Gaussian Processes

Marc Deisenroth
Centre for Artificial Intelligence
Department of Computer Science
University College London

AIMS Rwanda and AIMS Ghana
March/April 2020
http://www.gaussianprocess.org/
Problem Setting

Objective

For a set of observations $y_i = f(x_i) + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma^2_\varepsilon)$, find a distribution over functions $p(f)$ that explains the data

Probabilistic regression problem
Problem Setting

Objective

For a set of observations $y_i = f(x_i) + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$, find a distribution over functions $p(f)$ that explains the data.

**Probabilistic regression problem**
Some Application Areas

- Reinforcement learning and robotics
- Bayesian optimization (experimental design)
- Geostatistics
- Sensor networks
- Time-series modeling and forecasting
- High-energy physics
- Medical applications
Bayesian Linear Regression: Model

Prior \[ p(\theta) = \mathcal{N}(m_0, S_0) \]

Likelihood \[ p(y|x, \theta) = \mathcal{N}(y | \phi^T(x)\theta, \sigma^2) \]

\[ \implies y = \phi^T(x)\theta + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \]

- Parameter \( \theta \) becomes a latent (random) variable
- Distribution \( p(\theta) \) induces a distribution over plausible functions
- Choose a conjugate Gaussian prior
  - Gaussian posterior \( p(\theta|X, y) = \mathcal{N}(\theta | m_N, S_N) \)
  - Closed-form computations (e.g., predictions, marginal likelihood)
Consider a linear regression setting

\[ y = a + bx + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2_n) \]

\[ p(a, b) = \mathcal{N}(0, I) \]
Consider a linear regression setting

\[
y = f(x) + \epsilon = a + bx + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2_n)
\]

\[
p(a, b) = \mathcal{N}(0, I)
\]

\[
f_i(x) = a_i + b_i x, \quad [a_i, b_i] \sim p(a, b)
\]
Sampling from the Posterior over Functions

Consider a linear regression setting

\[ y = f(x) + \epsilon = a + bx + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2) \]

\[ p(a, b) = \mathcal{N}(0, I) \]

\[ X = [x_1, \ldots, x_N], \quad y = [y_1, \ldots, y_N] \quad \text{Training data} \]
Sampling from the Posterior over Functions

Consider a linear regression setting

\[ y = f(x) + \epsilon = a + bx + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2) \]

\[ p(a, b) = \mathcal{N}(0, I) \]

\[ p(a, b | X, y) = \mathcal{N}(m_N, S_N) \quad \text{Posterior} \]
Sampling from the Posterior over Functions

Consider a linear regression setting

\[ y = f(x) + \epsilon = a + bx + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2) \]

\[ [a_i, b_i] \sim p(a, b | X, y) \]

\[ f_i = a_i + b_i x \]

Marc Deisenroth (UCL)

Gaussian Processes

March/April 2020
Instead of sampling parameters, which induce a distribution over functions, **sample functions directly**

- Place a prior on functions
- Make assumptions on the distribution of functions
Instead of sampling parameters, which induce a distribution over functions, **sample functions directly**
- Place a prior on functions
- Make assumptions on the distribution of functions

Intuition: function = infinitely long vector of function values
- Make assumptions on the distribution of function values
Instead of sampling parameters, which induce a distribution over functions, sample functions directly.

- Place a prior on functions
- Make assumptions on the distribution of functions

Intuition: function = infinitely long vector of function values

- Make assumptions on the distribution of function values

Gaussian process
1 Gaussian Process: Definition
2 Regression as Inference
   - GP Prior
   - Likelihood
   - Marginal Likelihood
   - Posterior
   - Predictions
3 Model Selection
   - GP Training
   - Training
4 Limitations and Guidelines
5 Application Areas

BLR:
\[ p(y_* | x_*) = \int p(y_* | x_*, \Theta) p(\Theta) d\Theta \]

consider all plausible (\infty)
values/settings of \( \Theta \)

GP:
\[ p(y_* | x_*) = \int p(y_* | x_*, \Theta) p(\Theta) d\Theta \]

consider all plausible (\infty)
values/settings of \( \Theta \)

\[ f(\cdot) = \sin(\cdot) \]
\[ f: \mathbb{R}^d \rightarrow \mathbb{R} \]
Gaussian Process: Definition
- We will place a distribution \( p(f) \) on functions \( f \).
- Informally, a function can be considered an infinitely long vector of function values \( f = [f_1, f_2, f_3, ...] \).
Gaussian Process

- We will place a distribution $p(f)$ on functions $f$
- Informally, a function can be considered an infinitely long vector of function values $f = [f_1, f_2, f_3, ...]$
- A Gaussian process is a generalization of a multivariate Gaussian distribution to infinitely many variables.
We will place a distribution $p(f)$ on functions $f$

Informally, a function can be considered an infinitely long vector of function values $f = [f_1, f_2, f_3, ...]$

A Gaussian process is a generalization of a multivariate Gaussian distribution to infinitely many variables.

**Definition (Rasmussen & Williams, 2006)**

A Gaussian process (GP) is a collection of random variables $f_1, f_2, ..., $ any finite number of which is Gaussian distributed.

- Training data is finite
- Test data is finite
- Locations at which we want to evaluate $f(x)$

```python
def f(x):
    return np.sin(x)

xx = np.linspace(-10, 10, 50)
f(xx)
```
Gaussian Process

- We will place a distribution $p(f)$ on functions $f$
- Informally, a function can be considered an infinitely long vector of function values $f = [f_1, f_2, f_3, ...]$
- A Gaussian process is a generalization of a multivariate Gaussian distribution to infinitely many variables.

Definition (Rasmussen & Williams, 2006)

A Gaussian process (GP) is a collection of random variables $f_1, f_2, ..., $ any finite number of which is Gaussian distributed.

- A Gaussian distribution is specified by a mean vector $\mu$ and a covariance matrix $\Sigma$
- A Gaussian process is specified by a mean function $m(\cdot)$ and a covariance function (kernel) $k(\cdot, \cdot)$

▶ More on this later
Regression as Inference
Objective

For a set of observations $y_i = f(x_i) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2_n)$, find a (posterior) distribution over functions $p(f(\cdot) | X, y)$ that explains the data. Here: $X$ training inputs, $y$ training targets.
Objective

For a set of observations \( y_i = f(x_i) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2) \), find a (posterior) distribution over functions \( p(f(\cdot)|X, y) \) that explains the data. Here: \( X \) training inputs, \( y \) training targets.

Training data: \( X, y \). Bayes’ theorem yields

\[
p(f(\cdot)|X, y) = \frac{p(y|f(\cdot), X) \ p(f(\cdot))}{p(y|X)}
\]
Objective

For a set of observations $y_i = f(x_i) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$, find a (posterior) distribution over functions $p(f(\cdot)|X, y)$ that explains the data. Here: $X$ training inputs, $y$ training targets.

Training data: $X, y$. Bayes’ theorem yields

$$p(f(\cdot)|X, y) = \frac{p(y|f(\cdot), X) p(f(\cdot))}{p(y|X)}$$

Prior: $p(f(\cdot)) = GP(m, k)$ ➤ Specify mean $m$ function and kernel $k$. 

Marc Deisenroth (UCL)  Gaussian Processes  March/April 2020  16
Objective

For a set of observations $y_i = f(x_i) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$, find a (posterior) distribution over functions $p(f(\cdot) | X, y)$ that explains the data. Here: $X$ training inputs, $y$ training targets.

Training data: $X, y$. Bayes’ theorem yields

$$p(f(\cdot) | X, y) = \frac{p(y | f(\cdot), X) \ p(f(\cdot))}{p(y | X)}$$

Prior: $p(f(\cdot)) = GP(m, k)$ ➭ Specify mean $m$ function and kernel $k$.

Likelihood (noise model): $p(y | f(\cdot), X) = \mathcal{N}(f(X), \sigma_n^2 I)$
Objective

For a set of observations $y_i = f(x_i) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$, find a (posterior) distribution over functions $p(f(\cdot) | X, y)$ that explains the data. Here: $X$ training inputs, $y$ training targets.

Training data: $X, y$. Bayes’ theorem yields

$$p(f(\cdot) | X, y) = \frac{p(y | f(\cdot), X) p(f(\cdot))}{p(y | X)}$$

Prior: $p(f(\cdot)) = GP(m, k)$ $\quad$ Specify mean $m$ function and kernel $k$.

Likelihood (noise model): $p(y | f(\cdot), X) = \mathcal{N}(f(X), \sigma_n^2 I)$

Marginal likelihood (evidence): $p(y | X) = \int p(y | f(\cdot), X)p(f(\cdot) | X)df$
Objective

For a set of observations \( y_i = f(x_i) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2) \), find a (posterior) distribution over functions \( p(f(\cdot)|X, y) \) that explains the data. Here: \( X \) training inputs, \( y \) training targets

Training data: \( X, y \). Bayes’ theorem yields

\[
p(f(\cdot)|X, y) = \frac{p(y|f(\cdot), X) \cdot p(f(\cdot))}{p(y|X)}
\]

Prior: \( p(f(\cdot)) = GP(m, k) \) ➤ Specify mean \( m \) function and kernel \( k \).

Likelihood (noise model): \( p(y|f(\cdot), X) = \mathcal{N}(f(X), \sigma_n^2 I) \)

Marginal likelihood (evidence): \( p(y|X) = \int p(y|f(\cdot), X) p(f(\cdot)|X) df \)

Posterior: \( p(f(\cdot)|y, X) = GP(m_{post}, k_{post}) \)
GP Prior

\[
p(f(\cdot) | X, y) = \frac{p(y | f(\cdot), X) \, p(f(\cdot))}{p(y | X)}
\]

Bayesian linear regression:

- Prior \( p(\theta) \) on the parameters \( \theta \) allows us to encode some properties of the parameters (e.g., range, reasonable values, ...)
- Every sample \( \theta_i \sim p(\theta) \) induces a function \( f_i(\cdot) := \theta_i^T \phi(\cdot) \)
GP Prior

$$p(f(\cdot)|X, y) = \frac{p(y|f(\cdot), X) p(f(\cdot))}{p(y|X)}$$

Bayesian linear regression:

- Prior $p(\theta)$ on the parameters $\theta$ allows us to encode some properties of the parameters (e.g., range, reasonable values, ...)
- Every sample $\theta_i \sim p(\theta)$ induces a function $f_i(\cdot) := \theta_i^T \phi(\cdot)$

Gaussian process:

- GP prior: $p(f(\cdot))$
- Function plays the role of the parameters
  - Every sample $f_i(\cdot) \sim GP$ is a function
GP Prior (2)

- Bayesian prior specifies assumptions on the quantity of interest
- What assumptions could we make on the underlying function?
- What characterizes the function we want to model?

- continuity
- differentiability
- function is positive
- function varies slowly
  (no rapid change in curve)
- strictly monotonic (invertible)
- symmetry
- bounded function values
- differentiability
- symmetry
- periodicity
- bounded function values

\[
\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \sim \mathcal{N}(\mu, \Sigma) \\
\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = W \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \\
f_1 \sim \text{GP}: \mathbb{R}^D \rightarrow \mathbb{R} \\
f_2 \sim \text{GP}: \mathbb{R}^D \rightarrow \mathbb{R} \\
\text{multi-output Gaussian process}
\]

\[
\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \sim \mathcal{N}(\mu, \Sigma) \\
\text{cov}(f_1, f_2) \\
\text{cov}(f_3, f_4) \\
\text{cov}(f_1, f_3) \\
\text{cov}(f_2, f_4) \\
\text{cov}(f_1, f_4) \\
S = \begin{bmatrix} \text{var}[f_1] & \text{cov}[f_1, f_2] & \cdots \\ \text{cov}[f_1, f_3] & \text{var}[f_2] & \cdots \\ \vdots & \vdots & \ddots \\ \text{cov}[f_1, f_4] & \text{cov}[f_2, f_4] & \cdots & \text{var}[f_4] \end{bmatrix}
\]
Bayesian prior specifies assumptions on the quantity of interest.

What assumptions could we make on the underlying function?

What characterizes the function we want to model?

- Mean function
- Covariance function
The average function of the distribution over functions

- Allows us to bias the model (can make sense in application-specific settings)
Can be a parametrized function, e.g., linear, exponential, or neural network. Example: $m_\theta(x) = \theta^T \phi(x)$
Can be a parametrized function, e.g., linear, exponential, or neural network. Example: \( m_\theta(x) = \theta^T \phi(x) \)

Prior mean function \( m_\theta \) can incorporate problem-specific prior knowledge (e.g., in robotics, natural sciences)

Can simplify the learning problem
Can be a parametrized function, e.g., linear, exponential, or neural network. Example: $m_\theta(x) = \theta^T \phi(x)$

- Prior mean function $m_\theta$ can incorporate **problem-specific prior knowledge** (e.g., in robotics, natural sciences)
- Can simplify the learning problem
- Often: “Agnostic” mean function in the absence of data or prior knowledge: $m(\cdot) \equiv 0$ everywhere (for symmetry reasons)
Covariance function (kernel) is symmetric and positive semi-definite.
Covariance function (kernel) is symmetric and positive semi-definite

Compute covariances/correlations between (unknown) function values by just looking at the corresponding inputs:

$$\text{Cov}[f(x_i), f(x_j)] = k(x_i, x_j)$$

- Kernel trick (Schölkopf & Smola, 2002)
Covariance Function

- Covariance function (kernel) is symmetric and positive semi-definite
- Compute covariances/correlations between (unknown) function values by just looking at the corresponding inputs:

\[ \text{Cov}[f(x_i), f(x_j)] = k(x_i, x_j) \]

Kernel trick (Schölkopf & Smola, 2002)
- Encodes high-level structural assumptions (e.g., smoothness, periodicity) of the function we want to model
Gaussian Covariance Function

\[ k_{Gauss}(x_i, x_j) = \sigma_f^2 \exp \left( - (x_i - x_j) \times (x_i - x_j)/\ell^2 \right) \]

- Assumption on latent function: **Smooth (\( \infty \) differentiable)**
Gaussian Covariance Function

\[ k_{\text{Gauss}}(x_i, x_j) = \sigma_f^2 \exp \left( -\frac{(x_i - x_j)^\top (x_i - x_j)}{\ell^2} \right) \]

- Assumption on latent function: **Smooth (\(\infty\) differentiable)**
- \(\sigma_f\): Amplitude of the latent function

\(\ell\approx\text{standard deviation in a Gaussian distribution}\)

\[ k(x_i, x_j) = \text{cov}(f(x_i), f(x_j)) \approx 0 \text{ if } x_i \text{ and } x_j \text{ are far from each other} \]

\(L\) depends on \(\ell\)
Gaussian Covariance Function

\[ k_{Gauss}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp \left( - (\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_j)/\ell^2 \right) \]

- Assumption on latent function: **Smooth** (\( \infty \) differentiable)
- \( \sigma_f \): **Amplitude** of the latent function
- \( \ell \): **Length-scale**. How far do we have to move in input space before the function value changes significantly, i.e., when do function values become uncorrelated?

**Smoothness parameter**
Amplitude Parameter $\sigma_f^2$

\[ k_{Gauss}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp \left( - (\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_j)/\ell^2 \right) \]

- Controls the amplitude (vertical magnitude) of the function we wish to model
Amplitude Parameter $\sigma_f^2$

\[ k_{Gauss}(x_i, x_j) = \sigma_f^2 \exp \left( - (x_i - x_j)^\top (x_i - x_j)/\ell^2 \right) \]

- **Controls the amplitude** (vertical magnitude) of the function we wish to model
Amplitude Parameter $\sigma_f^2$

$$k_{Gauss}(x_i, x_j) = \sigma_f^2 \exp \left( - (x_i - x_j)^\top (x_i - x_j)/\ell^2 \right)$$

- **Controls the amplitude** (vertical magnitude) of the function we wish to model
Amplitude Parameter $\sigma_f^2$

$$k_{\text{Gauss}}(x_i, x_j) = \sigma_f^2 \exp \left( - (x_i - x_j)^\top (x_i - x_j) / \ell^2 \right)$$

- Controls the amplitude (vertical magnitude) of the function we wish to model
Length-Scale $\ell$

$$k_{\text{Gauss}}(x_i, x_j) = \sigma_f^2 \exp \left( - (x_i - x_j)^\top (x_i - x_j) / \ell^2 \right)$$

- How “wiggly” is the function?
- How much information we can transfer to other function values?
  - Correlation between function values
- How far do we have to move in input space from $x$ to $x'$ to make $f(x)$ and $f(x')$ uncorrelated?
Length-Scale $\ell$ (2)

$$k_{\text{Gauss}}(x_i, x_j) = \sigma_f^2 \exp\left( - \frac{(x_i - x_j)^\top (x_i - x_j)}{\ell^2} \right)$$

- Correlation between function values $f(x)$ and $f(x')$ depends on the (scaled) distance $\|\tau\|/\ell = \|x - x'\|/\ell$ of the corresponding inputs.
- What does a short/long length-scale $\ell$ imply?
Length-Scale $\ell$ (3)

$$k_{Gauss}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp \left( - (\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_j)/\ell^2 \right)$$

Samples from a GP prior with lengthscale 0.05

Explore interactive diagrams at https://drafts.distill.pub/gp/
\[ k_{\text{Gauss}}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp \left( -\frac{(\mathbf{x}_i - \mathbf{x}_j)^\top(\mathbf{x}_i - \mathbf{x}_j)}{\ell^2} \right) \]
\[ k_{\text{Gauss}}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp \left( - (\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_j)/\ell^2 \right) \]
Length-Scale $\ell$ (3)

$$k_{Gauss}(x_i, x_j) = \sigma_f^2 \exp \left( - (x_i - x_j)^\top (x_i - x_j) / \ell^2 \right)$$

Explore interactive diagrams at https://drafts.distill.pub/gp/
Matérn Covariance Function

\[ k_{\text{Mat},3/2}(x_i, x_j) = \sigma_f^2 \left( 1 + \sqrt{3} \frac{||x_i - x_j||}{\ell} \right) \exp \left( -\sqrt{3} \frac{||x_i - x_j||}{\ell} \right) \]

- Assumption on latent function: 1-times differentiable
- \( \sigma_f \): Amplitude of the latent function
- \( \ell \): Length-scale. How far do we have to move in input space before the function value changes significantly?
Periodic Covariance Function

\[ k_{\text{per}}(x_i, x_j) = \sigma_f^2 \exp \left( - \frac{2 \sin^2 \left( \frac{\kappa(x_i - x_j)}{2\pi} \right)}{\ell^2} \right) \]

\[ = k_{\text{Gauss}}(u(x_i), u(x_j)), \quad u(x) = \begin{bmatrix} \cos(\kappa x) \\ \sin(\kappa x) \end{bmatrix} \]

- Assumption on latent function: periodic
- Periodicity parameter \( \kappa \)
Creating New Covariance Functions

Assume $k_1$ and $k_2$ are valid covariance functions and $u(\cdot)$ is a (nonlinear) transformation of the input space. Then

- $k_1 + k_2$ is a valid covariance function
Creating New Covariance Functions

Assume $k_1$ and $k_2$ are valid covariance functions and $u(\cdot)$ is a (nonlinear) transformation of the input space. Then

- $k_1 + k_2$ is a valid covariance function
- $k_1 k_2$ is a valid covariance function
Assume $k_1$ and $k_2$ are valid covariance functions and $u(\cdot)$ is a (nonlinear) transformation of the input space. Then

- $k_1 + k_2$ is a valid covariance function
- $k_1 k_2$ is a valid covariance function
- $k(u(x), u(x'))$ is a valid covariance function (MacKay, 1998)
  - Periodic covariance function
  - Manifold Gaussian process (Calandra et al., 2016)
  - Deep kernel learning (Wilson et al., 2016)
Assume $k_1$ and $k_2$ are valid covariance functions and $u(\cdot)$ is a (nonlinear) transformation of the input space. Then

- $k_1 + k_2$ is a valid covariance function
- $k_1 k_2$ is a valid covariance function
- $k(u(x), u(x'))$ is a valid covariance function (MacKay, 1998)
  - Periodic covariance function
  - Manifold Gaussian process (Calandra et al., 2016)
  - Deep kernel learning (Wilson et al., 2016)
- Automatic Statistician (Lloyd et al., 2014)
\[
p(f(\cdot)|X, y) = \frac{p(y|f(\cdot), X) p(f(\cdot))}{p(y|X)}
\]
(Gaussian) Likelihood

\[ p(f(\cdot)|X, y) = \frac{p(y|f(\cdot), X) \ p(f(\cdot))}{p(y|X) } \]

Gaussian likelihood in linear regression:

\[ p(y|x, \theta) = \mathcal{N}(y | \theta^\top x, \sigma^2) \]

- **Function** (not a distribution) of the parameters
- Describes how parameters and observed data are connected
- Tells us how to transform parameters into (noisy) data
(Gaussian) Likelihood

\[
p(f(\cdot) | X, y) = \frac{p(y | f(\cdot), X) \cdot p(f(\cdot))}{p(y | X)}
\]

Gaussian likelihood in linear regression:

\[
p(y | x, \theta) = \mathcal{N}(y | \theta^\top x, \sigma^2)
\]

- **Function** (not a distribution) of the parameters
- Describes how parameters and observed data are connected
- Tells us how to transform parameters into (noisy) data

Gaussian likelihood in Gaussian processes:

\[
p(y | x, f(\cdot)) = \mathcal{N}(y | f(x), \sigma^2)
\]

- Parameters are the function \( f \) itself
Marginal Likelihood

\[ p(f(\cdot) | X, y) = \frac{p(y|f(\cdot), X) p(f(\cdot))}{p(y|X)} \]

Bayesian linear regression with a Gaussian prior \( p(\theta) = \mathcal{N}(0, I) \):

\[ p(y|X) = \int p(y|X, \theta) p(\theta) d\theta \]

- Normalizes the posterior distribution
Marginal Likelihood

\[
p(f(\cdot)|X, y) = \frac{p(y|f(\cdot), X) \cdot p(f(\cdot))}{p(y|X)}
\]

Bayesian linear regression with a Gaussian prior \(p(\theta) = \mathcal{N}(0, I)\):

\[
p(y|X) = \int p(y|X, \theta)p(\theta)d\theta
\]

\[
= \mathcal{N}(y | 0, \Phi \Phi^T + \sigma^2 I)
\]

- Normalizes the posterior distribution
- Can be computed analytically
Marginal Likelihood

\[
p(f(\cdot)|X, y) = \frac{p(y|f(\cdot), X) p(f(\cdot))}{p(y|X)}
\]

Bayesian linear regression with a Gaussian prior \( p(\theta) = \mathcal{N}(0, I) \):

\[
p(y|X) = \int p(y|X, \theta)p(\theta)d\theta
= \mathcal{N}(y | 0, \Phi \Phi^T + \sigma^2 I)
= \mathbb{E}_\theta[p(y|X, \theta)]
\]

- Normalizes the posterior distribution
- Can be computed analytically
- Expected likelihood (under the parameter prior)
- Expected predictive distribution of the training targets \( y \) (under the parameter prior)
Marginal Likelihood (2)

Gaussian process marginal likelihood

\[ p(y|X) = \int p(y|X, \theta) p(f(\cdot)) df \]

- Normalizes the posterior distribution
Gaussian process marginal likelihood

\[
p(y|X) = \int p(y|X, \theta)p(f(\cdot))df
\]

\[
= \mathcal{N}(y | 0, K + \sigma^2 I)
\]

- Normalizes the posterior distribution
- Can be computed analytically
Gaussian process marginal likelihood

\[
p(y|X) = \int p(y|X, \theta)p(f(\cdot))df
\]

\[
= \mathcal{N}(y | 0, K + \sigma^2 I)
\]

\[
= \mathbb{E}_f[p(y|X, f(\cdot))]
\]

- Normalizes the posterior distribution
- Can be computed analytically
- Expected likelihood (under the GP prior)
- Expected predictive distribution of the training targets \( y \) (under the GP prior)
Gaussian process marginal likelihood

\[ p(y|X) = \int p(y|X, \theta)p(f(\cdot))df \]

\[ = \mathcal{N}(y | 0, K + \sigma^2 I) \]

\[ = \mathbb{E}_f[p(y|X, f(\cdot))] \]

- Normalizes the posterior distribution
- Can be computed analytically
- Expected likelihood (under the GP prior)
- Expected predictive distribution of the training targets \( y \) (under the GP prior)

\[ \log p(y|X) = -\frac{1}{2}y^\top(K + \sigma^2 I)^{-1}y - \frac{1}{2} \log |K + \sigma^2 I| - \frac{N}{2} \log(2\pi) \]

\[ K_{ij} = k(x_i, x_j), \quad i, j = 1, \ldots, N \]
Posterior over functions (with training data $X, y$):

$$p(f(\cdot)|X, y) = \frac{p(y|f(\cdot), X)p(f(\cdot))}{p(y|X)}$$
GP Posterior

Posterior over functions (with training data $X, y$):

\[
p(f(\cdot)|X, y) = \frac{p(y|f(\cdot), X) \ p(f(\cdot))}{p(y|X)}
\]

Using the properties of Gaussians, we obtain (with $K := k(X, X)$)

\[
p(y|f(\cdot), X) \ p(f(\cdot)) = \mathcal{N}(y | f(X), \sigma_n^2 I) \ GP(m(\cdot), k(\cdot, \cdot))
\]
GP Posterior

Posterior over functions (with training data $X, y$):

$$p(f(\cdot)|X, y) = \frac{p(y|f(\cdot), X) p(f(\cdot))}{p(y|X)}$$

Using the properties of Gaussians, we obtain (with $K := k(X, X)$)

$$p(y|f(\cdot), X) p(f(\cdot)) = \mathcal{N}(y | f(X), \sigma_n^2 I) \text{ GP}(m(\cdot), k(\cdot, \cdot))$$

$$= Z \times \text{ GP}(m_{\text{post}}(\cdot), k_{\text{post}}(\cdot, \cdot))$$

$$m_{\text{post}}(\cdot) = m(\cdot) + k(\cdot, X)(K + \sigma_n^2 I)^{-1}(y - m(X))$$

$$k_{\text{post}}(\cdot, \cdot) = k(\cdot, \cdot) - k(\cdot, X)(K + \sigma_n^2 I)^{-1}k(X, \cdot)$$
Posterior over functions (with training data $X, y$):

$$p(f(\cdot)|X, y) = \frac{p(y|f(\cdot), X) \ p(f(\cdot))}{p(y|X)}$$

Using the properties of Gaussians, we obtain (with $K := k(X, X)$)

$$p(y|f(\cdot), X) \ p(f(\cdot)) = \mathcal{N}(y | f(X), \sigma^2_nI) \ GP(m(\cdot), k(\cdot, \cdot))$$

$$= Z \times GP(m_{\text{post}}(\cdot), k_{\text{post}}(\cdot, \cdot))$$

$$m_{\text{post}}(\cdot) = m(\cdot) + k(\cdot, X)(K + \sigma^2_nI)^{-1}(y - m(X))$$

$$k_{\text{post}}(\cdot, \cdot) = k(\cdot, \cdot) - k(\cdot, X)(K + \sigma^2_nI)^{-1}k(X, \cdot)$$

Marginal likelihood:

$$Z = p(y|X) = \int p(y|f(\cdot), X) \ p(f(\cdot)) \ df = \mathcal{N}(y | m(X), K + \sigma^2_nI)$$
Sampling from the GP Prior

- GP is a distribution over functions
  - A sample from a GP will be an entire function
Sampling from the GP Prior

- GP is a distribution over functions
  - A sample from a GP will be an entire function
- In practice, we cannot sample functions directly
Sampling from the GP Prior

- GP is a distribution over functions
  - A sample from a GP will be an entire function
- In practice, we cannot sample functions directly
- Instead: function = collection of function values
Sampling from the GP Prior

- GP is a distribution over functions
  - A sample from a GP will be an entire function
- In practice, we cannot sample functions directly
- Instead: function = collection of function values
- Determine function values at a finite set of input locations
  \[ X_* = [\mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_*^{(K)}] \]
Without any training data, the predictive distribution at test points \( X_\ast \) is

\[
p(f(X_\ast)|X_\ast) = \mathcal{N}(\mathbb{E}_f[f(X_\ast)], \mathbb{V}_f[f(X_\ast)])
\]

\[
= \mathcal{N}(m_{\text{prior}}(X_\ast), k_{\text{prior}}(X_\ast, X_\ast))
\]
Without any training data, the predictive distribution at test points \( X_\ast \) is

\[
p(f(X_\ast)|X_\ast) = \mathcal{N}(\mathbb{E}_f[f(X_\ast)], \mathbb{V}_f[f(X_\ast)])
\]

\[
= \mathcal{N}(m_{\text{prior}}(X_\ast), k_{\text{prior}}(X_\ast, X_\ast))
\]

Exploited: Definition of GP that all function values are jointly Gaussian distributed
Without any training data, the predictive distribution at test points $X_*$ is

$$p(f(X_*)|X_*) = \mathcal{N} (\mathbb{E}_f[f(X_*)], \mathbb{V}_f[f(X_*)])$$

$$= \mathcal{N} (m_{\text{prior}}(X_*), k_{\text{prior}}(X_*, X_*))$$

Exploited: Definition of GP that all function values are jointly Gaussian distributed

Generate “function draws” (samples from the GP prior)

$$f_k(X_*) \sim \mathcal{N} (m_{\text{prior}}(X_*), k_{\text{prior}}(X_*, X_*))$$
Goal: Generate random functions $f_k$, so that

$$f_k(X_*) \sim \mathcal{N}(m_{\text{prior}}(X_*), k_{\text{prior}}(X_*, X_*))$$
Goal: Generate random functions $f_k$, so that

$$f_k(X_*) \sim \mathcal{N}(m_{\text{prior}}(X_*), k_{\text{prior}}(X_*, X_*))$$

Define $m_* := m_{\text{prior}}(X_*)$ and $K_{**} := k_{\text{prior}}(X_*, X_*)$. Then

$$f_k(X_*) \sim \mathcal{N}(m_*, K_{**})$$

Sample from a multivariate Gaussian
GP Predictions (Posterior)

\[ y = f(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2) \]

- **Objective:** Find \( p(f(X_\star)|X, y, X_\star) \) for training data \( X, y \) and test inputs \( X_\star \).
- GP prior at training inputs: \( p(f|X) = \mathcal{N}(m(X), K) \)
- Gaussian Likelihood: \( p(y|f, X) = \mathcal{N}(f(X), \sigma_n^2 I) \)
\[ y = f(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2) \]

- **Objective:** Find \( p(f(X_*)|X, y, X_*) \) for training data \( X, y \) and test inputs \( X_* \).
- GP prior at training inputs: \( p(f|X) = \mathcal{N}(m(X), K) \)
- Gaussian Likelihood: \( p(y|f, X) = \mathcal{N}(f(X), \sigma_n^2 I) \)
- With \( f \sim \text{GP} \) it follows that \( f, f_* \) are jointly Gaussian distributed:

\[
p(f, f_*|X, X_*) = \mathcal{N} \left( \begin{bmatrix} m(X) \\ m(X_*) \end{bmatrix}, \begin{bmatrix} K & k(X_*, X) \\ k(X*, X_*) & k(X_*, X_*) \end{bmatrix} \right) \]

\[
p(x_n, f_*) = \mathcal{N}(m, \Sigma)
\]

\[
f := [f_1, \ldots, f_N] = [f(x_1), \ldots, f(x_N)] \in \mathbb{R}^N
\]

\[
f_* := [f_1, \ldots, f_K] = [f(x_1), \ldots, f(x_K)] \in \mathbb{R}^K
\]

\[
\text{cov}(f(x), f(x_*)) = \text{var}(f(x)) = K \quad \text{var}(f(x_*)) = K(X_*, X_*)
\]
\[ y = f(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2) \]

- **Objective:** Find \( p(f(X_*)|X, y, X_*) \) for training data \( X, y \) and test inputs \( X_* \).
- **GP prior at training inputs:** \( p(f|X) = \mathcal{N}(m(X), K) \)
- **Gaussian Likelihood:** \( p(y|f, X) = \mathcal{N}(f(X), \sigma_n^2 I) \)
- With \( f \sim GP \) it follows that \( f, f_* \) are jointly Gaussian distributed:

\[
p(f, f_*|X, X_*) = \mathcal{N}
\begin{bmatrix}
m(X) \\
m(X_*)
\end{bmatrix},
\begin{bmatrix}
K & k(X, X_*) \\
k(X_*, X) & k(X_*, X_*)
\end{bmatrix}
\]

- Due to the Gaussian likelihood, we also get \( (f \text{ is unobserved}) \)

\[
p(y, f_*|X, X_*) = \mathcal{N}
\begin{bmatrix}
m(X) \\
m(X_*)
\end{bmatrix},
\begin{bmatrix}
K + \sigma_n^2 I & k(X, X_*) \\
k(X_*, X) & k(X_*, X_*)
\end{bmatrix}
\]
GP Posterior Predictions

Prior evaluated at $X, X_*$:

$$p(y, f_*|X, X_*) = \mathcal{N}\left( \begin{bmatrix} m(X) \\ m(X_*) \end{bmatrix}, \begin{bmatrix} K + \sigma_n^2 I & k(X, X_*) \\ k(X_*, X) & k(X_*, X_*) \end{bmatrix} \right)$$

Posterior predictive distribution $p(f_*|X, y, X_*)$ at test inputs $X_*$
GP Posterior Predictions

Prior evaluated at $X, X_*$:

$$p(y, f_*|X, X_*) = \mathcal{N}\left(\begin{bmatrix} m(X) \\ m(X_*) \end{bmatrix}, \begin{bmatrix} K + \sigma_n^2 I & k(X, X_*) \\ k(X_*, X) & k(X_*, X_*) \end{bmatrix}\right)$$

Posterior predictive distribution $p(f_*|X, y, X_*)$ at test inputs $X_*$ obtained by Gaussian conditioning:

$$p(f_*|X, y, X_*) = \mathcal{N}\left(\begin{bmatrix} \mathbb{E}[f_*|X, y, X_*] \\ \mathbb{V}[f_*|X, y, X_*] \end{bmatrix}, \begin{bmatrix} \mathbb{E}[f_*|X, y, X_*] \\ \mathbb{V}[f_*|X, y, X_*] \end{bmatrix}\right)$$

$$\mathbb{E}[f_*|X, y, X_*] = m(X_*) + k(X_*, X)(K + \sigma_n^2 I)^{-1}(y - m(X))$$

prior mean

“Kalman gain”

error
GP Posterior Predictions

Prior evaluated at $X, X_*$:

$$p(y, f_*|X, X_*) = \mathcal{N} \left( \begin{bmatrix} m(X) \\ m(X_*) \end{bmatrix}, \begin{bmatrix} K + \sigma_n^2 I & k(X, X_*) \\ k(X_*, X) & k(X_*, X_*) \end{bmatrix} \right)$$

Posterior predictive distribution $p(f_*|X, y, X_*)$ at test inputs $X_*$ obtained by Gaussian conditioning:

$$p(f_*|X, y, X_*) = \mathcal{N} \left( \begin{bmatrix} \mathbb{E}[f_*|X, y, X_*] \\ \mathbb{V}[f_*|X, y, X_*] \end{bmatrix}, \begin{bmatrix} \mathbb{E}[f_*|X, y, X_*] \\ \mathbb{V}[f_*|X, y, X_*] \end{bmatrix} \right)$$

$$\mathbb{E}[f_*|X, y, X_*] = m(X_*) + k(X_*, X)(K + \sigma_n^2 I)^{-1}(y - m(X))$$

prior mean

"Kalman gain"

error

$$\mathbb{V}[f_*|X, y, X_*] = k(X_*, X_*) - k(X_*, X)(K + \sigma_n^2 I)^{-1}k(X, X_*), \quad \geq 0$$

prior variance

Marc Deisenroth (UCL) Gaussian Processes March/April 2020
Sanity Check

- GP posterior (from earlier):

\[
p(f(\cdot)|X, y) = GP(m_{\text{post}}(\cdot), k_{\text{post}}(\cdot, \cdot))
\]

\[
m_{\text{post}}(\cdot) = m(\cdot) + k(\cdot, X)(K + \sigma_n^2 I)^{-1}(y - m(X))
\]

\[
k_{\text{post}}(\cdot, \cdot) = k(\cdot, \cdot) - k(\cdot, X)(K + \sigma_n^2 I)^{-1}k(X, \cdot)
\]
Sanity Check

- GP posterior (from earlier):

\[
p(f(\cdot)|X, y) = GP(m_{\text{post}}(\cdot), k_{\text{post}}(\cdot, \cdot))
\]
\[
m_{\text{post}}(\cdot) = m(\cdot) + k(\cdot, X)(K + \sigma_n^2 I)^{-1}(y - m(X))
\]
\[
k_{\text{post}}(\cdot, \cdot) = k(\cdot, \cdot) - k(\cdot, X)(K + \sigma_n^2 I)^{-1}k(X, \cdot)
\]

- GP posterior predictions at \(X_*\):

\[
p(f_*|X, y, X_*) = \mathcal{N}(\mathbb{E}[f_*|X, y, X_*], \mathbb{V}[f_*|X, y, X_*])
\]
\[
\mathbb{E}[f_*|X, y, X_*] = m(X_*) + k(X_*, X)(K + \sigma_n^2 I)^{-1}(y - m(X))
\]
\[
\mathbb{V}[f_*|X, y, X_*] = k(X_*, X_*) - k(X_*, X)(K + \sigma_n^2 I)^{-1}k(X, X_*)
\]
Sanity Check

- GP posterior (from earlier):

\[
p(f(\cdot)|X, y) = GP(m_{\text{post}}(\cdot), k_{\text{post}}(\cdot, \cdot))
\]

\[
m_{\text{post}}(\cdot) = m(\cdot) + k(\cdot, X)(K + \sigma_n^2 I)^{-1}(y - m(X))
\]

\[
k_{\text{post}}(\cdot, \cdot) = k(\cdot, \cdot) - k(\cdot, X)(K + \sigma_n^2 I)^{-1}k(X, \cdot)
\]

- GP posterior predictions at \(X_*\):

\[
p(f_*|X, y, X_*) = \mathcal{N}(E[f_*|X, y, X_*], V[f_*|X, y, X_*])
\]

\[
E[f_*|X, y, X_*] = m(X_*) + k(X_*, X)(K + \sigma_n^2 I)^{-1}(y - m(X))
\]

\[
V[f_*|X, y, X_*] = k(X_*, X_*) - k(X_*, X)(K + \sigma_n^2 I)^{-1}k(X, X_*)
\]

Predictions

Make predictions by evaluating the GP posterior mean and covariance function at a finite number of inputs \(X_*\).
Predictive (marginal) mean and variance:

\[
\begin{align*}
\mathbb{E}[f(x_*)|x_*, \emptyset] &= m(x_*) = 0 \\
\text{Var}[f(x_*)|x_*, \emptyset] &= \sigma^2(x_*) = k(x_*, x_*)
\end{align*}
\]
Illustration: Inference with Gaussian Processes

Prior belief about the function

Predictive (marginal) mean and variance:

\[
\mathbb{E}[f(\mathbf{x}_*)|\mathbf{x}_*, \emptyset] = m(\mathbf{x}_*) = 0
\]

\[
\mathbb{V}[f(\mathbf{x}_*)|\mathbf{x}_*, \emptyset] = \sigma^2(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*)
\]
Predictive (marginal) mean and variance:

\[ \mathbb{E}[f(x_*)|x_*, X, y] = m(x_*) = k(x_*, X)(K + \sigma_n^2 I)^{-1}y \]

\[ \mathbb{V}[f(x_*)|x_*, X, y] = k(x_*, x_*) - k(x_*, X)(K + \sigma_n^2 I)^{-1}k(X, x_*) \]
Illustration: Inference with Gaussian Processes

Posterior belief about the function

Predictive (marginal) mean and variance:

\[
\begin{align*}
\mathbb{E}[f(x_*)| x_*, X, y] &= m(x_*) = k(x_*, X)(K + \sigma_n^2 I)^{-1}y \\
\mathbb{V}[f(x_*)| x_*, X, y] &= k(x_*, x_*) - k(x_*, X)(K + \sigma_n^2 I)^{-1}k(X, x_*)
\end{align*}
\]
Predictive (marginal) mean and variance:

\[
\begin{align*}
\mathbb{E}[f(x_*)|x_*, X, y] &= m(x_*) = k(x_*, X)(K + \sigma_n^2 I)^{-1}y \\
\mathbb{V}[f(x_*)|x_*, X, y] &= k(x_*, x_*) - k(x_*, X)(K + \sigma_n^2 I)^{-1}k(X, x_*)
\end{align*}
\]
Illustration: Inference with Gaussian Processes

Predictive (marginal) mean and variance:

\[
E[f(x_*)|x_*, X, y] = m(x_*) = k(x_*, X)(K + \sigma_n^2 I)^{-1}y
\]

\[
V[f(x_*)|x_*, X, y] = k(x_*, x_*) - k(x_*, X)(K + \sigma_n^2 I)^{-1}k(X, x_*)
\]
Predictive (marginal) mean and variance:

\[
\begin{align*}
\mathbb{E}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] &= m(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{X})(\mathbf{K} + \sigma_n^2 I)^{-1} \mathbf{y} \\
\text{Var}[f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{X}, \mathbf{y}] &= k(\mathbf{x}_*, \mathbf{x}_*) - k(\mathbf{x}_*, \mathbf{X})(\mathbf{K} + \sigma_n^2 I)^{-1} k(\mathbf{X}, \mathbf{x}_*)
\end{align*}
\]
Illustration: Inference with Gaussian Processes

Posterior belief about the function

Predictive (marginal) mean and variance:

\[
\begin{align*}
\mathbb{E}[f(x_*)|x_*, X, y] &= m(x_*) = k(x_*, X)(K + \sigma_n^2 I)^{-1}y \\
\text{Var}[f(x_*)|x_*, X, y] &= k(x_*, x_*) - k(x_*, X)(K + \sigma_n^2 I)^{-1}k(X, x_*)
\end{align*}
\]
Illustration: Inference with Gaussian Processes

Predictive (marginal) mean and variance:

\[
E[f(x_*)|x_*, X, y] = m(x_*) = k(x_*, X)(K + \sigma_n^2 I)^{-1}y \\
V[f(x_*)|x_*, X, y] = k(x_*, x_*) - k(x_*, X)(K + \sigma_n^2 I)^{-1}k(X, x_*)
\]
Illustration: Inference with Gaussian Processes

Predictive (marginal) mean and variance:

\[
\mathbb{E}[f(x_*)|x_*, X, y] = m(x_*) = k(X, x_*)^\top(K + \sigma_n^2I)^{-1}y
\]

\[
\text{Var}[f(x_*)|x_*, X, y] = \sigma^2(x_*) = k(x_*, x_*) - k(X, x_*)^\top(K + \sigma_n^2I)^{-1}k(X, x_*)
\]
Predictive (marginal) mean and variance:

\[
\mathbb{E}[f(x_*)|x_*, X, y] = m(x_*) = k(X, x_*)^\top (K + \sigma_n^2 I)^{-1} y
\]

\[
\mathbb{V}[f(x_*)|x_*, X, y] = \sigma^2(x_*) = k(x_*, x_*) - k(X, x_*)^\top (K + \sigma_n^2 I)^{-1} k(X, x_*)
\]


References II


