

Dr Ben Calderhead

b.calderhead@imperial.ac.uk

IS10
Learning

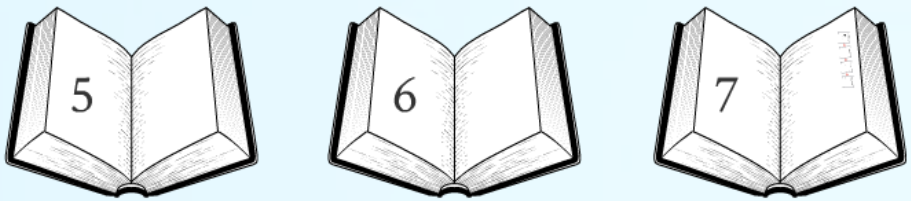
2018

re 6

lderhead

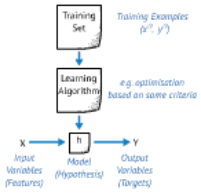
mperial.ac.uk

Non-parametric Methods



Introduction to Gaussian Processes

Nonparametric Regression



We'll consider continuous output variables, i.e. *regression*.

Gaussian Processes

A Gaussian process is an infinite collection of random variables, any *finite* number of which have a (consistent) multivariate Gaussian distribution.

We can think of this as a generalisation of a multivariate Gaussian to *infinite* dimensions.

A Gaussian process defines a distribution over functions.

- We can consider a function to approximately be an infinitely long vector.

Gaussian Process versus Gaussian Distribution

Gaussian distribution: $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$

Gaussian process: $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$

Gaussian Process versus Gaussian Distribution

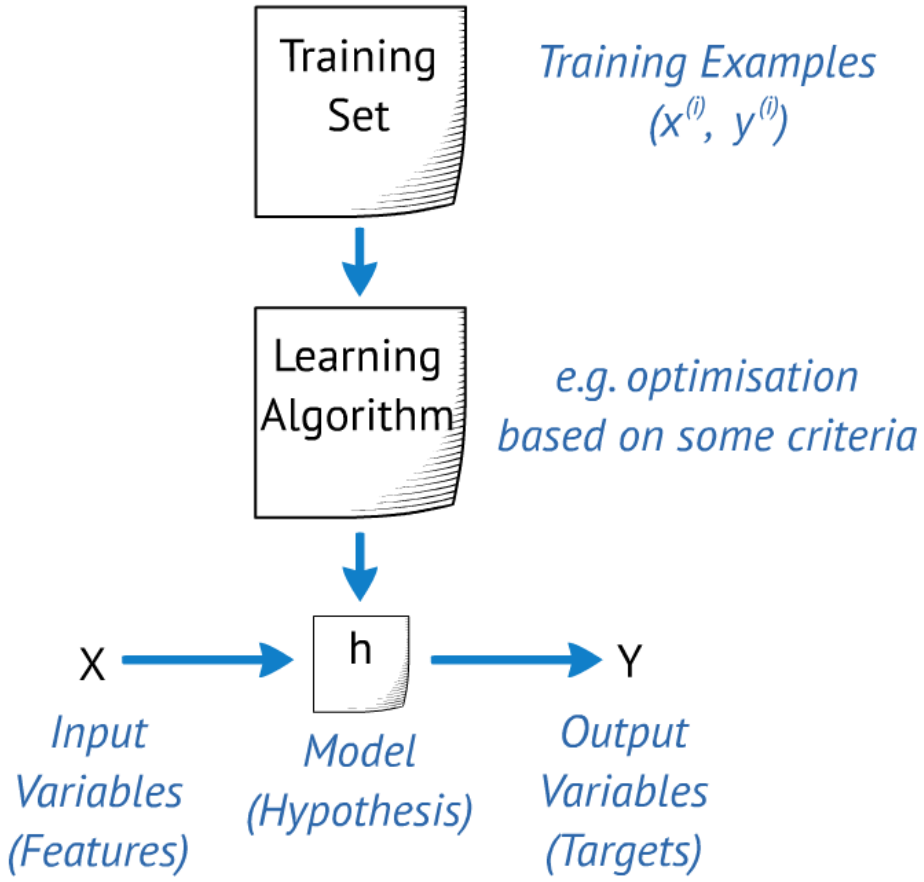
A Gaussian distribution is a distribution over a variable of a given dimension.

- It is fully specified by a mean vector and a covariance matrix: $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$.
- For each index i denoting each dimension of $\mathbf{x}$, the mean and variance is defined directly.

A GP is a distribution over functions and so the index set is of infinite dimension.

- It is fully specified by a mean function and a covariance function: $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$.
- For each dimension of f, the mean and (co-)variance is specified via a function indexed by $\mathbf{x}$, where e.g. $\mathbf{x} \in \mathbb{R}$.

Nonparametric Regression



We'll consider continuous output variables, i.e. *regression*.

Gaussian Processes

A Gaussian process is an infinite collection of random variables, any *finite* number of which have a (consistent) multivariate Gaussian distribution.

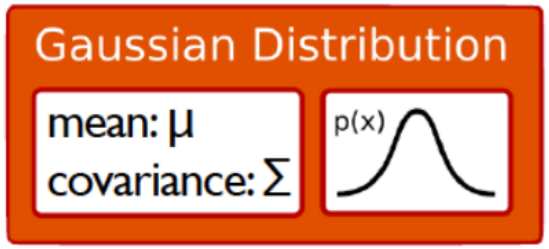
We can think of this as a generalisation of a multivariate Gaussian to *infinite* dimensions.

A Gaussian process defines a distribution over functions.

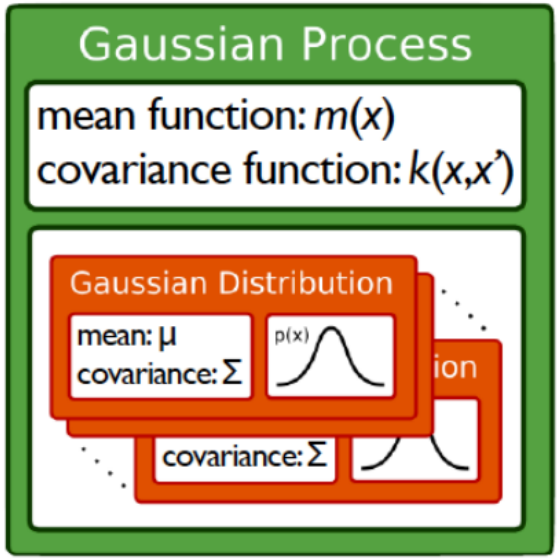
- We can consider a function to approximately be an infinitely long vector.

Gaussian Process versus Gaussian Distribution

Gaussian distribution: $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$



Gaussian process: $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$



Gaussian Process versus Gaussian Distribution

A Gaussian distribution is a distribution over a variable of a given dimension.

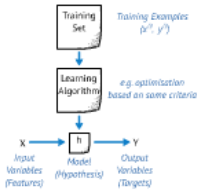
- It is fully specified by a mean vector and a covariance matrix: $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.
- For each index i denoting each dimension of $\mathbf{x}$, the mean and variance is defined directly.

A GP is a distribution over functions and so the index set is of infinite dimension.

- It is fully specified by a mean function and a covariance function: $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$.
- For each dimension of f, the mean and (co-)variance is specified via a function indexed by $\mathbf{x}$, where e.g. $\mathbf{x} \in \mathbb{R}$.

Introduction to Gaussian Processes

Nonparametric Regression



We'll consider continuous output variables, i.e. *regression*.

Gaussian Processes

A Gaussian process is an infinite collection of random variables, any *finite* number of which have a (consistent) multivariate Gaussian distribution.

We can think of this as a generalisation of a multivariate Gaussian to *infinite* dimensions.

A Gaussian process defines a distribution over functions.

- We can consider a function to approximately be an infinitely long vector.

Gaussian Process versus Gaussian Distribution

Gaussian distribution: $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$

Gaussian process: $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$

Gaussian Process versus Gaussian Distribution

A Gaussian distribution is a distribution over a variable of a given dimension.

- It is fully specified by a mean vector and a covariance matrix: $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$.
- For each index i denoting each dimension of $\mathbf{x}$, the mean and variance is defined directly.

A GP is a distribution over functions and so the index set is of infinite dimension.

- It is fully specified by a mean function and a covariance function: $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$.
- For each dimension of f, the mean and (co-)variance is specified via a function indexed by $\mathbf{x}$, where e.g. $\mathbf{x} \in \mathbb{R}$.

Multivariate Gaussian Distributions

Multivariate Gaussian

Although the Gaussian process is an *infinite dimensional* object, it turns out that we only ever need to deal with finitely many points at any one time.

Multivariate Gaussian distributions are all we require!

A vector-valued random variable $\mathbf{x} \in \mathbb{R}^n$ is said to have a multivariate normal (or Gaussian) distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^n$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbf{S}^n_{++}$ if

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{2/n}|\boldsymbol{\Sigma}|^{1/n}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}(\mathbf{x}-\boldsymbol{\mu})\right)$$

where $\mathbf{S}^n_{++}$ denotes the space of $n \times n$ positive-definite symmetric matrices.

Multivariate Gaussian

Gaussian random variables are extremely useful in machine learning and statistics for the following main reasons.

- Measurement error can often be considered to be the accumulation of a large number of small independent random perturbations - by the *Central Limit Theorem*, summations of independent random variables will tend towards being Gaussian distributed.
- They are convenient because many of the integrals involving Gaussian distributions that arise in practice may be written as simple closed form solutions.

Multivariate Gaussian

The following properties of multivariate Gaussians are vital for manipulating Gaussian processes analytically.

Consider a random vector $\mathbf{x} \in \mathbb{R}^n$ with $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. We can then consider the variable $\mathbf{x}$ that has been partitioned into two sets $\mathbf{x}_A = [x_1, \ldots, x_r]^T \in \mathbb{R}^r$ and $\mathbf{x}_B = [x_{r+1}, \ldots, x_n]^T \in \mathbb{R}^{(n-r)}$. We therefore have

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_A \\ \mathbf{x}_B \end{pmatrix} \qquad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{AA} & \boldsymbol{\Sigma}_{AB} \\ \boldsymbol{\Sigma}_{BA} & \boldsymbol{\Sigma}_{BB} \end{pmatrix}$$

We note that $\boldsymbol{\Sigma}_{AB} = \boldsymbol{\Sigma}_{BA}^T$, since $\boldsymbol{\Sigma} = E[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T] = \boldsymbol{\Sigma}^T$.

Multivariate Gaussian

The Gaussian density function is correctly normalised.

$$\int_{\mathbf{x}} p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) d\mathbf{x} = 1$$

Multivariate Gaussian

8.1.8 Product of gaussian densities

Let $\mathcal{N}_x(\mathbf{m}, \boldsymbol{\Sigma})$ denote a density of $\mathbf{x}$, then

$$\mathcal{N}_x(\mathbf{m}_1, \boldsymbol{\Sigma}_1) \cdot \mathcal{N}_x(\mathbf{m}_2, \boldsymbol{\Sigma}_2) = c_c \mathcal{N}_x(\mathbf{m}_c, \boldsymbol{\Sigma}_c) \tag{371}$$

$$\begin{aligned} c_c &= \mathcal{N}_{\mathbf{m}_2}(\mathbf{m}_2, (\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)) \\ &= \frac{1}{\sqrt{\det(2\pi(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2))}} \exp\left[-\frac{1}{2}(\mathbf{m}_1 - \mathbf{m}_2)^T(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)^{-1}(\mathbf{m}_1 - \mathbf{m}_2)\right] \\ \mathbf{m}_c &= (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1}(\boldsymbol{\Sigma}_1^{-1}\mathbf{m}_1 + \boldsymbol{\Sigma}_2^{-1}\mathbf{m}_2) \\ \boldsymbol{\Sigma}_c &= (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1} \end{aligned}$$

but note that the product is not normalized as a density of $\mathbf{x}$.

Matrix Cookbook...

Multivariate Gaussian

The marginal densities

$$\begin{aligned} p(\mathbf{x}_A) &= \int_{\mathbf{x}_B} p(\mathbf{x}_A, \mathbf{x}_B|\boldsymbol{\mu}, \boldsymbol{\Sigma}) d\mathbf{x}_B \\ p(\mathbf{x}_B) &= \int_{\mathbf{x}_A} p(\mathbf{x}_A, \mathbf{x}_B|\boldsymbol{\mu}, \boldsymbol{\Sigma}) d\mathbf{x}_A \end{aligned}$$

are also Gaussian with the following form,

$$\begin{aligned} \mathbf{x}_A &\sim \mathcal{N}(\boldsymbol{\mu}_A, \boldsymbol{\Sigma}_{AA}) \\ \mathbf{x}_B &\sim \mathcal{N}(\boldsymbol{\mu}_B, \boldsymbol{\Sigma}_{BB}) \end{aligned}$$

Multivariate Gaussian

The conditional densities

$$\begin{aligned} p(\mathbf{x}_A|\mathbf{x}_B) &= \frac{p(\mathbf{x}_A, \mathbf{x}_B|\boldsymbol{\mu}, \boldsymbol{\Sigma})}{\int_{\mathbf{x}_A} p(\mathbf{x}_A, \mathbf{x}_B|\boldsymbol{\mu}, \boldsymbol{\Sigma}) d\mathbf{x}_A} \\ p(\mathbf{x}_B|\mathbf{x}_A) &= \frac{p(\mathbf{x}_A, \mathbf{x}_B|\boldsymbol{\mu}, \boldsymbol{\Sigma})}{\int_{\mathbf{x}_B} p(\mathbf{x}_A, \mathbf{x}_B|\boldsymbol{\mu}, \boldsymbol{\Sigma}) d\mathbf{x}_B} \end{aligned}$$

are also Gaussian with the following form,

$$\begin{aligned} \mathbf{x}_A|\mathbf{x}_B &\sim \mathcal{N}\left(\boldsymbol{\mu}_A + \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_{BB}^{-1}(\mathbf{x}_B - \boldsymbol{\mu}_B), \boldsymbol{\Sigma}_{AA} - \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_{BB}^{-1}\boldsymbol{\Sigma}_{BA}\right) \\ \mathbf{x}_B|\mathbf{x}_A &\sim \mathcal{N}\left(\boldsymbol{\mu}_B + \boldsymbol{\Sigma}_{BA}\boldsymbol{\Sigma}_{AA}^{-1}(\mathbf{x}_A - \boldsymbol{\mu}_A), \boldsymbol{\Sigma}_{BB} - \boldsymbol{\Sigma}_{BA}\boldsymbol{\Sigma}_{AA}^{-1}\boldsymbol{\Sigma}_{AB}\right) \end{aligned}$$

Multivariate Gaussian

Although the Gaussian process is an *infinite dimensional* object, it turns out that we only ever need to deal with finitely many points at any one time.

Multivariate Gaussian distributions are all we require!

A vector-valued random variable $\mathbf{x} \in \mathbb{R}^n$ is said to have a multivariate normal (or Gaussian) distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^n$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbf{S}^n_{++}$ if

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{2/n}|\boldsymbol{\Sigma}|^{1/n}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}(\mathbf{x}-\boldsymbol{\mu})\right)$$

where $\mathbf{S}^n_{++}$ denotes the space of $n \times n$ positive-definite symmetric matrices.

Multivariate Gaussian

Gaussian random variables are extremely useful in machine learning and statistics for the following main reasons.

- Measurement error can often be considered to be the accumulation of a large number of small independent random perturbations - by the *Central Limit Theorem*, summations of independent random variables will tend towards being Gaussian distributed.
- They are convenient because many of the integrals involving Gaussian distributions that arise in practice may be written as simple closed form solutions.

Multivariate Gaussian

The following properties of multivariate Gaussians are vital for manipulating Gaussian processes analytically.

Consider a random vector $\mathbf{x} \in \mathbb{R}^n$ with $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. We can then consider the variable $\mathbf{x}$ that has been partitioned into two sets $\mathbf{x}_A = [x_1, \ldots, x_r]^T \in \mathbb{R}^r$ and $\mathbf{x}_B = [x_{r+1}, \ldots, x_n]^T \in \mathbb{R}^{(n-r)}$. We therefore have

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_A \\ \mathbf{x}_B \end{pmatrix} \qquad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{AA} & \boldsymbol{\Sigma}_{AB} \\ \boldsymbol{\Sigma}_{BA} & \boldsymbol{\Sigma}_{BB} \end{pmatrix}$$

We note that $\boldsymbol{\Sigma}_{AB} = \boldsymbol{\Sigma}_{BA}^T$, since $\boldsymbol{\Sigma} = E[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T] = \boldsymbol{\Sigma}^T$.

Multivariate Gaussian

The Gaussian density function is correctly normalised.

$$\int_{\mathbf{x}} p(\mathbf{x}|\boldsymbol{\mu}, \Sigma) d\mathbf{x} = 1$$

Multivariate Gaussian

8.1.8 Product of gaussian densities

Let $\mathcal{N}_{\mathbf{x}}(\mathbf{m}, \mathbf{\Sigma})$ denote a density of $\mathbf{x}$, then

$$\mathcal{N}_{\mathbf{x}}(\mathbf{m}_1, \mathbf{\Sigma}_1) \cdot \mathcal{N}_{\mathbf{x}}(\mathbf{m}_2, \mathbf{\Sigma}_2) = c_c \mathcal{N}_{\mathbf{x}}(\mathbf{m}_c, \mathbf{\Sigma}_c) \tag{371}$$

$$\begin{aligned} c_c &= \mathcal{N}_{\mathbf{m}_1}(\mathbf{m}_2, (\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)) \\ &= \frac{1}{\sqrt{\det(2\pi(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2))}} \exp\left[-\frac{1}{2}(\mathbf{m}_1 - \mathbf{m}_2)^T(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)^{-1}(\mathbf{m}_1 - \mathbf{m}_2)\right] \\ \mathbf{m}_c &= (\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1})^{-1}(\mathbf{\Sigma}_1^{-1}\mathbf{m}_1 + \mathbf{\Sigma}_2^{-1}\mathbf{m}_2) \\ \mathbf{\Sigma}_c &= (\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1})^{-1} \end{aligned}$$

but note that the product is not normalized as a density of $\mathbf{x}$.

Matrix Cookbook...

Multivariate Gaussian

The marginal densities

$$
\begin{aligned}
p(\mathbf{x}_A) &= \int_{\mathbf{x}_B} p(\mathbf{x}_A, \mathbf{x}_B | \mu, \Sigma) d\mathbf{x}_B \\
p(\mathbf{x}_B) &= \int_{\mathbf{x}_A} p(\mathbf{x}_A, \mathbf{x}_B | \mu, \Sigma) d\mathbf{x}_A
\end{aligned}
$$

are also Gaussian with the following form,

$$
\begin{aligned}
\mathbf{x}_A &\sim \mathcal{N}(\mu_A, \Sigma_{AA}) \\
\mathbf{x}_B &\sim \mathcal{N}(\mu_B, \Sigma_{BB})
\end{aligned}
$$

Multivariate Gaussian

The conditional densities

$$
\begin{aligned}
p(\mathbf{x}_A|\mathbf{x}_B) &= \frac{p(\mathbf{x}_A, \mathbf{x}_B|\mu, \Sigma)}{\int_{\mathbf{x}_A} p(\mathbf{x}_A, \mathbf{x}_B|\mu, \Sigma) d\mathbf{x}_A} \\
p(\mathbf{x}_B|\mathbf{x}_A) &= \frac{p(\mathbf{x}_A, \mathbf{x}_B|\mu, \Sigma)}{\int_{\mathbf{x}_B} p(\mathbf{x}_A, \mathbf{x}_B|\mu, \Sigma) d\mathbf{x}_B}
\end{aligned}
$$

are also Gaussian with the following form,

$$
\begin{aligned}
\mathbf{x}_A|\mathbf{x}_B &\sim \mathcal{N}\left(\mu_A + \Sigma_{AB}\Sigma_{BB}^{-1}(\mathbf{x}_B - \mu_B), \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}\right) \\
\mathbf{x}_B|\mathbf{x}_A &\sim \mathcal{N}\left(\mu_B + \Sigma_{BA}\Sigma_{AA}^{-1}(\mathbf{x}_A - \mu_A), \Sigma_{BB} - \Sigma_{BA}\Sigma_{AA}^{-1}\Sigma_{AB}\right)
\end{aligned}
$$

Multivariate Gaussian Distributions

Multivariate Gaussian

Although the Gaussian process is an *infinite dimensional* object, it turns out that we only ever need to deal with finitely many points at any one time.

Multivariate Gaussian distributions are all we require!

A vector-valued random variable $\mathbf{x} \in \mathbb{R}^n$ is said to have a multivariate normal (or Gaussian) distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^n$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbf{S}^n_{++}$ if

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{2/n}|\boldsymbol{\Sigma}|^{1/n}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}(\mathbf{x}-\boldsymbol{\mu})\right)$$

where $\mathbf{S}^n_{++}$ denotes the space of $n \times n$ positive-definite symmetric matrices.

Multivariate Gaussian

Gaussian random variables are extremely useful in machine learning and statistics for the following main reasons.

- Measurement error can often be considered to be the accumulation of a large number of small independent random perturbations - by the *Central Limit Theorem*, summations of independent random variables will tend towards being Gaussian distributed.
- They are convenient because many of the integrals involving Gaussian distributions that arise in practice may be written as simple closed form solutions.

Multivariate Gaussian

The following properties of multivariate Gaussians are vital for manipulating Gaussian processes analytically.

Consider a random vector $\mathbf{x} \in \mathbb{R}^n$ with $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. We can then consider the variable $\mathbf{x}$ that has been partitioned into two sets $\mathbf{x}_A = [x_1, \ldots, x_r]^T \in \mathbb{R}^r$ and $\mathbf{x}_B = [x_{r+1}, \ldots, x_n]^T \in \mathbb{R}^{(n-r)}$. We therefore have

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_A \\ \mathbf{x}_B \end{pmatrix} \qquad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{AA} & \boldsymbol{\Sigma}_{AB} \\ \boldsymbol{\Sigma}_{BA} & \boldsymbol{\Sigma}_{BB} \end{pmatrix}$$

We note that $\boldsymbol{\Sigma}_{AB} = \boldsymbol{\Sigma}_{BA}^T$, since $\boldsymbol{\Sigma} = E[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T] = \boldsymbol{\Sigma}^T$.

Multivariate Gaussian

The Gaussian density function is correctly normalised.

$$\int_{\mathbf{x}} p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) d\mathbf{x} = 1$$

Multivariate Gaussian

8.1.8 Product of gaussian densities

Let $\mathcal{N}_x(\mathbf{m}, \boldsymbol{\Sigma})$ denote a density of $\mathbf{x}$, then

$$\mathcal{N}_x(\mathbf{m}_1, \boldsymbol{\Sigma}_1) \cdot \mathcal{N}_x(\mathbf{m}_2, \boldsymbol{\Sigma}_2) = c_c \mathcal{N}_x(\mathbf{m}_c, \boldsymbol{\Sigma}_c) \tag{371}$$

$$\begin{aligned} c_c &= \mathcal{N}_{\mathbf{m}_2}(\mathbf{m}_2, (\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)) \\ &= \frac{1}{\sqrt{\det(2\pi(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2))}} \exp\left[-\frac{1}{2}(\mathbf{m}_1 - \mathbf{m}_2)^T(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)^{-1}(\mathbf{m}_1 - \mathbf{m}_2)\right] \\ \mathbf{m}_c &= (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1}(\boldsymbol{\Sigma}_1^{-1}\mathbf{m}_1 + \boldsymbol{\Sigma}_2^{-1}\mathbf{m}_2) \\ \boldsymbol{\Sigma}_c &= (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})^{-1} \end{aligned}$$

but note that the product is not normalized as a density of $\mathbf{x}$.

Matrix Cookbook...

Multivariate Gaussian

The marginal densities

$$\begin{aligned} p(\mathbf{x}_A) &= \int_{\mathbf{x}_B} p(\mathbf{x}_A, \mathbf{x}_B|\boldsymbol{\mu}, \boldsymbol{\Sigma}) d\mathbf{x}_B \\ p(\mathbf{x}_B) &= \int_{\mathbf{x}_A} p(\mathbf{x}_A, \mathbf{x}_B|\boldsymbol{\mu}, \boldsymbol{\Sigma}) d\mathbf{x}_A \end{aligned}$$

are also Gaussian with the following form,

$$\begin{aligned} \mathbf{x}_A &\sim \mathcal{N}(\boldsymbol{\mu}_A, \boldsymbol{\Sigma}_{AA}) \\ \mathbf{x}_B &\sim \mathcal{N}(\boldsymbol{\mu}_B, \boldsymbol{\Sigma}_{BB}) \end{aligned}$$

Multivariate Gaussian

The conditional densities

$$\begin{aligned} p(\mathbf{x}_A|\mathbf{x}_B) &= \frac{p(\mathbf{x}_A, \mathbf{x}_B|\boldsymbol{\mu}, \boldsymbol{\Sigma})}{\int_{\mathbf{x}_A} p(\mathbf{x}_A, \mathbf{x}_B|\boldsymbol{\mu}, \boldsymbol{\Sigma}) d\mathbf{x}_A} \\ p(\mathbf{x}_B|\mathbf{x}_A) &= \frac{p(\mathbf{x}_A, \mathbf{x}_B|\boldsymbol{\mu}, \boldsymbol{\Sigma})}{\int_{\mathbf{x}_B} p(\mathbf{x}_A, \mathbf{x}_B|\boldsymbol{\mu}, \boldsymbol{\Sigma}) d\mathbf{x}_B} \end{aligned}$$

are also Gaussian with the following form,

$$\begin{aligned} \mathbf{x}_A|\mathbf{x}_B &\sim \mathcal{N}\left(\boldsymbol{\mu}_A + \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_{BB}^{-1}(\mathbf{x}_B - \boldsymbol{\mu}_B), \boldsymbol{\Sigma}_{AA} - \boldsymbol{\Sigma}_{AB}\boldsymbol{\Sigma}_{BB}^{-1}\boldsymbol{\Sigma}_{BA}\right) \\ \mathbf{x}_B|\mathbf{x}_A &\sim \mathcal{N}\left(\boldsymbol{\mu}_B + \boldsymbol{\Sigma}_{BA}\boldsymbol{\Sigma}_{AA}^{-1}(\mathbf{x}_A - \boldsymbol{\mu}_A), \boldsymbol{\Sigma}_{BB} - \boldsymbol{\Sigma}_{BA}\boldsymbol{\Sigma}_{AA}^{-1}\boldsymbol{\Sigma}_{AB}\right) \end{aligned}$$

Gaussian Process Priors

Gaussian Process Prior

We have quite a lot of freedom when choosing the mean and covariance functions that will define our prior over functions.

Usually the mean is set to be the zero function.

The covariance function is much more interesting however, since it defines how the points in our function covary with one another.

The only requirement of a covariance function is that it is positive semi-definite.

Covariance Functions

If k, k_1 and k_2 are positive semi-definite, then we can create new valid covariance functions according the to following functions, which are also positive semi-definite.

1. $ak(\mathbf{x}, \mathbf{x}')R$, for $a \geq 0$
2. $k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}')$
3. $k_1(\mathbf{x}, \mathbf{x}')k_2(\mathbf{x}, \mathbf{x}')$
4. $P(k(\mathbf{x}, \mathbf{x}'))$, where $P()$ is some polynomial function with positive coefficients
5. $exp(k(\mathbf{x}, \mathbf{x}'))$
6. $f(\mathbf{x})k(\mathbf{x}, \mathbf{x}')f(\mathbf{x}')$
7. $k(\phi(\mathbf{x}), \phi(\mathbf{x}'))$

Squared Exponential Covariance Function

The squared exponential covariance function is a common choice,

$$k(x, x') = \sigma_0 \exp\left\{-\frac{1}{2}\left(\frac{x - x'}{\lambda}\right)^2\right\}$$

Intuitively, we see that function variables close to each other in the input space are highly correlated, whereas those far from each other are uncorrelated.

There are hyperparameters that change the output of the covariance function: σ controls the amplitude, λ controls the lengthscale.

The squared exponential produces very smooth functions, since it is *infinitely differentiable*.

Matern Covariance Function

The Matern exponential covariance function is another common choice,

$$k(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}|x - x'|}{\lambda}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}|x - x'|}{\lambda}\right)$$

where K_ν is a modified Bessel function.

There are hyperparameters that change the output of the covariance function: ν controls the roughness of the sample functions, λ controls the lengthscale.

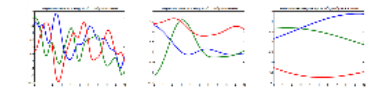
With $\nu = \infty$ we recover the squared exponential covariance, with $\nu = \frac{1}{2}$ we obtain an Ornstein-Uhlenbeck process.

Periodic Covariance Function

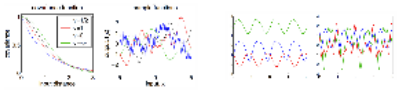
We can even model periodic functions,

$$k(x, x') = \exp\left(\frac{-2\sin^2(x - x')}{\lambda^2}\right)$$

Covariance Function Comparison



Squared Exponential



Matern

Periodic

Gaussian Process Prior

We have quite a lot of freedom when choosing the mean and covariance functions that will define our prior over functions.

Usually the mean is set to be the zero function.

The covariance function is much more interesting however, since it defines how the points in our function covary with one another.

The only requirement of a covariance function is that it is positive semi-definite.

Covariance Functions

If k, k_1 and k_2 are positive semi-definite, then we can create new valid covariance functions according the to following functions, which are also positive semi-definite.

1. $ak(\mathbf{x}, \mathbf{x}')R$, for $a \geq 0$
2. $k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}')$
3. $k_1(\mathbf{x}, \mathbf{x}')k_2(\mathbf{x}, \mathbf{x}')$
4. $P(k(\mathbf{x}, \mathbf{x}'))$, where $P()$ is some polynomial function with positive coefficients
5. $exp(k(\mathbf{x}, \mathbf{x}'))$
6. $f(\mathbf{x})k(\mathbf{x}, \mathbf{x}')f(\mathbf{x}')$
7. $k(\phi(\mathbf{x}), \phi(\mathbf{x}'))$

Squared Exponential Covariance Function

The squared exponential covariance function is a common choice,

$$k(x, x') = \sigma_0 \exp\left\{-\frac{1}{2}\left(\frac{x - x'}{\lambda}\right)^2\right\}$$

Intuitively, we see that function variables close to each other in the input space are highly correlated, whereas those far from each other are uncorrelated.

There are hyperparameters that change the output of the covariance function: σ controls the amplitude, λ controls the lengthscale.

The squared exponential produces very smooth functions, since it is *infinitely differentiable*.

Matern Covariance Function

The Matern exponential covariance function is another common choice,

$$k(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}|x - x'|}{\lambda} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu}|x - x'|}{\lambda} \right)$$

where K_{ν} is a modified Bessel function.

There are hyperparameters that change the output of the covariance function: ν controls the roughness of the sample functions, λ controls the lengthscale.

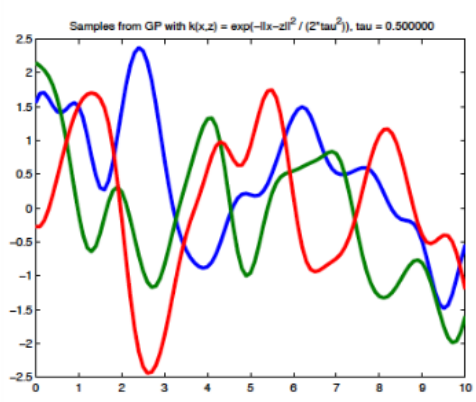
With $\nu = \infty$ we recover the squared exponential covariance, with $\nu = \frac{1}{2}$ we obtain an Ornstein-Uhlenbeck process.

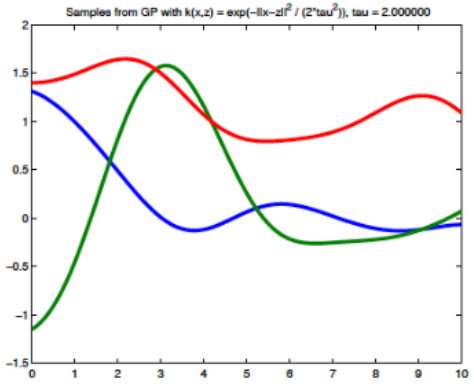
Periodic Covariance Function

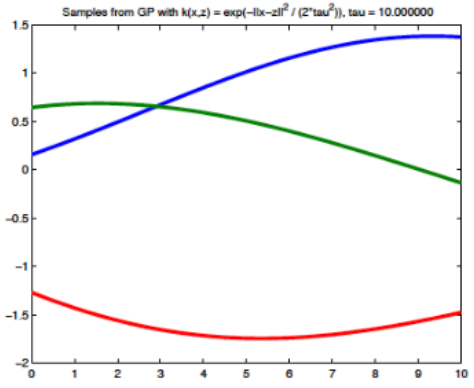
We can even model periodic functions,

$$k(x, x') = \exp\left(\frac{-2\sin^2(x - x')}{\lambda^2}\right)$$

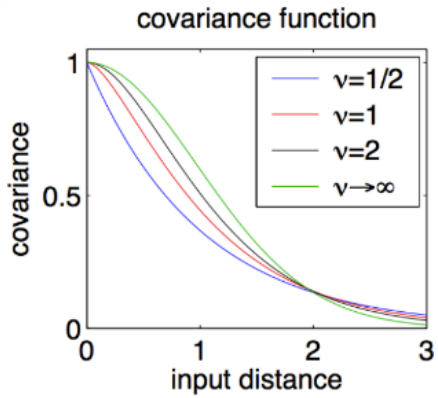
Covariance Function Comparison

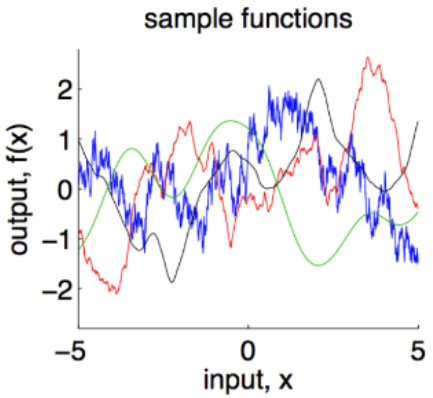




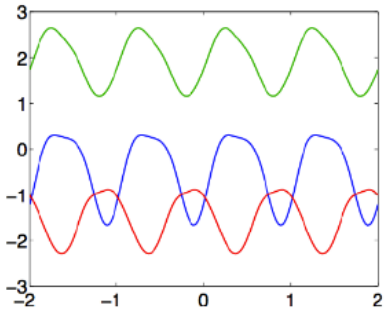


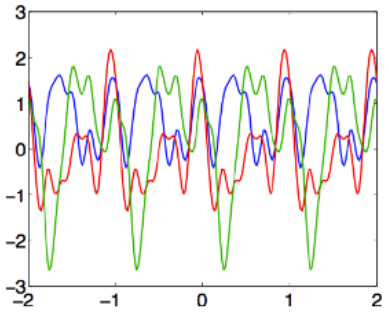
Squared Exponential





Matern





Periodic

Gaussian Process Priors

Gaussian Process Prior

We have quite a lot of freedom when choosing the mean and covariance functions that will define our prior over functions.

Usually the mean is set to be the zero function.

The covariance function is much more interesting however, since it defines how the points in our function covary with one another.

The only requirement of a covariance function is that it is positive semi-definite.

Covariance Functions

If k, k_1 and k_2 are positive semi-definite, then we can create new valid covariance functions according the to following functions, which are also positive semi-definite.

1. $ak(\mathbf{x}, \mathbf{x}')R$, for $a \geq 0$
2. $k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}')$
3. $k_1(\mathbf{x}, \mathbf{x}')k_2(\mathbf{x}, \mathbf{x}')$
4. $P(k(\mathbf{x}, \mathbf{x}'))$, where $P()$ is some polynomial function with positive coefficients
5. $exp(k(\mathbf{x}, \mathbf{x}'))$
6. $f(\mathbf{x})k(\mathbf{x}, \mathbf{x}')f(\mathbf{x}')$
7. $k(\phi(\mathbf{x}), \phi(\mathbf{x}'))$

Squared Exponential Covariance Function

The squared exponential covariance function is a common choice,

$$k(x, x') = \sigma_0 \exp\left\{-\frac{1}{2}\left(\frac{x - x'}{\lambda}\right)^2\right\}$$

Intuitively, we see that function variables close to each other in the input space are highly correlated, whereas those far from each other are uncorrelated.

There are hyperparameters that change the output of the covariance function: σ controls the amplitude, λ controls the lengthscale.

The squared exponential produces very smooth functions, since it is *infinitely differentiable*.

Matern Covariance Function

The Matern exponential covariance function is another common choice,

$$k(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}|x - x'|}{\lambda}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}|x - x'|}{\lambda}\right)$$

where K_ν is a modified Bessel function.

There are hyperparameters that change the output of the covariance function: ν controls the roughness of the sample functions, λ controls the lengthscale.

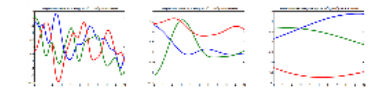
With $\nu = \infty$ we recover the squared exponential covariance, with $\nu = \frac{1}{2}$ we obtain an Ornstein-Uhlenbeck process.

Periodic Covariance Function

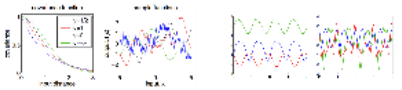
We can even model periodic functions,

$$k(x, x') = \exp\left(\frac{-2\sin^2(x - x')}{\lambda^2}\right)$$

Covariance Function Comparison



Squared Exponential



Matern

Periodic

Making Predictions with Gaussian Processes

Gaussian Process Regression

We can therefore model each datapoint as some function of the inputs plus noise, such that

$$y_i = f(x_i) + \epsilon_i$$

where we can define our prior over functions as a Gaussian process,

$$\mathbf{f}(\mathbf{x}) \sim \mathcal{GP}(m = \mathbf{0}, k)$$

and we can assume Gaussian noise for the likelihood,

$$p(\mathbf{y}|\mathbf{f}) \sim \mathcal{N}(\mathbf{f}, \sigma^2)$$

Conveniently, because everything is Gaussian, we can integrate over all the functions to obtain the marginal likelihood,

$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})d\mathbf{f} = \mathcal{N}(\mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I})$$

where $\mathbf{K}$ is the matrix generated by the covariance function at the given inputs.

Gaussian Process Regression

Let's assume we now have a training set with N measurements $\mathbf{y}_N$ observed with input variables $\mathbf{x}_N$. We then wish to make T predictions of $\mathbf{y}_T$ for some other input variable $\mathbf{x}_T$.

We employ the notation $\mathbf{K}_N$ to denote the matrix obtained by evaluating the covariance function at the N inputs $\mathbf{x}_N$, and $\mathbf{K}_{NT}$ denotes the cross covariance between the inputs $\mathbf{x}_N$ and $\mathbf{x}_T$.

We can consider the marginal likelihood at the joint training and test points,

$$p(\mathbf{y}, \mathbf{y}_T) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{N+T} + \sigma^2\mathbf{I})$$

where

$$\mathbf{K}_{N+T} = \begin{pmatrix} \mathbf{K}_N & \mathbf{K}_{NT} \\ \mathbf{K}_{TN} & \mathbf{K}_T \end{pmatrix}$$

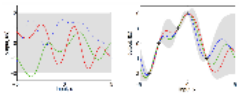
Gaussian Process Regression

Finally, given this partitioned Gaussian distribution, we can condition the test variables on the training variables, such that

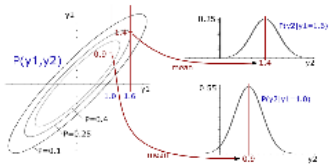
$$p(\mathbf{y}_T|\mathbf{y}) \sim \mathcal{N}(\mu_T, \Sigma_T)$$

where

$$\begin{aligned} \mu_T &= \mathbf{K}_{TN}\left(\mathbf{K}_N + \sigma^2\mathbf{I}\right)^{-1}\mathbf{y} \\ \Sigma_T &= \mathbf{K}_T - \mathbf{K}_{TN}\left(\mathbf{K}_N + \sigma^2\mathbf{I}\right)^{-1}\mathbf{K}_{NT} \end{aligned}$$



Gaussian Process Regression



Choosing Hyperparameters

Other non-Bayesian methods (such as Support Vector Machines) require the use of cross validation to choose the appropriate parameters for obtaining good predictive performance.

The advantage of Gaussian process approach is that we can choose the hyperparameters and covariances directly from the data!

In particular we can minimise the negative log marginal likelihood with respect to all hyperparameters $\boldsymbol{\theta}$,

$$\mathcal{L} = -\log p(\mathbf{y}|\boldsymbol{\theta}) = \frac{1}{2}\log\det\mathbf{C}(\boldsymbol{\theta}) + \frac{1}{2}\mathbf{y}^\top\mathbf{C}(\boldsymbol{\theta})\mathbf{y} + \frac{N}{2}\log(2\pi)$$

where $\mathbf{C}(\boldsymbol{\theta}) = \mathbf{K} + \sigma^2\mathbf{I}$.

Relation to Linear Regression

Let's consider modelling functions defined by a weighted sum of a finite set of basis function, such that

$$f(x) = \sum_{m=1}^{M} \theta_m\phi_m(x) = \boldsymbol{\theta}^\top\boldsymbol{\phi}(x)$$

We can place a Gaussian prior on the weights, so that $p(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, \Sigma_{\boldsymbol{\theta}})$.

This defines a *Gaussian process* with a covariance function given by,

$$K(x, x') = E[f(x), f(x')] = \boldsymbol{\phi}(x)^\top\Sigma_{\boldsymbol{\theta}}\boldsymbol{\phi}(x')$$

Relation to Linear Regression

There is in fact a direct correspondence between covariance functions and basis functions given by Mercer's theorem, which states that we can always decompose a covariance function into eigenfunctions and eigenvectors,

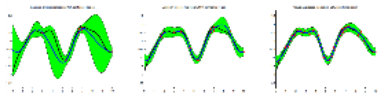
$$K(x, x') = \sum_{i=1}^{\infty} \lambda_i\varphi_i(x)\varphi_i(x')$$

If the sum is finite then it can be defined analogously using linear regression.

Often the sum is infinite, and there are no closed form solutions for the eigenvectors and eigenvalues.

We see how this is another form of the *kernel trick* - the covariance function effectively transforms the input space into a high or even infinite dimensional feature space, yet we can still compute in the lower dimensional space.

Example: Increasing Data



Summary

Gaussian processes are a very powerful approach for performing nonparametric supervised learning.

We can use the kernel trick to allow us to do linear regression effectively using an infinite number of basis functions, while retaining analytic tractability.

The main challenge is the computational scaling; it is cubic in the number of observations.

Gaussian Process Regression

We can therefore model each datapoint as some function of the inputs plus noise, such that

$$y_i = f(x_i) + \epsilon_i$$

where we can define our prior over functions as a Gaussian process,

$$\mathbf{f}(\mathbf{x}) \sim \mathcal{GP}(m = \mathbf{0}, k)$$

and we can assume Gaussian noise for the likelihood,

$$p(\mathbf{y}|\mathbf{f}) \sim \mathcal{N}(\mathbf{f}, \sigma^2)$$

Conveniently, because everything is Gaussian, we can integrate over all the functions to obtain the marginal likelihood,

$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})d\mathbf{f} = \mathcal{N}(\mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I})$$

where $\mathbf{K}$ is the matrix generated by the covariance function at the given inputs.

Gaussian Process Regression

Let's assume we now have a training set with N measurements $\mathbf{y}_N$ observed with input variables $\mathbf{x}_N$. We then wish to make T predictions of $\mathbf{y}_T$ for some other input variable $\mathbf{x}_T$.

We employ the notation $\mathbf{K}_N$ to denote the matrix obtained by evaluating the covariance function at the N inputs $\mathbf{x}_N$, and $\mathbf{K}_{NT}$ denotes the cross covariance between the inputs $\mathbf{x}_N$ and $\mathbf{x}_T$.

We can consider the marginal likelihood at the joint training and test points,

$$p(\mathbf{y}, \mathbf{y}_T) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{N+T} + \sigma^2 \mathbf{I})$$

where

$$\mathbf{K}_{N+T} = \begin{pmatrix} \mathbf{K}_N & \mathbf{K}_{NT} \\ \mathbf{K}_{TN} & \mathbf{K}_T \end{pmatrix}$$

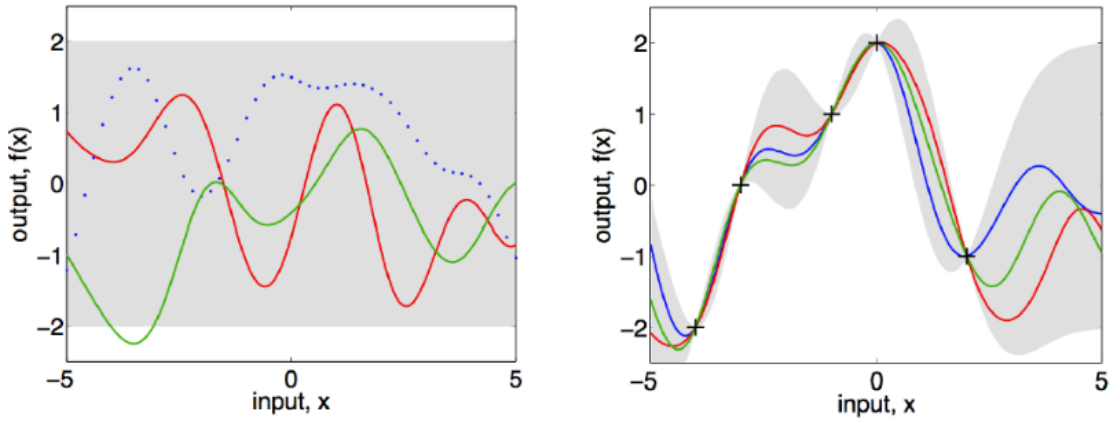
Gaussian Process Regression

Finally, given this partitioned Gaussian distribution, we can condition the test variables on the training variables, such that

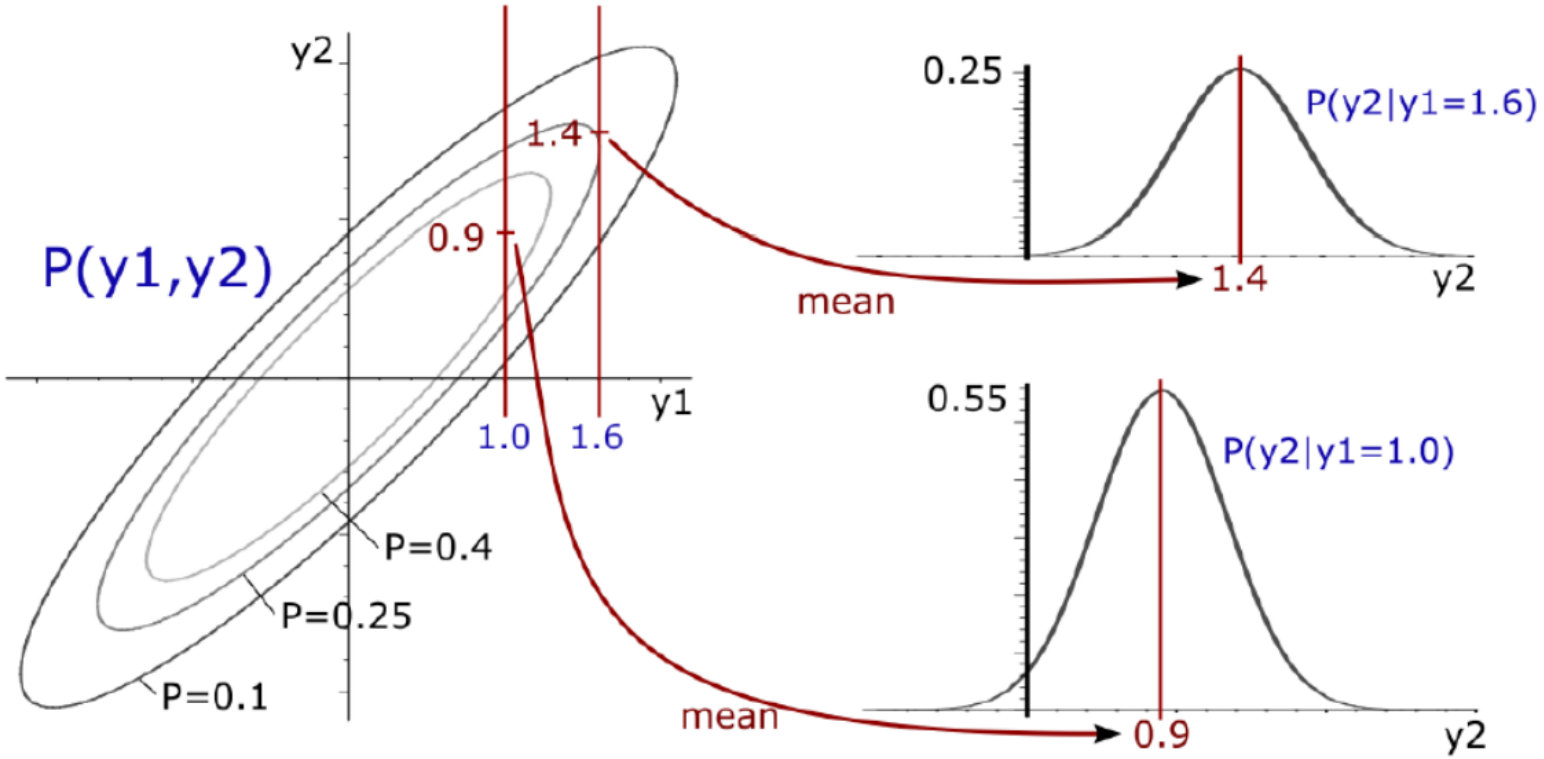
$$p(\mathbf{y}_T|\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_T, \boldsymbol{\Sigma}_T)$$

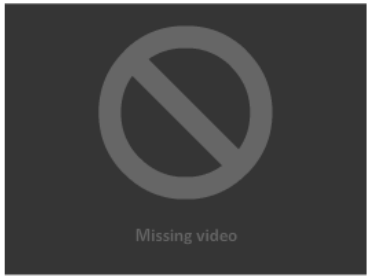
where

$$\begin{aligned} \boldsymbol{\mu}_T &= \mathbf{K}_{TN}\left(\mathbf{K}_N + \sigma^2\mathbf{I}\right)^{-1}\mathbf{y} \\ \boldsymbol{\Sigma}_T &= \mathbf{K}_T - \mathbf{K}_{TN}\left(\mathbf{K}_N + \sigma^2\mathbf{I}\right)^{-1}\mathbf{K}_{NT} \end{aligned}$$



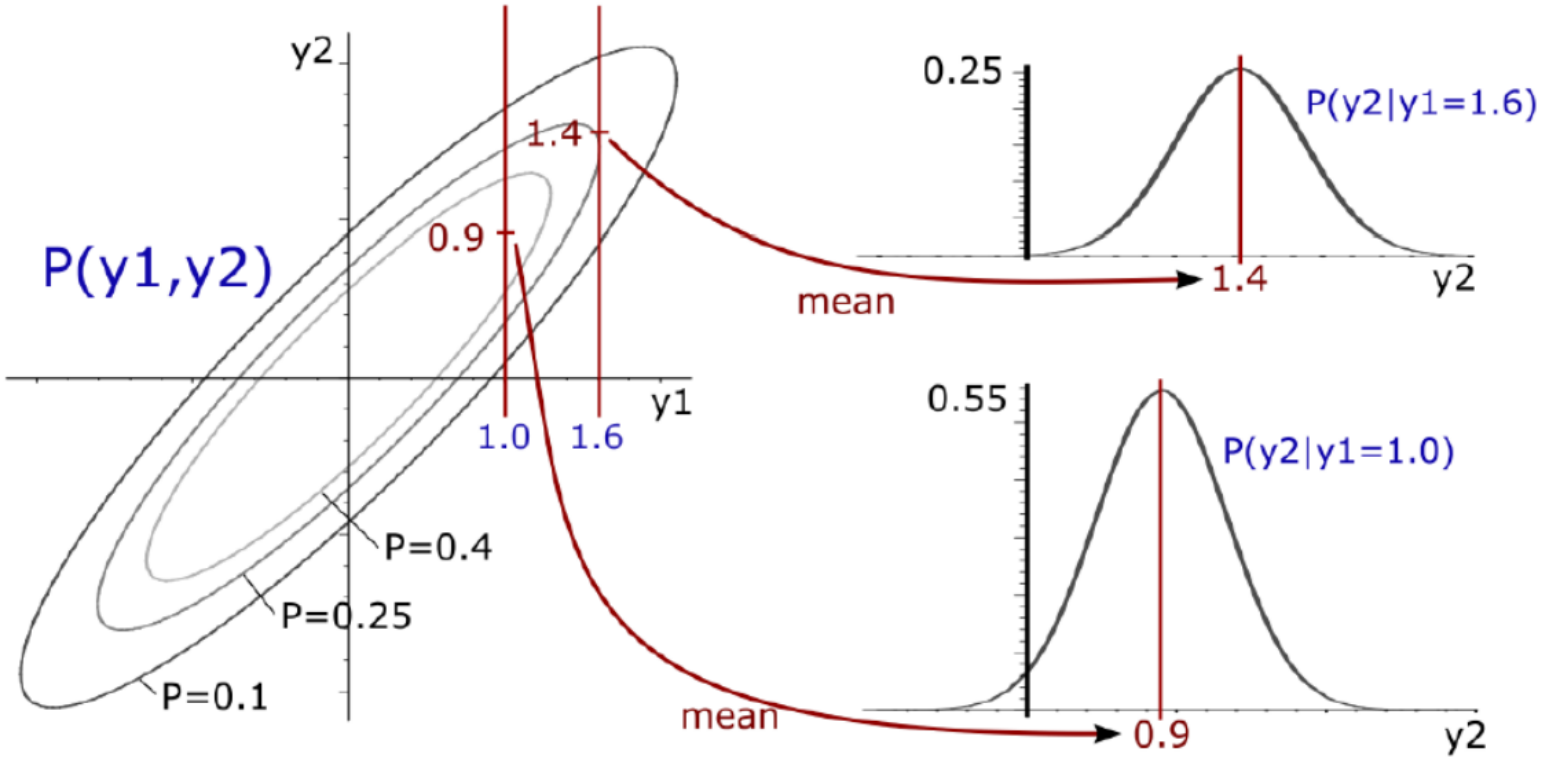
Gaussian Process Regression

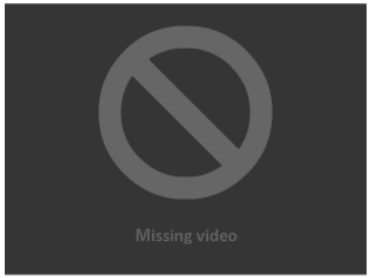




Missing video

Gaussian Process Regression





Choosing Hyperparameters

Other non-Bayesian methods (such as Support Vector Machines) require the use of cross validation to choose the appropriate parameters for obtaining good predictive performance.

The advantage of Gaussian process approach is that we can choose the hyperparameters and covariances directly from the data!

In particular we can minimise the negative log marginal likelihood with respect to all hyperparameters $\boldsymbol{\theta}$,

$$\mathcal{L} = -\log p(\mathbf{y}|\boldsymbol{\theta}) = \frac{1}{2}\log\det\mathbf{C}(\boldsymbol{\theta}) + \frac{1}{2}\mathbf{y}^T\mathbf{C}(\boldsymbol{\theta})\mathbf{y} + \frac{N}{2}\log(2\pi)$$

where $\mathbf{C}(\boldsymbol{\theta}) = \mathbf{K} + \sigma^2\mathbf{I}$.

Relation to Linear Regression

Let's consider modelling functions defined by a weighted sum of a finite set of basis function, such that

$$f(x) = \sum_{m=1}^{M} \theta_m \phi_m(x) = \boldsymbol{\theta}^T \boldsymbol{\phi}(x)$$

We can place a Gaussian prior on the weights, so that $p(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}})$.

This defines a *Gaussian process* with a covariance function given by,

$$K(x, x') = E[f(x), f(x')] = \phi(x)^T \boldsymbol{\Sigma}_{\boldsymbol{\theta}} \phi(x')$$

Relation to Linear Regression

There is in fact a direct correspondence between covariance functions and basis functions given by Mercer's theorem, which states that we can always decompose a covariance function into eigenfunctions and eigenvectors,

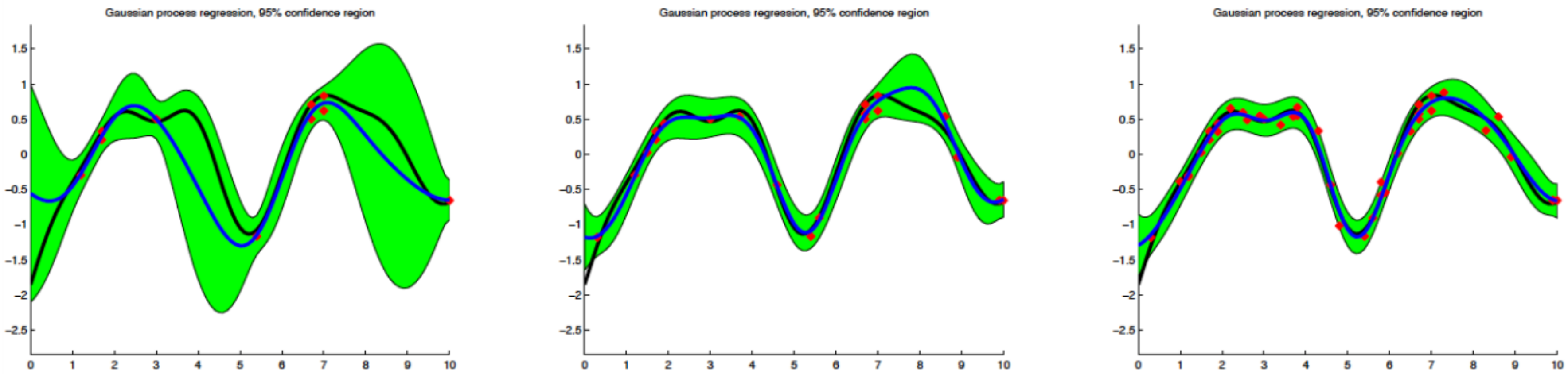
$$K(x, x') = \sum_{i=1}^{\infty} \lambda_i \varphi_i(x) \varphi_i(x')$$

If the sum is finite then it can be defined analogously using linear regression.

Often the sum is infinite, and there are no closed form solutions for the eigenvectors and eigenvalues.

We see how this is another form of the *kernel trick* - the covariance function effectively transforms the input space into a high or even infinite dimensional feature space, yet we can still compute in the lower dimensional space.

Example: Increasing Data



Summary

Gaussian processes are a very powerful approach for performing nonparametric supervised learning.

We can use the kernel trick to allow us to do linear regression effectively using an infinite number of basis functions, while retaining analytic tractability.

The main challenge is the computational scaling; it is cubic in the number of observations.

Making Predictions with Gaussian Processes

Gaussian Process Regression

We can therefore model each datapoint as some function of the inputs plus noise, such that

$$y_i = f(x_i) + \epsilon_i$$

where we can define our prior over functions as a Gaussian process,

$$\mathbf{f}(\mathbf{x}) \sim \mathcal{GP}(m = \mathbf{0}, k)$$

and we can assume Gaussian noise for the likelihood,

$$p(\mathbf{y}|\mathbf{f}) \sim \mathcal{N}(\mathbf{f}, \sigma^2)$$

Conveniently, because everything is Gaussian, we can integrate over all the functions to obtain the marginal likelihood,

$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})d\mathbf{f} = \mathcal{N}(\mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I})$$

where $\mathbf{K}$ is the matrix generated by the covariance function at the given inputs.

Gaussian Process Regression

Let's assume we now have a training set with N measurements $\mathbf{y}_N$ observed with input variables $\mathbf{x}_N$. We then wish to make T predictions of $\mathbf{y}_T$ for some other input variable $\mathbf{x}_T$.

We employ the notation $\mathbf{K}_N$ to denote the matrix obtained by evaluating the covariance function at the N inputs $\mathbf{x}_N$, and $\mathbf{K}_{NT}$ denotes the cross covariance between the inputs $\mathbf{x}_N$ and $\mathbf{x}_T$.

We can consider the marginal likelihood at the joint training and test points,

$$p(\mathbf{y}, \mathbf{y}_T) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{N+T} + \sigma^2\mathbf{I})$$

where

$$\mathbf{K}_{N+T} = \begin{pmatrix} \mathbf{K}_N & \mathbf{K}_{NT} \\ \mathbf{K}_{TN} & \mathbf{K}_T \end{pmatrix}$$

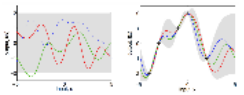
Gaussian Process Regression

Finally, given this partitioned Gaussian distribution, we can condition the test variables on the training variables, such that

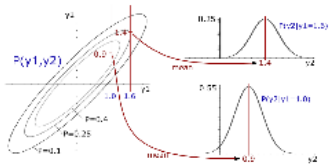
$$p(\mathbf{y}_T|\mathbf{y}) \sim \mathcal{N}(\mu_T, \Sigma_T)$$

where

$$\begin{aligned} \mu_T &= \mathbf{K}_{TN}\left(\mathbf{K}_N + \sigma^2\mathbf{I}\right)^{-1}\mathbf{y} \\ \Sigma_T &= \mathbf{K}_T - \mathbf{K}_{TN}\left(\mathbf{K}_N + \sigma^2\mathbf{I}\right)^{-1}\mathbf{K}_{NT} \end{aligned}$$



Gaussian Process Regression



Choosing Hyperparameters

Other non-Bayesian methods (such as Support Vector Machines) require the use of cross validation to choose the appropriate parameters for obtaining good predictive performance.

The advantage of Gaussian process approach is that we can choose the hyperparameters and covariances directly from the data!

In particular we can minimise the negative log marginal likelihood with respect to all hyperparameters $\boldsymbol{\theta}$,

$$\mathcal{L} = -\log p(\mathbf{y}|\boldsymbol{\theta}) = \frac{1}{2}\log\det\mathbf{C}(\boldsymbol{\theta}) + \frac{1}{2}\mathbf{y}^\top\mathbf{C}(\boldsymbol{\theta})\mathbf{y} + \frac{N}{2}\log(2\pi)$$

where $\mathbf{C}(\boldsymbol{\theta}) = \mathbf{K} + \sigma^2\mathbf{I}$.

Relation to Linear Regression

Let's consider modelling functions defined by a weighted sum of a finite set of basis function, such that

$$f(x) = \sum_{m=1}^{M} \theta_m\phi_m(x) = \boldsymbol{\theta}^\top\boldsymbol{\phi}(x)$$

We can place a Gaussian prior on the weights, so that $p(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, \Sigma_{\boldsymbol{\theta}})$.

This defines a *Gaussian process* with a covariance function given by,

$$K(x, x') = E[f(x), f(x')] = \boldsymbol{\phi}(x)^\top\Sigma_{\boldsymbol{\theta}}\boldsymbol{\phi}(x')$$

Relation to Linear Regression

There is in fact a direct correspondence between covariance functions and basis functions given by Mercer's theorem, which states that we can always decompose a covariance function into eigenfunctions and eigenvectors,

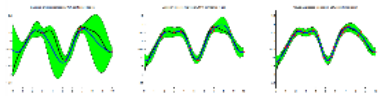
$$K(x, x') = \sum_{i=1}^{\infty} \lambda_i\varphi_i(x)\varphi_i(x')$$

If the sum is finite then it can be defined analogously using linear regression.

Often the sum is infinite, and there are no closed form solutions for the eigenvectors and eigenvalues.

We see how this is another form of the *kernel trick* - the covariance function effectively transforms the input space into a high or even infinite dimensional feature space, yet we can still compute in the lower dimensional space.

Example: Increasing Data



Summary

Gaussian processes are a very powerful approach for performing nonparametric supervised learning.

We can use the kernel trick to allow us to do linear regression effectively using an infinite number of basis functions, while retaining analytic tractability.

The main challenge is the computational scaling; it is cubic in the number of observations.

M5MS10
Machine Learning

Spring 2018

Lecture 6

Dr Ben Calderhead

b.calderhead@imperial.ac.uk