Special course on Gaussian processes: Session #5

Markus Heinonen
Aalto University
users.aalto.fi/heinom10

Feb 6, 2019

Outline

1. Part 0: Recap
2. Part 1: spectral kernels
3. Part 2: Non-stationary and heteroscedastic GPs

Kernel method

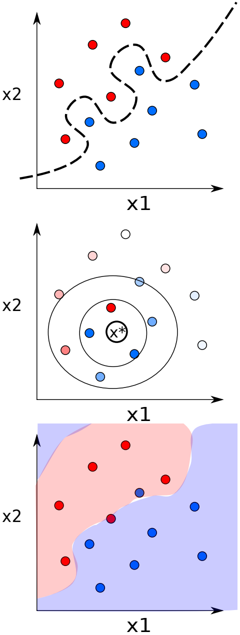
- Kernel ridge regression

$$f(\mathbf{x}^*) = \sum_{i=1}^{N} \underbrace{\alpha_i}_{\text{weight}} \underbrace{K(\mathbf{x}^*, \mathbf{x}_i)}_{\text{similarity}}$$

$$\boldsymbol{\alpha} = (K_{XX} \underbrace{+\lambda I}_{\text{regulariser}})^{-1}\mathbf{y} \quad \in \mathbb{R}^N$$

- Gaussian kernel (similarity)

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



Kernel "trick"

- Why do we get non-linearity?
- Basis expansion

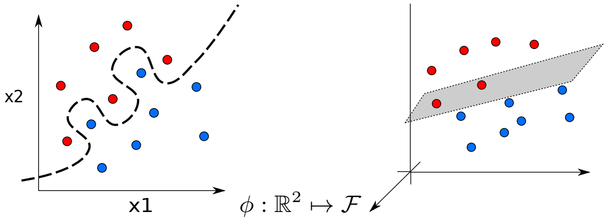
$$K(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$$

  with

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})$$

  - Gaussian kernel considers infinite number of monomials x^i

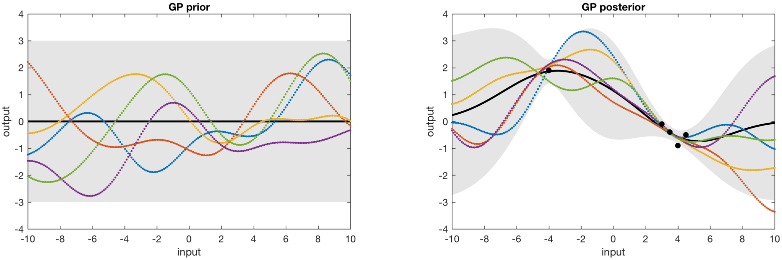
$$\phi_{gauss}(x) = e^{-x^2/2\ell^2} \left[1, \frac{1}{\sqrt{1!\ell^2}} x, \frac{1}{\sqrt{2!\ell^4}} x^2, \ldots \right.$$



Gaussian process prior

- Bayesian non-parametric kernel model for learning from data
- Key idea: function prior $f(x) \sim \mathcal{GP}(m(x), K_\theta(x, x'))$ that encodes

$$
p\begin{pmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{pmatrix} = \mathcal{N}\left(\underbrace{\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix}}_{\mathbf{f}} \middle| \underbrace{\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_N) \end{bmatrix}}_{\mathbf{m}}, \underbrace{\begin{bmatrix} K_\theta(x_1, x_1) & \cdots & K_\theta(1, x_N) \\ \vdots & \ddots & \vdots \\ K_\theta(N, x_1) & \cdots & K_\theta(N, x_N) \end{bmatrix}}_{K_\theta} \right)
$$



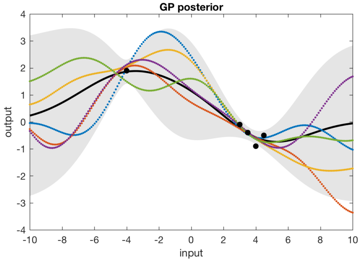
Gaussian process posterior (regression)

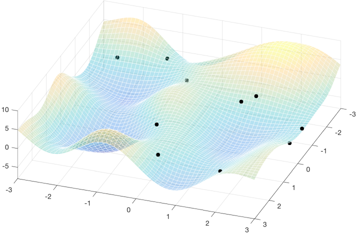
- Observed noisy data values $\mathbf{y} = (y_1, \dots, y_N)$ at N inputs $X = (\mathbf{x}_1, \dots, \mathbf{x}_N)$
- Assume Gaussian likelihood $\mathcal{N}(y_i | f(x_i), \sigma_n^2)$ and prior $f(\mathbf{x}) \sim \mathcal{GP}(0, K_\theta)$
- Posterior $p(\mathbf{f}_\star | \mathbf{y}, X) \sim \mathcal{N}(\boldsymbol{\mu}_\star, \Sigma_\star)$ for $N_\star$ new test points $X_\star = (x_1^\star, \dots, x_{N_\star}^\star)$ with

$$\mathbb{E}[\mathbf{f}_\star | \mathbf{y}, X] = \boldsymbol{\mu}_\star = K(X_\star, X) \underbrace{(K(X, X) + \sigma_n^2 I)^{-1} \mathbf{y}}_{\boldsymbol{\alpha}}$$

$$\text{Cov}[\mathbf{f}_\star | \mathbf{y}, X] = \Sigma_\star = K(X_\star, X_\star) - K(X_\star, X)(K(X, X) + \sigma_n^2 I)^{-1} K(X, X_\star)$$

  - The mean is equal to non-probabilistic kernel regression $f(x) = \sum_i \alpha_i K(x, x_i)$ with $\lambda = \sigma_n^2$
  - GP model "adds variances" to kernel machines





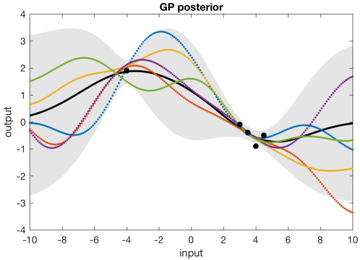
Gaussian process posterior (regression)

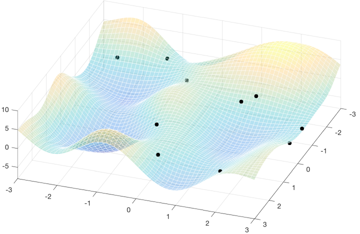
- Observed noisy data values $\mathbf{y} = (y_1, \ldots, y_N)$ at N inputs $X = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$
- Assume Gaussian likelihood $\mathcal{N}(y_i | f(x_i), \sigma_n^2)$ and prior $f(\mathbf{x}) \sim \mathcal{GP}(0, K_\theta)$
- Posterior $p(\mathbf{f}_\star | \mathbf{y}, X) \sim \mathcal{N}(\boldsymbol{\mu}_\star, \Sigma_\star)$ for $N_\star$ new test points $X_\star = (x_1^\star, \ldots, x_{N_\star}^\star)$ with

$$\mathbb{E}[\mathbf{f}_\star | \mathbf{y}, X] = \boldsymbol{\mu}_\star = K(X_\star, X) \underbrace{(K(X, X) + \sigma_n^2 I)^{-1} \mathbf{y}}_{\boldsymbol{\alpha}}$$

$$\text{Cov}[\mathbf{f}_\star | \mathbf{y}, X] = \Sigma_\star = K(X_\star, X_\star) - K(X_\star, X)(K(X, X) + \sigma_n^2 I)^{-1} K(X, X_\star)$$

  - The mean is equal to non-probabilistic kernel regression $f(x) = \sum_i \alpha_i K(x, x_i)$ with $\lambda = \sigma_n^2$
  - GP model "adds variances" to kernel machines





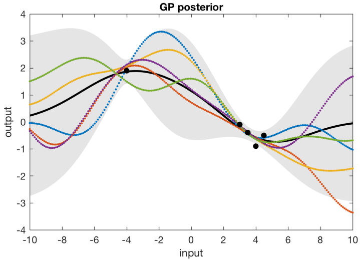
Gaussian process posterior (regression)

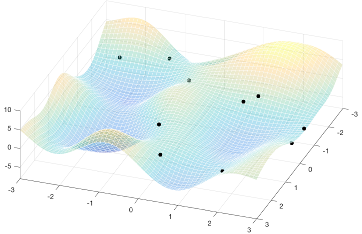
- Observed noisy data values $\mathbf{y} = (y_1, \ldots, y_N)$ at N inputs $X = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$
- Assume Gaussian likelihood $\mathcal{N}(y_i | f(x_i), \sigma_n^2)$ and prior $f(\mathbf{x}) \sim \mathcal{GP}(0, K_\theta)$
- Posterior $p(\mathbf{f}_\star | \mathbf{y}, X) \sim \mathcal{N}(\boldsymbol{\mu}_\star, \Sigma_\star)$ for $N_\star$ new test points $X_\star = (x_1^\star, \ldots, x_{N_\star}^\star)$ with

$$\mathbb{E}[\mathbf{f}_\star | \mathbf{y}, X] = \boldsymbol{\mu}_\star = K(X_\star, X) \underbrace{(K(X, X) + \sigma_n^2 I)^{-1} \mathbf{y}}_{\boldsymbol{\alpha}}$$

$$\text{Cov}[\mathbf{f}_\star | \mathbf{y}, X] = \Sigma_\star = K(X_\star, X_\star) - K(X_\star, X)(K(X, X) + \sigma_n^2 I)^{-1} K(X, X_\star)$$

  - The mean is equal to non-probabilistic kernel regression $f(x) = \sum_i \alpha_i K(x, x_i)$ with $\lambda = \sigma_n^2$
  - GP model "adds variances" to kernel machines





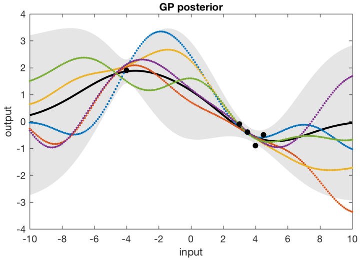
Gaussian process posterior (regression)

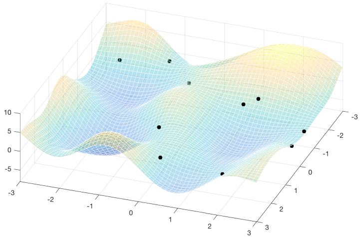
- Observed noisy data values $\mathbf{y} = (y_1, \ldots, y_N)$ at N inputs $X = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$
- Assume Gaussian likelihood $\mathcal{N}(y_i | f(x_i), \sigma_n^2)$ and prior $f(\mathbf{x}) \sim \mathcal{GP}(0, K_\theta)$
- Posterior $p(\mathbf{f}_\star | \mathbf{y}, X) \sim \mathcal{N}(\boldsymbol{\mu}_\star, \Sigma_\star)$ for $N_\star$ new test points $X_\star = (x_1^\star, \ldots, x_{N_\star}^\star)$ with

$$\mathbb{E}[\mathbf{f}_\star | \mathbf{y}, X] = \boldsymbol{\mu}_\star = K(X_\star, X) \underbrace{(K(X, X) + \sigma_n^2 I)^{-1} \mathbf{y}}_{\boldsymbol{\alpha}}$$

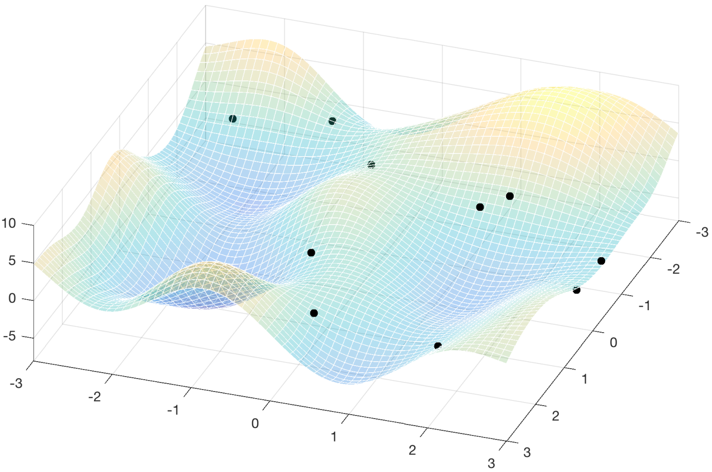
$$\text{Cov}[\mathbf{f}_\star | \mathbf{y}, X] = \Sigma_\star = K(X_\star, X_\star) - K(X_\star, X)(K(X, X) + \sigma_n^2 I)^{-1} K(X, X_\star)$$

- The mean is equal to non-probabilistic kernel regression $f(x) = \sum_i \alpha_i K(x, x_i)$ with $\lambda = \sigma_n^2$
- GP model "adds variances" to kernel machines





2D posterior example



How to learn a kernel?

- Choose a prior with maximum amount of functions that match the data $\mathcal{D}$

$$\log p(\mathbf{y}|\theta) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\theta)d\mathbf{f}$$
$$= -\frac{1}{2}\underbrace{\mathbf{y}^T(K_\theta + \sigma^2 I)^{-1}\mathbf{y}}_{\text{data fit}} - \frac{1}{2}\underbrace{\log|K_\theta + \sigma^2 I|}_{\text{model complexity}} - \frac{N}{2}\log 2\pi$$

- Integral has convenient only with Gaussian likelihoods (ie. regression)
- Non-Gaussian likelihoods warrant eg. variational inference
- Minimizes overfitting
  - Determinant captures the volume of the data cloud in the kernel feature space
  - Finds a simple basis for the data
- Extremely powerful formalism to learn kernels
  - No need for model selection cross-validation
  - We can differentiate $\log p(\mathbf{y}|\theta)$ and apply gradient optimisation for parameters θ

How to learn a kernel?

- Choose a prior with maximum amount of functions that match the data $\mathcal{D}$

$$\log p(\mathbf{y}|\theta) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\theta)d\mathbf{f}$$
$$= -\frac{1}{2}\underbrace{\mathbf{y}^T(K_\theta + \sigma^2 I)^{-1}\mathbf{y}}_{\text{data fit}} - \frac{1}{2}\underbrace{\log|K_\theta + \sigma^2 I|}_{\text{model complexity}} - \frac{N}{2}\log 2\pi$$

- Integral has convenient only with Gaussian likelihoods (ie. regression)
- Non-Gaussian likelihoods warrant eg. variational inference
- Minimizes overfitting
  - Determinant captures the volume of the data cloud in the kernel feature space
  - Finds a simple basis for the data
- Extremely powerful formalism to learn kernels
  - No need for model selection cross-validation
  - We can differentiate $\log p(\mathbf{y}|\theta)$ and apply gradient optimisation for parameters θ

How to learn a kernel?

- Choose a prior with maximum amount of functions that match the data $\mathcal{D}$

$$\log p(\mathbf{y}|\theta) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\theta)d\mathbf{f}$$

$$= -\frac{1}{2}\underbrace{\mathbf{y}^T(K_\theta + \sigma^2 I)^{-1}\mathbf{y}}_{\text{data fit}} - \frac{1}{2}\underbrace{\log|K_\theta + \sigma^2 I|}_{\text{model complexity}} - \frac{N}{2}\log 2\pi$$

- Integral has convenient only with Gaussian likelihoods (ie. regression)
- Non-Gaussian likelihoods warrant eg. variational inference
- Minimizes overfitting
  - Determinant captures the volume of the data cloud in the kernel feature space
  - Finds a simple basis for the data
- Extremely powerful formalism to learn kernels
  - No need for model selection cross-validation
  - We can differentiate $\log p(\mathbf{y}|\theta)$ and apply gradient optimisation for parameters θ

Recap (regression setting)

1. Gaussian process prior on inputs $\mathbf{x} \in \mathbb{R}^D$, output $y \in \mathbb{R}$,

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), K(\mathbf{x}, \mathbf{x}')) \tag{1}$$
$$\Leftrightarrow \tag{2}$$
$$p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{m}, K_{XX}) \tag{3}$$
$$\mathbb{E}[f(\mathbf{x})] = m(\mathbf{x}) \tag{4}$$
$$\textbf{cov}[f(\mathbf{x}), f(\mathbf{x}')] = K(\mathbf{x}, \mathbf{x}') \tag{5}$$

for inputs $X = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T \in \mathbb{R}^{N \times D}$, functions $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))^T \in \mathbb{R}^N$ and means $\mathbf{m} = (m(\mathbf{x}_1), \ldots, m(\mathbf{x}_N))^T \in \mathbb{R}^N$,

2. Predictive (regression) posterior $f(\mathbf{x})|(X, \mathbf{y}) \sim \mathcal{N}(\mu(\mathbf{x}), \sigma(\mathbf{x})^2)$

$$\mu(\mathbf{x}) = K_{\mathbf{x}X}(K_{XX} + \sigma_n^2 I_N)^{-1}\mathbf{y} \tag{6}$$
$$\sigma(\mathbf{x})^2 = K_{\mathbf{x}\mathbf{x}} - K_{\mathbf{x}X}(K_{XX} + \sigma_n^2 I_N)^{-1}K_{X\mathbf{x}} \tag{7}$$

3. Optimization criteria ('loss function') for hyperparameters θ

$$p(\mathbf{y}|\theta) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\theta)d\mathbf{x} = \mathcal{N}(\mathbf{y}|\mathbf{0}, K_\theta(X, X) + \sigma_n^2 I_N)$$

Recap (regression setting)

1. Gaussian process prior on inputs $\mathbf{x} \in \mathbb{R}^D$, output $y \in \mathbb{R}$,

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), K(\mathbf{x}, \mathbf{x}')) \tag{1}$$
$$\Leftrightarrow \tag{2}$$
$$p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{m}, K_{XX}) \tag{3}$$
$$\mathbb{E}[f(\mathbf{x})] = m(\mathbf{x}) \tag{4}$$
$$\textbf{cov}[f(\mathbf{x}), f(\mathbf{x}')] = K(\mathbf{x}, \mathbf{x}') \tag{5}$$

for inputs $X = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T \in \mathbb{R}^{N \times D}$, functions $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))^T \in \mathbb{R}^N$ and means $\mathbf{m} = (m(\mathbf{x}_1), \ldots, m(\mathbf{x}_N))^T \in \mathbb{R}^N$,

2. Predictive (regression) posterior $f(\mathbf{x})|(X, \mathbf{y}) \sim \mathcal{N}(\mu(\mathbf{x}), \sigma(\mathbf{x})^2)$

$$\mu(\mathbf{x}) = K_{\mathbf{x}X}(K_{XX} + \sigma_n^2 I_N)^{-1}\mathbf{y} \tag{6}$$
$$\sigma(\mathbf{x})^2 = K_{\mathbf{xx}} - K_{\mathbf{x}X}(K_{XX} + \sigma_n^2 I_N)^{-1} K_{X\mathbf{x}} \tag{7}$$

3. Optimization criteria ('loss function') for hyperparameters θ

$$p(\mathbf{y}|\theta) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\theta)d\mathbf{x} = \mathcal{N}(\mathbf{y}|\mathbf{0}, K_\theta(X, X) + \sigma_n^2 I_N)$$

Recap (regression setting)

1. Gaussian process prior on inputs $\mathbf{x} \in \mathbb{R}^D$, output $y \in \mathbb{R}$,

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), K(\mathbf{x}, \mathbf{x}')) \tag{1}$$

$$\Leftrightarrow \tag{2}$$

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{m}, K_{XX}) \tag{3}$$

$$\mathbb{E}[f(\mathbf{x})] = m(\mathbf{x}) \tag{4}$$

$$\textbf{cov}[f(\mathbf{x}), f(\mathbf{x}')] = K(\mathbf{x}, \mathbf{x}') \tag{5}$$

   for inputs $X = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T \in \mathbb{R}^{N \times D}$, functions $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))^T \in \mathbb{R}^N$ and means $\mathbf{m} = (m(\mathbf{x}_1), \ldots, m(\mathbf{x}_N))^T \in \mathbb{R}^N$,

2. Predictive (regression) posterior $f(\mathbf{x})|(X, \mathbf{y}) \sim \mathcal{N}(\mu(\mathbf{x}), \sigma(\mathbf{x})^2)$

$$\mu(\mathbf{x}) = K_{\mathbf{x}X}(K_{XX} + \sigma_n^2 I_N)^{-1}\mathbf{y} \tag{6}$$

$$\sigma(\mathbf{x})^2 = K_{\mathbf{x}\mathbf{x}} - K_{\mathbf{x}X}(K_{XX} + \sigma_n^2 I_N)^{-1} K_{X\mathbf{x}} \tag{7}$$

3. Optimization criteria ('loss function') for hyperparameters θ

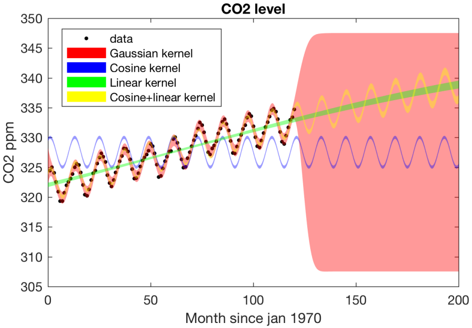
$$p(\mathbf{y}|\theta) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\theta)d\mathbf{x} = \mathcal{N}(\mathbf{y}|\mathbf{0}, K_\theta(X, X) + \sigma_n^2 I_N)$$

Outline

1. Part 0: Recap
2. Part 1: spectral kernels
3. Part 2: Non-stationary and heteroscedastic GPs

Which kernel to choose?

- Gaussian kernel $K_g(x,x') = \exp\left(-\frac{(x-x')^2}{2\ell^2}\right)$
- Periodic kernel $K_{cos}(x,x') = \exp\left(-\frac{2\sin^2(\pi|x-x'|/p}{\ell^2})\right)$
- Linear kernel $K_{lin}(x,x') = xx' + c$
- Kernel sum $K(x,x') = K_g(x,x') + K_{lin}(x,x')$



- Spectral kernels can learn arbitrary kernel forms
  - The topic of today's lecture

Fourier transforms

- Fourier transform $S(\omega)$ of a function $f(x)$,

$$S(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \omega} dx$$

  where
  - i is the imaginary number with $i^2 = -1$ and $i^0 = 1$
  - ω is a frequency
- Inverse Fourier transform $f(x)$ of spectral density $S(\omega)$,

$$f(x) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i x \omega} d\omega$$

- Euler's identity helps compute Fouriers in practise

$$e^{ix} = \underbrace{\cos x}_{\text{real part}} + \underbrace{i \cdot sinx}_{\text{complex part}}$$

  where the complex part is often designed to cancel out (or simply ignored)
- Hence,

$$e^{-2\pi i x \omega} = \cos(2\pi x \omega) - i \sin(2\pi x \omega)$$
$$e^{2\pi i x \omega} = \cos(2\pi x \omega) + i \sin(2\pi x \omega)$$

Fourier transforms

- Fourier transform $S(\omega)$ of a function $f(x)$,

$$S(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \omega} dx$$

  where
  - i is the imaginary number with $i^2 = -1$ and $i^0 = 1$
  - ω is a frequency
- Inverse Fourier transform $f(x)$ of spectral density $S(\omega)$,

$$f(x) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i x \omega} d\omega$$

- Euler's identity helps compute Fouriers in practise

$$e^{ix} = \underbrace{\cos x}_{\text{real part}} + \underbrace{i \cdot sinx}_{\text{complex part}}$$

  where the complex part is often designed to cancel out (or simply ignored)
- Hence,

$$e^{-2\pi i x \omega} = \cos(2\pi x \omega) - i \sin(2\pi x \omega)$$

$$e^{2\pi i x \omega} = \cos(2\pi x \omega) + i \sin(2\pi x \omega)$$

Fourier transforms

- Fourier transform $S(\omega)$ of a function $f(x)$,

$$S(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \omega} dx$$

  where
  - i is the imaginary number with $i^2 = -1$ and $i^0 = 1$
  - ω is a frequency
- Inverse Fourier transform $f(x)$ of spectral density $S(\omega)$,

$$f(x) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i x \omega} d\omega$$

- Euler's identity helps compute Fouriers in practise

$$e^{ix} = \underbrace{\cos x}_{\text{real part}} + \underbrace{i \cdot sinx}_{\text{complex part}}$$

  where the complex part is often designed to cancel out (or simply ignored)
- Hence,

$$e^{-2\pi i x \omega} = \cos(2\pi x \omega) - i \sin(2\pi x \omega)$$
$$e^{2\pi i x \omega} = \cos(2\pi x \omega) + i \sin(2\pi x \omega)$$

Fourier transforms

- Fourier transform $S(\omega)$ of a function $f(x)$,

$$S(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \omega} dx$$

where

  - i is the imaginary number with $i^2 = -1$ and $i^0 = 1$
  - ω is a frequency

- Inverse Fourier transform $f(x)$ of spectral density $S(\omega)$,

$$f(x) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i x \omega} d\omega$$

- Euler's identity helps compute Fouriers in practise

$$e^{ix} = \underbrace{\cos x}_{\text{real part}} + \underbrace{i \cdot sinx}_{\text{complex part}}$$

where the complex part is often designed to cancel out (or simply ignored)

- Hence,

$$e^{-2\pi i x \omega} = \cos(2\pi x \omega) - i \sin(2\pi x \omega)$$
$$e^{2\pi i x \omega} = \cos(2\pi x \omega) + i \sin(2\pi x \omega)$$

Fourier duals

- Let's apply Fouriers to the function $K(\tau) \equiv K(x - x') = K(x, x')$, where $\tau = x - x'$

Theorem (Bochner)

Any stationary kernel $K : \mathbb{R}^D \mapsto \mathbb{R}$ and its spectral density $S : \mathbb{R}^D \mapsto \mathbb{R}$ are Fourier duals

$$K(x - x') \equiv K(\tau) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i \omega^T \tau} d\omega \qquad \text{(Inverse Fourier Transform)}$$

$$S(\omega) = \int_{-\infty}^{\infty} K(\tau) e^{-2\pi i \omega^T \tau} d\tau, \qquad \text{(Fourier Transform)}$$

where $\tau = \mathbf{x} - \mathbf{x}'$.

1. All stationary kernels have spectral density $S(\omega)$ where ω is a frequency
   - If someone gives you a kernel $K(\tau)$, we can solve what frequencies it considers by solving the (FT)
   - Studying known kernel's frequency representations usually of theoretical interest
2. All spectral densities define a covariance function $K(\tau)$
   - If someone gives you a spectral density $S(\omega)$, we can solve its similarity function (=kernel) by solving the (IFT)
   - If we change the spectral density, we get a new kernel
   - $\Rightarrow$ kernel learning (!)

Fourier duals

- Let's apply Fouriers to the function $K(\tau) \equiv K(x - x') = K(x, x')$, where $\tau = x - x'$

Theorem (Bochner)

Any stationary kernel $K : \mathbb{R}^D \mapsto \mathbb{R}$ and its spectral density $S : \mathbb{R}^D \mapsto \mathbb{R}$ are Fourier duals

$$K(x - x') \equiv K(\tau) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i \omega^T \tau} d\omega \qquad \text{(Inverse Fourier Transform)}$$

$$S(\omega) = \int_{-\infty}^{\infty} K(\tau) e^{-2\pi i \omega^T \tau} d\tau, \qquad \text{(Fourier Transform)}$$

where $\tau = \mathbf{x} - \mathbf{x}'$.

1. All stationary kernels have spectral density $S(\omega)$ where ω is a frequency
   - If someone gives you a kernel $K(\tau)$, we can solve what frequencies it considers by solving the (FT)
   - Studying known kernel's frequency representations usually of theoretical interest
2. All spectral densities define a covariance function $K(\tau)$
   - If someone gives you a spectral density $S(\omega)$, we can solve its similarity function (=kernel) by solving the (IFT)
   - If we change the spectral density, we get a new kernel
   - $\Rightarrow$ kernel learning (!)

Fourier duals

- Let's apply Fouriers to the function $K(\tau) \equiv K(x - x') = K(x, x')$, where $\tau = x - x'$

Theorem (Bochner)

Any stationary kernel $K : \mathbb{R}^D \mapsto \mathbb{R}$ and its spectral density $S : \mathbb{R}^D \mapsto \mathbb{R}$ are Fourier duals

$$K(x - x') \equiv K(\tau) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i \omega^T \tau} d\omega \qquad \text{(Inverse Fourier Transform)}$$

$$S(\omega) = \int_{-\infty}^{\infty} K(\tau) e^{-2\pi i \omega^T \tau} d\tau, \qquad \text{(Fourier Transform)}$$

where $\tau = \mathbf{x} - \mathbf{x}'$.

1. All stationary kernels have spectral density $S(\omega)$ where ω is a frequency
   - If someone gives you a kernel $K(\tau)$, we can solve what frequencies it considers by solving the (FT)
   - Studying known kernel's frequency representations usually of theoretical interest
2. All spectral densities define a covariance function $K(\tau)$
   - If someone gives you a spectral density $S(\omega)$, we can solve its similarity function (=kernel) by solving the (IFT)
   - If we change the spectral density, we get a new kernel
   - $\Rightarrow$ kernel learning (!)

Fourier duals

- Let's apply Fouriers to the function $K(\tau) \equiv K(x - x') = K(x, x')$, where $\tau = x - x'$

Theorem (Bochner)

Any stationary kernel $K : \mathbb{R}^D \mapsto \mathbb{R}$ and its spectral density $S : \mathbb{R}^D \mapsto \mathbb{R}$ are Fourier duals

$$K(x - x') \equiv K(\tau) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i \omega^T \tau} d\omega \qquad \text{(Inverse Fourier Transform)}$$

$$S(\omega) = \int_{-\infty}^{\infty} K(\tau) e^{-2\pi i \omega^T \tau} d\tau, \qquad \text{(Fourier Transform)}$$

where $\tau = \mathbf{x} - \mathbf{x}'$.

1. All stationary kernels have spectral density $S(\omega)$ where ω is a frequency
   - If someone gives you a kernel $K(\tau)$, we can solve what frequencies it considers by solving the (FT)
   - Studying known kernel's frequency representations usually of theoretical interest
2. All spectral densities define a covariance function $K(\tau)$
   - If someone gives you a spectral density $S(\omega)$, we can solve its similarity function (=kernel) by solving the (IFT)
   - If we change the spectral density, we get a new kernel
   - $\Rightarrow$ kernel learning (!)

Kernel sinusoid representation

- Assume symmetric frequency distribution $S(\omega) = S(-\omega)$
- Euler's identity $e^{\pm ix} = \cos x \pm i \sin x$
- Sine identity $\sin(-x) = -\sin(x)$
- Then we can solve the inverse Fourier as

$$\begin{aligned}
K(\tau) &= \int_{-\infty}^{\infty} S(\omega) e^{2\pi i \tau \omega} d\omega \\
&= \int_{-\infty}^{\infty} S(\omega) \cos(2\pi\tau\omega) d\omega + \int_{-\infty}^{\infty} i \cdot S(\omega) \sin(2\pi\tau\omega) d\omega \\
&= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) + \int_{-\infty}^{0} i \cdot S(\omega) \sin(2\pi\tau\omega) d\omega + \int_{0}^{\infty} i \cdot S(\omega) \sin(2\pi\tau\omega) d\omega \\
&= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) + \int_{0}^{\infty} iS(-\omega) \sin(2\pi\tau(-\omega)) d\omega + \int_{0}^{\infty} iS(\omega) \sin(2\pi\tau\omega) d\omega \\
&= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) + \int_{0}^{\infty} -iS(\omega) \sin(2\pi\tau\omega) d\omega + \int_{0}^{\infty} iS(\omega) \sin(2\pi\tau\omega) d\omega \\
&= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega)
\end{aligned}$$

- Hence, all stationary kernels are $S(\omega)$-weighted combinations of sinusoids $\cos(2\pi\tau\omega)$

Kernel sinusoid representation

- Assume symmetric frequency distribution $S(\omega) = S(-\omega)$
- Euler's identity $e^{\pm ix} = \cos x \pm i \sin x$
- Sine identity $\sin(-x) = -\sin(x)$
- Then we can solve the inverse Fourier as

$$\begin{aligned}
K(\tau) &= \int_{-\infty}^{\infty} S(\omega) e^{2\pi i \tau \omega} d\omega \\
&= \int_{-\infty}^{\infty} S(\omega) \cos(2\pi\tau\omega) d\omega + \int_{-\infty}^{\infty} i \cdot S(\omega) \sin(2\pi\tau\omega) d\omega \\
&= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) + \int_{-\infty}^{0} i \cdot S(\omega) \sin(2\pi\tau\omega) d\omega + \int_{0}^{\infty} i \cdot S(\omega) \sin(2\pi\tau\omega) d\omega \\
&= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) + \int_{0}^{\infty} iS(-\omega) \sin(2\pi\tau(-\omega)) d\omega + \int_{0}^{\infty} iS(\omega) \sin(2\pi\tau\omega) d\omega \\
&= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) + \int_{0}^{\infty} -iS(\omega) \sin(2\pi\tau\omega) d\omega + \int_{0}^{\infty} iS(\omega) \sin(2\pi\tau\omega) d\omega \\
&= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega)
\end{aligned}$$

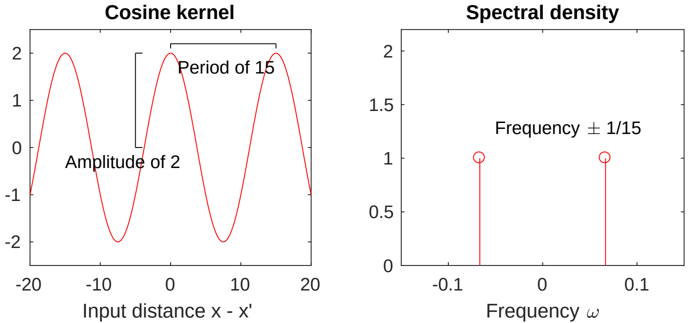
- Hence, all stationary kernels are $S(\omega)$-weighted combinations of sinusoids $\cos(2\pi\tau\omega)$

Kernel sinusoid representation

- General kernel definition

$$K(\tau) = \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega)$$

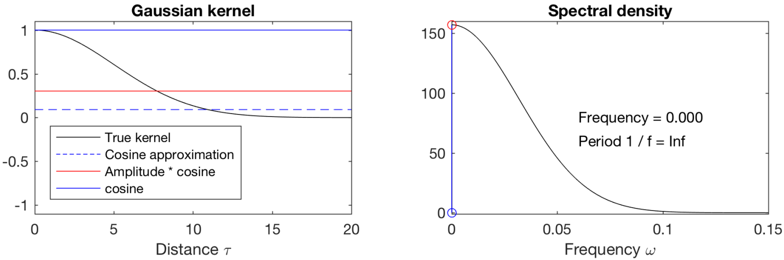
- Frequency ω is inverse of period $1/\omega$
- Frequencies are symmetric $S(\omega) = S(-\omega)$
- With $S(\omega) = \delta_{1/15}(\omega)$, the kernel becomes $K(\tau) = \cos(2\pi\tau\frac{1}{15})$



Gaussian kernel sinusoids

- Gaussian kernel $K_{SE}(\tau) = \exp(-\tau^2/\ell^2)$ fourier representation

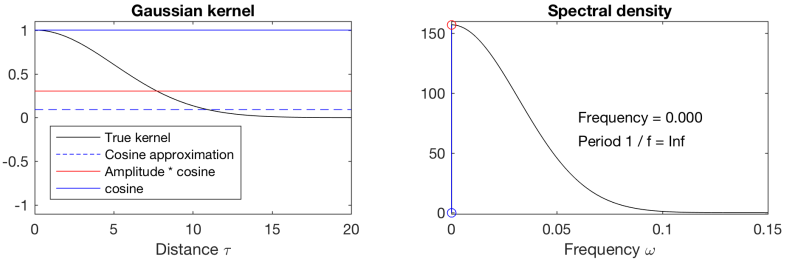
$$S_{SE}(\omega) = \int_{-\infty}^{\infty} K_{SE}(\tau) e^{-2\pi i \omega^T \tau} d\tau$$
$$= 2\pi\ell^2 \exp(-2\pi^2\ell^2\omega^2)$$
$$K_{SE}(\tau) = \int_0^{\infty} \underbrace{S_{SE}(\omega)}_{\text{amplitudes}} \cdot \underbrace{\cos(2\pi\tau\omega)}_{\text{sinusoids}} d\omega$$
$$\approx \sum_{\omega} S_{SE}(\omega) \cdot \cos(2\pi\tau\omega)$$



Gaussian kernel sinusoids

- Gaussian kernel $K_{SE}(\tau) = \exp(-\tau^2/\ell^2)$ fourier representation

$$S_{SE}(\omega) = \int_{-\infty}^{\infty} K_{SE}(\tau) e^{-2\pi i \omega^T \tau} d\tau$$
$$= 2\pi\ell^2 \exp(-2\pi^2\ell^2\omega^2)$$
$$K_{SE}(\tau) = \int_0^{\infty} \underbrace{S_{SE}(\omega)}_{\text{amplitudes}} \cdot \underbrace{\cos(2\pi\tau\omega)}_{\text{sinusoids}} d\omega$$
$$\approx \sum_{\omega} S_{SE}(\omega) \cdot \cos(2\pi\tau\omega)$$

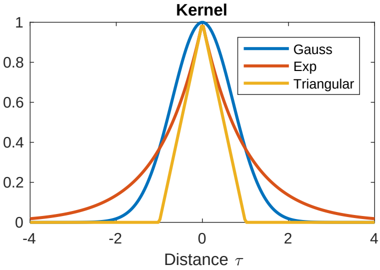


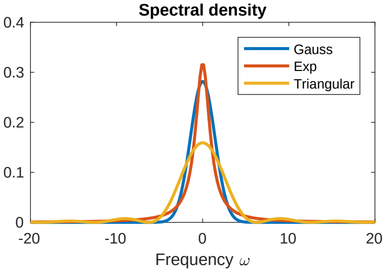
Some spectral densities

$$K_{gauss}(\tau) = \exp(-\frac{\tau^2}{\ell^2}) \qquad S_{gauss}(\omega) = \frac{\sqrt{\ell}}{2\sqrt{\pi}} \exp(-\ell\omega^2/4)$$

$$K_{exp}(\tau) = \exp(-|\tau|/\ell) \qquad S_{exp}(\omega) = 1/(\pi/\ell + \pi\ell\omega^2)$$

$$K_{tri}(\tau) = 0.5(1 - |\tau|)_+ \qquad S_{tri}(\omega) = (1 - \cos\omega)/(\pi\omega^2)$$





- Can we construct new kernels from custom spectral densities?

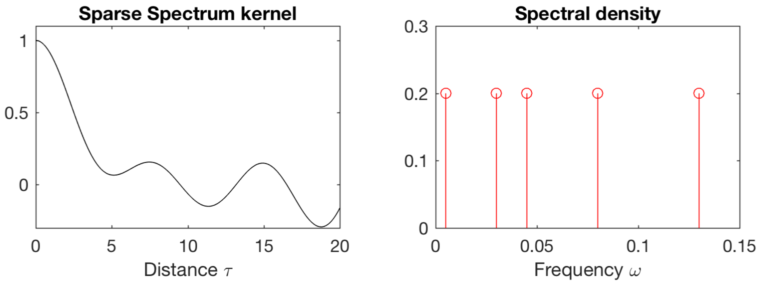
Lazaro-Gredilla: Sparse Spectrum (SS) kernel

- Define Q real frequencies $(\omega_1, \ldots, \omega_Q)^T \in \mathbb{R}^Q$ with Fourier dual[1]

$$S(\omega) := \frac{1}{Q} \sum_{i=1}^{Q} \delta(\omega = \omega_i)$$

$$\Rightarrow K(\tau) = \frac{1}{Q} \sum_{i=1}^{Q} \cos(2\pi\tau\omega_i)$$

- Highly structured covariance, no decay, prone to overfitting



[1] Lazaro-Gredilla, Quinonero-Candela, Rasmussen, Figueiras-Vida (JMLR 2010) Sparse spectrum gaussian process regression

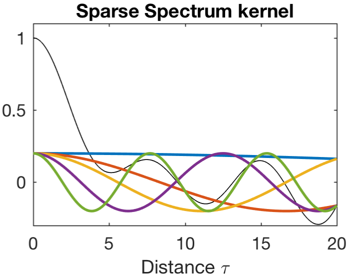
Lazaro-Gredilla: Sparse Spectrum (SS) kernel

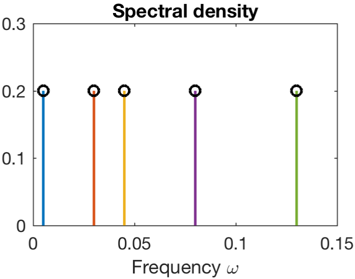
- Define Q real frequencies $(\omega_1, \ldots, \omega_Q)^T \in \mathbb{R}^Q$ with Fourier dual[1]

$$S(\omega) := \frac{1}{Q} \sum_{i=1}^{Q} \delta(\omega = \omega_i)$$

$$\Rightarrow K(\tau) = \frac{1}{Q} \sum_{i=1}^{Q} \cos(2\pi\tau\omega_i)$$

- Highly structured covariance, no decay, prone to overfitting





[1] Lazaro-Gredilla, Quinonero-Candela, Rasmussen, Figueiras-Vida (JMLR 2010) Sparse spectrum gaussian process regression

Wilson: Spectral Mixture (SM) kernel

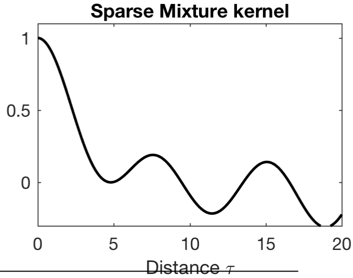
- Define mixture of Q Gaussians $\{a_i \mathcal{N}(\mu_i, \sigma_i^2)\}_{i=1}^{Q}$ [2]

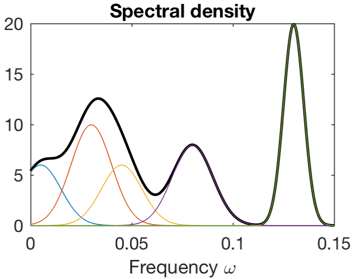
$$S(\omega) := \sum_{i=1}^{Q} a_i \mathcal{N}(\omega | \mu_i, \sigma_i^2)$$

$$\Rightarrow K(\tau) = \int_{-\infty}^{\infty} S(\omega) \cos(2\pi\tau\omega) d\omega$$

$$= \sum_{i=1}^{Q} a_i \underbrace{\exp(-2\pi^2 \sigma_i^2 \tau^2)}_{\text{smooth decay}} \underbrace{\cos(2\pi\tau\mu_i)}_{\text{periodic}}$$

- Dense in the set of stationary kernels $\Rightarrow$ can generate **any** stationary kernel





[2]Wilson, Adams (ICML 2013) Gaussian process kernels for pattern discovery and extrapolation

Wilson: Spectral Mixture (SM) kernel

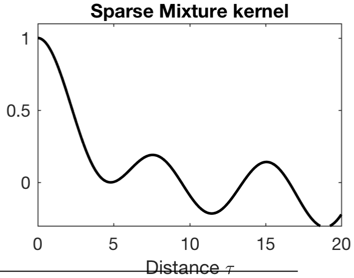
- Define mixture of Q Gaussians $\{a_i \mathcal{N}(\mu_i, \sigma_i^2)\}_{i=1}^{Q}$ [2]

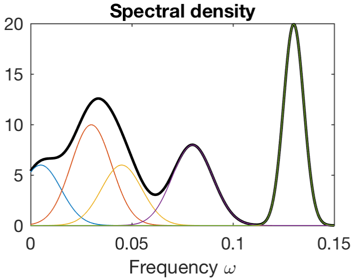
$$S(\omega) := \sum_{i=1}^{Q} a_i \mathcal{N}(\omega | \mu_i, \sigma_i^2)$$

$$\Rightarrow K(\tau) = \int_{-\infty}^{\infty} S(\omega) \cos(2\pi\tau\omega) d\omega$$

$$= \sum_{i=1}^{Q} a_i \underbrace{\exp(-2\pi^2 \sigma_i^2 \tau^2)}_{\text{smooth decay}} \underbrace{\cos(2\pi\tau\mu_i)}_{\text{periodic}}$$

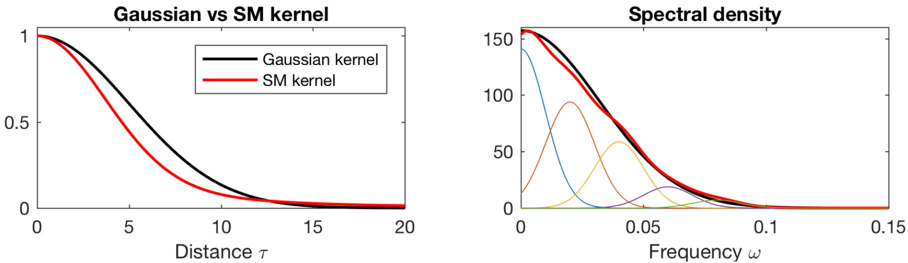
- Dense in the set of stationary kernels $\Rightarrow$ can generate **any** stationary kernel





[2] Wilson, Adams (ICML 2013) Gaussian process kernels for pattern discovery and extrapolation

Wilson: Spectral Mixture (SM) kernel

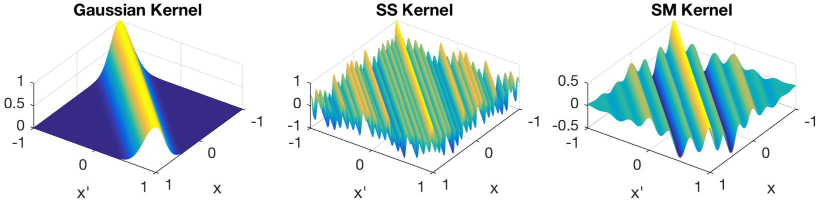


- Approximate gaussian kernel with SM kernel with $Q = 5$ components, i.e.

$$\sum_{i=1}^{Q} a_i \exp(-2\pi^2 \sigma_i^2 \tau^2) \cos(2\pi\tau\mu_i) \approx \exp\left(\frac{(x - x')^2}{2\ell^2}\right)$$

for certain a_i, μ_i, σ_i

Spectral kernels



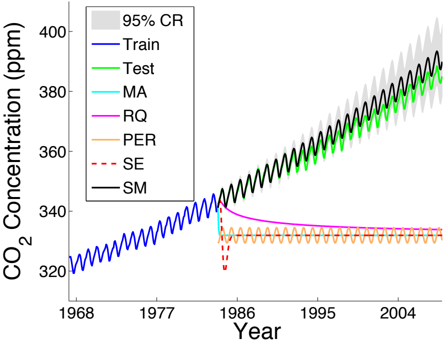
- Image from Remes, Heinonen, Kaski: Non-stationary spectral kernels, NIPS'17

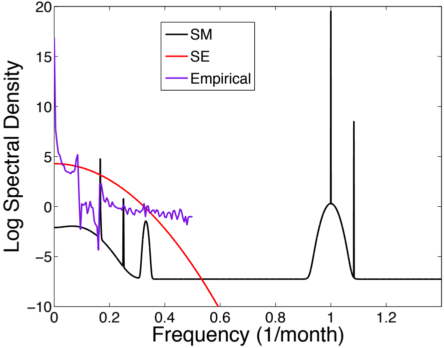
SM kernel inference

- Optimize $3Q$ hyperparameters $\theta = \{a_i, \mu_i, \sigma_i\}_{i=1}^{Q}$ of kernel $K_\theta(x - x') = \sum_{i=1}^{Q} a_i \exp(-2\pi^2 {\sigma_i}^2 \tau^2) \cos(2\pi\tau\mu_i)$ by maximizing

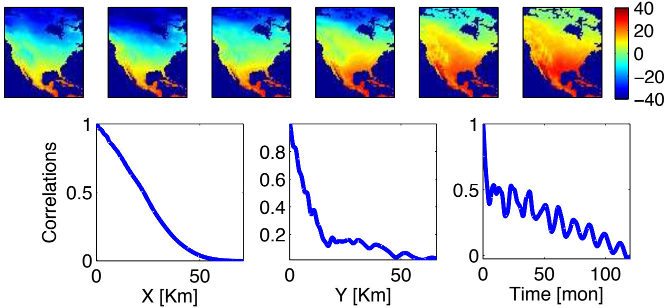
$$\log p(\mathbf{y}|\theta) = -\frac{1}{2} \underbrace{\mathbf{y}^T (K_\theta + \sigma^2 I)^{-1} \mathbf{y}}_{\text{data fit}} - \frac{1}{2} \underbrace{\log |K_\theta + \sigma^2 I|}_{\text{model complexity}} - \frac{N}{2} \log 2\pi$$

- After kernel is fixed, predictions have closed form

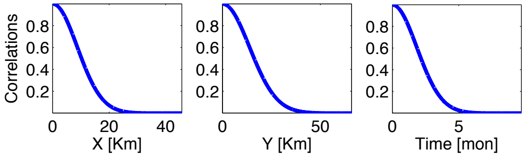




Spatio-temporal temperatures



(a) Learned GPatt Kernel for Temperatures



(b) Learned GP-SE Kernel for Temperatures

- SM kernel induces only stationary covariances, but temperatures are non-stationary

Outline

1. Part 0: Recap
2. Part 1: spectral kernels
3. Part 2: Non-stationary and heteroscedastic GPs

Heteroscedastic Gaussian process

- Standard Gaussian process assumes additive zero-mean noise model

$$y(\mathbf{x}) = f(\mathbf{x}) + \varepsilon(\cancel{\mathbf{x}}) \tag{8}$$

$$\varepsilon(\cancel{\mathbf{x}}) \sim \mathcal{N}(0, \sigma_n^2) \tag{9}$$

where all noises are zero mean with constant variance σ_n^2

- Heteroscedastic model assumes input-dependent noise:

$$\varepsilon(\mathbf{x}) \sim \mathcal{N}(0, \sigma_n(\mathbf{x})^2)$$

- More complex (non-Gaussian) noise models are sometimes used
- The function $\sigma_n(\mathbf{x})^2$ can be another Gaussian process (!)

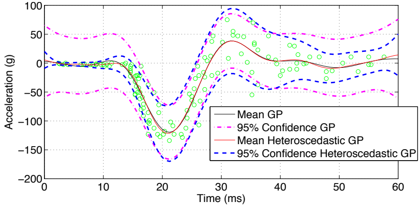


Figure 1. Silverman's (1985) motorcycle benchmark is an example for input dependent noise. It consists of a sequence of accelerometer readings through time following a simulated motor-cycle crash.

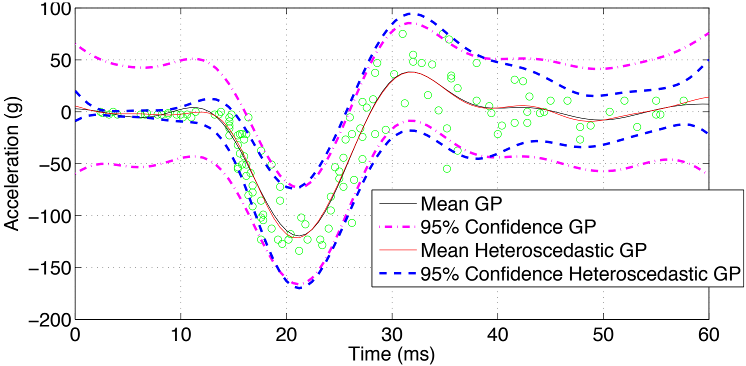


Figure 1. Silverman's (1985) motorcycle benchmark is an example for input dependent noise. It consists of a sequence of accelerometer readings through time following a simulated motor-cycle crash.

[3] Kersting et al (2007): Most Likely Heteroscedastic Gaussian process regression

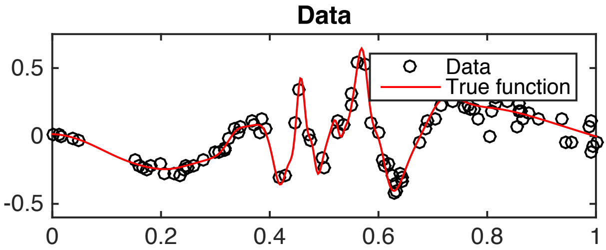
Stationary kernels

- Stationary kernels are translation-invariant:

$$K(x, x') = K(x + a, x' + a) \tag{10}$$
$$K(x, x') = K(x - x') \tag{11}$$

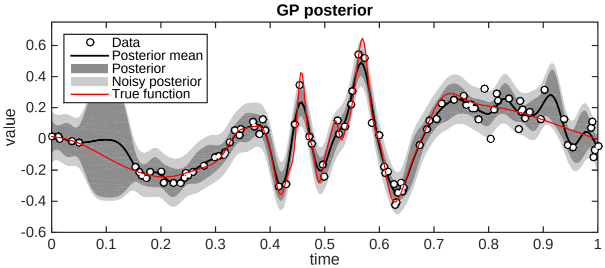
  for any a
  - Stationary kernels are function of vector distance $x - x'$
  - For instance if input variable is 'age' in years, then a stationary kernel has property $K(1, 2) = K(80, 81)$
  - Strange to assume that 1 and 2 year olds are as similar to each other as 80 and 81 year olds
- Non-stationary kernel is not translation invariant, i.e. we can have $K(1, 2) \neq K(80, 81)$
- Simplest non-stationary kernel is the dot product, $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}$ since
  - $\mathbf{x} = [1, 1]^T$, $\mathbf{x}' = [2, 2]$, $K(\mathbf{x}, \mathbf{x}') = 1 \cdot 2 + 1 \cdot 2 = 4$
  - $\mathbf{x} = [10, 10]^T$, $\mathbf{x}' = [11, 11]$, $K(\mathbf{x}, \mathbf{x}') = 10 \cdot 11 + 10 \cdot 11 = 120$

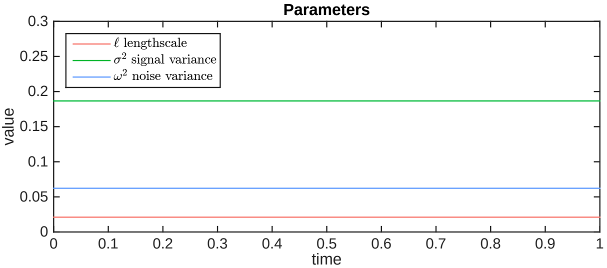
Problem with stationary functions



- Simple dataset

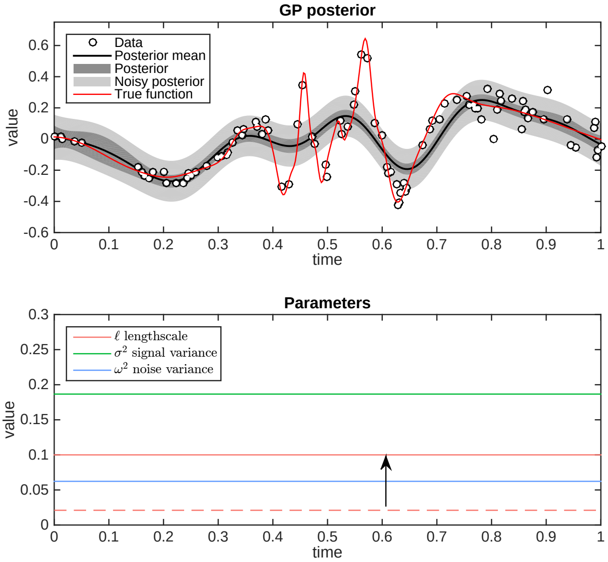
Problem with stationary functions





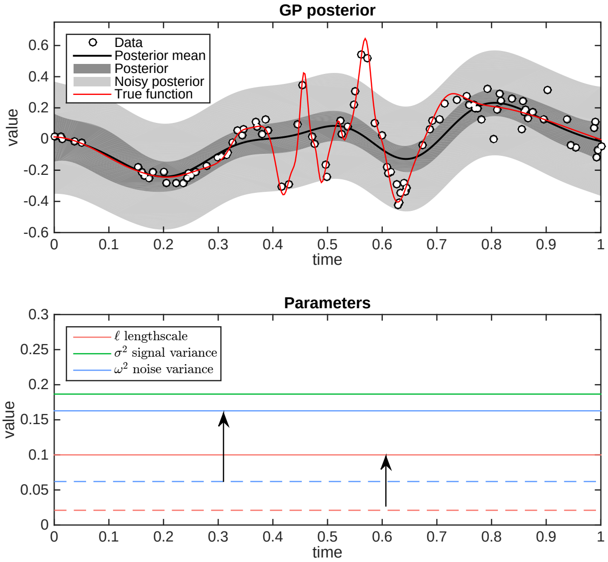
- Optimal Gaussian process fit
- Bad fit in the beginning

Problem with stationary functions



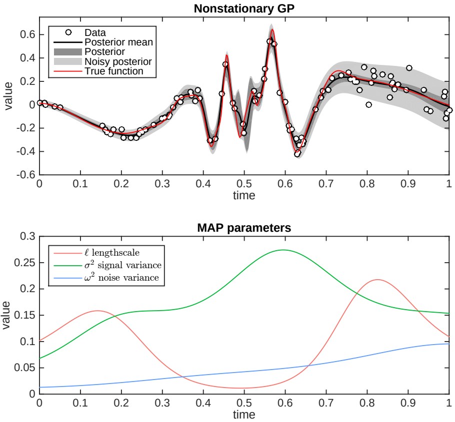
- Let's increase lengthscale to get smoother model
- Initial fit fixed, now ill fit in the middle

Problem with stationary functions



- Let's increase noise level to to match data
- $\Rightarrow$ We need input-dependent parameters

Non-stationary solution[4]



- Function process

$$y(x) = f(x) + \varepsilon(x)$$
$$f(x) \sim \mathcal{GP}(0, \sigma(x)\sigma(x')K_{\ell(\cdot)}(x, x'))$$
$$\varepsilon(x) \sim \mathcal{N}(0, \omega(x)^2)$$

- Parameter processes

$$\ell(x) \sim \mathcal{GP}(\mu_\ell, K_\ell(x, x'))$$
$$\sigma(x) \sim \mathcal{GP}(\mu_\sigma, K_\sigma(x, x'))$$
$$\omega(x) \sim \mathcal{GP}(\mu_\omega, K_\omega(x, x'))$$

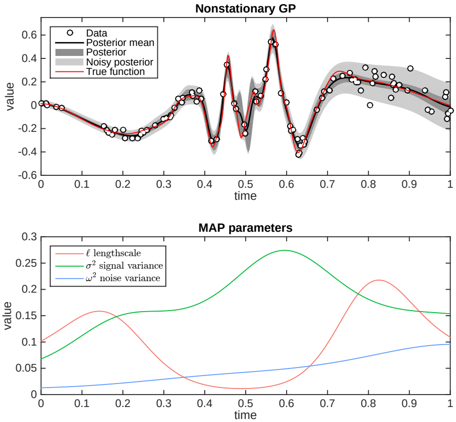
- Kernel

$$K(x, x') = \sqrt{\frac{2\ell(x)\ell(x')}{\ell(x)^2 + \ell(x')^2}} \exp\left(-\frac{(x - x')^2}{\ell(x)^2 + \ell(x')^2}\right)$$

- Explicit **function** representation through smoothness, scale and noise functions

[4] Heinonen et al. Non-stationary Gaussian process regression with Hamiltonian Monte Carlo. AISTATS 2016

Non-stationary inference

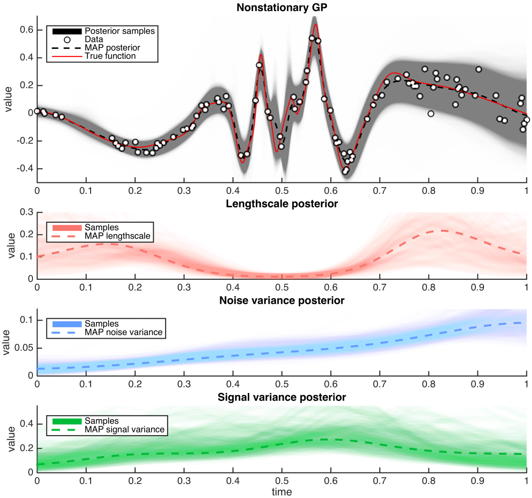


- Marginal joint likelihood

$$\mathcal{L} = p(\mathbf{y}, \boldsymbol{\ell}, \boldsymbol{\omega}, \boldsymbol{\sigma}) = p(\mathbf{y}|\boldsymbol{\ell}, \boldsymbol{\omega}, \boldsymbol{\sigma})p(\boldsymbol{\ell})p(\boldsymbol{\sigma})p(\boldsymbol{\omega})$$
$$= \mathcal{N}(\mathbf{y}|\mathbf{0}, \boldsymbol{\sigma}\boldsymbol{\sigma}^T \circ K_{\boldsymbol{\ell}} + diag(\boldsymbol{\omega}))\mathcal{N}(\boldsymbol{\ell}|\mu_\ell, K_\ell)\mathcal{N}(\boldsymbol{\sigma}|\mu_\sigma, K_\sigma)\mathcal{N}(\boldsymbol{\omega}|\mu_\omega, K_\omega)$$

- We optimize $\mathcal{L}$ for MAP estimates $\hat{\boldsymbol{\ell}}, \hat{\boldsymbol{\sigma}}, \hat{\boldsymbol{\omega}}$.
- The predictive posterior $p(\mathbf{f}|\hat{\boldsymbol{\ell}}, \hat{\boldsymbol{\sigma}}, \hat{\boldsymbol{\omega}}, \mathbf{y})$ is of standard form, except our kernel is $\hat{\boldsymbol{\sigma}}\hat{\boldsymbol{\sigma}}^T \circ K_{\hat{\boldsymbol{\ell}}}$

Inference



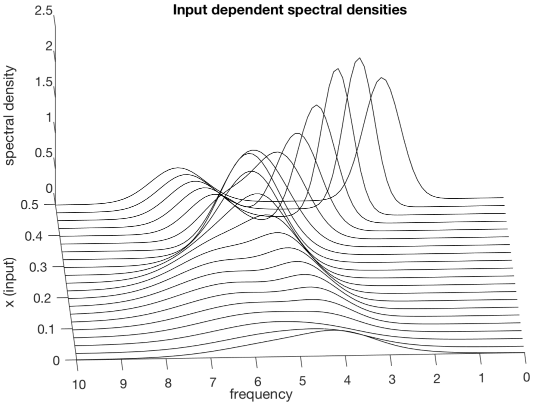
- Sample exact posterior with HMC[5]

$$p(\mathbf{f}, \boldsymbol{\ell}, \boldsymbol{\sigma}, \boldsymbol{\omega}; \mathbf{y})$$

[5] Heinonen et al. Non-stationary Gaussian process regression with Hamiltonian Monte Carlo. AISTATS 2016

Non-stationary spectral kernels

- We have seen how to learn arbitrary stationary kernels via spectral learning
- We have seen how to learn (non-stationary) Gaussian kernel with parameter functions
- What about non-stationary spectral kernels?
- Model input-dependent frequencies, or spectrograms $S(x, \omega)$
  - E.g. wavelets are time-dependent frequencies in signal processing



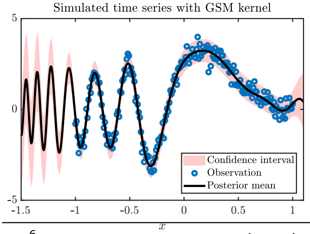
Generalised Spectral Mixture (GSM) kernel[6][7]

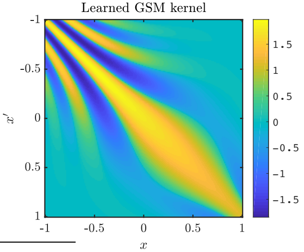
- Non-stationary spectral kernel can be derived:

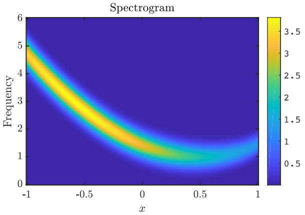
$$K_{\mathbf{w},\boldsymbol{\mu},\boldsymbol{\sigma}}(x,x') \propto \sum_{i=1}^{Q} w_i(x)w_i(x') \underbrace{\exp\left(-\frac{(x-x')^2}{\ell_i(x)^2+\ell_i(x')^2)}\right)}_{\text{Exponential kernel}} \underbrace{\cos(2\pi(\mu_i(x)x-\mu_i(x')x'))}_{\text{periodic}}$$

with

$$\log w_i(x) \sim \mathcal{GP}(0, K_w)$$
$$\log \mu_i(x) \sim \mathcal{GP}(0, K_\mu)$$
$$\log \ell_i(x) \sim \mathcal{GP}(0, K_\sigma)$$







[6] Remes, Heinonen, Kaski (2017): Non-stationary spectral kernels

[7] Shen, Heinonen, Kaski (2019): Harmonizable mixture kernels with variational Fourier features

Summary

- Performance of GP has crucial dependency on how well the kernel matches the data
- Gaussian kernel is a convenient 'default' kernel that can interpolate well
  - Advantage: simple, efficient, easy-to-learn, universal
  - Disadvantage: cannot fit periodic data, stationary only
- Spectral kernels can extrapolate repeating patterns
  - Advantage: can learn arbitrary periodic or non-periodic stationary patterns
  - Disadvantage: slower to learn, high possibility to overfit
- Non-stationary Gaussian kernel can learn adaptive interpolations
  - Advantage: can learn smoothly changing smoothness / variance
  - Disadvantage: slower to learn, more possibilities to overfit
- Non-stationary spectral kernels can learn rich frequency representations
  - Advantage: can learn smoothly changing smoothness / variance
  - Disadvantage: complex modelling of the kernel, computer intensive optimization, major risk of overfitting
  - Active research field

Summary

- Performance of GP has crucial dependency on how well the kernel matches the data
- Gaussian kernel is a convenient 'default' kernel that can interpolate well
  - Advantage: simple, efficient, easy-to-learn, universal
  - Disadvantage: cannot fit periodic data, stationary only
- Spectral kernels can extrapolate repeating patterns
  - Advantage: can learn arbitrary periodic or non-periodic stationary patterns
  - Disadvantage: slower to learn, high possibility to overfit
- Non-stationary Gaussian kernel can learn adaptive interpolations
  - Advantage: can learn smoothly changing smoothness / variance
  - Disadvantage: slower to learn, more possibilities to overfit
- Non-stationary spectral kernels can learn rich frequency representations
  - Advantage: can learn smoothly changing smoothness / variance
  - Disadvantage: complex modelling of the kernel, computer intensive optimization, major risk of overfitting
  - Active research field

Summary

- Performance of GP has crucial dependency on how well the kernel matches the data
- Gaussian kernel is a convenient 'default' kernel that can interpolate well
  - Advantage: simple, efficient, easy-to-learn, universal
  - Disadvantage: cannot fit periodic data, stationary only
- Spectral kernels can extrapolate repeating patterns
  - Advantage: can learn arbitrary periodic or non-periodic stationary patterns
  - Disadvantage: slower to learn, high possibility to overfit
- Non-stationary Gaussian kernel can learn adaptive interpolations
  - Advantage: can learn smoothly changing smoothness / variance
  - Disadvantage: slower to learn, more possibilities to overfit
- Non-stationary spectral kernels can learn rich frequency representations
  - Advantage: can learn smoothly changing smoothness / variance
  - Disadvantage: complex modelling of the kernel, computer intensive optimization, major risk of overfitting
  - Active research field

Summary

- Performance of GP has crucial dependency on how well the kernel matches the data
- Gaussian kernel is a convenient 'default' kernel that can interpolate well
  - Advantage: simple, efficient, easy-to-learn, universal
  - Disadvantage: cannot fit periodic data, stationary only
- Spectral kernels can extrapolate repeating patterns
  - Advantage: can learn arbitrary periodic or non-periodic stationary patterns
  - Disadvantage: slower to learn, high possibility to overfit
- Non-stationary Gaussian kernel can learn adaptive interpolations
  - Advantage: can learn smoothly changing smoothness / variance
  - Disadvantage: slower to learn, more possibilities to overfit
- Non-stationary spectral kernels can learn rich frequency representations
  - Advantage: can learn smoothly changing smoothness / variance
  - Disadvantage: complex modelling of the kernel, computer intensive optimization, major risk of overfitting
  - Active research field