

Introduction aux Processus Gaussiens (1/2)

Application aux données temporelles

Clément Lejeune

company,
dept

4 Juillet 2024

Objectifs de cette présentation:

- Pédagogique (pas un état de l'art)

Objectifs de cette présentation:

- Pédagogique (pas un état de l'art)
- Comprendre la philosophie (Bayésienne et non-paramétrique) des processus Gaussiens (*GP*)

Objectifs de cette présentation:

- Pédagogique (pas un état de l'art)
- Comprendre la philosophie (Bayésienne et non-paramétrique) des processus Gaussiens (*GP*)
- Comprendre leurs intérêts probabilistes/prédictifs

Objectifs de cette présentation:

- Pédagogique (pas un état de l'art)
- Comprendre la philosophie (Bayésienne et non-paramétrique) des processus Gaussiens (*GP*)
- Comprendre leurs intérêts probabilistes/prédictifs
- Comprendre les limites des *GP* "naïfs"

De quoi va-t-on parler ?

- Théorie: lois Gaussiennes, noyaux, probabilités conditionnelles

Objectifs de cette présentation:

- Pédagogique (pas un état de l'art)
- Comprendre la philosophie (Bayésienne et non-paramétrique) des processus Gaussiens (*GP*)
- Comprendre leurs intérêts probabilistes/prédictifs
- Comprendre les limites des *GP* "naïfs"

De quoi va-t-on parler ?

- Théorie: lois Gaussiennes, noyaux, probabilités conditionnelles
- Applicatif: forecasting, prédiction d'incertitude, complétion de valeurs manquantes, lissage

Gaussien: vecteur vs. processus

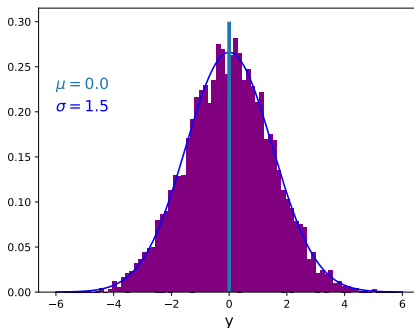
- 1 Contenu
- 2 Gaussien: vecteur vs. processus
 - Construction d'un GP
 - Le kernel
- 3 Prédire avec un GP
 - Règle de Bayes
 - prédiction = data \times hypothèse
 - Exemples
 - Application: prédiction de température maritime
- 4 Avantages / Limites des GP naïfs
 - Avantages
 - Limites
 - Remèdes

Construction d'un GP

Loi Gaussienne unidimensionnelle:

$$y \sim \mathcal{N}(m, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-m)^2}{2\sigma^2}}$$

- ① m : espérance (aka moyenne) de y
- ② $\sigma > 0$: écart-type



Construction d'un GP

Loi Gaussienne multidimensionnelle (*vecteur* Gaussien): Distribution *jointe* des composantes d'un vecteur d -dimensionnel dont les *marginales* sont Gaussiennes (unidimensionnelles).

$$\mathbf{y} := [y_1, \dots, y_d]^\top \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d \det |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu})}$$

- ① $\boldsymbol{\mu} \in \mathbb{R}^d$: *vecteur moyen* $\implies \mu_j$: moyenne de la Gaussienne y_j
- ② $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ définie positive¹: *matrice de covariance*

¹i.e. $\mathbf{a}^\top \boldsymbol{\Sigma} \mathbf{a} > 0$ (donc symétrique)

Construction d'un GP

Loi Gaussienne multidimensionnelle (*vecteur* Gaussien): Distribution *jointe* des composantes d'un vecteur d -dimensionnel dont les *marginales* sont Gaussiennes (unidimensionnelles).

$$\mathbf{y} := [y_1, \dots, y_d]^\top \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d \det |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu})}$$

① $\boldsymbol{\mu} \in \mathbb{R}^d$: *vecteur moyen* $\implies \mu_j$: moyenne de la Gaussienne y_j

② $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ définie positive¹: *matrice de covariance*

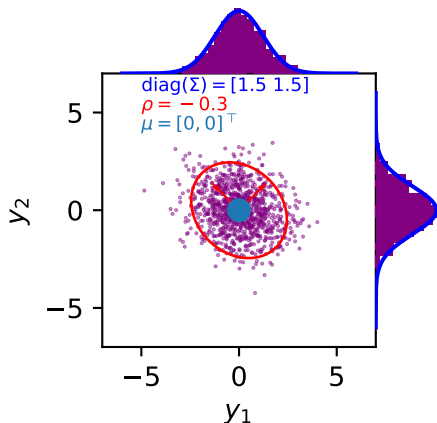
$$\boldsymbol{\Sigma} = \begin{pmatrix} \Sigma_1^2 & \Sigma_{12} & \cdots & \Sigma_{1d} \\ \Sigma_{21} & \ddots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{d1} & \cdots & \cdots & \Sigma_d^2 \end{pmatrix} \implies \begin{cases} \Sigma_j : \text{écart-type de } y_j \\ \Sigma_{ij} : \text{covariance entre les} \\ \text{marginales } y_i \text{ et } y_j \end{cases}$$

¹i.e. $\mathbf{a}^\top \boldsymbol{\Sigma} \mathbf{a} > 0$ (donc symétrique)

Construction d'un GP

Cas $d = 2$:

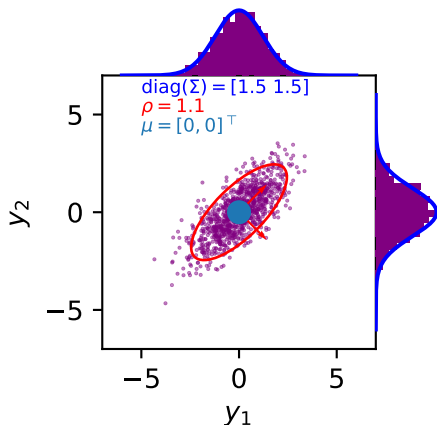
$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \Sigma_1 & \Sigma_{12} \\ \Sigma_{21} & \Sigma_2 \end{pmatrix} = \begin{pmatrix} 1.5 & \rho = -0.3 \\ \rho = -0.3 & 1.5 \end{pmatrix}$$



Construction d'un GP

Cas $d = 2$:

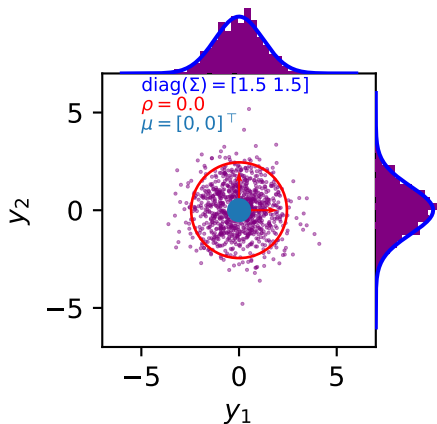
$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \Sigma_1 & \Sigma_{12} \\ \Sigma_{21} & \Sigma_2 \end{pmatrix} = \begin{pmatrix} 1.5 & \rho = 1.1 \\ \rho = 1.1 & 1.5 \end{pmatrix}$$



Construction d'un GP

Cas $d = 2$:

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \Sigma_1 & \Sigma_{12} \\ \Sigma_{21} & \Sigma_2 \end{pmatrix} = \begin{pmatrix} 1.5 & \rho = 0 \\ \rho = 0 & 1.5 \end{pmatrix}$$



Construction d'un GP

Cas $d = 2$:

$$\mu = \begin{pmatrix} 0 \\ 0 \\ \dots \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 1.5 & 0.99 & 0.98 & 0.96 & 0.94 \\ 0.99 & 1.5 & 0.99 & 0.98 & 0.96 \\ 0.98 & 0.99 & 1.5 & 0.99 & 0.98 \\ 0.96 & 0.98 & 0.99 & 1.5 & 0.99 \\ 0.94 & 0.96 & 0.98 & 0.99 & 1.5 \end{pmatrix}$$

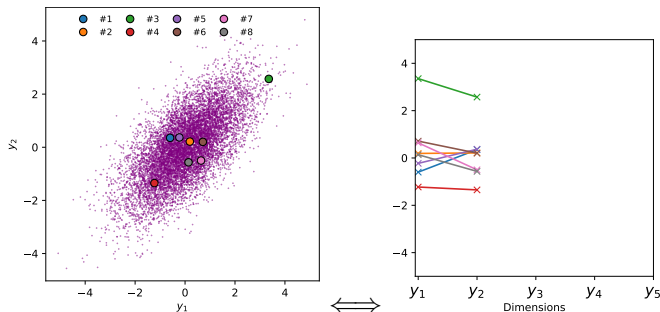


Figure: Dimensions $j = 1, 2$. Gauche: 10^4 + 8 échantillons. Droite: Réindexation des 8 *mêmes* échantillons.

Construction d'un GP

Cas $d = 5$:

$$\boldsymbol{\mu} = \begin{pmatrix} 0 \\ 0 \\ \dots \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} 1.5 & 0.99 & 0.98 & 0.96 & 0.94 \\ 0.99 & 1.5 & 0.99 & 0.98 & 0.96 \\ 0.98 & 0.99 & 1.5 & 0.99 & 0.98 \\ 0.96 & 0.98 & 0.99 & 1.5 & 0.99 \\ 0.94 & 0.96 & 0.98 & 0.99 & 1.5 \end{pmatrix}$$

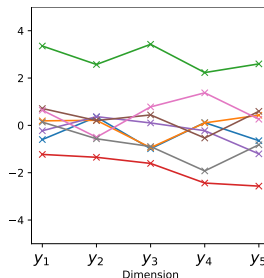


Figure: Réindéxation de 8 échantillons, dimensions $j = 1, \dots, 5$. Chaque "courbe" est un tirage de $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

Construction d'un GP

Cas $d = 50$:

$$\mu = \begin{pmatrix} 0 \\ 0 \\ \dots \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 1.5 & 0.998 & 0.993 & 0.986 & 0.975 & 0.962 & 0.946 & 0.927 & \dots \\ 0.998 & 1.5 & 0.998 & 0.993 & 0.986 & 0.975 & 0.962 & 0.946 & \dots \\ 0.993 & 0.998 & 1.5 & 0.998 & 0.993 & 0.986 & 0.975 & 0.962 & \dots \\ 0.986 & 0.993 & 0.998 & 1.5 & 0.998 & 0.993 & 0.986 & 0.975 & \dots \\ 0.975 & 0.986 & 0.993 & 0.998 & 1.5 & 0.998 & 0.993 & 0.986 & \dots \\ 0.962 & 0.975 & 0.986 & 0.993 & 0.998 & 1.5 & 0.998 & 0.993 & \dots \\ 0.946 & 0.962 & 0.975 & 0.986 & 0.993 & 0.998 & 1.5 & 0.998 & \dots \\ 0.927 & 0.946 & 0.962 & 0.975 & 0.986 & 0.993 & 0.998 & 1.5 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 1.5 \end{pmatrix}$$

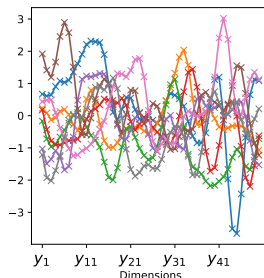


Figure: Réindéxation de 8 échantillons, dimensions $j = 1, \dots, 50$

Construction d'un GP

Cas $d = \infty$: chaque courbe est une *collection infinie* de valeurs, que l'on peut voir comme une fonction:

- ① $\boldsymbol{\mu}$: vecteur de taille infinie \Leftrightarrow fonction moyenne $\mathbb{E}(y(\mathbf{x})) = m(\mathbf{x})$
- ② $\boldsymbol{\Sigma}$: matrice de taille infinie \Leftrightarrow noyau de covariance ou "*kernel*"
 $\mathbb{C}(y(\mathbf{x}), y(\mathbf{x}')) = k(\mathbf{x}, \mathbf{x}')$

Construction d'un GP

Cas $d = \infty$: chaque courbe est une *collection infinie* de valeurs, que l'on peut voir comme une fonction:

- ① μ : vecteur de taille infinie \Leftrightarrow fonction moyenne $\mathbb{E}(y(\mathbf{x})) = m(\mathbf{x})$
- ② Σ : matrice de taille infinie \Leftrightarrow noyau de covariance ou "*kernel*"
 $\mathbb{C}(y(\mathbf{x}), y(\mathbf{x}')) = k(\mathbf{x}, \mathbf{x}')$

Processus Gaussien $y(\mathbf{x}) \sim GP(m, k)$ Rasmussen **and** Williams 2006

Un processus Gaussien (GP) est une distribution de probabilités sur un espace de *fonctions*, $y(\mathbf{x}) : \mathcal{X} \rightarrow \mathcal{Y}$, telle que toute collection finie $[y(x_i), \dots, y(x_j), \dots, y(x_k)], \forall i, j, k$ forme un vecteur Gaussien.

Construction d'un GP

- GP peut être vu comme une réindéxation d'un vecteur "infini"

Construction d'un GP

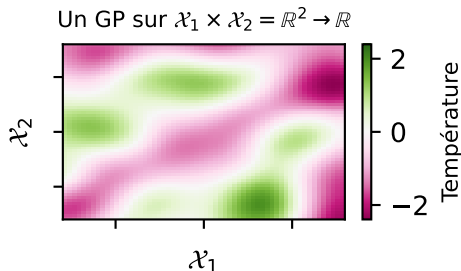
- GP peut être vu comme une réindéxation d'un vecteur "infini"
- Cas temporel: $y(t) : \mathbb{R}^+ \rightarrow \mathbb{R}$, *n'importe qu'elle discrétisation* de y doit former un vecteur Gaussien, en prenant par ex 2 entrées de y :

$$y(\mathbf{x}) \sim GP(m, k) \implies y([t_i, t_j]) \sim \mathcal{N}\left(\begin{pmatrix} \mu_i \\ \mu_j \end{pmatrix}, \begin{pmatrix} \Sigma_i^2 & \Sigma_{ij} \\ \Sigma_{ij} & \Sigma_j^2 \end{pmatrix}\right)$$

- Exemple visu: cf. exemples introductifs

Construction d'un GP

- Cas spatial: $y(\mathbf{x}) : \mathbb{R}^2 \rightarrow \mathcal{Y} = \mathbb{R}$ (ex: température en 2D)
- Visu d'un échantillon:



Construction d'un GP

- On impose souvent une forme simple sur $\mathbb{E}(y(x)) = m(x)$ comme:
 $\beta^\top \mathbf{x}, \mathbf{0}$

Construction d'un GP

- On impose souvent une forme simple sur $\mathbb{E}(y(x)) = m(x)$ comme:
 $\beta^\top \mathbf{x}, \mathbf{0}$
- $y : \mathcal{X} \rightarrow \mathcal{Y}$ est une *fonction*: il suffit de bien choisir \mathcal{X} et \mathcal{Y}
 - $\mathcal{X} = \mathbb{R}^+$ et $\mathcal{Y} = \mathbb{R} \implies y = \textit{série temporelle}$ Rasmussen **and** Williams 2006

Construction d'un GP

- On impose souvent une forme simple sur $\mathbb{E}(y(x)) = m(x)$ comme:
 $\beta^\top \mathbf{x}, \mathbf{0}$
- $y : \mathcal{X} \rightarrow \mathcal{Y}$ est une *fonction*: il suffit de bien choisir \mathcal{X} et \mathcal{Y}
 - $\mathcal{X} = \mathbb{R}^+$ et $\mathcal{Y} = \mathbb{R} \implies y =$ *série temporelle* Rasmussen **and** Williams 2006
 - $\mathcal{X} = [a, b] \times [c, d]$ (ou \mathbb{R}^2) et $\mathcal{Y} = \mathbb{R} \implies y =$ *champ spatial* (météo, hydro, etc)

Construction d'un GP

- On impose souvent une forme simple sur $\mathbb{E}(y(x)) = m(x)$ comme:
 $\beta^\top \mathbf{x}, \mathbf{0}$
- $y : \mathcal{X} \rightarrow \mathcal{Y}$ est une *fonction*: il suffit de bien choisir \mathcal{X} et \mathcal{Y}
 - $\mathcal{X} = \mathbb{R}^+$ et $\mathcal{Y} = \mathbb{R} \implies y = \textit{série temporelle}$ Rasmussen **and** Williams 2006
 - $\mathcal{X} = [a, b] \times [c, d]$ (ou \mathbb{R}^2) et $\mathcal{Y} = \mathbb{R} \implies y = \textit{champ spatial}$ (météo, hydro, etc)
 - $\mathcal{X} = [a, b] \times [c, d] \times \mathbb{R}^+$ et $\mathcal{Y} = \mathbb{R} \implies y = \textit{série temporelle d'images}$

Construction d'un GP

- On impose souvent une forme simple sur $\mathbb{E}(y(x)) = m(x)$ comme:
 $\beta^\top \mathbf{x}, \mathbf{0}$
- $y : \mathcal{X} \rightarrow \mathcal{Y}$ est une *fonction*: il suffit de bien choisir \mathcal{X} et \mathcal{Y}
 - $\mathcal{X} = \mathbb{R}^+$ et $\mathcal{Y} = \mathbb{R} \implies y = \text{*série temporelle*}$ Rasmussen **and** Williams 2006
 - $\mathcal{X} = [a, b] \times [c, d]$ (ou \mathbb{R}^2) et $\mathcal{Y} = \mathbb{R} \implies y = \text{*champ spatial*}$ (météo, hydro, etc)
 - $\mathcal{X} = [a, b] \times [c, d] \times \mathbb{R}^+$ et $\mathcal{Y} = \mathbb{R} \implies y = \text{*série temporelle d'images*}$
 - $\mathcal{X} = \mathbb{R}^d$ et $\mathcal{Y} = \{1 \dots K\} \implies y = \text{*classificateur*}$ d'images/pixels (ex: segmentation) Hensman **and others** 2015
 - \mathcal{X} et \mathcal{Y} peuvent être des espaces de graphes Borovitskiy **and others** 2021
 - ... etc, etc, etc.

Gaussien: vecteur vs. processus

1 Contenu

2 Gaussien: vecteur vs. processus

- Construction d'un GP
- Le kernel

3 Prédire avec un GP

- Règle de Bayes
- prédiction = data \times hypothèse
- Exemples
- Application: prédiction de température maritime

4 Avantages / Limites des GP naïfs

- Avantages
- Limites
- Remèdes

Le kernel

- $k(\mathbf{x}, \mathbf{x}') = \mathbb{C}(y(\mathbf{x}), y(\mathbf{x}'))$ représente la *similarité de y* entre deux entrées \mathbf{x} et \mathbf{x}' , donc doit générer une matrice de covariance valide²

² $\sum_{i,j}^n a_i a_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0, \forall \mathbf{a} = [a_1, \dots, a_n]$

³ C-à-d être un noyau reproduisant d'un espace de Hilbert: $y(x_0) = \sum_{i=1}^{\infty} y(x_i) k(x_i, x_0)$

Le kernel

- $k(\mathbf{x}, \mathbf{x}') = \mathbb{C}(y(\mathbf{x}), y(\mathbf{x}'))$ représente la *similarité de y* entre deux entrées \mathbf{x} et \mathbf{x}' , donc doit générer une matrice de covariance valide²
- k définit *implicitement* les propriétés fonctionnelles de y ³

² $\sum_{i,j}^n a_i a_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0, \forall \mathbf{a} = [a_1, \dots, a_n]$

³ C-à-d être un noyau reproduisant d'un espace de Hilbert: $y(x_0) = \sum_{i=1}^{\infty} y(x_i) k(x_i, x_0)$

Le kernel

- $k(\mathbf{x}, \mathbf{x}') = \mathbb{C}(y(\mathbf{x}), y(\mathbf{x}'))$ représente la *similarité de y* entre deux entrées \mathbf{x} et \mathbf{x}' , donc doit générer une matrice de covariance valide²
- k définit *implicitement* les propriétés fonctionnelles de y ³
- Kernel: fonction qui dépend aussi *d'hyperparamètres* θ

² $\sum_{i,j}^n a_i a_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0, \forall \mathbf{a} = [a_1, \dots, a_n]$

³ C-à-d être un noyau reproduisant d'un espace de Hilbert: $y(x_0) = \sum_{i=1}^{\infty} y(x_i) k(x_i, x_0)$

Le kernel

- $k(\mathbf{x}, \mathbf{x}') = \mathbb{C}(y(\mathbf{x}), y(\mathbf{x}'))$ représente la *similarité de y* entre deux entrées \mathbf{x} et \mathbf{x}' , donc doit générer une matrice de covariance valide²
- k définit *implicitement* les propriétés fonctionnelles de y ³
- Kernel: fonction qui dépend aussi *d'hyperparamètres* θ
- Et donc le nerf d'un GP est essentiellement dans son kernel (noyau)

² $\sum_{i,j}^n a_i a_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0, \forall \mathbf{a} = [a_1, \dots, a_n]$

³ C-à-d être un noyau reproduisant d'un espace de Hilbert: $y(x_0) = \sum_{i=1}^{\infty} y(x_i) k(x_i, x_0)$

Le kernel

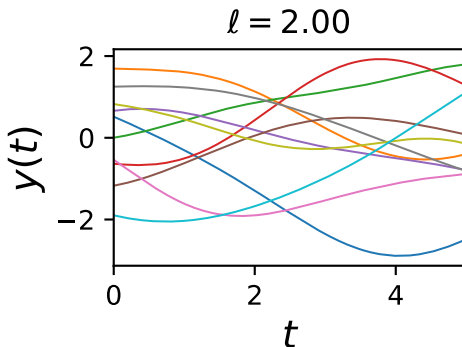
- Quelques exemples de $GP(0, k)$ défini sur $\mathcal{X} = \mathbb{R}^+$ et $\mathcal{Y} = \mathbb{R}$ (série temporelle)
- kernel de série temporelle \leftrightarrow fonction d'*autocovariance*
- Visualiser: bonne manière de comprendre un kernel

Le kernel

- Exponentiel quadratique (aka "Radial basis", "Gaussian", "Heat" kernel): $k(t, t') = e^{-\frac{(t-t')^2}{2\ell^2}}$
- $GP(0, k)$ génère des y infiniment différentiables par rapport à t
- ℓ contrôle le lissé (ici au sens oscillatoire) des générées

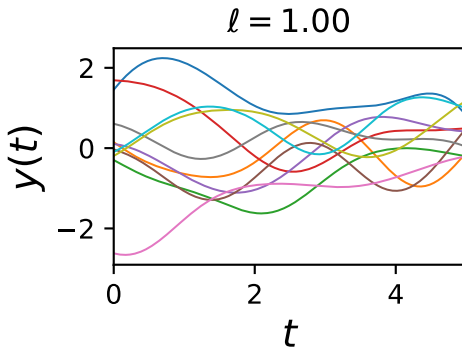
Le kernel

- Exponentiel quadratique (aka "Radial basis", "Gaussian", "Heat" kernel): $k(t, t') = e^{-\frac{(t-t')^2}{2\ell^2}}$
- $GP(0, k)$ génère des y infiniment différentiables par rapport à t
- ℓ contrôle le lissé (ici au sens oscillatoire) des générées



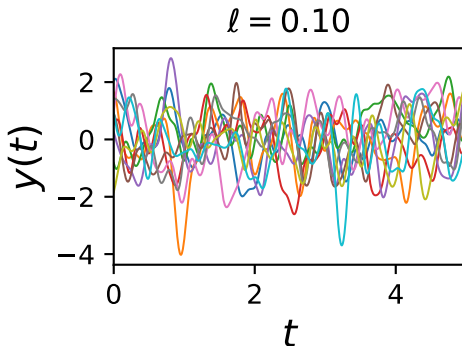
Le kernel

- Exponentiel quadratique (aka "Radial basis", "Gaussian", "Heat" kernel): $k(t, t') = e^{-\frac{(t-t')^2}{2\ell^2}}$
- $GP(0, k)$ génère des y infiniment différentiables par rapport à t
- ℓ contrôle le lissé (ici au sens oscillatoire) des générées



Le kernel

- Exponentiel quadratique (aka "Radial basis", "Gaussian", "Heat" kernel): $k(t, t') = e^{-\frac{(t-t')^2}{2\ell^2}}$
- $GP(0, k)$ génère des y infiniment différentiables par rapport à t
- ℓ contrôle le lissé (ici au sens oscillatoire) des générées

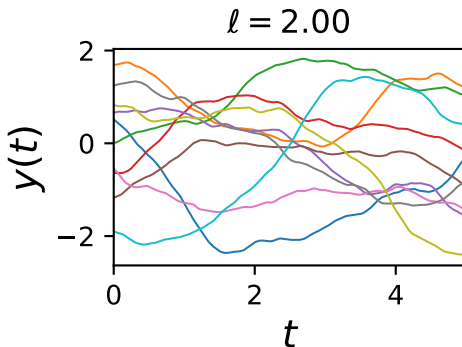


Le kernel

- kernel de Matérn: $k(t, t') = (1 + \sqrt{\frac{3|t-t'|}{\ell}})e^{-\sqrt{\frac{3|t-t'|}{\ell}}}$
- Génère des y différentiables une seule fois par rapport à t
- ℓ : même interprétation que le kernel exponentiel quadratique

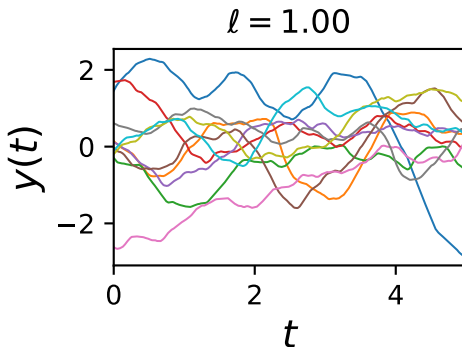
Le kernel

- kernel de Matérn: $k(t, t') = (1 + \sqrt{\frac{3|t-t'|}{\ell}})e^{-\sqrt{\frac{3|t-t'|}{\ell}}}$
- Génère des y différentiables une seule fois par rapport à t
- ℓ : même interprétation que le kernel exponentiel quadratique



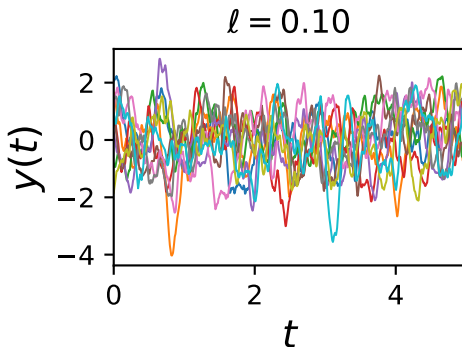
Le kernel

- kernel de Matérn: $k(t, t') = (1 + \sqrt{\frac{3|t-t'|}{\ell}})e^{-\sqrt{\frac{3|t-t'|}{\ell}}}$
- Génère des y différentiables une seule fois par rapport à t
- ℓ : même interprétation que le kernel exponentiel quadratique



Le kernel

- kernel de Matérn: $k(t, t') = (1 + \sqrt{\frac{3|t-t'|}{\ell}})e^{-\sqrt{\frac{3|t-t'|}{\ell}}}$
- Génère des y différentiables une seule fois par rapport à t
- ℓ : même interprétation que le kernel exponentiel quadratique

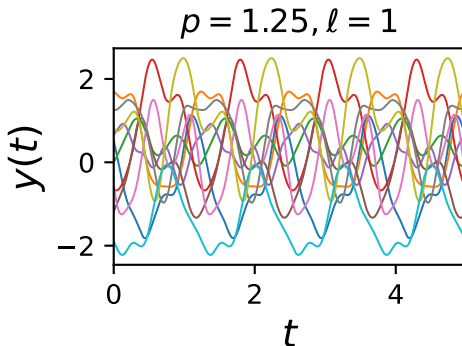


Le kernel

- kernel localement-périodique: $k(t, t') = e^{-\frac{2 \sin^2(\pi(t-t')/T_0)}{\ell^2}}$
- Génère des y périodiques ET lisses

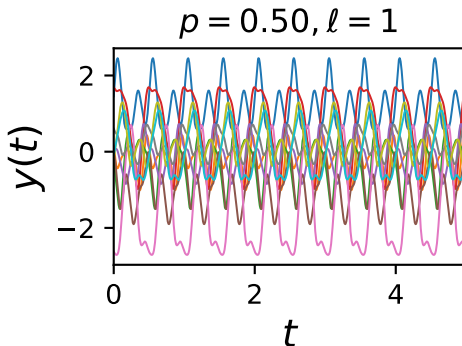
Le kernel

- kernel localement-périodique: $k(t, t') = e^{-\frac{2 \sin^2(\pi(t-t')/T_0)}{\ell^2}}$
- Génère des y périodiques ET lisses



Le kernel

- kernel localement-périodique: $k(t, t') = e^{-\frac{2 \sin^2(\pi(t-t')/T_0)}{\ell^2}}$
- Génère des y périodiques ET lisses

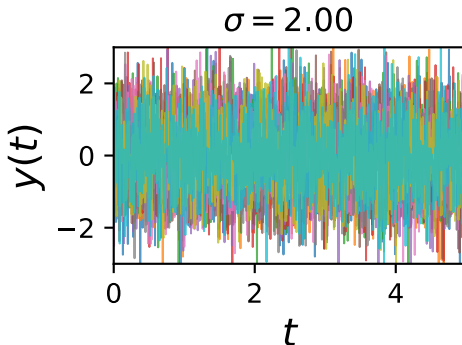


Le kernel

- kernel de bruit blanc:

$$k(t, t') = \begin{cases} \sigma^2 & \text{si } t = t' \\ 0 & \text{sinon} \end{cases}$$

- Aucune corrélation. σ contrôle l'amplitude (variance) du bruit

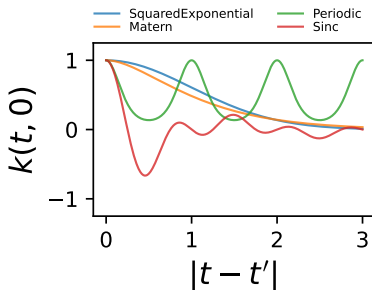


- Aussi, des kernels dans le domaine fréquentiel (Fourier) Parra **and** Tobar 2017⁴

⁴ex: $\sum_{q=0}^Q \sigma_q^2 e^{-2\pi^2 \Sigma_q (t-t')^2} \cos(2\pi \mu_q (t - t'))$

Le kernel

- Aussi, des kernels dans le domaine fréquentiel (Fourier) Parra **and** Tobar 2017⁴
- Famille des *kernels stationnaires*, $k(t, t')$ ne dépend que de $t - t' \leftrightarrow$ "plus les inputs sont éloignées, moins les sorties associées sont corrélées"



⁴ ex: $\sum_{q=0}^Q \sigma_q^2 e^{-2\pi^2 \Sigma_q (t-t')^2} \cos(2\pi \mu_q (t - t'))$

- Flexibilité des kernels: $k_1 + k_2$, $k_1 \times k_2$, $g(t) \times k$ (g est déterministe)

Le kernel

- Flexibilité des kernels: $k_1 + k_2$, $k_1 \times k_2$, $g(t) \times k$ (g est déterministe)
 - $k_1 + k_2$: "Ou" logique
 - $k_1 \times k_2$: "Et" logique
- Depuis quelques années: construction de kernels par des deep-nets
Wilson **and others** 2016, pratique mais peu interprétable

- Flexibilité des kernels: $k_1 + k_2$, $k_1 \times k_2$, $g(t) \times k$ (g est déterministe)
 - $k_1 + k_2$: "Ou" logique
 - $k_1 \times k_2$: "Et" logique
- Depuis quelques années: construction de kernels par des deep-nets
Wilson **and others** 2016, pratique mais peu interprétable
- Plus récemment: construction de kernel à partir d'équations différentielles

Prédire avec un GP

- 1 Contenu
- 2 Gaussien: vecteur vs. processus
 - Construction d'un GP
 - Le kernel
- 3 Prédire avec un GP
 - Règle de Bayes
 - prédiction = data \times hypothèse
 - Exemples
 - Application: prédiction de température maritime
- 4 Avantages / Limites des GP naïfs
 - Avantages
 - Limites
 - Remèdes

Règle de Bayes

- Règle de Bayes: $p(y_2|y_1) = \frac{p(y_1, y_2)}{p(y_1)} = \frac{p(y_1|y_2)p(y_2)}{p(y_1)}$

Règle de Bayes

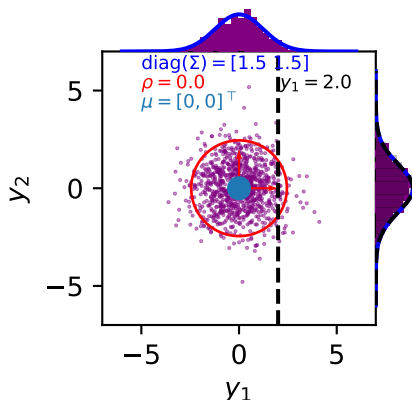
- Règle de Bayes: $p(y_2|y_1) = \frac{p(y_1, y_2)}{p(y_1)} = \frac{p(y_1|y_2)p(y_2)}{p(y_1)}$
- $\mathbf{y} = [y_1, y_2]^\top \sim \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_1 & \rho \\ \rho & \Sigma_2 \end{pmatrix}\right)$

Règle de Bayes

- Règle de Bayes: $p(y_2|y_1) = \frac{p(y_1, y_2)}{p(y_1)} = \frac{p(y_1|y_2)p(y_2)}{p(y_1)}$
- $\mathbf{y} = [y_1, y_2]^\top \sim \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_1 & \rho \\ \rho & \Sigma_2 \end{pmatrix}\right)$
- $p(y_2|y_1 = u) = \mathcal{N}(\mu_2 + \rho \Sigma_1^{-1}(u - \mu_1), \Sigma_2 - \rho^2 \Sigma_1^{-1})$

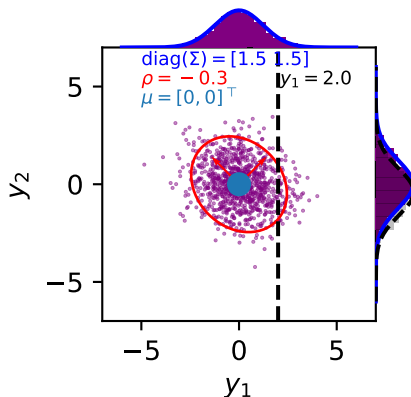
Règle de Bayes

- Règle de Bayes: $p(y_2|y_1) = \frac{p(y_1, y_2)}{p(y_1)} = \frac{p(y_1|y_2)p(y_2)}{p(y_1)}$
- $\mathbf{y} = [y_1, y_2]^\top \sim \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_1 & \rho \\ \rho & \Sigma_2 \end{pmatrix}\right)$
- $p(y_2|y_1 = u) = \mathcal{N}(\mu_2 + \rho \Sigma_1^{-1}(u - \mu_1), \Sigma_2 - \rho^2 \Sigma_1^{-1})$



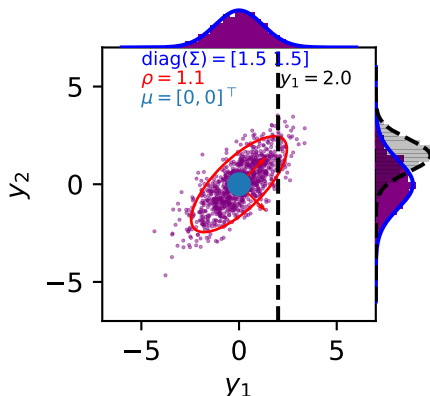
Règle de Bayes

- Règle de Bayes: $p(y_2|y_1) = \frac{p(y_1, y_2)}{p(y_1)} = \frac{p(y_1|y_2)p(y_2)}{p(y_1)}$
- $\mathbf{y} = [y_1, y_2]^\top \sim \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_1 & \rho \\ \rho & \Sigma_2 \end{pmatrix}\right)$
- $p(y_2|y_1 = u) = \mathcal{N}(\mu_2 + \rho \Sigma_1^{-1}(u - \mu_1), \Sigma_2 - \rho^2 \Sigma_1^{-1})$



Règle de Bayes

- Règle de Bayes: $p(y_2|y_1) = \frac{p(y_1, y_2)}{p(y_1)} = \frac{p(y_1|y_2)p(y_2)}{p(y_1)}$
- $\mathbf{y} = [y_1, y_2]^\top \sim \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_1 & \rho \\ \rho & \Sigma_2 \end{pmatrix}\right)$
- $p(y_2|y_1 = u) = \mathcal{N}(\mu_2 + \rho \Sigma_1^{-1}(u - \mu_1), \Sigma_2 - \rho^2 \Sigma_1^{-1})$



Règle de Bayes

- Règle de Bayes: $p(y_2|y_1) = \frac{p(y_1, y_2)}{p(y_1)} = \frac{p(y_1|y_2)p(y_2)}{p(y_1)}$
- $\mathbf{y} = [y_1, y_2]^\top \sim \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_1 & \rho \\ \rho & \Sigma_2 \end{pmatrix}\right)$

$$p(y_2|y_1 = u) = \mathcal{N}(\mu_2 + \underbrace{\rho \Sigma_1^{-1}(u - \mu_1)}_{\geq 0}, \Sigma_2 - \underbrace{\rho^2 \Sigma_1^{-1}}_{\geq 0})$$

Conditionner $y_2|y_1 \leftrightarrow$ réduire l'incertitude (variance) de y_2

Conditionnement entre variables Gaussiennes: reste Gaussien (conjugué)

Prédire avec un GP

- 1 Contenu
- 2 Gaussien: vecteur vs. processus
 - Construction d'un GP
 - Le kernel
- 3 Prédire avec un GP
 - Règle de Bayes
 - prédiction = data \times hypothèse
 - Exemples
 - Application: prédiction de température maritime
- 4 Avantages / Limites des GP naïfs
 - Avantages
 - Limites
 - Remèdes

prédiction = data \times hypothèse

- Data **entraînement**: $\{y_i, x_i\}_{1 \leq i \leq n} = [\mathbf{y}, \mathbf{x}]$, test: y_0, \mathbf{x}_0

prédiction = data \times hypothèse

- Data **entraînement**: $\{y_i, x_i\}_{1 \leq i \leq n} = [\mathbf{y}, \mathbf{x}]$, test: y_0, x_0
- Modèle de **prédiction** (régression): $y_0 = f_\theta(x_0)$

prédiction = data \times hypothèse

- Data **entraînement**: $\{y_i, x_i\}_{1 \leq i \leq n} = [\mathbf{y}, \mathbf{x}]$, test: y_0, \mathbf{x}_0
- Modèle de **prédiction** (régression): $y_0 = f_\theta(\mathbf{x}_0)$
- **Hypothèses**: $f_\theta \sim GP(0, k_\theta)$ et $y_i = f_\theta(x_i)$:

$$\underbrace{[\mathbf{y}, f_\theta(\mathbf{x}_0)]^\top}_{(n+1) \times 1} \sim \mathcal{N}(\mathbf{0}, \begin{pmatrix} \mathbf{K}_y & \mathbf{k}_0 \\ \mathbf{k}_0^\top & k_\theta(\mathbf{x}_0, \mathbf{x}_0) \end{pmatrix})$$

prédiction = data \times hypothèse

- Data **entraînement**: $\{y_i, x_i\}_{1 \leq i \leq n} = [\mathbf{y}, \mathbf{x}]$, test: y_0, \mathbf{x}_0
- Modèle de **prédiction** (régression): $y_0 = f_\theta(\mathbf{x}_0)$
- **Hypothèses**: $f_\theta \sim GP(0, k_\theta)$ et $y_i = f_\theta(x_i)$:

$$\underbrace{[\mathbf{y}, f_\theta(\mathbf{x}_0)]^\top}_{(n+1) \times 1} \sim \mathcal{N}(\mathbf{0}, \begin{pmatrix} \mathbf{K}_y & \mathbf{k}_0 \\ \mathbf{k}_0^\top & k_\theta(\mathbf{x}_0, \mathbf{x}_0) \end{pmatrix})$$

$$\mathbf{K}_y = \begin{pmatrix} k_\theta(\mathbf{x}_1, \mathbf{x}_1) & \dots & k_\theta(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k_\theta(\mathbf{x}_n, \mathbf{x}_1) & \dots & k_\theta(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix}$$

prédiction = data \times hypothèse

- Data **entraînement**: $\{y_i, x_i\}_{1 \leq i \leq n} = [\mathbf{y}, \mathbf{x}]$, test: y_0, x_0
- Modèle de **prédiction** (régression): $y_0 = f_\theta(x_0)$
- **Hypothèses**: $f_\theta \sim GP(0, k_\theta)$ et $y_i = f_\theta(x_i)$:

$$\underbrace{[\mathbf{y}, f_\theta(x_0)]^\top}_{(n+1) \times 1} \sim \mathcal{N}(\mathbf{0}, \begin{pmatrix} \mathbf{K}_y & \mathbf{k}_0 \\ \mathbf{k}_0^\top & k_\theta(x_0, x_0) \end{pmatrix})$$

$$\mathbf{k}_0 = [k_\theta(x_0, x_1), \dots, k_\theta(x_0, x_n)]^\top$$

prédiction = data \times hypothèse

- Data **entraînement**: $\{y_i, x_i\}_{1 \leq i \leq n} = [\mathbf{y}, \mathbf{x}]$, test: y_0, \mathbf{x}_0
- Modèle de **prédiction** (régression): $y_0 = f_\theta(\mathbf{x}_0)$
- **Hypothèses**: $f_\theta \sim GP(0, k_\theta)$ et $y_i = f_\theta(x_i)$:

$$\underbrace{[\mathbf{y}, f_\theta(\mathbf{x}_0)]^\top}_{(n+1) \times 1} \sim \mathcal{N}(\mathbf{0}, \begin{pmatrix} \mathbf{K}_{\mathbf{y}} & \mathbf{k}_0 \\ \mathbf{k}_0^\top & k_\theta(\mathbf{x}_0, \mathbf{x}_0) \end{pmatrix})$$

- But: apprendre $f_\theta \rightarrow$ prédire $f_\theta(\mathbf{x}_0) = y_0$
- Intuitivement: générer des **prédictions** avec $f_\theta \sim GP(0, k_\theta)$ ET garder ce qui fit à $[\mathbf{y}, \mathbf{x}]$ et à \mathbf{x}_0 :

prédiction = data \times hypothèse

- Data **entraînement**: $\{y_i, x_i\}_{1 \leq i \leq n} = [\mathbf{y}, \mathbf{x}]$, test: y_0, \mathbf{x}_0
- Modèle de **prédiction** (régression): $y_0 = f_\theta(\mathbf{x}_0)$
- **Hypothèses**: $f_\theta \sim GP(0, k_\theta)$ et $y_i = f_\theta(x_i)$:

$$\underbrace{[\mathbf{y}, f_\theta(\mathbf{x}_0)]^\top}_{(n+1) \times 1} \sim \mathcal{N}(\mathbf{0}, \begin{pmatrix} \mathbf{K}_\mathbf{y} & \mathbf{k}_0 \\ \mathbf{k}_0^\top & k_\theta(\mathbf{x}_0, \mathbf{x}_0) \end{pmatrix})$$

- But: apprendre $f_\theta \rightarrow$ prédire $f_\theta(\mathbf{x}_0) = y_0$
- Intuitivement: générer des **prédictions** avec $f_\theta \sim GP(0, k_\theta)$ ET garder ce qui fit à $[\mathbf{y}, \mathbf{x}]$ et à \mathbf{x}_0 :

$$\underbrace{p(\text{hypothèse} | \text{data})}_{\text{posterior}} = \frac{\overbrace{p(\text{data} | \text{hypothèse})}^{\text{likelihood}} \overbrace{p(\text{hypothèse})}^{\text{prior}}}{\underbrace{p(\text{data})}_{\text{evidence}}}$$

prédiction = data \times hypothèse

- Data **entraînement**: $\{y_i, x_i\}_{1 \leq i \leq n} = [\mathbf{y}, \mathbf{x}]$, test: y_0, \mathbf{x}_0
- Modèle de **prédiction** (régression): $y_0 = f_\theta(\mathbf{x}_0)$
- **Hypothèses**: $f_\theta \sim GP(0, k_\theta)$ et $y_i = f_\theta(x_i)$:

$$\underbrace{[\mathbf{y}, f_\theta(\mathbf{x}_0)]^\top}_{(n+1) \times 1} \sim \mathcal{N}(\mathbf{0}, \begin{pmatrix} \mathbf{K}_{\mathbf{y}} & \mathbf{k}_0 \\ \mathbf{k}_0^\top & k_\theta(\mathbf{x}_0, \mathbf{x}_0) \end{pmatrix})$$

- But: apprendre $f_\theta \rightarrow$ prédire $f_\theta(\mathbf{x}_0) = y_0$
- Intuitivement: générer des **prédictions** avec $f_\theta \sim GP(0, k_\theta)$ ET garder ce qui fit à $[\mathbf{y}, \mathbf{x}]$ et à \mathbf{x}_0 :
- Revient à générer selon:

$$\overbrace{p(f_\theta(\mathbf{x}), f_\theta(\mathbf{x}_0) | \mathbf{y}, \mathbf{x}, \mathbf{x}_0)}^{\text{posterior}}$$

prédiction = data \times hypothèse

- Data **entraînement**: $\{y_i, x_i\}_{1 \leq i \leq n} = [\mathbf{y}, \mathbf{x}]$, test: y_0, \mathbf{x}_0
- Modèle de **prédiction** (régression): $y_0 = f_\theta(\mathbf{x}_0)$
- **Hypothèses**: $f_\theta \sim GP(0, k_\theta)$ et $y_i = f_\theta(x_i)$:

$$\underbrace{[\mathbf{y}, f_\theta(\mathbf{x}_0)]^\top}_{(n+1) \times 1} \sim \mathcal{N}(\mathbf{0}, \begin{pmatrix} \mathbf{K}_\mathbf{y} & \mathbf{k}_0 \\ \mathbf{k}_0^\top & k_\theta(\mathbf{x}_0, \mathbf{x}_0) \end{pmatrix})$$

- But: apprendre $f_\theta \rightarrow$ prédire $f_\theta(\mathbf{x}_0) = y_0$
- Intuitivement: générer des **prédictions** avec $f_\theta \sim GP(0, k_\theta)$ ET garder ce qui fit à $[\mathbf{y}, \mathbf{x}]$ et à \mathbf{x}_0 :
- Revient à générer selon:

$$\underbrace{p(f_\theta(\mathbf{x}), f_\theta(\mathbf{x}_0) | \mathbf{y}, \mathbf{x}, \mathbf{x}_0)}_{\text{posterior}} = \frac{\overbrace{p(\mathbf{y} | f_\theta(\mathbf{x}))}^{\text{likelihood}} \overbrace{p(f_\theta(\mathbf{x}), f_\theta(\mathbf{x}_0))}^{\text{prior}}}{p(\mathbf{y} | \mathbf{x})} = GP(m_*, k_{*, \theta})$$

prédiction = data \times hypothèse

Prédiction (aka inférence):

- $m_*, k_{*,\theta}$?
- $p(f_\theta|y, \mathbf{x}) = GP(m_*, k_{*,\theta})$

prédiction = data \times hypothèse

Prédiction (aka inférence):

- $m_*, k_{*,\theta}$?
- $p(f_\theta|y, \mathbf{x}) = GP(m_*, k_{*,\theta})$
- $m_*(x_0) = 0 + \mathbf{k}_0^\top \mathbf{K}_y^{-1} \mathbf{y}$

prédiction = data \times hypothèse

Prédiction (aka inférence):

- $m_*, k_{*,\theta}$?
- $p(f_\theta|y, \mathbf{x}) = GP(m_*, k_{*,\theta})$
- $m_*(x_0) = 0 + \mathbf{k}_0^\top \mathbf{K}_y^{-1} \mathbf{y}$

prédiction = data \times hypothèse

Prédiction (aka inférence):

- $m_*, k_{*,\theta}$?
- $p(f_\theta|y, \mathbf{x}) = GP(m_*, k_{*,\theta})$
- $m_*(x_0) = 0 + \mathbf{k}_0^\top \mathbf{K}_y^{-1} \mathbf{y}$
- $k_{*,\theta}(x_0, x'_0) = k_\theta(x_0, x'_0) -$

prédiction = data \times hypothèse

Prédiction (aka inférence):

- $m_*, k_{*,\theta}$?
- $p(f_\theta|y, \mathbf{x}) = GP(m_*, k_{*,\theta})$
- $m_*(x_0) = 0 + \mathbf{k}_0^\top \mathbf{K}_y^{-1} y$
- $k_{*,\theta}(x_0, x'_0) = k_{\theta}(x_0, x'_0) - \mathbf{k}_0^\top \mathbf{K}_y^{-1} \mathbf{k}_0$

prédiction = data \times hypothèse

Prédiction (aka inférence):

- $m_*, k_{*,\theta}$?
- $p(f_\theta|y, \mathbf{x}) = GP(m_*, k_{*,\theta})$
- $m_*(x_0) = 0 + \mathbf{k}_0^\top \mathbf{K}_y^{-1} y$
- $k_{*,\theta}(x_0, x'_0) = k_{\theta}(x_0, x'_0) - \mathbf{k}_0^\top \mathbf{K}_y^{-1} \mathbf{k}_0$
- La distribution de probabilités de la prédiction y_0 (connaissant les data d'entraînement) est caractérisée !
- Pour n'importe quel x_0 : accès à la prédiction moyenne, incertitude (intervalle de confiance), quantiles, etc., de y_0

prédiction = data \times hypothèse

Prédiction (aka inférence):

- $m_*, k_{*,\theta}$?
- $p(f_\theta|y, \mathbf{x}) = GP(m_*, k_{*,\theta})$
- $m_*(x_0) = 0 + \mathbf{k}_0^\top \mathbf{K}_y^{-1} \mathbf{y}$
- $k_{*,\theta}(x_0, x'_0) = k_{\theta}(x_0, x'_0) - \mathbf{k}_0^\top \mathbf{K}_y^{-1} \mathbf{k}_0$
- La distribution de probabilités de la prédiction y_0 (connaissant les data d'entraînement) est caractérisée !
- Pour n'importe quel x_0 : accès à la prédiction moyenne, incertitude (intervalle de confiance), quantiles, etc., de y_0
- Modèle de prédiction = { data d'entraînement, hyperparamètres }

prédiction = data \times hypothèse

Prédiction (aka inférence):

- $m_*, k_{*,\theta}$?
- $p(f_\theta|y, \mathbf{x}) = GP(m_*, k_{*,\theta})$
- $m_*(x_0) = 0 + \mathbf{k}_0^\top \mathbf{K}_y^{-1} \mathbf{y}$
- $k_{*,\theta}(x_0, x'_0) = k_{\theta}(x_0, x'_0) - \mathbf{k}_0^\top \mathbf{K}_y^{-1} \mathbf{k}_0$
- La distribution de probabilités de la prédiction y_0 (connaissant les data d'entraînement) est caractérisée !
- Pour n'importe quel x_0 : accès à la prédiction moyenne, incertitude (intervalle de confiance), quantiles, etc., de y_0
- Modèle de prédiction = { data d'entraînement, hyperparamètres }
- La prédiction dépend aussi de θ (hyperparamètre) !

prédiction = data \times hypothèse

Entraînement:

- Meilleur θ ?

prédiction = data \times hypothèse

Entraînement:

- Meilleur θ ?
- Maximum de *likelihood*(θ) $:= p(y|f_{\theta}(\mathbf{x}))$?

Entraînement:

- Meilleur θ ?
- Maximum de *likelihood*(θ) $:= p(y|f_{\theta}(\mathbf{x}))$?
- NON !! Restreint l'hypothèse uniquement à $[y, \mathbf{x}]$ (overfitting)

Entraînement:

- Meilleur θ ?
- Maximum de *likelihood*(θ) := $p(y|f_{\theta}(\mathbf{x}))$?
- NON !! Restreint l'hypothèse uniquement à $[y, \mathbf{x}]$ (overfitting)
- Mieux: hypothèse sur les data + voisinage
- Maximum de *likelihood marginale*:

Entraînement:

- Meilleur θ ?
- Maximum de *likelihood*(θ) := $p(y|f_{\theta}(\mathbf{x}))$?
- NON !! Restreint l'hypothèse uniquement à $[y, \mathbf{x}]$ (overfitting)
- Mieux: hypothèse sur les data + voisinage
- Maximum de *likelihood marginale*:

$$= p(y|f_{\theta}(\mathbf{x}))p(f_{\theta}(\mathbf{x}))$$

- $\leftrightarrow \theta$ telle que les données d'entraînement y soient le plus probables au regard de toutes les des hypothèses possibles et des inputs \mathbf{x}

prédiction = data \times hypothèse

Entraînement:

- Meilleur θ ?
- Maximum de *likelihood*(θ) := $p(y|f_\theta(\mathbf{x}))$?
- NON !! Restreint l'hypothèse uniquement à $[y, \mathbf{x}]$ (overfitting)
- Mieux: hypothèse sur les data + voisinage
- Maximum de *likelihood marginale*:

$$p(y|\mathbf{x}, \theta) = \int_{\mathbb{R}^n} p(y|\mathbf{x}, f_\theta(\mathbf{x}))p(f_\theta(\mathbf{x}))d\mathbf{f}_\theta$$

- $\leftrightarrow \theta$ telle que les données d'entraînement y soient le plus probables au regard de toutes les des hypothèses possibles et des inputs \mathbf{x}

prédiction = data \times hypothèse

Entraînement:

- Meilleur θ ?
- Maximum de *likelihood*(θ) := $p(y|f_\theta(\mathbf{x}))$?
- NON !! Restreint l'hypothèse uniquement à $[y, \mathbf{x}]$ (overfitting)
- Mieux: hypothèse sur les data + voisinage
- Maximum de *likelihood marginale*:

$$\begin{aligned} p(y|\mathbf{x}, \theta) &= \int_{\mathbb{R}^n} p(y|\mathbf{x}, f_\theta(\mathbf{x}))p(f_\theta(\mathbf{x}))df_\theta \\ &= \int \mathcal{N}(f_\theta(\mathbf{x}), I_d)\mathcal{N}(\mathbf{0}, \mathbf{K}_y)df_\theta \end{aligned}$$

- $\leftrightarrow \theta$ telle que les données d'entraînement y soient le plus probables au regard de toutes les des hypothèses possibles et des inputs \mathbf{x}



$$\begin{aligned}\log \textit{likelihoodmarg}(\theta) &= \log p(\mathbf{y}|\mathbf{x}, \theta) (= \log \prod_i p(y_i|x_i, \theta)) \\ &= \log \mathcal{N}(\mathbf{0}, \mathbf{K}_y + I_d) \\ &= - \underbrace{\mathbf{y}^\top \mathbf{K}_y^{-1} \mathbf{y}}_{\textit{fidélité}} - \underbrace{\frac{1}{2} \log |\mathbf{K}_y|}_{\textit{pénalité}} - \frac{n}{2} \log 2\pi\end{aligned}$$



$$\begin{aligned}\log \textit{likelihoodmarg}(\theta) &= \log p(\mathbf{y}|\mathbf{x}, \theta) (= \log \prod_i p(y_i|x_i, \theta)) \\ &= \log \mathcal{N}(\mathbf{0}, \mathbf{K}_y + I_d) \\ &= - \underbrace{\mathbf{y}^\top \mathbf{K}_y^{-1} \mathbf{y}}_{\textit{fidélité}} - \underbrace{\frac{1}{2} \log |\mathbf{K}_y|}_{\textit{pénalité}} - \frac{n}{2} \log 2\pi\end{aligned}$$

- Rappel: \mathbf{K}_y matrice $n \times n$ des $k_\theta(x_i, x_j) = \mathbb{C}(f_\theta(x_i), f_\theta(x_j)) \stackrel{\text{ex}}{=} e^{-\frac{(x_i - x_j)^2}{2\ell^2}}$



$$\begin{aligned}\log \textit{likelihood}_{\textit{marg}}(\theta) &= \log p(\mathbf{y}|\mathbf{x}, \theta) (= \log \prod_i p(y_i|x_i, \theta)) \\ &= \log \mathcal{N}(\mathbf{0}, \mathbf{K}_y + I_d) \\ &= - \underbrace{\mathbf{y}^\top \mathbf{K}_y^{-1} \mathbf{y}}_{\textit{fidélité}} - \underbrace{\frac{1}{2} \log |\mathbf{K}_y|}_{\textit{pénalité}} - \frac{n}{2} \log 2\pi\end{aligned}$$

- Rappel: \mathbf{K}_y matrice $n \times n$ des $k_\theta(x_i, x_j) = \mathbb{C}(f_\theta(x_i), f_\theta(x_j)) \stackrel{\text{ex}}{=} e^{-\frac{(x_i - x_j)^2}{2\ell^2}}$
- Apprentissage de θ par descente de gradient de $-\log \textit{likelihood}_{\textit{marg}}$ (large sous-domaine de la communauté GP)

Méthodo générale:

- 1 Collecter des données: $\{y_i, x_i\}_{1 \leq i \leq n} = [\mathbf{y}, \mathbf{x}]$

Méthodo générale:

- 1 Collecter des données: $\{y_i, x_i\}_{1 \leq i \leq n} = [\mathbf{y}, \mathbf{x}]$
- 2 Formuler des hypothèses: choisir/combinaison un/des kernels k_θ

Méthodo générale:

- 1 Collecter des données: $\{y_i, x_i\}_{1 \leq i \leq n} = [\mathbf{y}, \mathbf{x}]$
- 2 Formuler des hypothèses: choisir/combinaire un/des kernels k_θ
- 3 **Entraîner** le modèle: estimer les hyperparamètres θ

Prédire avec un GP

Méthodo générale:

- 1 Collecter des données: $\{y_i, x_i\}_{1 \leq i \leq n} = [\mathbf{y}, \mathbf{x}]$
- 2 Formuler des **hypothèses**: choisir/combinaire un/des kernels k_θ
- 3 **Entraîner** le modèle: estimer les hyperparamètres θ
- 4 **Prédire** ("inférence"): évaluer m_* et $k_{*,\theta}$, générer des données, etc.

Méthodo générale:

- 1 Collecter des données: $\{y_i, x_i\}_{1 \leq i \leq n} = [\mathbf{y}, \mathbf{x}]$
- 2 Formuler des **hypothèses**: choisir/combinaire un/des kernels k_θ
- 3 **Entraîner** le modèle: estimer les hyperparamètres θ
- 4 **Prédire** ("inférence"): évaluer m_* et $k_{*,\theta}$, générer des données, etc.
- 5 Critique/évaluation du modèle

Prédire avec un GP

1 Contenu

2 Gaussien: vecteur vs. processus

- Construction d'un GP
- Le kernel

3 Prédire avec un GP

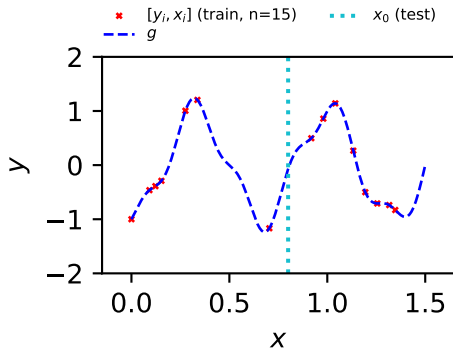
- Règle de Bayes
- prédiction = data \times hypothèse
- Exemples
- Application: prédiction de température maritime

4 Avantages / Limites des GP naïfs

- Avantages
- Limites
- Remèdes

Exemples

- **train** = $\{y_i, x_i\}_{i \leq n}$ où $y_i = g(x_i) = -\cos(3\pi x_i) + \frac{1}{4} \sin(8\pi x_i)$
- g inconnue



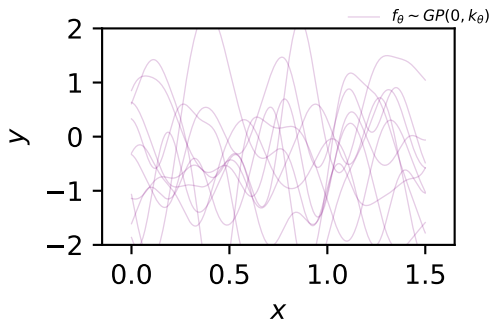
- $f_\theta(x_0 = 0.8) = \tilde{y}_0$? (*interpolation*)

Exemples

- **Hypothèses:** $y_i = f(x_i)$ et f est lisse \rightarrow kernel exponentiel quadratique
- $k_\theta(x, x') = \sigma_k^2 e^{-\frac{(x-x')^2}{2\ell^2}}, \theta = [\sigma_k, \ell]$

Exemples

- **Hypothèses:** $y_i = f(x_i)$ et f est lisse \rightarrow kernel exponentiel quadratique
- $k_\theta(x, x') = \sigma_k^2 e^{-\frac{(x-x')^2}{2\ell^2}}$, $\theta = [\sigma_k, \ell]$
- $[\sigma_k, \ell]$ inconnus (ici fixés arbitrairement)



Entraînement

- θ ? minimisation de $-\log$ likelihood marginale: $\sigma_k \approx 0.92, \ell \approx 0.11$

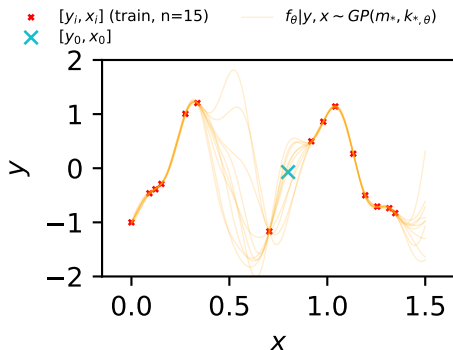
Entraînement

- θ ? minimisation de $-\log$ likelihood marginale: $\sigma_k \approx 0.92, \ell \approx 0.11$
- $m_*, k_{*,\theta}$ et θ appris

Exemples

Entraînement

- θ ? minimisation de $-\log$ likelihood marginale: $\sigma_k \approx 0.92, \ell \approx 0.11$
- $m_*, k_{*,\theta}$ et θ appris
- Inférence: générer selon la distribution posterior $GP(m_*, k_{*,\theta})$

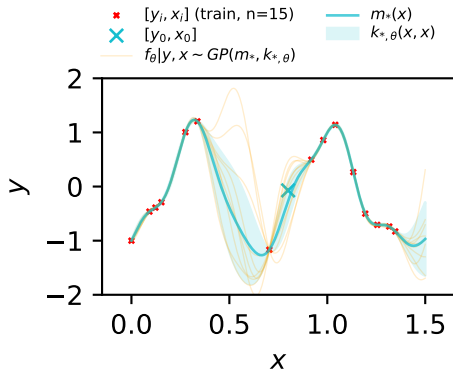


Inférence

- \tilde{y}_0 : moyenne de $p(f_\theta|y, \mathbf{x}, x_0) = GP(m_*, k_{*,\theta})$ en $x_0 = 0.8$
- $m_*(x_0) \pm \frac{1}{2}\sqrt{k_{*,\theta}(x_0, x_0)}$: $= -0.22 \pm 0.16$ (95%)

Exemples

- Quid d'autres valeurs de x_0 ?



- Incertitude sur "toute" la fonction

Exemples

- **Hypothèses** précédentes: "pas de bruit" (hypothèse forte), "lisse" (hypothèse large)

Exemples

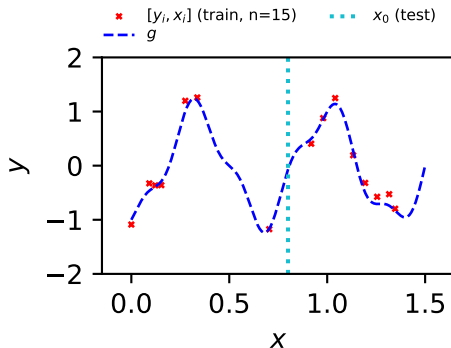
- **Hypothèses** précédentes: "pas de bruit" (hypothèse forte), "lisse" (hypothèse large)
- **Hypothèses** plus réalistes ?

Exemples

- **Hypothèses** précédentes: "pas de bruit" (hypothèse forte), "lisse" (hypothèse large)
- **Hypothèses** plus réalistes ?
- Bruit de mesure $y_i = f_\theta(x_i) + \epsilon$, lisse, quasi-périodique:
 $k_\theta = k_\epsilon + \sigma_{exp2} k_{exp2} \times \sigma_{per} k_{per}$
 - k_ϵ : bruit
 - $k_{exp2} \times k_{per}$: Lisse ET quasi-périodique
 - $\theta = [\sigma_\epsilon, \sigma_{exp2}, \ell, \sigma_{per}, T_0, \ell_{per}]$

Exemples

- Même données + bruit blanc ($\sigma = 0.3$)
- $f_{\theta}(x_0 = 0.8) = \tilde{y}_0$ (*smoothing*) ?



Exemples

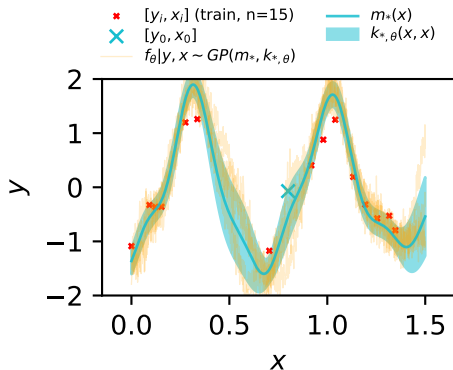
- θ ? minimisation de $-\log$ likelihood: $\sigma_{\epsilon} \approx 0.24$, $\sigma_{exp2} \approx 1$, $\ell \approx 1$, $\sigma_{per} \approx 1$, $T_0 = 0.7$, $\ell_{per} \approx 1$

Exemples

- θ ? minimisation de $-\log$ likelihood: $\sigma_{\epsilon} \approx 0.24$, $\sigma_{exp2} \approx 1$, $\ell \approx 1$, $\sigma_{per} \approx 1$, $T_0 = 0.7$, $\ell_{per} \approx 1$
- m_* , $k_{*,\theta}$ et θ appris

Exemples

- θ ? minimisation de $-\log$ likelihood: $\sigma_\epsilon \approx 0.24$, $\sigma_{exp2} \approx 1$, $\ell \approx 1$, $\sigma_{per} \approx 1$, $T_0 = 0.7$, $\ell_{per} \approx 1$
- m_* , $k_{*,\theta}$ et θ appris
- Génération selon la distribution posterior $GP(m_*, k_{*,\theta})$



Exemples

- Pour $x > 1.5$? (*forecast*)

Exemples

- Pour $x > 1.5$? (*forecast*)
- Comparaison des deux hypothèses

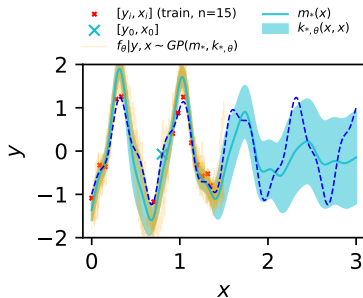
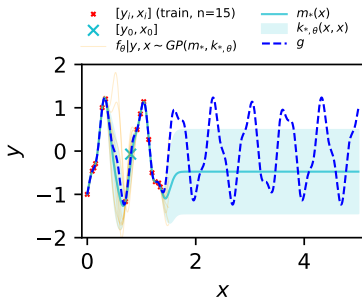


Figure: (a): sans bruit, lisse (b): bruit, lisse et périodique

Prédire avec un GP

1 Contenu

2 Gaussien: vecteur vs. processus

- Construction d'un GP
- Le kernel

3 Prédire avec un GP

- Règle de Bayes
- prédiction = data \times hypothèse
- Exemples
- Application: prédiction de température maritime

4 Avantages / Limites des GP naïfs

- Avantages
- Limites
- Remèdes

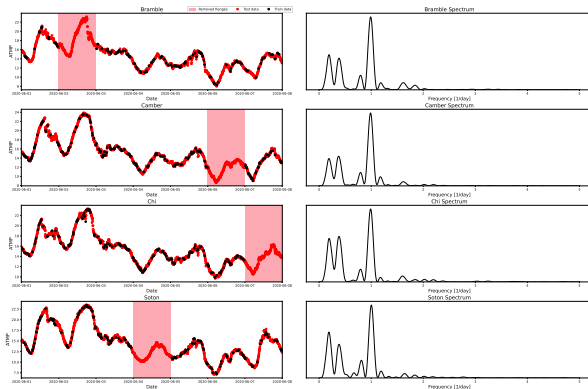
Application: prédiction de température maritime

- Séries temporelles de température sur la côté sud anglaise⁵: 4 stations sur 7 jours, $\Delta t \approx 5\text{min}$



Application: prédiction de température maritime

- Séries temporelles de température sur la côte sud anglaise⁵: 4 stations sur 7 jours, $\Delta t \approx 5\text{min}$
- But: estimation du bruit, forecasting et complétion de valeurs manquantes



⁵ Accessible ici

Application: prédiction de température maritime

- Série temporelle multivariée, GP défini de $\mathcal{X} = \mathbb{R}^+$ (temps) vers $\mathcal{Y} = \mathbb{R}^4$ (stations)

Application: prédiction de température maritime

- Série temporelle multivariée, GP défini de $\mathcal{X} = \mathbb{R}^+$ (temps) vers $\mathcal{Y} = \mathbb{R}^4$ (stations)
- Hypothèses: les stations sont corrélées, bruitées et périodiques
- Deux niveaux de corrélation: **intra**-station et **inter**-station

$$\mathbf{K}(t, t') = \begin{pmatrix} k_{11}(t, t') & k_{12}(t, t') & k_{13}(t, t') & k_{14}(t, t') \\ \vdots & k_{22}(t, t') & k_{23}(t, t') & k_{24}(t, t') \\ \vdots & \vdots & \ddots & \vdots \\ k_{41}(t, t') & \dots & \dots & k_{44}(t, t') \end{pmatrix}$$

- Chaque station doit être prédite avec par elle-même + les autres

Application: prédiction de température maritime

- Kernels "spectraux"⁶Parra **and** Tobar 2017: fit les puissances spectrales pour estimer les périodicités intra/inter et filtrer le bruit (hautes fréquences)
- Entraînement: $n = 644$, test: $n = 6845$, 84 hyperparamètres (notebook dispo: détails optim etc.)
- ...
- 1500 itérations (1min 23s)
- Comparaison avec le même kernel sans corrélation inter-station

⁶de la forme $k_{ij}(t, t') = \alpha_{ij} e^{-\frac{(|t-t'| + \theta_{ij})^2}{2\Sigma_{ij}}} \cos((|t - t'| + \theta_{ij})\mu_{ij} + \phi_{ij})$

Application: prédiction de température maritime

	MAE	MAPE	RMSE
Bramble (NaN)	2.10	10.37	2.79
Camber (NaN)	1.58	13.82	1.74
Chi (forecast)	1.76	12.53	2.01
Soton (NaN)	1.68	15.00	2.13

kernel *sans* corrélation inter-station

	MAE	MAPE	RMSE
Bramble	0.65	3.44	0.89
Camber	0.61	5.27	0.72
Chi	0.61	4.72	0.71
Soton	0.45	3.55	0.54

kernel *avec* corrélation inter-station

Table: Erreurs de prédiction

Application: prédiction de température maritime

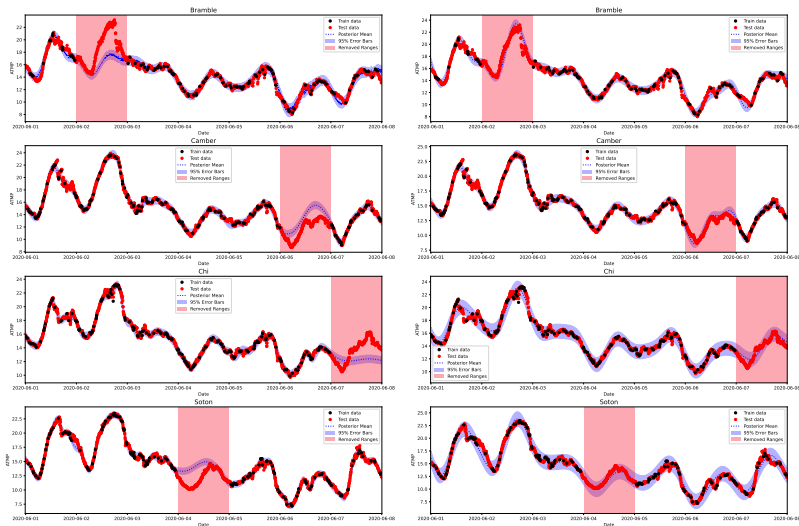


Figure: Prédiction (rouge) sans (resp avec) corrélation inter-stations à gauche (resp droite)

Application: prédiction de température maritime

- Kernel appris par chaque modèle:

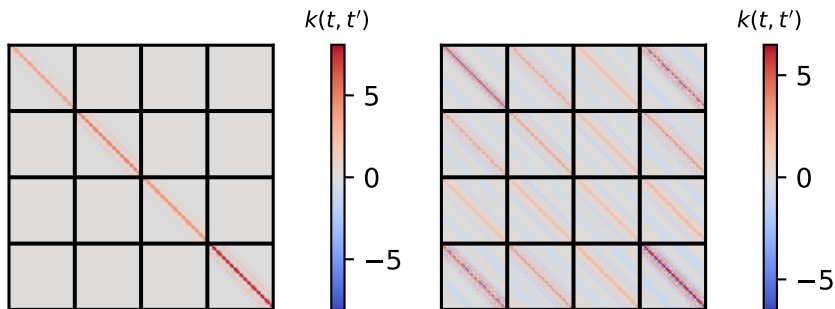


Figure: Kernel sans (resp avec) corrélation inter-stations à gauche (resp droite)

Application: prédiction de température maritime

- Kernel appris par chaque modèle:

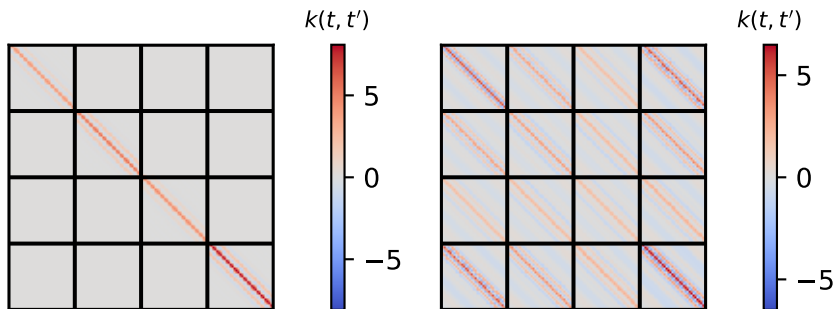


Figure: Kernel sans (resp avec) corrélation inter-stations à gauche (resp droite)

- Le modèle corrèle les prédictions de chaque station à court-terme

Application: prédiction de température maritime

- Prédiction probabilistes:

Application: prédiction de température maritime

- Prédiction probabilistes:
- Quelle est la probabilité qu'il fasse entre 13° et 14° à Bramble au 08/06/2020 à minuit ?
- Prédiction du modèle: 59%

Application: prédiction de température maritime

- Prédiction probabilistes:
- Quelle est la probabilité qu'il fasse entre 13° et 14° à Bramble au 08/06/2020 à minuit ?
- Prédiction du modèle: 59%
- A l'inverse: à la même date, quel intervalle à 90% ?
- Prédiction du modèle: $[14.2; 14.9]^{\circ}$

Avantages / Limites des GP naïfs

1 Contenu

2 Gaussien: vecteur vs. processus

- Construction d'un GP
- Le kernel

3 Prédire avec un GP

- Règle de Bayes
- prédiction = data \times hypothèse
- Exemples
- Application: prédiction de température maritime

4 Avantages / Limites des GP naïfs

- Avantages
- Limites
- Remèdes

Avantages

- *Probabiliste*: génératif, fourni une confiance (aka incertitude)

Avantages

- *Probabiliste*: génératif, fourni une confiance (aka incertitude)
- *Bayésien*: "contrôle" des hypothèses (a priori)

Avantages

- *Probabiliste*: génératif, fourni une confiance (aka incertitude)
- *Bayésien*: "contrôle" des hypothèses (a priori)
- *Non-paramétrique*: $\text{prédiction} = \text{data} \times \text{hypothèse}$
 - ① Paramètres du *GP* (moyenne, covariance) = les/des données d'entraînement, de taille infinie (non-bornée)

Avantages

- *Probabiliste*: génératif, fourni une confiance (aka incertitude)
- *Bayésien*: "contrôle" des hypothèses (a priori)
- *Non-paramétrique*: $\text{prédiction} = \text{data} \times \text{hypothèse}$
 - 1 Paramètres du *GP* (moyenne, covariance) = les/des données d'entraînement, de taille infinie (non-bornée)
 - 2 + les hyperparamètres (ce qui règle l'hypothèse)

Avantages

- *Probabiliste*: génératif, fourni une confiance (aka incertitude)
- *Bayésien*: "contrôle" des hypothèses (a priori)
- *Non-paramétrique*: $\text{prédiction} = \text{data} \times \text{hypothèse}$
 - 1 Paramètres du *GP* (moyenne, covariance) = les/des **données d'entraînement**, de taille infinie (non-bornée)
 - 2 + les **hyperparamètres** (ce qui règle l'hypothèse)
- Le modèle ne se "réduit" pas à un ensemble *fini* de paramètres

Avantages

- *Probabiliste*: génératif, fourni une confiance (aka incertitude)
- *Bayésien*: "contrôle" des hypothèses (a priori)
- *Non-paramétrique*: $\text{prédiction} = \text{data} \times \text{hypothèse}$
 - ① Paramètres du *GP* (moyenne, covariance) = les/des données d'entraînement, de taille infinie (non-bornée)
 - ② + les hyperparamètres (ce qui règle l'hypothèse)
- Le modèle ne se "réduit" pas à un ensemble fini de paramètres
- Cadre naturel des données spatio/temporelles: échantillonnage irrégulier (\neq SARIMA), décomposition de la variabilité

Avantages

- *Probabiliste*: génératif, fourni une confiance (aka incertitude)
- *Bayésien*: "contrôle" des hypothèses (a priori)
- *Non-paramétrique*: $\text{prédiction} = \text{data} \times \text{hypothèse}$
 - ① Paramètres du *GP* (moyenne, covariance) = les/des données d'entraînement, de taille infinie (non-bornée)
 - ② + les hyperparamètres (ce qui règle l'hypothèse)
- Le modèle ne se "réduit" pas à un ensemble fini de paramètres
- Cadre naturel des données spatio/temporelles: échantillonnage irrégulier (\neq SARIMA), décomposition de la variabilité
- Extensible à plus de dimensions d'entrée que le temps

Avantages / Limites des GP naïfs

- 1 Contenu
- 2 Gaussien: vecteur vs. processus
 - Construction d'un GP
 - Le kernel
- 3 Prédire avec un GP
 - Règle de Bayes
 - prédiction = data \times hypothèse
 - Exemples
 - Application: prédiction de température maritime
- 4 Avantages / Limites des GP naïfs
 - Avantages
 - **Limites**
 - Remèdes

Scalabilité:

- Bottleneck = inverser la matrice de covariance (kernel)
- La distribution prédictive $GP(m_*, k_{*,\theta})$ est lourde à calculer ($n \gg 10^4$)
 - **Entraînement** (log-likelihood): calcul: $\mathcal{O}(n^3)$ /itération. Mémoire: $\mathcal{O}(n^2)$
 - **Prédiction** (inférence): calcul $\mathcal{O}(n^3)$ (une fois),

GP pour les données non-Gaussiennes ?

- Monde Gaussien: symétrique, pas d'outliers, valeurs réelles

GP pour les données non-Gaussiennes ?

- Monde Gaussien: symétrique, pas d'outliers, valeurs réelles

$$\overbrace{p(f_\theta | y, \mathbf{x}, x_0)}^{\text{posterior}} = \frac{\overbrace{p(y | \mathbf{x}, f_\theta)}^{\text{likelihood}} \overbrace{p(f_\theta)}^{\text{prior=Gaussien}}}{\underbrace{p(y | \mathbf{x})}_{\text{Gaussien}}}$$

GP pour les données non-Gaussiennes ?

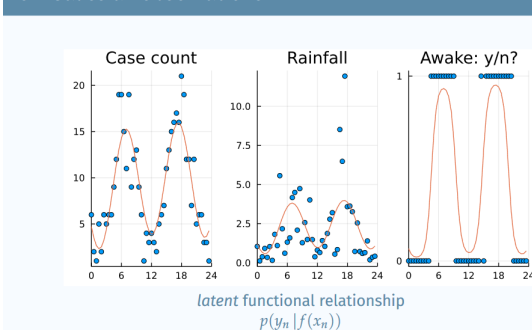
- Monde Gaussien: symétrique, pas d'outliers, valeurs réelles

$$\overbrace{p(f_\theta | y, \mathbf{x}, x_0)}^{\text{posterior=Gaussien}} = \frac{\overbrace{p(y | \mathbf{x}, f_\theta)}^{\text{likelihood=Gaussien}} \overbrace{p(f_\theta)}^{\text{prior=Gaussien}}}{\underbrace{p(y | \mathbf{x})}_{\text{Gaussien}}} = GP(m_*, k_{*,\theta})$$

GP pour les données non-Gaussiennes ?

- Mondes *non-Gaussiens*: valeurs positives (durées, pluviométrie), comptage (nb d'accidents), valeurs extrêmes (rendements financiers), taux, catégorielles, etc.

Non-Gaussian observations



Source: Gaussian Process Summer School 22'

GP pour les données non-Gaussiennes ?

- Mondes *non-Gaussiens*: valeurs positives (durées, pluviométrie), comptage (nb d'accidents), valeurs extrêmes (rendements financiers), taux, catégorielles, etc.
- Principe: transformer $\mathbb{E}(y)$ (espace latent) puis faire une hypothèse de GP sur la résultante. Prédiction en opération inverse.

GP pour les données non-Gaussiennes ?

- Mondes *non-Gaussiens*: valeurs positives (durées, pluviométrie), comptage (nb d'accidents), valeurs extrêmes (rendements financiers), taux, catégorielles, etc.
- Principe: transformer $\mathbb{E}(y)$ (espace latent) puis faire une hypothèse de GP sur la résultante. Prédiction en opération inverse.
- Exemple (classification): modéliser $\text{inv-logit}[p(y = \text{Awake})]$ avec un *GP*

GP pour les données non-Gaussiennes ?

- Mondes non-Gaussiens: valeurs positives (durées, pluviométrie), comptage (nb d'accidents), valeurs extrêmes (rendements financiers), taux, catégorielles, etc.
- Principe: transformer $\mathbb{E}(y)$ (espace latent) puis faire une hypothèse de GP sur la résultante. Prédiction en opération inverse.
- Exemple (classification): modéliser $\text{inv-logit}[p(y = \text{Awake})]$ avec un *GP*
- Conséquence:

GP pour les données non-Gaussiennes ?

- Mondes *non-Gaussiens*: valeurs positives (durées, pluviométrie), comptage (nb d'accidents), valeurs extrêmes (rendements financiers), taux, catégorielles, etc.
- Principe: transformer $\mathbb{E}(y)$ (espace latent) puis faire une hypothèse de GP sur la résultante. Prédiction en opération inverse.
- Exemple (classification): modéliser $\text{inv-logit}[p(y = \text{Awake})]$ avec un GP
- Conséquence:

$$\begin{array}{c} \text{posterior}=\text{non-Gaussien} \\ \underbrace{p(f_\theta | y, \mathbf{x})} \end{array} = \frac{\overbrace{p(y | \mathbf{x}, f_\theta)}^{\text{likelihood}=\text{non-Gaussien}} \overbrace{p(f_\theta)}^{\text{prior=Gaussien}}}{\underbrace{p(y | \mathbf{x})}_{\text{non-Gaussien}}} = \cancel{GP(m_*, k_*, \theta)}$$

Avantages / Limites des GP naïfs

1 Contenu

2 Gaussien: vecteur vs. processus

- Construction d'un GP
- Le kernel

3 Prédire avec un GP

- Règle de Bayes
- prédiction = data \times hypothèse
- Exemples
- Application: prédiction de température maritime

4 Avantages / Limites des GP naïfs

- Avantages
- Limites
- Remèdes

$GP(m_*, k_{*,\theta})$ est incalculable analytiquement ou trop coûteuse,

- Méthodes variationnelles: Bruinsma **and others** 2020; Hensman **and others** 2015
- $GP(m_*, k_{*,\theta})$ est *approximée* par une distribution "variationnelle" q_ψ , plus simple à apprendre

⁷Ex: divergence KL.

$GP(m_*, k_{*,\theta})$ est incalculable analytiquement ou trop coûteuse,

- Méthodes variationnelles: Bruinsma **and others** 2020; Hensman **and others** 2015
- $GP(m_*, k_{*,\theta})$ est *approximée* par une distribution "variationnelle" q_ψ , plus simple à apprendre
- (hyper-) paramètres appris par minimisation d'une "différence"⁷ entre q_ψ et $GP(m_*, k_{*,\theta})$

⁷Ex: divergence KL.

$GP(m_*, k_{*,\theta})$ est incalculable analytiquement ou trop coûteuse,

- Méthodes variationnelles: Bruinsma **and others** 2020; Hensman **and others** 2015
- $GP(m_*, k_{*,\theta})$ est *approximée* par une distribution "variationnelle" q_ψ , plus simple à apprendre
- (hyper-) paramètres appris par minimisation d'une "différence"⁷ entre q_ψ et $GP(m_*, k_{*,\theta})$
- Le modèle d'inférence est donc appris en résolvant un problème d'*optimisation*: gradient stochastique, différentiation automatique (PyTorch, Tensorflow, JAX), etc.

⁷Ex: divergence KL.

Méthodes d'*échantillonnage* (MCMC, importance sampling, MH, etc):
Rasmussen **and** Williams 2006

- Principe: remplacer $GP(m_*, k_{*,\theta})$ par un échantillonneur peu coûteux

Méthodes d'*échantillonnage* (MCMC, importance sampling, MH, etc):
Rasmussen **and** Williams 2006

- Principe: remplacer $GP(m_*, k_{*,\theta})$ par un échantillonneur peu coûteux
- $m_*, k_{*,\theta}$ restent inconnus et sont estimés en échantillonnant (beaucoup) puis en moyennant (loi forte des grands nombres)

Méthodes d'*échantillonnage* (MCMC, importance sampling, MH, etc):
Rasmussen **and** Williams 2006

- Principe: remplacer $GP(m_*, k_{*,\theta})$ par un échantillonneur peu coûteux
- $m_*, k_{*,\theta}$ restent inconnus et sont estimés en échantillonnant (beaucoup) puis en moyennant (loi forte des grands nombres)

Méthodes de *Kalman* ("State-space" model): Solin **and** Särkkä 2014;
Titsias **and others** 2024

- (uniquement pour les données temporelles)

Méthodes d'*échantillonnage* (MCMC, importance sampling, MH, etc):
Rasmussen **and** Williams 2006

- Principe: remplacer $GP(m_*, k_{*,\theta})$ par un échantillonneur peu coûteux
- $m_*, k_{*,\theta}$ restent inconnus et sont estimés en échantillonnant (beaucoup) puis en moyennant (loi forte des grands nombres)

Méthodes de *Kalman* ("State-space" model): Solin **and** Särkkä 2014;
Titsias **and others** 2024

- (uniquement pour les données temporelles)
- Réécriture du GP en équation différentielle stochastique

Méthodes d'*échantillonnage* (MCMC, importance sampling, MH, etc):
Rasmussen **and** Williams 2006

- Principe: remplacer $GP(m_*, k_{*,\theta})$ par un échantillonneur peu coûteux
- $m_*, k_{*,\theta}$ restent inconnus et sont estimés en échantillonnant (beaucoup) puis en moyennant (loi forte des grands nombres)

Méthodes de *Kalman* ("State-space" model): Solin **and** Särkkä 2014;
Titsias **and others** 2024

- (uniquement pour les données temporelles)
- Réécriture du GP en équation différentielle stochastique
- Apprentissage: ~~$\mathcal{O}(n^3)$~~ , calcul $\mathcal{O}(n)$

Méthodes d'*échantillonnage* (MCMC, importance sampling, MH, etc):
Rasmussen **and** Williams 2006

- Principe: remplacer $GP(m_*, k_{*,\theta})$ par un échantillonneur peu coûteux
- $m_*, k_{*,\theta}$ restent inconnus et sont estimés en échantillonnant (beaucoup) puis en moyennant (loi forte des grands nombres)

Méthodes de *Kalman* ("State-space" model): Solin **and** Särkkä 2014;
Titsias **and others** 2024

- (uniquement pour les données temporelles)
- Réécriture du GP en équation différentielle stochastique
- Apprentissage: ~~$\mathcal{O}(n^3)$~~ , calcul $\mathcal{O}(n)$
- Réécriture dépend du kernel: état de recherche

Prochain épisode

- *GP* pour les données non-Gaussiennes: méthodes et applications
- Méthodes scalables (apprentissage + inférence) pour les *GP*

Merci pour votre attention.
Questions ?

Références I

-  Borovitskiy, Viacheslav **and others** (2021). “Matérn Gaussian Processes on Graphs”. **in**.
-  Bruinsma, Wessel P **and others** (2020). “Scalable Exact Inference in Multi-Output Gaussian Processes”. **in**.
-  Hensman, James **and others** (2015). “Scalable Variational Gaussian Process Classification”. **in**.
-  Parra, Gabriel **and** Felipe Tobar (2017). “Spectral mixture kernels for multi-output Gaussian processes”. **in**.
-  Rasmussen, Carl Edward **and** Christopher K.I Williams (2006). *Gaussian Processes for Machine Learning*. MIT Press. ISBN: 026218253X.
-  Solin, Arno **and** Simo Särkkä (2014). “Explicit Link Between Periodic Covariance Functions and State Space Models”. **in**.
-  Titsias, Michalis K **and others** (2024). “Kalman Filter for Online Classification of Non-Stationary Data”. **in**.

Références II



Wilson, Andrew Gordon **and others** (2016). “Deep Kernel Learning”.
in.