Motivation

Goals of this lecture

- Understand what a Gaussian Process (GP) is.
- Learn how GPs can be used for regression.

More specific to GPs, you will learn:

- What a covariance matrix means from a GP point of view.
- How a GP defines a prior over functions, and its relationship to its covariance matrix and correlation terms.
- What “conditioning on the measurements” means, in a probabilistic sense as well as mathematically.

Note: GPs for classification are outside the scope of this lecture. But if you understand regression GPs it won't be too difficult to learn how classification GPs work. Please see Rasmussen and William's “Gaussian Processes for Machine Learning” book.
Motivation: why Gaussian Processes?
Motivation

Say we want to estimate a scalar function $f(x)$ from training data $\mathcal{D} = \{x_i, f_i\}_{i=1}^N$, with $y_i = f(x_i) + \epsilon$.
Motivation

Say we want to estimate a scalar function $f(x)$ from training data $\mathcal{D} = \{x_i, f_i\}_{i=1}^{N}$, with $y_i = f(x_i) + \epsilon$.
Say we want to estimate a scalar function \( f(x) \) from training data \( \mathcal{D} = \{x_i, f_i\}_{i=1}^{N} \), with \( y_i = f(x_i) + \epsilon \).
Motivation

Say we want to estimate a scalar function $f(x)$ from training data $\mathcal{D} = \{x_i, f_i\}_{i=1}^{N}$, with $y_i = f(x_i) + \epsilon$.
Gaussian Processes let us place a prior on the 'shape' of $f(x)$.
And this prior is formulated probabilistically.
Let's get started!
Just before...

\[ X \sim \mathcal{N}(\mu, \sigma^2) \]
Let's get back to finding $f(x)$, but now

- From a single observation $\{x_1, f_1\}$
- We want to predict $f_\ast = f(x_\ast)$

Intuition?

$$X_\ast \sim X_1 \Rightarrow f_\ast \sim f_1$$

$$|X_\ast - x_1| >> 0 \Rightarrow f_\ast?$$
Gaussian Processes

Now, in a probabilistic manner...

The estimated $f_*$ is now a **Random Variable**, with a corresponding **PDF**.

Intuition?

$x_* \sim X_r$

Uncertainty $\rightarrow$ PDF

2nd depends on $x_1, p_1, x_*$
Gaussian Processes

Getting there...

The estimated \( f_* \) is now a Gaussian RV: \( f_* \sim \mathcal{N}(\mu_*, \sigma_*^2) \)

Intuition?

\( X_* \sim X_1 \Rightarrow M_* \sim F_1 \)

\( |X_* - X_1| \gg \sigma / \sqrt{n} \)

\( \mathbb{M}_*(X_1, F_1, X_*) \)
What about multiple training points?

\[ F_* \sim N(\mu_*, \sigma_*^2) \]

\[ \mu_*(x_1, f_1, x_2, f_2, x_3, f_3, x_*) \]

\[ \sigma_*^2 \]

**Comment:** \( \mu_*, \sigma_*^2 \) depend both on the data and the pred point \( x_* \).
Gaussian Processes

So far we expressed $f_*$ as a function of the training data.

But Gaussian Processes work in a slightly different way...
Gaussian Processes

We take \( f_1 \) and \( f_\ast \) to be RVs, with a joint Gaussian pdf

\[
p(f_1, f_\ast \mid x_1, x_\ast)
\]
Gaussian Processes

We take $f_1$ and $f_*$ to be RVs, with a joint Gaussian pdf

$$p(f_1, f_* | x_1, x_*)$$

More precisely,

$$\begin{bmatrix} f_1 \\ f_* \end{bmatrix} \sim \mathcal{N} \left( \mathbf{0}, K(x_1, x_*) \right)$$

$$X_1 \sim x_* \Rightarrow \nabla_{1*} \uparrow$$

$$|X_1 - x_*| >> 0 \Rightarrow \nabla_{1*} \rightarrow 0$$
Gaussian Processes

Incorporating the measurement:
Gaussian Processes

Incorporating the measurement:

Our model is \( p(f_1, f_\star | x_1, x_\star) \)

prior

(before any measurements)
Gaussian Processes

Incorporating the measurement:

But we know \( f(x_1) = f_1 \! \)

Our model is \( p(f_1, f_* | x_1, x_*) \)
Gaussian Processes

Incorporating the measurement:

Our model is $p(f_1, f_* | x_1, x_*)$

But we know $f(x_1) = f_1$

And we want to estimate
Gaussian Processes

Incorporating the measurement:

But we know \( f(x_1) = f_1 \! \)

Our model is \( p(f_1, f_* | x_1, x_*) \)

And we want to estimate

What we want is \( p(f_* | f_1, x_1, x_*) \)
Gaussian Processes

Incorporating the measurement:

But we know \( f(x_1) = f_1 \! \)

Our model is \( p(f_1, f_* | x_1, x_* \) \)

And we want to estimate

What we want is \( p(f_* | f_1, x_1, x_* \) \)
Gaussian Processes

Summarizing our 2-point Gaussian Process:

- Our model or prior is

\[
p(f_1, f_* \mid x_1, x_*) = \mathcal{N}(0, K(x_1, x_*))
\]

\[
K = [K_{mn}] = [k(x_m, x_n)]
\]

- If we have a measurement \( f(x_1) = f_1 \), we can condition on it to estimate \( f_* \):

\[
p(f_* \mid f_1, x_1, x_*) = \mathcal{N}(\mu_*, \sigma_*^2)
\]

We get a probability distribution as the output.
Exercise

Let’s use the RBF kernel

\[ k(x_n, x_m) = e^{-(x_n - x_m)^2 / 2} \]

\[ K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_*) \\ k(x_*, x_1) & k(x_*, x_*) \end{bmatrix} \]

\[ K = [K_{mn}] = [k(x_m, x_n)] \]

1) How many rows and columns \( K \) has? \( 2 \times 2 \)

2) Compute \( K \) for \( x_1 = 1, x_* = 2 \).

3) Are \( f_1 \) and \( f_* \) strongly correlated?

\[ K = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \]

(Hint: \( e^{-1/2} \approx 0.6 \))

4) What if \( x_* = 10 \)?
Time to get to the Real GP
A Gaussian process defines a prior over functions $f$,

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

Defining the kernel function $k(x_n, x_m)$ defines the prior

1. We can sample functions from this prior

2. We can use the prior + measurements to generate predictions
1) A prior over functions

Given $k(x_n, x_m)$, we can sample functions from $p(f|X) = \mathcal{N}(f|0, K(X))$.

Example 1: RBF kernel $k(x_n, x_m) = e^{-||x_n - x_m||^2/L^2}$
1) A prior over functions

Given $k(x_n, x_m)$, we can sample functions from $p(f|X) = \mathcal{N}(f|0, K(X))$

Example 1: RBF kernel $k(x_n, x_m) = e^{-||x_n - x_m||^2/L^2}$

$L = 10$
1) A prior over functions

Given $k(x_n, x_m)$, we can sample functions from $p(f|X) = \mathcal{N}(f|0, K(X))$

Example 1: RBF kernel $k(x_n, x_m) = e^{-\|x_n - x_m\|^2 / L^2}$
1) A prior over functions

Example 1: RBF kernel, What will happen if $L \to 0$?

$$k(x_n, x_m) = e^{-\|x_n - x_m\|^2 / L^2}$$
1) A prior over functions

Example 1: RBF kernel, **What will happen if \( L \to 0? \)**

What will happen to the correlation between different points?

\[
k(x_n, x_m) = e^{-||x_n - x_m||^2 / L^2}
\]

\( L \to 0 \)

\[
k = I
\]
1) A prior over functions

Given $k(x_n, x_m)$, we can sample functions from $p(f|X) = \mathcal{N}(f|0, K(X))$

**Example 2:** Quadratic kernel $k(x_n, x_m) = (1 + x_n^T x_m)^2$
2) Incorporating \textit{noise-free} measurements

Notation:

- \( f, X \): training data
- \( f_*, X_* \): prediction

\[
\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N} \left( 0, \begin{bmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix} \right)
\]
2) Incorporating *noise-free* measurements

Notation:

- $f, X$: training data
- $f_*, X_*$: prediction

\[
\begin{bmatrix}
    f \\
    f_*
\end{bmatrix} \sim \mathcal{N}(0, \begin{bmatrix}
    K(X, X) & K(X, X_*) \\
    K(X_*, X) & K(X_*, X_*)
\end{bmatrix})
\]

Conditioning on $f$ (training data) we get

\[
f_* \mid f, X_*, X \sim \mathcal{N}(\mu, \Sigma)
\]

with

\[
\mu = K(X_*, X) K(X, X)^{-1} f \\
\Sigma = K(X_*, X_*) - K(X_*, X) K(X, X)^{-1} K(X, X_*)
\]
3) Incorporating *noisy* measurements (as in real life)

Assume measurements $y$ are noisy such that $y = f(x) + \epsilon$

and $\epsilon$ is i.i.d. with $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$

Therefore, $\text{cov}(y) = K(X, X) + \sigma_n^2 I$, and we can write

$$
\begin{bmatrix}
  y \\
  f_*
\end{bmatrix} \sim \mathcal{N}
\left(0, 
\begin{bmatrix}
  K(X, X) + \sigma_n^2 I & K(X, X_*) \\
  K(X_*, X) & K(X_*, X_*)
\end{bmatrix}
\right)
$$
3) Incorporating *noisy* measurements (as in real life)

Assume measurements $y$ are noisy such that $y = f(x) + \epsilon$

and $\epsilon$ is i.i.d. with $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$

Therefore, $\text{cov}(y) = K(X, X) + \sigma_n^2 I$, and we can write

$$
\begin{bmatrix}
  y \\
  f_*
\end{bmatrix} \sim \mathcal{N}(0, \begin{bmatrix}
  K(X, X) + \sigma_n^2 I & K(X, X_*) \\
  K(X_*, X) & K(X_*, X_*)
\end{bmatrix})
$$

Conditioning on $y$ (training data) we get

$$
\begin{align*}
  f_* | y, X_*, X & \sim \mathcal{N}(\mu', \Sigma') \\
\end{align*}
$$

with

$$
\begin{align*}
  \mu' &= K(X_*, X)\left[K(X, X) + \sigma_n^2 I\right]^{-1} y \\
  \Sigma' &= K(X_*, X_*) - K(X_*, X)\left[K(X, X) + \sigma_n^2 I\right]^{-1} K(X, X_*)
\end{align*}
$$
Demo Time
A Real Example [Rasmussen, Williams, Gaussian Processes for Machine Learning]
A Real Example [Rasmussen, Williams, Gaussian Processes for Machine Learning]
Gaussian Processes

A Real Example [Rasmussen, Williams, Gaussian Processes for Machine Learning]

**Kernel Design:**

\[
k(x, x') = k_1(x, x') + k_2(x, x') + k_3(x, x') + k_4(x, x')
\]

- **Long-term smoothness**
  \[
k_1(x, x') = \theta_1^2 \exp\left(-\frac{(x - x')^2}{\theta_2^2}\right)
\]
  \[
  \text{RBF}
  \]

- **Seasonal trend (periodicity)**
  \[
k_2(x, x') = \theta_3^2 \exp\left(-\frac{-2 \sin^2(\pi(x - x'))}{\theta_5^2}\right) \exp\left(-\frac{1}{2} \frac{(x - x')^2}{\theta_4^2}\right)
\]

- **Short- and medium-term anomaly**
  \[
k_3(x, x') = \theta_6^2 \left(1 + \frac{(x - x')^2}{2\theta_8 \theta_7^2}\right)^{-\theta_8}
\]

- **Noise**
  \[
k_4(x, x') = \theta_9^2 \exp\left(-\frac{(x - x')^2}{2\theta_{10}^2}\right) + \theta_{11}^2 \delta_{x,x'}
\]

11 parameters
Gaussian Processes

What about Classification?

So far $f(x)$ is a real function, not optimal for classification

what would you suggest doing?

$F(x) \sim N(\mu(x), K(x))$

$P(F*|F--) \sim N(-)$

$P(y=1|F) = \Phi(F)$

$F \sim N(\mu, \Sigma)$

$\Phi(F)$ is not

Regression

Classification

GPML matlab toolbox
Gaussian Processes

Summary

• GPs place a prior over functions through $p(f|X) = \mathcal{N}(f|0, K(X))$
  $\rightarrow K(X)$ defines 'shape' and prior knowledge about our problem

• Prediction = Prior | Measurements
  (Tightly linked to Bayesian Estimation)

• GPs can be applied to classification
  (and many other applications, eg. Dimensionality Reduction, Latent Variable Models...