1. Khái niệm cơ bản về Gaussian Processes

1.1 Gaussian là gì?

- Bạn đã biết **Gaussian distribution** (phân phối chuẩn) rồi, đúng không?
- Nó là dạng phân phối chuông quen thuộc, đặc trưng bởi:
    - **Mean (μ)**: giá trị trung bình
    - **Variance (σ²)**: độ phân tán

Công thức phân phối chuẩn 1 chiều:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

1.2 Gaussian đa chiều (Multivariate Gaussian)

- Khi mở rộng Gaussian ra nhiều chiều (n chiều), ta có **Multivariate Gaussian distribution**.
- Thay vì chỉ mean và variance, bây giờ cần:
    - **Mean vector** $\mu \in \mathbb{R}^n$
    - **Covariance matrix** $\Sigma \in \mathbb{R}^{n \times n}$

Công thức:

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

1.3 Gaussian Process là gì?

Gaussian Process (GP) là một *khái niệm tổng quát hóa* của phân phối Gaussian đa chiều cho... **vô hạn chiều**.

Định nghĩa đơn giản:

> **Gaussian Process là một phân phối xác suất trên toàn bộ các hàm số.**
> Nghĩa là, bất kỳ tập con hữu hạn nào của các điểm đầu vào, đầu ra tương ứng của chúng sẽ có phân phối Gaussian.

Hình dung:

- Thay vì nói "vector này có phân phối Gaussian", ta nói "hàm số này có phân phối Gaussian."
- Một GP được xác định bởi:
    - **Mean function** $m(x)$: mean của output tại mỗi điểm x.
    - **Covariance function** $k(x, x')$: độ liên hệ (covariance) giữa output tại x và x'.

Cách viết GP:

$$f(x) \sim □□(m(x), k(x, x'))$$

1.4 Vì sao Gaussian Process mạnh?

- **Bayesian by nature**: Luôn có phân phối xác suất cho mọi dự đoán, thể hiện được **sự không chắc chắn (uncertainty)**.
- **Không cần chỉ rõ dạng hàm số**: Thay vì giả định mô hình là linear/quadratic/... GP tự tìm hàm tối ưu qua dữ liệu.
- **Linh hoạt**: Có thể điều chỉnh bằng kernel để phù hợp với dữ liệu phức tạp.

1.5 Ảnh minh họa trực quan

Ví dụ, nếu bạn chọn GP với mean function bằng 0 và kernel RBF:

- Trước khi nhìn dữ liệu, các hàm mẫu từ GP sẽ như thế này:

Gaussian Process prior

- Sau khi quan sát một vài điểm dữ liệu, GP sẽ update thành posterior:

Gaussian Process posterior

(Bạn thấy không, đường dự đoán đi qua các điểm dữ liệu với vùng "confidence" hẹp lại.)

🎯 Kết luận mục 1

- GP = Phân phối xác suất trên tập các hàm số.
- Xác định bằng mean function và kernel (covariance function).
- Mọi tập con hữu hạn các điểm đều có phân phối Gaussian.
- Mạnh mẽ ở chỗ nó dự đoán kèm theo độ không chắc chắn (uncertainty).

Rất tuyệt, mình sẽ dẫn bạn vào **Mục 2: Toán học nền tảng của Gaussian Processes** nhé.
Phần này mình sẽ trình bày chậm rãi, dễ hiểu, có ví dụ minh họa.

---

2. Toán học nền tảng của Gaussian Processes

2.1 GP được xác định bởi gì?

Như đã nói ở mục 1, một GP hoàn toàn xác định bởi:

- **Hàm trung bình** $m(x)$
- **Hàm hiệp phương sai** (kernel) $k(x, x')$

Công thức:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x'))$$

Trong đó:

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

👉 *Hàm mean* cho ta biết kỳ vọng tại mỗi điểm, *hàm kernel* cho ta biết mức độ liên quan giữa các điểm.

2.2 Dự đoán với Gaussian Process (Posterior Prediction)

Giả sử bạn đã có:

- Dữ liệu huấn luyện: $X = x_1, x_2, \ldots, x_n$, nhãn $\mathbf{y} = y_1, \ldots, y_n$
- Bạn muốn dự đoán output tại điểm mới x_*.

Quy trình:

1. Tính **ma trận kernel**:
   - $K(X, X)$: giữa các điểm training với nhau (size $n \times n$)
   - $K(X, x_*)$: giữa các điểm training và điểm cần dự đoán (size $n \times 1$)
   - $ K(x_, *x_*) $: giữa điểm cần dự đoán với chính nó (scalar)
2. Dự đoán mean và variance:
   - Mean:

$$\mu_* = K(X, x_*)^T K(X, X)^{-1} \mathbf{y}$$

- Variance:

$
\sigma_^2 = *K(x_*, x_) - *K(X, x_*)^T K(X, X)^{-1} K(X, x_*)
$

👉 Ý nghĩa:

- $\mu_làgiátrịdựđoántrungbìnhtại *x_* $.
- $\sigma_^2làmứcđộkhôngchắcchắntại *x_* $ (khoảng tin cậy).

2.3 Các Kernel phổ biến trong GP

Hàm kernel (hay còn gọi là *covariance function*) quyết định tính chất của hàm số mà GP mô hình hóa.

Một số kernel phổ biến:

Tên	**Công thức**	**Đặc điểm**
RBF (Radial Basis Function) / Squared Exponential	$k(x, x') = \sigma_f^2 \exp\left(-\frac{(x-x')^2}{2\ell^2}\right)$	Hàm rất mượt, trơn
Matern	Phức tạp hơn, có thêm tham số điều khiển độ trơn	Điều chỉnh độ mượt

Tên	Công thức	Đặc điểm
Linear Kernel	$k(x, x') = \sigma_b^2 + \sigma_v^2 x x'$	Mô hình hóa quan hệ tuyến tính
Periodic Kernel	Công thức liên quan đến hàm sin	Bắt tính chu kỳ

Trong thực tế: thường dùng RBF hoặc Matern kernel.

2.4 Gaussian Noise trong GP

Trong thực tế, dữ liệu có nhiễu.
Ta mô hình hóa điều này bằng cách cộng thêm nhiễu Gaussian vào kernel:

$$K(X, X) \leftarrow K(X, X) + \sigma_n^2 I$$

Trong đó:

- σ_n^2 là variance của noise
- I là ma trận đơn vị

👉 Điều này làm GP trở nên "chịu đựng" noise tốt hơn!

2.5 Minh họa bằng ví dụ cụ thể

Giả sử:

- Bạn có 3 điểm training: $(0, 1), (1, 2), (2, 0.5)$
- Bạn muốn dự đoán tại $x_* = 1.5$
- Dùng RBF kernel với $\sigma_f = 1$, $\ell = 1$

Các bước:

- Tính ma trận kernel $K(X, X)$
- Tính vector kernel $K(X, x_*)$
- Áp dụng công thức mean, variance bên trên.

(Khi cần mình có thể đi tính cụ thể luôn nhé.)

🎯 Tóm lại Mục 2:

- Gaussian Process prediction dựa trên tính toán từ ma trận kernel.
- Dự đoán trả về **mean** và **uncertainty** (variance).
- Kernel quyết định tính chất của hàm số (trơn tru, gấp khúc, tuần hoàn...).
- Noise được mô hình hóa bằng cách cộng $\sigma_n^2 I$ vào kernel.

Tuyệt vời!
Giờ mình chuyển sang **Mục 3: Kernel Design** nhé.

🧩 Mục 3: Kernel Design (Thiết kế hàm Kernel)

Ý tưởng chính:
Trong Gaussian Process (GPs), **kernel function** (còn gọi là **covariance function**) định nghĩa độ tương quan giữa hai điểm bất kỳ x và x'.
Nó cho GP khả năng mô hình hóa nhiều dạng mối quan hệ khác nhau như: tuyến tính, phi tuyến, tuần hoàn,...

🌟 Các bước cần nắm:

1. **Kernel là gì?**
   - Kernel $k(x, x')$ đo "mức độ tương đồng" giữa hai điểm x và x'.
   - Kết quả $k(x, x')$ là một số thực.
2. **Tính chất của kernel:**
   - Phải **đối xứng**: $k(x, x') = k(x', x)$
   - Phải sinh ra **ma trận hiệp phương sai** (covariance matrix) **dương bán xác định** (positive semi-definite - PSD).
3. **Một số kernel cơ bản:**
   - **Linear Kernel**:
     $k(x, x') = x^\top x'$
     ⇒ Giống như mô hình tuyến tính.
   - **RBF (Radial Basis Function) Kernel / Gaussian Kernel**:
     $k(x, x') = \exp\left(-\frac{||x-x'||^2}{2l^2}\right)$
     - l là tham số điều chỉnh mức độ "mịn" (smoothness).
   - **Polynomial Kernel**:
     $k(x, x') = (x^\top x' + c)^d$
     - c là hệ số điều chỉnh, d là bậc đa thức.
   - **Periodic Kernel**:
     $k(x, x') = \exp\left(-\frac{2\sin^2(\pi(x-x')/p)}{l^2}\right)$
     - p là chu kỳ.
4. **Kernel tự xây dựng (Custom Kernel):**
   - Bạn có thể kết hợp cộng/trừ/nhân các kernel để tạo kernel mới!

- Ví dụ:
  $k_{new}(x, x') = k_1(x, x') + k_2(x, x')$
  hoặc
  $k_{new}(x, x') = k_1(x, x') \times k_2(x, x')$

🧠 Hiểu trực giác:

- Nếu $k(x, x')$ lớn ⇒ x và x' rất giống nhau ⇒ giá trị $f(x)$ và $f(x')$ cũng gần nhau.
- Nếu $k(x, x')$ nhỏ ⇒ x và x' ít liên quan ⇒ giá trị $f(x)$ và $f(x')$ có thể khác xa.

📝 Một bài tập nhỏ (làm tay):

Giả sử dùng **RBF Kernel** với $l = 1$:

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

Tính $k(2, 3)$.

Giải:
$||2 - 3||^2 = 1^2 = 1$
$k(2, 3) = \exp\left(-\frac{1}{2}\right) = \exp(-0.5) \approx 0.6065$

Vậy: $k(2, 3) \approx 0.6065$.

Rất gọn gàng và chuyên nghiệp luôn! 🔥
Bây giờ ta đã hoàn thành:

- Giai đoạn 1: **Giới thiệu**
- Giai đoạn 2: **Toán học nền tảng**
- Giai đoạn 3: **Kernel Design**

🎯 Tiếp theo sẽ là **Mục 4: Implement GP cơ bản (Regression)**

Cụ thể trong mục này, chúng ta sẽ:

1. **Viết code** cho Gaussian Process **Regression** (bản cực kỳ cơ bản).
2. Tự tay implement các bước:
   - Tính ma trận Kernel $K(X, X)$, $ K(X, X_) , *K(X_*, X_*) $
   - Tính toán:
     - **Posterior mean**: $\mu_* = K(X_*, X)K(X, X)^{-1}y$
     - **Posterior covariance**: $ \Sigma_* = K(X_, X_) - K(X_, *X) K(X, X)^{-1} K(X, X_*) $
3. **Tạo một bài toán Regression đơn giản**, ví dụ:
   - $y = \sin(x)$ trên khoảng $[0, 5]$ với vài điểm noise.
4. **Plot** ra:

- Predictive mean
- Predictive variance (±2σ)

🧠 Mục tiêu sau bước này:

- Bạn hiểu rõ **cơ chế dự đoán** của Gaussian Process Regression.
- Bạn tự tay build được một GP nhỏ, **không dùng thư viện như sklearn hay GPyTorch**.

📋 Checklist chi tiết cho Mục 4:

STT	Công việc	Trạng thái
1	Viết hàm RBF Kernel	☐
2	Tính $K(X, X)$, $ K(X,X_) , K(X_,X_*) $	☐
3	Tính Posterior Mean và Covariance	☐
4	Viết function `predict(X_train, y_train, X_test)`	☐
5	Tạo dataset toy $y = \sin(x)$	☐
6	Plot kết quả (mean ± 2σ)	☐

Bạn có muốn mình dẫn dắt từng bước một ngay bây giờ không?
👉 Nếu đồng ý, mình sẽ bắt đầu bằng việc viết **hàm RBF Kernel** trước nhé. 🚀
(hoặc nếu bạn muốn điều chỉnh thứ tự thì cũng được nha!)