CEE 616: Probabilistic Machine Learning

M4 Nonparametric Methods:
L4b: Gaussian Processes

Jimi Oke

UMass Amherst

College of Engineering

Thu, Nov 12, 2025

Outline

Gaussian processes

Gaussian processes

Given a domain $\mathcal{X}$, a Gaussian process (GP) defines a joint distribution over functions of the form $f : \mathcal{X} \in \mathbb{R}$.

Gaussian processes

Given a domain $\mathcal{X}$, a Gaussian process (GP) defines a joint distribution over functions of the form $f : \mathcal{X} \in \mathbb{R}$.

Assumptions

Gaussian processes

Given a domain $\mathcal{X}$, a Gaussian process (GP) defines a joint distribution over functions of the form $f : \mathcal{X} \in \mathbb{R}$.

Assumptions

- Function values for $M > 0$ inputs $\boldsymbol{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_M)]$ is jointly Gaussian

Gaussian processes

Given a domain $\mathcal{X}$, a Gaussian process (GP) defines a joint distribution over functions of the form $f : \mathcal{X} \in \mathbb{R}$.

Assumptions

- Function values for $M > 0$ inputs $\boldsymbol{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_M)]$ is jointly Gaussian
  - Mean:

Gaussian processes

Given a domain $\mathcal{X}$, a Gaussian process (GP) defines a joint distribution over functions of the form $f : \mathcal{X} \in \mathbb{R}$.

Assumptions

- Function values for $M > 0$ inputs $\boldsymbol{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_M)]$ is jointly Gaussian
  - Mean: $\boldsymbol{\mu} = m(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M)$,

Gaussian processes

Given a domain $\mathcal{X}$, a Gaussian process (GP) defines a joint distribution over functions of the form $f : \mathcal{X} \in \mathbb{R}$.

Assumptions

- Function values for $M > 0$ inputs $\boldsymbol{f} = [f(\boldsymbol{x}_1), \dots, f(\boldsymbol{x}_M)]$ is jointly Gaussian
  - Mean: $\boldsymbol{\mu} = m(\boldsymbol{x}_1, \dots, \boldsymbol{x}_M)$, where m is a mean function

Gaussian processes

Given a domain $\mathcal{X}$, a Gaussian process (GP) defines a joint distribution over functions of the form $f : \mathcal{X} \in \mathbb{R}$.

Assumptions

- Function values for $M > 0$ inputs $\boldsymbol{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_M)]$ is jointly Gaussian
  - Mean: $\boldsymbol{\mu} = m(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M)$, where m is a mean function
  - Covariance:

Gaussian processes

Given a domain $\mathcal{X}$, a Gaussian process (GP) defines a joint distribution over functions of the form $f : \mathcal{X} \in \mathbb{R}$.

Assumptions

- Function values for $M > 0$ inputs $\boldsymbol{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_M)]$ is jointly Gaussian
  - Mean: $\boldsymbol{\mu} = m(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M)$, where m is a mean function
  - Covariance: $\boldsymbol{\Sigma}_{ij} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$,

Gaussian processes

Given a domain $\mathcal{X}$, a Gaussian process (GP) defines a joint distribution over functions of the form $f : \mathcal{X} \in \mathbb{R}$.

Assumptions

- Function values for $M > 0$ inputs $\boldsymbol{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_M)]$ is jointly Gaussian
  - Mean: $\boldsymbol{\mu} = m(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M)$, where m is a mean function
  - Covariance: $\boldsymbol{\Sigma}_{ij} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where $\mathcal{K}$ is a Mercer kernel

Gaussian processes

Given a domain $\mathcal{X}$, a Gaussian process (GP) defines a joint distribution over functions of the form $f : \mathcal{X} \in \mathbb{R}$.

Assumptions

- Function values for $M > 0$ inputs $\boldsymbol{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_M)]$ is jointly Gaussian
  - Mean: $\boldsymbol{\mu} = m(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M)$, where m is a mean function
  - Covariance: $\boldsymbol{\Sigma}_{ij} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where $\mathcal{K}$ is a Mercer kernel

More formally, a GP can be considered as a finite number of r.v.'s which have a joint Gaussian distribution

Gaussian processes

Given a domain $\mathcal{X}$, a Gaussian process (GP) defines a joint distribution over functions of the form $f : \mathcal{X} \in \mathbb{R}$.

Assumptions

- Function values for $M > 0$ inputs $\boldsymbol{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_M)]$ is jointly Gaussian
  - Mean: $\boldsymbol{\mu} = m(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M)$, where m is a mean function
  - Covariance: $\boldsymbol{\Sigma}_{ij} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where $\mathcal{K}$ is a Mercer kernel

More formally, a GP can be considered as a finite number of r.v.'s which have a joint Gaussian distribution

$$\boldsymbol{f}(\boldsymbol{x}) = \mathcal{GP}(\boldsymbol{\mu}, \boldsymbol{\Sigma}_{ij}) \tag{1}$$

Gausssian processes and kernels

Gausssian processes and kernels

- The structure of the covariance $\boldsymbol{\Sigma}_{ij}$ is specified by a **kernel function** (which imposes a prior assumption of the distribution)

Gausssian processes and kernels

- The structure of the covariance $\mathbf{\Sigma}_{ij}$ is specified by a **kernel function** (which imposes a prior assumption of the distribution)
- Gaussian processes can be used for:

Gausssian processes and kernels

- The structure of the covariance $\mathbf{\Sigma}_{ij}$ is specified by a **kernel function** (which imposes a prior assumption of the distribution)
- Gaussian processes can be used for:
  - regression

Gausssian processes and kernels

- The structure of the covariance $\mathbf{\Sigma}_{ij}$ is specified by a **kernel function** (which imposes a prior assumption of the distribution)
- Gaussian processes can be used for:
  - regression
  - classification

Gausssian processes and kernels

- The structure of the covariance $\boldsymbol{\Sigma}_{ij}$ is specified by a **kernel function** (which imposes a prior assumption of the distribution)
- Gaussian processes can be used for:
  - regression
  - classification
  - clustering

Gausssian processes and kernels

- The structure of the covariance $\mathbf{\Sigma}_{ij}$ is specified by a **kernel function** (which imposes a prior assumption of the distribution)
- Gaussian processes can be used for:
  - regression
  - classification
  - clustering

Source: http://krasserm.github.io/2018/03/19/gaussian-processes/

Mercer kernel

Mercer kernel

- Nonparametric methods require an approach to map **prior knowledge** regarding the pairwise similarity of input vectors $(\boldsymbol{x}_i, \boldsymbol{x}_j)$

Mercer kernel

- Nonparametric methods require an approach to map **prior knowledge** regarding the pairwise similarity of input vectors $(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- This is achieved using a kernel function $\mathcal{K}$

Mercer kernel

- Nonparametric methods require an approach to map **prior knowledge** regarding the pairwise similarity of input vectors $(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- This is achieved using a kernel function $\mathcal{K}$
- A Mercer kernel is any symmetric function $\mathcal{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$ such that:

Mercer kernel

- Nonparametric methods require an approach to map **prior knowledge** regarding the pairwise similarity of input vectors $(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- This is achieved using a kernel function $\mathcal{K}$
- A Mercer kernel is any symmetric function $\mathcal{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$ such that:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) c_i c_j \geq 0 \tag{2}$$

Mercer kernel

- Nonparametric methods require an approach to map **prior knowledge** regarding the pairwise similarity of input vectors $(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- This is achieved using a kernel function $\mathcal{K}$
- A Mercer kernel is any symmetric function $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ such that:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) c_i c_j \geq 0 \tag{2}$$

for any set of N points $\boldsymbol{x}_i \in \mathcal{X}$ and any $c_i \in \mathbb{R}$

Gram matrix view

Gram matrix view

The Gram (or kernel) matrix is defined as:

Gram matrix view

The Gram (or kernel) matrix is defined as:

$$\boldsymbol{K} = \begin{pmatrix} \mathcal{K}(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & \mathcal{K}(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ \vdots & \ddots & \vdots \\ \mathcal{K}(\boldsymbol{x}_N, \boldsymbol{x}_1) & \cdots & \mathcal{K}(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{pmatrix} \tag{3}$$

If $\boldsymbol{K}$ is **positive definite** for any set of unique inputs $\boldsymbol{x}_i \in \mathcal{X}, n = 1:N$, then $\mathcal{K}$ is a Mercer kernel.

Gram matrix view

The Gram (or kernel) matrix is defined as:

$$\boldsymbol{K} = \begin{pmatrix} \mathcal{K}(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & \mathcal{K}(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ \vdots & \ddots & \vdots \\ \mathcal{K}(\boldsymbol{x}_N, \boldsymbol{x}_1) & \cdots & \mathcal{K}(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{pmatrix} \tag{3}$$

If $\boldsymbol{K}$ is **positive definite** for any set of unique inputs $\boldsymbol{x}_i \in \mathcal{X}, n = 1 : N$, then $\mathcal{K}$ is a Mercer kernel.

- Thus, a Mercer kernel is also known as a positive definite kernel

Mercer's theorem

Mercer's theorem

Any [symmetric] positive definite matrix $\boldsymbol{K}$ can be eigendecomposed as:

Mercer's theorem

Any [symmetric] positive definite matrix $\boldsymbol{K}$ can be eigendecomposed as:

$$\boldsymbol{K} = \boldsymbol{U}^\top \boldsymbol{\Lambda} \boldsymbol{U} \tag{4}$$

Mercer's theorem

Any [symmetric] positive definite matrix $\boldsymbol{K}$ can be eigendecomposed as:

$$\boldsymbol{K} = \boldsymbol{U}^\top \boldsymbol{\Lambda} \boldsymbol{U} \tag{4}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues

Mercer's theorem

Any [symmetric] positive definite matrix $\boldsymbol{K}$ can be eigendecomposed as:

$$\boldsymbol{K} = \boldsymbol{U}^\top \boldsymbol{\Lambda} \boldsymbol{U} \tag{4}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues $\lambda_i > 0$ and $\boldsymbol{U}$ the matrix of eigenvectors.

Mercer's theorem

Any [symmetric] positive definite matrix $\boldsymbol{K}$ can be eigendecomposed as:

$$\boldsymbol{K} = \boldsymbol{U}^\top \boldsymbol{\Lambda} \boldsymbol{U} \tag{4}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues $\lambda_i > 0$ and $\boldsymbol{U}$ the matrix of eigenvectors.

- Each element k_{ij} can be written as:

Mercer's theorem

Any [symmetric] positive definite matrix $\boldsymbol{K}$ can be eigendecomposed as:

$$\boldsymbol{K} = \boldsymbol{U}^\top \boldsymbol{\Lambda} \boldsymbol{U} \tag{4}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues $\lambda_i > 0$ and $\boldsymbol{U}$ the matrix of eigenvectors.

- Each element k_{ij} can be written as:

$$k_{ij} = (\boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{u}_i)^\top (\boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{u}_j)$$

Mercer's theorem

Any [symmetric] positive definite matrix $\boldsymbol{K}$ can be eigendecomposed as:

$$\boldsymbol{K} = \boldsymbol{U}^\top \boldsymbol{\Lambda} \boldsymbol{U} \tag{4}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues $\lambda_i > 0$ and $\boldsymbol{U}$ the matrix of eigenvectors.

- Each element k_{ij} can be written as:

$$k_{ij} = (\boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{u}_i)^\top (\boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{u}_j) \equiv \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j) \tag{5}$$

Mercer's theorem

Any [symmetric] positive definite matrix $\boldsymbol{K}$ can be eigendecomposed as:

$$\boldsymbol{K} = \boldsymbol{U}^\top \boldsymbol{\Lambda} \boldsymbol{U} \tag{4}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues $\lambda_i > 0$ and $\boldsymbol{U}$ the matrix of eigenvectors.

- Each element k_{ij} can be written as:

$$k_{ij} = (\boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{u}_i)^\top (\boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{u}_j) \equiv \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j) \tag{5}$$

  where $\boldsymbol{u}_i$ is the i-th column/row of $\boldsymbol{U}$.

Mercer's theorem

Any [symmetric] positive definite matrix $\boldsymbol{K}$ can be eigendecomposed as:

$$\boldsymbol{K} = \boldsymbol{U}^\top \boldsymbol{\Lambda} \boldsymbol{U} \tag{4}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues $\lambda_i > 0$ and $\boldsymbol{U}$ the matrix of eigenvectors.

- Each element k_{ij} can be written as:

$$k_{ij} = (\boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{u}_i)^\top (\boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{u}_j) \equiv \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j) \tag{5}$$

  where $\boldsymbol{u}_i$ is the i-th column/row of $\boldsymbol{U}$.
- We can then express each element as an inner product of feature vectors specified by $\boldsymbol{u}_i$:

Mercer's theorem

Any [symmetric] positive definite matrix $\boldsymbol{K}$ can be eigendecomposed as:

$$\boldsymbol{K} = \boldsymbol{U}^\top \boldsymbol{\Lambda} \boldsymbol{U} \tag{4}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues $\lambda_i > 0$ and $\boldsymbol{U}$ the matrix of eigenvectors.

- Each element k_{ij} can be written as:

$$k_{ij} = (\boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{u}_i)^\top (\boldsymbol{\Lambda}^{\frac{1}{2}} \boldsymbol{u}_j) \equiv \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j) \tag{5}$$

  where $\boldsymbol{u}_i$ is the i-th column/row of $\boldsymbol{U}$.
- We can then express each element as an inner product of feature vectors specified by $\boldsymbol{u}_i$:

$$k_{ij} = \sum_m \phi_m(\boldsymbol{x}_i) \phi_m(\boldsymbol{x}_j) \tag{6}$$

Popular Mercer kernels

Popular Mercer kernels

- **Squared exponential** (SE)/RBF kernel (below: $\ell = 1$)

Popular Mercer kernels

- **Squared exponential** (SE)/RBF kernel (below: $\ell = 1$)

Source: https://peterroelants.github.io/posts/gaussian-process-kernels/

Popular Mercer kernels

- **Squared exponential** (SE)/RBF kernel (below: $\ell = 1$)

Source: https://peterroelants.github.io/posts/gaussian-process-kernels/

- **Periodic** kernels (below: $\ell = 1, p = 1$)

Popular Mercer kernels

- **Squared exponential** (SE)/RBF kernel (below: $\ell = 1$)

Source: https://peterroelants.github.io/posts/gaussian-process-kernels/

- **Periodic** kernels (below: $\ell = 1, p = 1$)

Source: https://peterroelants.github.io/posts/gaussian-process-kernels/

Popular Mercer kernels

- **Squared exponential** (SE)/RBF kernel (below: $\ell = 1$)

Source: https://peterroelants.github.io/posts/gaussian-process-kernels/

- **Periodic** kernels (below: $\ell = 1, p = 1$)

Source: https://peterroelants.github.io/posts/gaussian-process-kernels/

- **Automatic relevancy determination** (ARD) kernel

Popular Mercer kernels

- **Squared exponential** (SE)/RBF kernel (below: $\ell = 1$)

Source: https://peterroelants.github.io/posts/gaussian-process-kernels/

- **Periodic** kernels (below: $\ell = 1, p = 1$)

Source: https://peterroelants.github.io/posts/gaussian-process-kernels/

- **Automatic relevancy determination** (ARD) kernel
- Kernels can be composed by addition, multiplication and other operations

Squared exponential kernel

Squared exponential kernel

The squared exponential (SE) kernel is given by:

Squared exponential kernel

The squared exponential (SE) kernel is given by:

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left[-\frac{||\boldsymbol{x} - \boldsymbol{x}'||^2}{2\ell^2}\right] \tag{7}$$

Squared exponential kernel

The squared exponential (SE) kernel is given by:

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left[-\frac{||\boldsymbol{x} - \boldsymbol{x}'||^2}{2\ell^2}\right] \tag{7}$$

where ℓ is the length scale (or **bandwidth**) parameter

Squared exponential kernel

The squared exponential (SE) kernel is given by:

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left[-\frac{||\boldsymbol{x} - \boldsymbol{x}'||^2}{2\ell^2}\right] \tag{7}$$

where ℓ is the length scale (or **bandwidth**) parameter

- Note that the SE kernel kernel measures similarity between $\boldsymbol{x}$ and $\boldsymbol{x}'$ using a scaled Euclidean distance

Squared exponential kernel

The squared exponential (SE) kernel is given by:

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left[-\frac{||\boldsymbol{x} - \boldsymbol{x}'||^2}{2\ell^2}\right] \tag{7}$$

where ℓ is the length scale (or **bandwidth**) parameter

- Note that the SE kernel kernel measures similarity between $\boldsymbol{x}$ and $\boldsymbol{x}'$ using a scaled Euclidean distance
- SE is also referred to as Gaussian, RBF or exponentiated quadratic

Joint Gaussian r.v.'s

Joint Gaussian r.v.'s

If two random vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are jointly Gaussian, then their joint probability is given by:

Joint Gaussian r.v.'s

If two random vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are jointly Gaussian, then their joint probability is given by:

$$p\begin{pmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{pmatrix} = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

Joint Gaussian r.v.'s

If two random vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are jointly Gaussian, then their joint probability is given by:

$$p\begin{pmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\end{pmatrix} = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}\left(\begin{pmatrix}\boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\end{pmatrix}, \begin{pmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12}\\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{pmatrix}\right) \tag{8}$$

Marginalization

Marginalization

Provides the distribution on each vector (partial information).

Marginalization

Provides the distribution on each vector (partial information).
The marginal distributions are given by:

Marginalization

Provides the distribution on each vector (partial information).
The marginal distributions are given by:

$$p(\boldsymbol{x}_1) \quad =$$

Marginalization

Provides the distribution on each vector (partial information).
The marginal distributions are given by:

$$p(\boldsymbol{x}_1) \quad = \quad \mathcal{N}(\boldsymbol{x}_1|\boldsymbol{m}u_1, \boldsymbol{\Sigma}_{11}) \tag{9}$$

Marginalization

Provides the distribution on each vector (partial information).
The marginal distributions are given by:

$$
\begin{aligned}
p(\boldsymbol{x}_1) &= \mathcal{N}(\boldsymbol{x}_1|\boldsymbol{m}u_1, \boldsymbol{\Sigma}_{11}) \qquad (9)\\
p(\boldsymbol{x}_2) &=
\end{aligned}
$$

Marginalization

Provides the distribution on each vector (partial information).
The marginal distributions are given by:

$$p(\boldsymbol{x}_1) = \mathcal{N}(\boldsymbol{x}_1 | \boldsymbol{m}u_1, \boldsymbol{\Sigma}_{11}) \quad (9)$$

$$p(\boldsymbol{x}_2) = \mathcal{N}(\boldsymbol{x}_2 | \boldsymbol{m}u_2, \boldsymbol{\Sigma}_{22}) \quad (10)$$

Marginalization

Provides the distribution on each vector (partial information).
The marginal distributions are given by:

$$p(\boldsymbol{x}_1) = \mathcal{N}(\boldsymbol{x}_1|\boldsymbol{m}u_1, \boldsymbol{\Sigma}_{11}) \quad (9)$$

$$p(\boldsymbol{x}_2) = \mathcal{N}(\boldsymbol{x}_2|\boldsymbol{m}u_2, \boldsymbol{\Sigma}_{22}) \quad (10)$$

Marginal distributions are also Gaussian.

Conditioning

Conditioning

The **posterior** conditional distribution is:

Conditioning

The **posterior** conditional distribution is:

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) = \mathcal{N}(\boldsymbol{x}_1|\boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}) \tag{11}$$

Conditioning

The **posterior** conditional distribution is:

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) = \mathcal{N}(\boldsymbol{x}_1|\boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}) \tag{11}$$

where:

$$\boldsymbol{\mu}_{1|2} \quad =$$

Conditioning

The **posterior** conditional distribution is:

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) = \mathcal{N}(\boldsymbol{x}_1|\boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}) \tag{11}$$

where:

$$\boldsymbol{\mu}_{1|2} \quad = \quad \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\Sigma_{22}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)$$

Conditioning

The **posterior** conditional distribution is:

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) = \mathcal{N}(\boldsymbol{x}_1|\boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}) \tag{11}$$

where:

$$\boldsymbol{\mu}_{1|2} \quad = \quad \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\Sigma_{22}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \quad \text{(posterior mean)}$$

Conditioning

The **posterior** conditional distribution is:

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) = \mathcal{N}(\boldsymbol{x}_1|\boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}) \tag{11}$$

where:

$$\begin{aligned}
\boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\Sigma_{22}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \quad \text{(posterior mean)} \\
\boldsymbol{\Sigma}_{1|2} &=
\end{aligned}$$

Conditioning

The **posterior** conditional distribution is:

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) = \mathcal{N}(\boldsymbol{x}_1|\boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}) \tag{11}$$

where:

$$\begin{aligned} \boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \quad \text{(posterior mean)} \\ \boldsymbol{\Sigma}_{1|2} &= \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} \end{aligned}$$

Conditioning

The **posterior** conditional distribution is:

$$p(\boldsymbol{x}_1|\boldsymbol{x}_2) = \mathcal{N}(\boldsymbol{x}_1|\boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}) \tag{11}$$

where:

$$\begin{aligned} \boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \quad \text{(posterior mean)} \\ \boldsymbol{\Sigma}_{1|2} &= \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} \quad \text{(posterior covariance)} \end{aligned}$$

Noise-free observations

Noise-free observations

In standard GP, we assume that observations in the training set are noise-free (exact output based on a known function f):

Noise-free observations

In standard GP, we assume that observations in the training set are noise-free (exact output based on a known function f):

$$\mathcal{D} = \{(\boldsymbol{x}_n, y_n) : n = 1 : N\} \tag{12}$$

Noise-free observations

In standard GP, we assume that observations in the training set are noise-free (exact output based on a known function f):

$$
\begin{aligned}
\mathcal{D} &= \{(\boldsymbol{x}_n, y_n) : n = 1 : N\} \qquad (12)\\
y_n &= f(\boldsymbol{x}_n)
\end{aligned}
$$

Noise-free observations

In standard GP, we assume that observations in the training set are noise-free (exact output based on a known function f):

$$\mathcal{D} = \{(\boldsymbol{x}_n, y_n) : n = 1 : N\} \tag{12}$$

$$y_n = f(\boldsymbol{x}_n) \quad \text{(noise-free observation)} \tag{13}$$

Noise-free observations

In standard GP, we assume that observations in the training set are noise-free (exact output based on a known function f):

$$\mathcal{D} = \{(\boldsymbol{x}_n, y_n) : n = 1 : N\} \tag{12}$$

$$y_n = f(\boldsymbol{x}_n) \quad \text{(noise-free observation)} \tag{13}$$

Source: https://www.aidanscannell.com/post/gaussian-process-regression/

Noise-free observations

In standard GP, we assume that observations in the training set are noise-free (exact output based on a known function f):

$$
\begin{aligned}
\mathcal{D} &= \{(\boldsymbol{x}_n, y_n) : n = 1 : N\} && (12)\\
y_n &= f(\boldsymbol{x}_n) \quad \text{(noise-free observation)} && (13)
\end{aligned}
$$

Source: https://www.aidanscannell.com/post/gaussian-process-regression/

To predict function outputs for new inputs $\boldsymbol{x}^*$, we estimate the *posterior conditional distribution*

Joint distribution

The joint distribution $p(\boldsymbol{f}_X, \boldsymbol{f}_* | \boldsymbol{X}, \boldsymbol{X}_*)$ of function outputs is given by:

Joint distribution

The joint distribution $p(\boldsymbol{f}_X, \boldsymbol{f}_* | \boldsymbol{X}, \boldsymbol{X}_*)$ of function outputs is given by:

$$\begin{pmatrix} \boldsymbol{f}_X \\ \boldsymbol{f}_* \end{pmatrix}$$

Joint distribution

The joint distribution $p(\boldsymbol{f}_X, \boldsymbol{f}_* | \boldsymbol{X}, \boldsymbol{X}_*)$ of function outputs is given by:

$$\begin{pmatrix} \boldsymbol{f}_X \\ \boldsymbol{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{pmatrix} \boldsymbol{K}_{X,X} & \boldsymbol{K}_{X,*} \\ \boldsymbol{K}_{X,*}^\top & \boldsymbol{K}_{*,*} \end{pmatrix} \right) \tag{14}$$

Joint distribution

The joint distribution $p(\boldsymbol{f}_X, \boldsymbol{f}_* | \boldsymbol{X}, \boldsymbol{X}_*)$ of function outputs is given by:

$$\begin{pmatrix} \boldsymbol{f}_X \\ \boldsymbol{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{pmatrix} \boldsymbol{K}_{X,X} & \boldsymbol{K}_{X,*} \\ \boldsymbol{K}_{X,*}^\top & \boldsymbol{K}_{*,*} \end{pmatrix} \right) \tag{14}$$

where:

- $\boldsymbol{f}_X$: function outputs over training set (interpolator): $[f(\boldsymbol{x}_1), \ldots, f(x_N)]$

Joint distribution

The joint distribution $p(\boldsymbol{f}_X, \boldsymbol{f}_* | \boldsymbol{X}, \boldsymbol{X}_*)$ of function outputs is given by:

$$\begin{pmatrix} \boldsymbol{f}_X \\ \boldsymbol{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{pmatrix} \boldsymbol{K}_{X,X} & \boldsymbol{K}_{X,*} \\ \boldsymbol{K}_{X,*}^\top & \boldsymbol{K}_{*,*} \end{pmatrix} \right) \tag{14}$$

where:

- $\boldsymbol{f}_X$: function outputs over training set (interpolator): $[f(\boldsymbol{x}_1), \ldots, f(x_N)]$
- $\boldsymbol{f}_*$: function outputs over test set: $[f(\boldsymbol{x}_1^*), \ldots, f(x_{N_*}^*)]$

Joint distribution

The joint distribution $p(\boldsymbol{f}_X, \boldsymbol{f}_* | \boldsymbol{X}, \boldsymbol{X}_*)$ of function outputs is given by:

$$\begin{pmatrix} \boldsymbol{f}_X \\ \boldsymbol{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{pmatrix} \boldsymbol{K}_{X,X} & \boldsymbol{K}_{X,*} \\ \boldsymbol{K}_{X,*}^\top & \boldsymbol{K}_{*,*} \end{pmatrix} \right) \tag{14}$$

where:

- $\boldsymbol{f}_X$: function outputs over training set (interpolator): $[f(\boldsymbol{x}_1), \ldots, f(x_N)]$
- $\boldsymbol{f}_*$: function outputs over test set: $[f(\boldsymbol{x}_1^*), \ldots, f(x_{N_*}^*)]$
- Test inputs: $\boldsymbol{X}_* \in \boldsymbol{R}^{N_* \times D}$;

Joint distribution

The joint distribution $p(\boldsymbol{f}_X, \boldsymbol{f}_* | \boldsymbol{X}, \boldsymbol{X}_*)$ of function outputs is given by:

$$\begin{pmatrix} \boldsymbol{f}_X \\ \boldsymbol{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{pmatrix} \boldsymbol{K}_{X,X} & \boldsymbol{K}_{X,*} \\ \boldsymbol{K}_{X,*}^\top & \boldsymbol{K}_{*,*} \end{pmatrix} \right) \tag{14}$$

where:

- $\boldsymbol{f}_X$: function outputs over training set (interpolator): $[f(\boldsymbol{x}_1), \ldots, f(x_N)]$
- $\boldsymbol{f}_*$: function outputs over test set: $[f(\boldsymbol{x}_1^*), \ldots, f(x_{N_*}^*)]$
- Test inputs: $\boldsymbol{X}_* \in \boldsymbol{R}^{N_* \times D}$; training inputs: $\boldsymbol{X} \in \mathbb{R}^{N \times D}$

Joint distribution

The joint distribution $p(\boldsymbol{f}_X, \boldsymbol{f}_* | \boldsymbol{X}, \boldsymbol{X}_*)$ of function outputs is given by:

$$\begin{pmatrix} \boldsymbol{f}_X \\ \boldsymbol{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{pmatrix} \boldsymbol{K}_{X,X} & \boldsymbol{K}_{X,*} \\ \boldsymbol{K}_{X,*}^\top & \boldsymbol{K}_{*,*} \end{pmatrix} \right) \tag{14}$$

where:

- $\boldsymbol{f}_X$: function outputs over training set (interpolator): $[f(\boldsymbol{x}_1), \ldots, f(x_N)]$
- $\boldsymbol{f}_*$: function outputs over test set: $[f(\boldsymbol{x}_1^*), \ldots, f(x_{N_*}^*)]$
- Test inputs: $\boldsymbol{X}_* \in \boldsymbol{R}^{N_* \times D}$; training inputs: $\boldsymbol{X} \in \mathbb{R}^{N \times D}$
- Mean vectors:

Joint distribution

The joint distribution $p(\boldsymbol{f}_X, \boldsymbol{f}_* | \boldsymbol{X}, \boldsymbol{X}_*)$ of function outputs is given by:

$$\begin{pmatrix} \boldsymbol{f}_X \\ \boldsymbol{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{pmatrix} \boldsymbol{K}_{X,X} & \boldsymbol{K}_{X,*} \\ \boldsymbol{K}_{X,*}^\top & \boldsymbol{K}_{*,*} \end{pmatrix} \right) \tag{14}$$

where:

- $\boldsymbol{f}_X$: function outputs over training set (interpolator): $[f(\boldsymbol{x}_1), \ldots, f(x_N)]$
- $\boldsymbol{f}_*$: function outputs over test set: $[f(\boldsymbol{x}_1^*), \ldots, f(x_{N_*}^*)]$
- Test inputs: $\boldsymbol{X}_* \in \boldsymbol{R}^{N_* \times D}$; training inputs: $\boldsymbol{X} \in \mathbb{R}^{N \times D}$
- Mean vectors: $\boldsymbol{\mu}_X = m(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ and

Joint distribution

The joint distribution $p(\boldsymbol{f}_X, \boldsymbol{f}_* | \boldsymbol{X}, \boldsymbol{X}_*)$ of function outputs is given by:

$$\begin{pmatrix} \boldsymbol{f}_X \\ \boldsymbol{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{pmatrix} \boldsymbol{K}_{X,X} & \boldsymbol{K}_{X,*} \\ \boldsymbol{K}_{X,*}^\top & \boldsymbol{K}_{*,*} \end{pmatrix} \right) \tag{14}$$

where:

- $\boldsymbol{f}_X$: function outputs over training set (interpolator): $[f(\boldsymbol{x}_1), \dots, f(x_N)]$
- $\boldsymbol{f}_*$: function outputs over test set: $[f(\boldsymbol{x}_1^*), \dots, f(x_{N_*}^*)]$
- Test inputs: $\boldsymbol{X}_* \in \boldsymbol{R}^{N_* \times D}$; training inputs: $\boldsymbol{X} \in \mathbb{R}^{N \times D}$
- Mean vectors: $\boldsymbol{\mu}_X = m(\boldsymbol{x}_1, \dots, \boldsymbol{x}_N)$ and $\boldsymbol{\mu}_* = m(\boldsymbol{x}_1^*, \dots, \boldsymbol{x}_{N_*}^*)$

Joint distribution

The joint distribution $p(\boldsymbol{f}_X, \boldsymbol{f}_* | \boldsymbol{X}, \boldsymbol{X}_*)$ of function outputs is given by:

$$\begin{pmatrix} \boldsymbol{f}_X \\ \boldsymbol{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{pmatrix} \boldsymbol{K}_{X,X} & \boldsymbol{K}_{X,*} \\ \boldsymbol{K}_{X,*}^\top & \boldsymbol{K}_{*,*} \end{pmatrix} \right) \tag{14}$$

where:

- $\boldsymbol{f}_X$: function outputs over training set (interpolator): $[f(\boldsymbol{x}_1), \dots, f(x_N)]$
- $\boldsymbol{f}_*$: function outputs over test set: $[f(\boldsymbol{x}_1^*), \dots, f(x_{N_*}^*)]$
- Test inputs: $\boldsymbol{X}_* \in \boldsymbol{R}^{N_* \times D}$; training inputs: $\boldsymbol{X} \in \mathbb{R}^{N \times D}$
- Mean vectors: $\boldsymbol{\mu}_X = m(\boldsymbol{x}_1, \dots, \boldsymbol{x}_N)$ and $\boldsymbol{\mu}_* = m(\boldsymbol{x}_1^*, \dots, \boldsymbol{x}_{N_*}^*)$
- Block covariance matrix:

Joint distribution

The joint distribution $p(\boldsymbol{f}_X, \boldsymbol{f}_* | \boldsymbol{X}, \boldsymbol{X}_*)$ of function outputs is given by:

$$\begin{pmatrix} \boldsymbol{f}_X \\ \boldsymbol{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{pmatrix} \boldsymbol{K}_{X,X} & \boldsymbol{K}_{X,*} \\ \boldsymbol{K}_{X,*}^\top & \boldsymbol{K}_{*,*} \end{pmatrix} \right) \tag{14}$$

where:

- $\boldsymbol{f}_X$: function outputs over training set (interpolator): $[f(\boldsymbol{x}_1), \ldots, f(x_N)]$
- $\boldsymbol{f}_*$: function outputs over test set: $[f(\boldsymbol{x}_1^*), \ldots, f(x_{N_*}^*)]$
- Test inputs: $\boldsymbol{X}_* \in \boldsymbol{R}^{N_* \times D}$; training inputs: $\boldsymbol{X} \in \mathbb{R}^{N \times D}$
- Mean vectors: $\boldsymbol{\mu}_X = m(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ and $\boldsymbol{\mu}_* = m(\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_{N_*}^*)$
- Block covariance matrix:
    - $\boldsymbol{K}_{X,X} = \mathcal{K}(\boldsymbol{X}, \boldsymbol{X}) \in \mathbb{R}^{N \times N}$

Joint distribution

The joint distribution $p(\boldsymbol{f}_X, \boldsymbol{f}_* | \boldsymbol{X}, \boldsymbol{X}_*)$ of function outputs is given by:

$$\begin{pmatrix} \boldsymbol{f}_X \\ \boldsymbol{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{pmatrix} \boldsymbol{K}_{X,X} & \boldsymbol{K}_{X,*} \\ \boldsymbol{K}_{X,*}^\top & \boldsymbol{K}_{*,*} \end{pmatrix} \right) \tag{14}$$

where:

- $\boldsymbol{f}_X$: function outputs over training set (interpolator): $[f(\boldsymbol{x}_1), \ldots, f(x_N)]$
- $\boldsymbol{f}_*$: function outputs over test set: $[f(\boldsymbol{x}_1^*), \ldots, f(x_{N_*}^*)]$
- Test inputs: $\boldsymbol{X}_* \in \boldsymbol{R}^{N_* \times D}$; training inputs: $\boldsymbol{X} \in \mathbb{R}^{N \times D}$
- Mean vectors: $\boldsymbol{\mu}_X = m(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ and $\boldsymbol{\mu}_* = m(\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_{N_*}^*)$
- Block covariance matrix:
  - $\boldsymbol{K}_{X,X} = \mathcal{K}(\boldsymbol{X}, \boldsymbol{X}) \in \mathbb{R}^{N \times N}$
  - $\boldsymbol{K}_{X,*} = \mathcal{K}(\boldsymbol{X}, \boldsymbol{X}_*) \in \mathbb{R}^{N \times N_*}$

Joint distribution

The joint distribution $p(\boldsymbol{f}_X, \boldsymbol{f}_* | \boldsymbol{X}, \boldsymbol{X}_*)$ of function outputs is given by:

$$\begin{pmatrix} \boldsymbol{f}_X \\ \boldsymbol{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{pmatrix} \boldsymbol{K}_{X,X} & \boldsymbol{K}_{X,*} \\ \boldsymbol{K}_{X,*}^\top & \boldsymbol{K}_{*,*} \end{pmatrix} \right) \tag{14}$$

where:

- $\boldsymbol{f}_X$: function outputs over training set (interpolator): $[f(\boldsymbol{x}_1), \ldots, f(x_N)]$
- $\boldsymbol{f}_*$: function outputs over test set: $[f(\boldsymbol{x}_1^*), \ldots, f(x_{N_*}^*)]$
- Test inputs: $\boldsymbol{X}_* \in \boldsymbol{R}^{N_* \times D}$; training inputs: $\boldsymbol{X} \in \mathbb{R}^{N \times D}$
- Mean vectors: $\boldsymbol{\mu}_X = m(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ and $\boldsymbol{\mu}_* = m(\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_{N_*}^*)$
- Block covariance matrix:
  - $\boldsymbol{K}_{X,X} = \mathcal{K}(\boldsymbol{X}, \boldsymbol{X}) \in \mathbb{R}^{N \times N}$
  - $\boldsymbol{K}_{X,*} = \mathcal{K}(\boldsymbol{X}, \boldsymbol{X}_*) \in \mathbb{R}^{N \times N_*}$
  - $\boldsymbol{K}_{*,*} = \mathcal{K}(\boldsymbol{X}_*, \boldsymbol{X}_*) \in \mathbb{R}^{N_* \times N_*}$

Posterior conditional distribution

Posterior conditional distribution

We sample predictions from the posterior conditional distributon, which is specified as:

Posterior conditional distribution

We sample predictions from the posterior conditional distributon, which is specified as:

$$p(\boldsymbol{f}_* | \boldsymbol{X}_*, \mathcal{D})$$

Posterior conditional distribution

We sample predictions from the posterior conditional distributon, which is specified as:

$$p(\boldsymbol{f}_*|\boldsymbol{X}_*, \mathcal{D}) \quad = \quad \mathcal{N}(\boldsymbol{f}_*|\boldsymbol{u}_{*|X}, \boldsymbol{\Sigma}_{*|X})$$

Posterior conditional distribution

We sample predictions from the posterior conditional distributon, which is specified as:

$$p(\boldsymbol{f}_*|\boldsymbol{X}_*, \mathcal{D}) \quad = \quad \mathcal{N}(\boldsymbol{f}_*|\boldsymbol{u}_{*|X}, \boldsymbol{\Sigma}_{*|X}) \tag{15}$$

Posterior conditional distribution

We sample predictions from the posterior conditional distributon, which is specified as:

$$
\begin{aligned}
p(\boldsymbol{f}_* | \boldsymbol{X}_*, \mathcal{D}) &= \mathcal{N}(\boldsymbol{f}_* | \boldsymbol{u}_{*|X}, \boldsymbol{\Sigma}_{*|X}) \qquad (15) \\
\boldsymbol{\mu}_{*|X} &
\end{aligned}
$$

Posterior conditional distribution

We sample predictions from the posterior conditional distributon, which is specified as:

$$p(\boldsymbol{f}_*|\boldsymbol{X}_*, \mathcal{D}) = \mathcal{N}(\boldsymbol{f}_*|\boldsymbol{u}_{*|X}, \boldsymbol{\Sigma}_{*|X}) \quad (15)$$

$$\boldsymbol{\mu}_{*|X} = m(\boldsymbol{X}_*) + \boldsymbol{K}_{X,*}^{\top} K_{X,X}^{-1}(\boldsymbol{f}_X - m(\boldsymbol{X})) \quad (16)$$

Posterior conditional distribution

We sample predictions from the posterior conditional distributon, which is specified as:

$$p(\boldsymbol{f}_*|\boldsymbol{X}_*, \mathcal{D}) = \mathcal{N}(\boldsymbol{f}_*|\boldsymbol{u}_{*|X}, \boldsymbol{\Sigma}_{*|X}) \quad (15)$$

$$\boldsymbol{\mu}_{*|X} = m(\boldsymbol{X}_*) + \boldsymbol{K}_{X,*}^{\top} K_{X,X}^{-1}(\boldsymbol{f}_X - m(\boldsymbol{X})) \quad (16)$$

$$\boldsymbol{\Sigma}_{*|X}$$

Posterior conditional distribution

We sample predictions from the posterior conditional distributon, which is specified as:

$$
\begin{aligned}
p(\boldsymbol{f}_*|\boldsymbol{X}_*, \mathcal{D}) &= \mathcal{N}(\boldsymbol{f}_*|\boldsymbol{u}_{*|X}, \boldsymbol{\Sigma}_{*|X}) && (15)\\
\boldsymbol{\mu}_{*|X} &= m(\boldsymbol{X}_*) + \boldsymbol{K}_{X,*}^\top K_{X,X}^{-1}(\boldsymbol{f}_X - m(\boldsymbol{X})) && (16)\\
\boldsymbol{\Sigma}_{*|X} &= \boldsymbol{K}_{*,*} - \boldsymbol{K}_{X,*}^\top \boldsymbol{K}_{X,X}^{-1} \boldsymbol{K}_{X,*} && (17)
\end{aligned}
$$

Noisy observations

Noisy observations

If we have a noisy realization of the underlying function, then:

Noisy observations

If we have a noisy realization of the underlying function, then:

$$y_n = f(\boldsymbol{x}_n) + \epsilon_n \tag{18}$$

Noisy observations

If we have a noisy realization of the underlying function, then:

$$y_n = f(\boldsymbol{x}_n) + \epsilon_n \tag{18}$$

where $\epsilon_n \sim \mathcal{N}(0, \sigma_y^2)$.

Noisy observations

If we have a noisy realization of the underlying function, then:

$$y_n = f(\boldsymbol{x}_n) + \epsilon_n \tag{18}$$

where $\epsilon_n \sim \mathcal{N}(0, \sigma_y^2)$.
The covariance of the observed noisy responses $\boldsymbol{y}$ is thus given by:

Noisy observations

If we have a noisy realization of the underlying function, then:

$$y_n = f(\boldsymbol{x}_n) + \epsilon_n \tag{18}$$

where $\epsilon_n \sim \mathcal{N}(0, \sigma_y^2)$.
The covariance of the observed noisy responses $\boldsymbol{y}$ is thus given by:

$$\text{Cov}[\boldsymbol{y}|\boldsymbol{X}] =$$

Noisy observations

If we have a noisy realization of the underlying function, then:

$$y_n = f(\boldsymbol{x}_n) + \epsilon_n \tag{18}$$

where $\epsilon_n \sim \mathcal{N}(0, \sigma_y^2)$.
The covariance of the observed noisy responses $\boldsymbol{y}$ is thus given by:

$$\text{Cov}[\boldsymbol{y}|\boldsymbol{X}] = \boldsymbol{K}_{X,X} + \sigma_y^2 \boldsymbol{I}_N$$

Noisy observations

If we have a noisy realization of the underlying function, then:

$$y_n = f(\boldsymbol{x}_n) + \epsilon_n \tag{18}$$

where $\epsilon_n \sim \mathcal{N}(0, \sigma_y^2)$.
The covariance of the observed noisy responses $\boldsymbol{y}$ is thus given by:

$$\text{Cov}[\boldsymbol{y}|\boldsymbol{X}] = \boldsymbol{K}_{X,X} + \sigma_y^2 \boldsymbol{I}_N \equiv \hat{\boldsymbol{K}}_{X,X} \tag{19}$$

Noisy observations

If we have a noisy realization of the underlying function, then:

$$y_n = f(\boldsymbol{x}_n) + \epsilon_n \tag{18}$$

where $\epsilon_n \sim \mathcal{N}(0, \sigma_y^2)$.
The covariance of the observed noisy responses $\boldsymbol{y}$ is thus given by:

$$\text{Cov}[\boldsymbol{y}|\boldsymbol{X}] = \boldsymbol{K}_{X,X} + \sigma_y^2 \boldsymbol{I}_N \equiv \hat{\boldsymbol{K}}_{X,X} \tag{19}$$

In scalar form, this can be written as:

Noisy observations

If we have a noisy realization of the underlying function, then:

$$y_n = f(\boldsymbol{x}_n) + \epsilon_n \tag{18}$$

where $\epsilon_n \sim \mathcal{N}(0, \sigma_y^2)$.
The covariance of the observed noisy responses $\boldsymbol{y}$ is thus given by:

$$\mathrm{Cov}[\boldsymbol{y}|\boldsymbol{X}] = \boldsymbol{K}_{X,X} + \sigma_y^2 \boldsymbol{I}_N \equiv \hat{\boldsymbol{K}}_{X,X} \tag{19}$$

In scalar form, this can be written as:

$$\mathrm{Cov}[y_i, y_j] =$$

Noisy observations

If we have a noisy realization of the underlying function, then:

$$y_n = f(\boldsymbol{x}_n) + \epsilon_n \tag{18}$$

where $\epsilon_n \sim \mathcal{N}(0, \sigma_y^2)$.
The covariance of the observed noisy responses $\boldsymbol{y}$ is thus given by:

$$\text{Cov}[\boldsymbol{y}|\boldsymbol{X}] = \boldsymbol{K}_{X,X} + \sigma_y^2 \boldsymbol{I}_N \equiv \hat{\boldsymbol{K}}_{X,X} \tag{19}$$

In scalar form, this can be written as:

$$\text{Cov}[y_i, y_j] = \text{Cov}[f_i, f_j] + \text{Cov}[\epsilon_i, \epsilon_j]$$

Noisy observations

If we have a noisy realization of the underlying function, then:

$$y_n = f(\boldsymbol{x}_n) + \epsilon_n \tag{18}$$

where $\epsilon_n \sim \mathcal{N}(0, \sigma_y^2)$.
The covariance of the observed noisy responses $\boldsymbol{y}$ is thus given by:

$$\text{Cov}[\boldsymbol{y}|\boldsymbol{X}] = \boldsymbol{K}_{X,X} + \sigma_y^2 \boldsymbol{I}_N \equiv \hat{\boldsymbol{K}}_{X,X} \tag{19}$$

In scalar form, this can be written as:

$$\text{Cov}[y_i, y_j] = \text{Cov}[f_i, f_j] + \text{Cov}[\epsilon_i, \epsilon_j] = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sigma_y^2 \delta_{ij} \tag{20}$$

Joint density

Joint density

The joint distribution of the observed data $\boldsymbol{y}$ and the noise-free function on test points $\boldsymbol{f}_*$ is:

Joint density

The joint distribution of the observed data $\boldsymbol{y}$ and the noise-free function on test points $\boldsymbol{f}_*$ is:

$$\begin{pmatrix} \boldsymbol{f}_X \\ \boldsymbol{f}_* \end{pmatrix}$$

Joint density

The joint distribution of the observed data $\boldsymbol{y}$ and the noise-free function on test points $\boldsymbol{f}_*$ is:

$$\begin{pmatrix} \boldsymbol{f}_X \\ \boldsymbol{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{pmatrix} \hat{\boldsymbol{K}}_{X,X} & \boldsymbol{K}_{X,*} \\ \boldsymbol{K}_{X,*}^\top & \boldsymbol{K}_{*,*} \end{pmatrix}\right) \tag{21}$$

Joint density

The joint distribution of the observed data $\boldsymbol{y}$ and the noise-free function on test points $\boldsymbol{f}_*$ is:

$$\begin{pmatrix} \boldsymbol{f}_X \\ \boldsymbol{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{pmatrix} \hat{\boldsymbol{K}}_{X,X} & \boldsymbol{K}_{X,*} \\ \boldsymbol{K}_{X,*}^\top & \boldsymbol{K}_{*,*} \end{pmatrix} \right) \tag{21}$$

Note that this has the same form as the joint distribution in the noise-free case, except: $\boldsymbol{K}_{X,X}$ is replaced by $\hat{\boldsymbol{K}}_{X,X}$, which is the covariance of the training inputs with a constant variance σ_y^2 added to all the diagonal terms.

Joint density

The joint distribution of the observed data $\boldsymbol{y}$ and the noise-free function on test points $\boldsymbol{f}_*$ is:

$$\begin{pmatrix} \boldsymbol{f}_X \\ \boldsymbol{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_* \end{pmatrix}, \begin{pmatrix} \hat{\boldsymbol{K}}_{X,X} & \boldsymbol{K}_{X,*} \\ \boldsymbol{K}_{X,*}^\top & \boldsymbol{K}_{*,*} \end{pmatrix} \right) \tag{21}$$

Note that this has the same form as the joint distribution in the noise-free case, except: $\boldsymbol{K}_{X,X}$ is replaced by $\hat{\boldsymbol{K}}_{X,X}$, which is the covariance of the training inputs with a constant variance σ_y^2 added to all the diagonal terms.

GPR example with noisy inputs. Source: https://www.aidanscannell.com/post/gaussian-process-regression/

Posterior predictive density

Posterior predictive density

The conditional posterior distribution (posterior predictive) is given by:

$$p(\boldsymbol{f}_*|\boldsymbol{X}_*, \mathcal{D})$$

Posterior predictive density

The conditional posterior distribution (posterior predictive) is given by:

$$p(\boldsymbol{f}_*|\boldsymbol{X}_*, \mathcal{D}) \quad = \quad \mathcal{N}(\boldsymbol{f}_*|\boldsymbol{u}_{*|X}, \boldsymbol{\Sigma}_{*|X})$$

Posterior predictive density

The conditional posterior distribution (posterior predictive) is given by:

$$p(\boldsymbol{f}_*|\boldsymbol{X}_*, \mathcal{D}) \quad = \quad \mathcal{N}(\boldsymbol{f}_*|\boldsymbol{u}_{*|X}, \boldsymbol{\Sigma}_{*|X}) \tag{22}$$

Posterior predictive density

The conditional posterior distribution (posterior predictive) is given by:

$$p(\boldsymbol{f}_*|\boldsymbol{X}_*, \mathcal{D}) \quad = \quad \mathcal{N}(\boldsymbol{f}_*|\boldsymbol{u}_{*|X}, \boldsymbol{\Sigma}_{*|X}) \tag{22}$$

$$\boldsymbol{\mu}_{*|X}$$

Posterior predictive density

The conditional posterior distribution (posterior predictive) is given by:

$$p(\boldsymbol{f}_*|\boldsymbol{X}_*, \mathcal{D}) = \mathcal{N}(\boldsymbol{f}_*|\boldsymbol{u}_{*|X}, \boldsymbol{\Sigma}_{*|X}) \tag{22}$$

$$\boldsymbol{\mu}_{*|X} = m(\boldsymbol{X}_*) + \boldsymbol{K}_{X,*}^{\top}\hat{\boldsymbol{K}}_{X,X}^{-1}(\boldsymbol{y} - m(\boldsymbol{X})) \tag{23}$$

Posterior predictive density

The conditional posterior distribution (posterior predictive) is given by:

$$
\begin{aligned}
p(\boldsymbol{f}_*|\boldsymbol{X}_*, \mathcal{D}) &= \mathcal{N}(\boldsymbol{f}_*|\boldsymbol{u}_{*|X}, \boldsymbol{\Sigma}_{*|X}) \qquad (22) \\
\boldsymbol{\mu}_{*|X} &= m(\boldsymbol{X}_*) + \boldsymbol{K}_{X,*}^{\top} \hat{\boldsymbol{K}}_{X,X}^{-1} (\boldsymbol{y} - m(\boldsymbol{X})) \qquad (23) \\
\boldsymbol{\Sigma}_{*|X} &
\end{aligned}
$$

Posterior predictive density

The conditional posterior distribution (posterior predictive) is given by:

$$
\begin{aligned}
p(\boldsymbol{f}_*|\boldsymbol{X}_*, \mathcal{D}) &= \mathcal{N}(\boldsymbol{f}_*|\boldsymbol{u}_{*|X}, \boldsymbol{\Sigma}_{*|X}) && (22)\\
\boldsymbol{\mu}_{*|X} &= m(\boldsymbol{X}_*) + \boldsymbol{K}_{X,*}^\top \hat{\boldsymbol{K}}_{X,X}^{-1}(\boldsymbol{y} - m(\boldsymbol{X})) && (23)\\
\boldsymbol{\Sigma}_{*|X} &= \boldsymbol{K}_{*,*} - \boldsymbol{K}_{X,*}^\top \hat{\boldsymbol{K}}_{X,X}^{-1}\boldsymbol{K}_{X,*} && (24)
\end{aligned}
$$

Optimizing kernel parameters

Optimizing kernel parameters

The value of kernel [hyper]parameters $\boldsymbol{\theta}$ affects prediction performance.

Optimizing kernel parameters

The value of kernel [hyper]parameters $\boldsymbol{\theta}$ affects prediction performance.

- In SE kernels, we want to optimize the length scale ℓ

Optimizing kernel parameters

The value of kernel [hyper]parameters $\boldsymbol{\theta}$ affects prediction performance.

- In SE kernels, we want to optimize the length scale ℓ
- For ARD kernels, we want to learn/optimize the characteristic length scale ℓ_d and the variance σ^2

Optimizing kernel parameters

The value of kernel [hyper]parameters $\boldsymbol{\theta}$ affects prediction performance.

- In SE kernels, we want to optimize the length scale ℓ
- For ARD kernels, we want to learn/optimize the characteristic length scale ℓ_d and the variance σ^2
- Hyperparameters can be learned via **maximum marginal likelihood**

Maximum marginal likelihood

Maximum marginal likelihood

Assuming the mean function is zero, the prior is given by:

Maximum marginal likelihood

Assuming the mean function is zero, the prior is given by:

$$p(\boldsymbol{f}|\boldsymbol{X},\theta) = \mathcal{N}(\boldsymbol{f}|\mathbf{0},\boldsymbol{K}) \tag{25}$$

Maximum marginal likelihood

Assuming the mean function is zero, the prior is given by:

$$p(\boldsymbol{f}|\boldsymbol{X}, \theta) = \mathcal{N}(\boldsymbol{f}|\mathbf{0}, \boldsymbol{K}) \tag{25}$$

and the likelihood of each observation (conditioned on the latent function $\boldsymbol{f}$) can be written as:

Maximum marginal likelihood

Assuming the mean function is zero, the prior is given by:

$$p(\boldsymbol{f}|\boldsymbol{X}, \theta) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{0}, \boldsymbol{K}) \tag{25}$$

and the likelihood of each observation (conditioned on the latent function $\boldsymbol{f}$) can be written as:

$$p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{X}) = \prod_{n=1}^{N} \mathcal{N}(y_n|f_n, \sigma_y^2) \tag{26}$$

Maximum marginal likelihood

Assuming the mean function is zero, the prior is given by:

$$p(\boldsymbol{f}|\boldsymbol{X}, \theta) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{0}, \boldsymbol{K}) \tag{25}$$

and the likelihood of each observation (conditioned on the latent function $\boldsymbol{f}$) can be written as:

$$p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{X}) = \prod_{n=1}^{N} \mathcal{N}(y_n|f_n, \sigma_y^2) \tag{26}$$

The marginal likelihood is thus given by:

Maximum marginal likelihood

Assuming the mean function is zero, the prior is given by:

$$p(\boldsymbol{f}|\boldsymbol{X}, \theta) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{0}, \boldsymbol{K}) \tag{25}$$

and the likelihood of each observation (conditioned on the latent function $\boldsymbol{f}$) can be written as:

$$p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{X}) = \prod_{n=1}^{N} \mathcal{N}(y_n|f_n, \sigma_y^2) \tag{26}$$

The marginal likelihood is thus given by:

$$p(\boldsymbol{y}|\boldsymbol{X}, \theta) = \int p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{X}) p(\boldsymbol{f}|\boldsymbol{X}, \theta) d\boldsymbol{f} \tag{27}$$

Maximum marginal likelihood

Assuming the mean function is zero, the prior is given by:

$$p(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{0}, \boldsymbol{K}) \tag{25}$$

and the likelihood of each observation (conditioned on the latent function $\boldsymbol{f}$) can be written as:

$$p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{X}) = \prod_{n=1}^{N} \mathcal{N}(y_n|f_n, \sigma_y^2) \tag{26}$$

The marginal likelihood is thus given by:

$$p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = \int p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{X})p(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{\theta})d\boldsymbol{f} \tag{27}$$

To find the optimal kernel parameters $\boldsymbol{\theta}$, we maximize the **log marginal likelihood**:

Maximum marginal likelihood

Assuming the mean function is zero, the prior is given by:

$$p(\boldsymbol{f}|\boldsymbol{X}, \theta) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{0}, \boldsymbol{K}) \tag{25}$$

and the likelihood of each observation (conditioned on the latent function $\boldsymbol{f}$) can be written as:

$$p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{X}) = \prod_{n=1}^{N} \mathcal{N}(y_n|f_n, \sigma_y^2) \tag{26}$$

The marginal likelihood is thus given by:

$$p(\boldsymbol{y}|\boldsymbol{X}, \theta) = \int p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{X}) p(\boldsymbol{f}|\boldsymbol{X}, \theta) d\boldsymbol{f} \tag{27}$$

To find the optimal kernel parameters $\boldsymbol{\theta}$, we maximize the **log marginal likelihood**:

$\log p(\boldsymbol{y}|\boldsymbol{X}, \theta) =$

Maximum marginal likelihood

Assuming the mean function is zero, the prior is given by:

$$p(\boldsymbol{f}|\boldsymbol{X}, \theta) = \mathcal{N}(\boldsymbol{f}|\mathbf{0}, \boldsymbol{K}) \tag{25}$$

and the likelihood of each observation (conditioned on the latent function $\boldsymbol{f}$) can be written as:

$$p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{X}) = \prod_{n=1}^{N} \mathcal{N}(y_n|f_n, \sigma_y^2) \tag{26}$$

The marginal likelihood is thus given by:

$$p(\boldsymbol{y}|\boldsymbol{X}, \theta) = \int p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{X}) p(\boldsymbol{f}|\boldsymbol{X}, \theta) d\boldsymbol{f} \tag{27}$$

To find the optimal kernel parameters $\boldsymbol{\theta}$, we maximize the **log marginal likelihood**:

$\log p(\boldsymbol{y}|\boldsymbol{X}, \theta) = \log \mathcal{N}(\boldsymbol{y}|\mathbf{0}, \hat{\boldsymbol{K}}_{X,X})$

Maximum marginal likelihood

Assuming the mean function is zero, the prior is given by:

$$p(\boldsymbol{f}|\boldsymbol{X}, \theta) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{0}, \boldsymbol{K}) \tag{25}$$

and the likelihood of each observation (conditioned on the latent function $\boldsymbol{f}$) can be written as:

$$p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{X}) = \prod_{n=1}^{N} \mathcal{N}(y_n|f_n, \sigma_y^2) \tag{26}$$

The marginal likelihood is thus given by:

$$p(\boldsymbol{y}|\boldsymbol{X}, \theta) = \int p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{X}) p(\boldsymbol{f}|\boldsymbol{X}, \theta) d\boldsymbol{f} \tag{27}$$

To find the optimal kernel parameters $\boldsymbol{\theta}$, we maximize the **log marginal likelihood**:

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \theta) = \log \mathcal{N}(\boldsymbol{y}|\boldsymbol{0}, \hat{\boldsymbol{K}}_{X,X}) = -fr12\boldsymbol{y}^\top \hat{\boldsymbol{K}}_{X,X}^{-1} \boldsymbol{y} - \frac{1}{2} \log |\hat{\boldsymbol{K}}_{X,X}| - \frac{N}{2} \log(2\pi) \tag{28}$$

Reading

- **PMLI 17.1-2**

Reading

- **PMLI 17.1-2**
- https://colab.research.google.com/github/krasserm/bayesian-machine-learning/blob/dev/gaussian-processes/gaussian_processes.ipynb

Reading

- **PMLI 17.1-2**
- https://colab.research.google.com/github/krasserm/bayesian-machine-learning/blob/dev/gaussian-processes/gaussian_processes.ipynb
- https://peterroelants.github.io/posts/gaussian-process-kernels/

Reading

- **PMLI 17.1-2**
- https://colab.research.google.com/github/krasserm/bayesian-machine-learning/blob/dev/gaussian-processes/gaussian_processes.ipynb
- https://peterroelants.github.io/posts/gaussian-process-kernels/
- https://github.com/aidanscannell/probabilistic-modelling/blob/master/notebooks/gaussian-process-regression.ipynb