CS-E4895 Gaussian processes
Lecture 5: Kernel learning

Markus Heinonen

Aalto University

Monday 13.3.2023

Agenda for today

1. Recap
2. What is a kernel? Which kernel to choose?
3. Structured kernels
4. Spectral kernels
5. Non-stationary kernels

Agenda for today

1. Recap
2. What is a kernel? Which kernel to choose?
3. Structured kernels
4. Spectral kernels
5. Non-stationary kernels

Recap: Gaussian process regression pipeline

1. Vector inputs $\mathbf{x} \in \mathbb{R}^D$, real outputs $y \in \mathbb{R}$
2. GP function prior

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big) \Rightarrow \begin{cases} \mathbb{E}[f(\mathbf{x})] = m(\mathbf{x}) \\ \mathbf{cov}[f(\mathbf{x}), f(\mathbf{x}')] = k(\mathbf{x}, \mathbf{x}') \\ p(\mathbf{f}) \sim \mathcal{N}(\mathbf{m}_X, \mathbf{K}_{XX}) \end{cases} \tag{1}$$

3. Data $(\mathbf{x}_i, y_i)_{i=1}^N = (X, \mathbf{y})$ observation model

$$y = f(\mathbf{x}) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2_{\text{obs}}) \tag{2}$$

4. Joint distribution

$$p\begin{pmatrix} \mathbf{y} \\ \mathbf{f}_* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{m}_X \\ \mathbf{m}_* \end{pmatrix}, \begin{pmatrix} \mathbf{K}_{XX} + \sigma^2_{\text{obs}} I & \mathbf{K}_{XX_*} \\ \mathbf{K}_{X_*X} & \mathbf{K}_{X_*X_*} \end{pmatrix}\right) \tag{3}$$

5. Predictive posterior

$$p(\mathbf{f}_*|\mathbf{y}) \sim \mathcal{N}\Big(\underbrace{\mathbf{K}_{X_*X}(\mathbf{K}_{XX} + \sigma_n^2 I_N)^{-1}(\mathbf{y} - \mathbf{m}_X) + \mathbf{m}_*}_{\mu_*}, \underbrace{\mathbf{K}_{X_*X_*} - \mathbf{K}_{X_*X}(\mathbf{K}_{XX} + \sigma_n^2 I_N)^{-1}\mathbf{K}_{XX_*}}_{\Sigma_*}\Big) \tag{4}$$

6. Marginal likelihood to optimise hyperparameters

$$\max \quad p(\mathbf{y}; \theta) = \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\theta) d\mathbf{x} = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{XX} + \sigma^2_{\text{obs}} I) \tag{5}$$

Recap: Gaussian process regression pipeline

1. Vector inputs $\mathbf{x} \in \mathbb{R}^D$, real outputs $y \in \mathbb{R}$
2. GP function prior

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big) \Rightarrow \begin{cases} \mathbb{E}[f(\mathbf{x})] = m(\mathbf{x}) \\ \mathbf{cov}[f(\mathbf{x}), f(\mathbf{x}')] = k(\mathbf{x}, \mathbf{x}') \\ p(\mathbf{f}) \sim \mathcal{N}(\mathbf{m}_X, \mathbf{K}_{XX}) \end{cases} \tag{1}$$

3. Data $(\mathbf{x}_i, y_i)_{i=1}^N = (X, \mathbf{y})$ observation model

$$y = f(\mathbf{x}) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2_{\text{obs}}) \tag{2}$$

4. Joint distribution

$$p\begin{pmatrix}\mathbf{y} \\ \mathbf{f}_*\end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix}\mathbf{m}_X \\ \mathbf{m}_*\end{pmatrix}, \begin{pmatrix}\mathbf{K}_{XX} + \sigma^2_{\text{obs}} I & \mathbf{K}_{XX_*} \\ \mathbf{K}_{X_*X} & \mathbf{K}_{X_*X_*}\end{pmatrix}\right) \tag{3}$$

5. Predictive posterior

$$p(\mathbf{f}_*|\mathbf{y}) \sim \mathcal{N}\Big(\underbrace{\mathbf{K}_{X_*X}(\mathbf{K}_{XX} + \sigma_n^2 I_N)^{-1}(\mathbf{y} - \mathbf{m}_X) + \mathbf{m}_*}_{\mu_*}, \underbrace{\mathbf{K}_{X_*X_*} - \mathbf{K}_{X_*X}(\mathbf{K}_{XX} + \sigma_n^2 I_N)^{-1}\mathbf{K}_{XX_*}}_{\Sigma_*}\Big) \tag{4}$$

6. Marginal likelihood to optimise hyperparameters

$$\max \quad p(\mathbf{y}; \theta) = \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\theta) d\mathbf{x} = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{XX} + \sigma^2_{\text{obs}} I) \tag{5}$$

Recap: Gaussian process regression pipeline

1. Vector inputs $\mathbf{x} \in \mathbb{R}^D$, real outputs $y \in \mathbb{R}$
2. GP function prior

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big) \Rightarrow \begin{cases} \mathbb{E}[f(\mathbf{x})] = m(\mathbf{x}) \\ \textbf{cov}[f(\mathbf{x}), f(\mathbf{x}')] = k(\mathbf{x}, \mathbf{x}') \\ p(\mathbf{f}) \sim \mathcal{N}(\mathbf{m}_X, \mathbf{K}_{XX}) \end{cases} \tag{1}$$

3. Data $(\mathbf{x}_i, y_i)_{i=1}^N = (X, \mathbf{y})$ observation model

$$y = f(\mathbf{x}) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2_{\text{obs}}) \tag{2}$$

4. Joint distribution

$$p\begin{pmatrix}\mathbf{y} \\ \mathbf{f}_*\end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix}\mathbf{m}_X \\ \mathbf{m}_*\end{pmatrix}, \begin{pmatrix}\mathbf{K}_{XX} + \sigma^2_{\text{obs}} I & \mathbf{K}_{XX_*} \\ \mathbf{K}_{X_*X} & \mathbf{K}_{X_*X_*}\end{pmatrix}\right) \tag{3}$$

5. Predictive posterior

$$p(\mathbf{f}_*|\mathbf{y}) \sim \mathcal{N}\Big(\underbrace{\mathbf{K}_{X_*X}(\mathbf{K}_{XX} + \sigma_n^2 I_N)^{-1}(\mathbf{y} - \mathbf{m}_X) + \mathbf{m}_*}_{\mu_*}, \underbrace{\mathbf{K}_{X_*X_*} - \mathbf{K}_{X_*X}(\mathbf{K}_{XX} + \sigma_n^2 I_N)^{-1}\mathbf{K}_{XX_*}}_{\Sigma_*}\Big) \tag{4}$$

6. Marginal likelihood to optimise hyperparameters

$$\max \quad p(\mathbf{y}; \theta) = \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\theta) d\mathbf{x} = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{XX} + \sigma^2_{\text{obs}} I) \tag{5}$$

Recap: Gaussian process regression pipeline

1. Vector inputs $\mathbf{x} \in \mathbb{R}^D$, real outputs $y \in \mathbb{R}$
2. GP function prior

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big) \Rightarrow \begin{cases} \mathbb{E}[f(\mathbf{x})] = m(\mathbf{x}) \\ \textbf{cov}[f(\mathbf{x}), f(\mathbf{x}')] = k(\mathbf{x}, \mathbf{x}') \\ p(\mathbf{f}) \sim \mathcal{N}(\mathbf{m}_X, \mathbf{K}_{XX}) \end{cases} \tag{1}$$

3. Data $(\mathbf{x}_i, y_i)_{i=1}^N = (X, \mathbf{y})$ observation model

$$y = f(\mathbf{x}) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2_{\text{obs}}) \tag{2}$$

4. Joint distribution

$$p\begin{pmatrix}\mathbf{y} \\ \mathbf{f}_*\end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix}\mathbf{m}_X \\ \mathbf{m}_*\end{pmatrix}, \begin{pmatrix}\mathbf{K}_{XX} + \sigma^2_{\text{obs}} I & \mathbf{K}_{XX_*} \\ \mathbf{K}_{X_*X} & \mathbf{K}_{X_*X_*}\end{pmatrix}\right) \tag{3}$$

5. Predictive posterior

$$p(\mathbf{f}_*|\mathbf{y}) \sim \mathcal{N}\Big(\underbrace{\mathbf{K}_{X_*X}(\mathbf{K}_{XX} + \sigma_n^2 I_N)^{-1}(\mathbf{y} - \mathbf{m}_X) + \mathbf{m}_*}_{\mu_*}, \underbrace{\mathbf{K}_{X_*X_*} - \mathbf{K}_{X_*X}(\mathbf{K}_{XX} + \sigma_n^2 I_N)^{-1}\mathbf{K}_{XX_*}}_{\Sigma_*}\Big) \tag{4}$$

6. Marginal likelihood to optimise hyperparameters

$$\max \quad p(\mathbf{y}; \theta) = \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\theta) d\mathbf{x} = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{XX} + \sigma^2_{\text{obs}} I) \tag{5}$$

Recap: Gaussian process regression pipeline

1. Vector inputs $\mathbf{x} \in \mathbb{R}^D$, real outputs $y \in \mathbb{R}$
2. GP function prior

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big) \Rightarrow \begin{cases} \mathbb{E}[f(\mathbf{x})] = m(\mathbf{x}) \\ \mathbf{cov}[f(\mathbf{x}), f(\mathbf{x}')] = k(\mathbf{x}, \mathbf{x}') \\ p(\mathbf{f}) \sim \mathcal{N}(\mathbf{m}_X, \mathbf{K}_{XX}) \end{cases} \tag{1}$$

3. Data $(\mathbf{x}_i, y_i)_{i=1}^N = (X, \mathbf{y})$ observation model

$$y = f(\mathbf{x}) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2_{\text{obs}}) \tag{2}$$

4. Joint distribution

$$p\begin{pmatrix}\mathbf{y} \\ \mathbf{f}_*\end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix}\mathbf{m}_X \\ \mathbf{m}_*\end{pmatrix}, \begin{pmatrix}\mathbf{K}_{XX} + \sigma^2_{\text{obs}} I & \mathbf{K}_{XX_*} \\ \mathbf{K}_{X_*X} & \mathbf{K}_{X_*X_*}\end{pmatrix}\right) \tag{3}$$

5. Predictive posterior

$$p(\mathbf{f}_*|\mathbf{y}) \sim \mathcal{N}\Big(\underbrace{\mathbf{K}_{X_*X}(\mathbf{K}_{XX} + \sigma_n^2 I_N)^{-1}(\mathbf{y} - \mathbf{m}_X) + \mathbf{m}_*}_{\mu_*}, \underbrace{\mathbf{K}_{X_*X_*} - \mathbf{K}_{X_*X}(\mathbf{K}_{XX} + \sigma_n^2 I_N)^{-1}\mathbf{K}_{XX_*}}_{\Sigma_*}\Big) \tag{4}$$

6. Marginal likelihood to optimise hyperparameters

$$\max \quad p(\mathbf{y}; \theta) = \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\theta) d\mathbf{x} = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{XX} + \sigma^2_{\text{obs}} I) \tag{5}$$

Recap: Gaussian process regression pipeline

1. Vector inputs $\mathbf{x} \in \mathbb{R}^D$, real outputs $y \in \mathbb{R}$
2. GP function prior

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big) \Rightarrow \begin{cases} \mathbb{E}[f(\mathbf{x})] = m(\mathbf{x}) \\ \mathbf{cov}[f(\mathbf{x}), f(\mathbf{x}')] = k(\mathbf{x}, \mathbf{x}') \\ p(\mathbf{f}) \sim \mathcal{N}(\mathbf{m}_X, \mathbf{K}_{XX}) \end{cases} \tag{1}$$

3. Data $(\mathbf{x}_i, y_i)_{i=1}^N = (X, \mathbf{y})$ observation model

$$y = f(\mathbf{x}) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \sigma^2_{\text{obs}}) \tag{2}$$

4. Joint distribution

$$p\begin{pmatrix}\mathbf{y} \\ \mathbf{f}_*\end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix}\mathbf{m}_X \\ \mathbf{m}_*\end{pmatrix}, \begin{pmatrix}\mathbf{K}_{XX} + \sigma^2_{\text{obs}} I & \mathbf{K}_{XX_*} \\ \mathbf{K}_{X_*X} & \mathbf{K}_{X_*X_*}\end{pmatrix}\right) \tag{3}$$

5. Predictive posterior

$$p(\mathbf{f}_*|\mathbf{y}) \sim \mathcal{N}\Big(\underbrace{\mathbf{K}_{X_*X}(\mathbf{K}_{XX} + \sigma_n^2 I_N)^{-1}(\mathbf{y} - \mathbf{m}_X) + \mathbf{m}_*}_{\mu_*}, \underbrace{\mathbf{K}_{X_*X_*} - \mathbf{K}_{X_*X}(\mathbf{K}_{XX} + \sigma_n^2 I_N)^{-1}\mathbf{K}_{XX_*}}_{\Sigma_*}\Big) \tag{4}$$

6. Marginal likelihood to optimise hyperparameters

$$\max \quad p(\mathbf{y}; \theta) = \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\theta) d\mathbf{x} = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{XX} + \sigma^2_{\text{obs}} I) \tag{5}$$

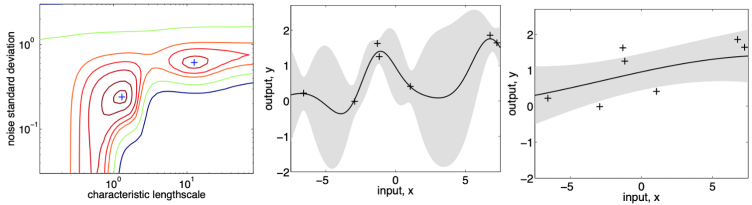
How to find hyperparameters?

- Marginal likelihood: Choose a prior with maximum amount of functions that match the data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$

$$\log p(\mathbf{y}|\theta) = \log \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\theta) d\mathbf{f} \tag{6}$$

$$= -\frac{1}{2} \underbrace{\mathbf{y}^T (K_\theta + \sigma^2 I)^{-1} \mathbf{y}}_{\text{data fit}} - \frac{1}{2} \underbrace{\log |K_\theta + \sigma^2 I|}_{\text{model complexity}} - \frac{N}{2} \log 2\pi \tag{7}$$

- Relatively robust against overfitting
- We can optimise kernel hyperparameters, and choose between kernel function forms by MLL
- No need for model selection cross-validation
-
- We can optimise with gradients $\nabla \log p(\mathbf{y}|\theta)$ (from autodiff)



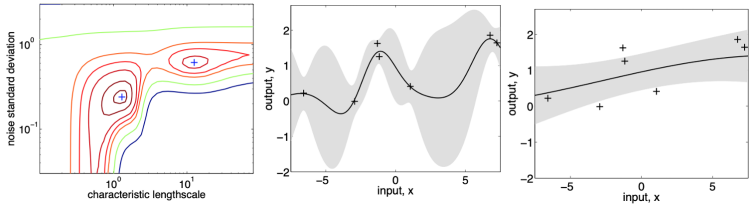
How to find hyperparameters?

- Marginal likelihood: Choose a prior with maximum amount of functions that match the data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$

$$\log p(\mathbf{y}|\theta) = \log \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\theta) d\mathbf{f} \tag{6}$$

$$= -\frac{1}{2} \underbrace{\mathbf{y}^T (K_\theta + \sigma^2 I)^{-1} \mathbf{y}}_{\text{data fit}} - \frac{1}{2} \underbrace{\log |K_\theta + \sigma^2 I|}_{\text{model complexity}} - \frac{N}{2} \log 2\pi \tag{7}$$

- Relatively robust against overfitting
- We can optimise kernel hyperparameters, and choose between kernel function forms by MLL
- No need for model selection cross-validation
-
- We can optimise with gradients $\nabla \log p(\mathbf{y}|\theta)$ (from autodiff)



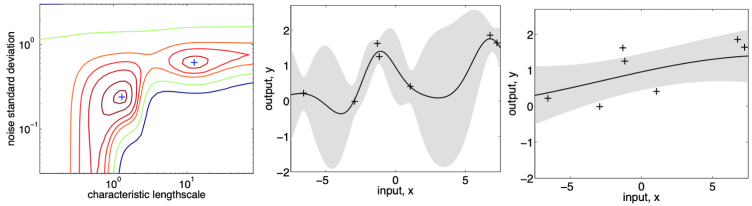
How to find hyperparameters?

- Marginal likelihood: Choose a prior with maximum amount of functions that match the data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$

$$\log p(\mathbf{y}|\theta) = \log \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\theta) d\mathbf{f} \tag{6}$$

$$= -\frac{1}{2} \underbrace{\mathbf{y}^T (K_\theta + \sigma^2 I)^{-1} \mathbf{y}}_{\text{data fit}} - \frac{1}{2} \underbrace{\log |K_\theta + \sigma^2 I|}_{\text{model complexity}} - \frac{N}{2} \log 2\pi \tag{7}$$

- Relatively robust against overfitting
- We can optimise kernel hyperparameters, and choose between kernel function forms by MLL
- No need for model selection cross-validation
-
- We can optimise with gradients $\nabla \log p(\mathbf{y}|\theta)$ (from autodiff)



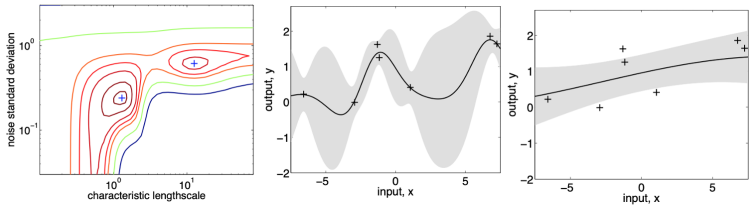
How to find hyperparameters?

- Marginal likelihood: Choose a prior with maximum amount of functions that match the data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$

$$\log p(\mathbf{y}|\theta) = \log \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\theta) d\mathbf{f} \tag{6}$$

$$= -\frac{1}{2} \underbrace{\mathbf{y}^T (K_\theta + \sigma^2 I)^{-1} \mathbf{y}}_{\text{data fit}} - \frac{1}{2} \underbrace{\log |K_\theta + \sigma^2 I|}_{\text{model complexity}} - \frac{N}{2} \log 2\pi \tag{7}$$

- Relatively robust against overfitting
- We can optimise kernel hyperparameters, and choose between kernel function forms by MLL
- No need for model selection cross-validation
-
- We can optimise with gradients $\nabla \log p(\mathbf{y}|\theta)$ (from autodiff)



Agenda for today

1. Recap
2. What is a kernel? Which kernel to choose?
3. Structured kernels
4. Spectral kernels
5. Non-stationary kernels

What is a kernel?

- Kernel is a covariance function

$$f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')) \quad \Rightarrow \quad \textbf{cov}[f(\mathbf{x}), f(\mathbf{x}')] = k(\mathbf{x}, \mathbf{x}') \tag{8}$$

  - Kernel is symmetric, $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$
  - Kernel is positive semidefinite (PSD)

$$\sum_i \sum_j k(\mathbf{x}_i, \mathbf{x}_j) c_i c_j \geq 0, \qquad \forall \mathbf{x} \in \mathbb{R}^D, c \in \mathbb{R} \tag{9}$$

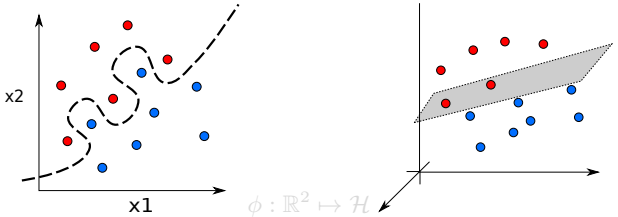
  - Non-negative eigenvalues
- A stationary kernel is a function of difference $\mathbf{x} - \mathbf{x}'$, ie. $k(\mathbf{x} - \mathbf{x}')$
  - Gaussian kernel is stationary: $\exp(-1/2\ell^2 || \mathbf{x} - \mathbf{x}' ||^2)$
- An isotropic kernel is a function of distance $|\mathbf{x} - \mathbf{x}'|$ (also radial basis function)
  - Gaussian kernel is isotropic: $\exp(-1/2\ell^2 \, ||\mathbf{x} - \mathbf{x}'||^2 \,)$

- Let's study a quadratic kernel over 2D inputs $\mathbf{a}, \mathbf{b} \in \mathbb{R}^2$

$$k(\mathbf{a}, \mathbf{b}) = (\mathbf{a}^T\mathbf{b})^2 \tag{10}$$
$$= (a_1 b_1 + a_2 b_2)^2 \tag{11}$$
$$= a_1^2 b_1^2 + 2a_1 a_2 b_1 b_2 + a_2^2 b_2^2 \tag{12}$$
$$= \underbrace{(a_1^2, \sqrt{2}a_1 a_2, a_2^2)}_{\phi(\mathbf{a})^T} \underbrace{(b_1^2, \sqrt{2}b_1 b_2, b_2^2)^T}_{\phi(\mathbf{b})}, \tag{13}$$

where $\phi(\mathbf{a}) = (a_1^2, \sqrt{2}a_1 a_2, a_2^2) \in \mathbb{R}^3$

- Linear in $\mathbb{R}^3$
- Non-linear in $\mathbb{R}^2$

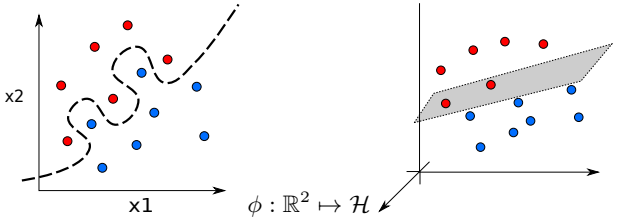


What is a kernel: Kernel trick

- Let's study a quadratic kernel over 2D inputs $\mathbf{a}, \mathbf{b} \in \mathbb{R}^2$

$$k(\mathbf{a}, \mathbf{b}) = (\mathbf{a}^T \mathbf{b})^2 \tag{10}$$
$$= (a_1 b_1 + a_2 b_2)^2 \tag{11}$$
$$= a_1^2 b_1^2 + 2 a_1 a_2 b_1 b_2 + a_2^2 b_2^2 \tag{12}$$
$$= \underbrace{(a_1^2, \sqrt{2} a_1 a_2, a_2^2)}_{\phi(\mathbf{a})^T} \underbrace{(b_1^2, \sqrt{2} b_1 b_2, b_2^2)^T}_{\phi(\mathbf{b})}, \tag{13}$$

where $\phi(\mathbf{a}) = (a_1^2, \sqrt{2} a_1 a_2, a_2^2) \in \mathbb{R}^3$
- Linear in $\mathbb{R}^3$
- Non-linear in $\mathbb{R}^2$



What is a kernel: Kernel trick II

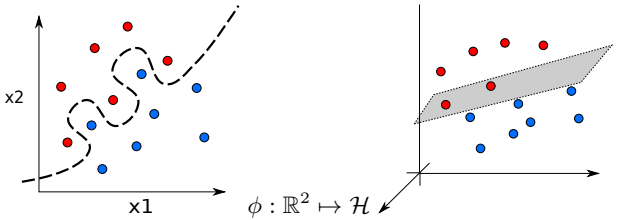
- Basis expansion ('Reproducing kernel Hilbert space, RKHS, Rasmussen 6.1.)

$$k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle. \tag{14}$$

- Example: Gaussian kernel considers infinite number of monomials x^i

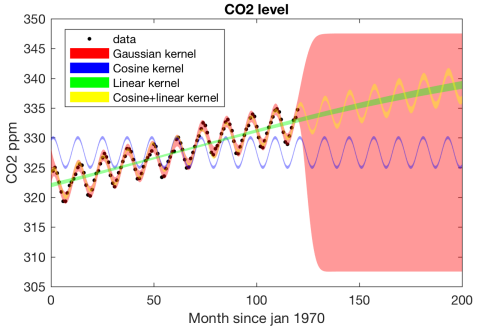
$$\phi_{\mathrm{SE}}(x) = e^{-x^2/2\ell^2} \left[1, \frac{1}{\sqrt{1!\ell^2}} x, \frac{1}{\sqrt{2!\ell^4}} x^2, \dots \right. \tag{15}$$

- Take-away: kernels extract features (implicitly)
- Take-away: there is a space where kernel is linear



How to choose a kernel?

- A kernel dictates the function space
- Properties of interest
  - Function smoothness
    - SE kernel is infinitely differentiable
    - Matern-p kernel is $\lfloor p \rfloor$-differentiable
  - Periodic kernels
  - Non-stationary kernels
- Spectral kernels can learn arbitrary kernel forms
- Non-stationary kernels can learn evolving processes
- Structured kernels can learn over discrete inputs
- Deep kernels can learn arbitrary feature representations



Choose from standard kernels

covariance function	expression	S	ND
constant	σ_0^2	$\surd$	
linear	$\sum_{d=1}^{D} \sigma_d^2 x_d x'_d$		
polynomial	$(\mathbf{x} \cdot \mathbf{x}' + \sigma_0^2)^p$		
squared exponential	$\exp(-\frac{r^2}{2\ell^2})$	$\surd$	$\surd$
Matérn	$\frac{1}{2^{\nu-1}\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{\ell} r\right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}}{\ell} r\right)$	$\surd$	$\surd$
exponential	$\exp(-\frac{r}{\ell})$	$\surd$	$\surd$
γ-exponential	$\exp\left(-(\frac{r}{\ell})^\gamma\right)$	$\surd$	$\surd$
rational quadratic	$(1 + \frac{r^2}{2\alpha\ell^2})^{-\alpha}$	$\surd$	$\surd$
neural network	$\sin^{-1}\left(\frac{2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}}'}{\sqrt{(1+2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}})(1+2\tilde{\mathbf{x}}'^\top \Sigma \tilde{\mathbf{x}}')}}\right)$		$\surd$

Gaussian kernel

- Gaussian is usually our first choice for $\mathbf{x} = (x_1, \ldots, x_D) \in \mathbb{R}^D$ inputs,

$$k_{\mathrm{SE}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}\frac{||\mathbf{x} - \mathbf{x}'||^2}{\ell^2}\right) \quad (16)$$

$$k_{\mathrm{SE-ARD}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{(x_d - x_d')^2}{\ell_d^2}\right) \quad (17)$$

- Automatic Relevance Detection (ARD) refers to covariate-specific lengthscales ℓ_d^2
- Inverse of lengthscale is input relevance (high lengthscales are less relevant)
- Optimise by MLL

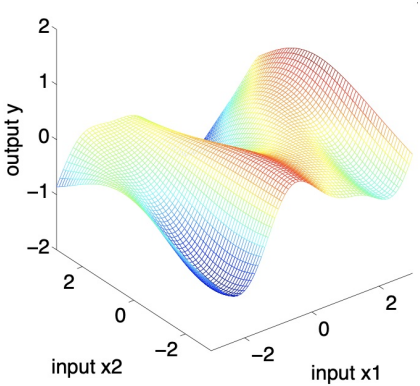


Fig 5.1 of Rasmussen, $(\ell_1, \ell_2) = (1, 3)$

Compositional kernel search[1]

- Dictionary of primitive kernels p

$$\mathrm{RBF}(\mathbf{x}, \mathbf{x}') = \exp(-||\mathbf{x} - \mathbf{x}'||^2/2\ell^2) \tag{18}$$

$$\mathrm{RQ}(\mathbf{x}, \mathbf{x}') = (1 + ||\mathbf{x} - \mathbf{x}'||^2/2\alpha\ell^2)^{1/\alpha} \tag{19}$$

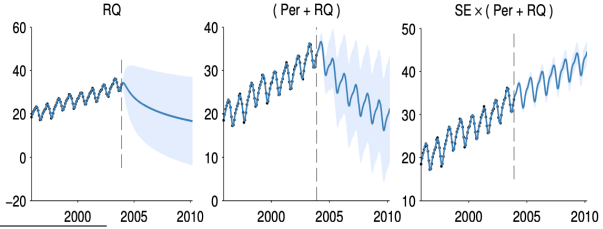
$$\mathrm{PER}(\mathbf{x}, \mathbf{x}') = \exp(-2\sin^2(\pi||\mathbf{x} - \mathbf{x}'||/p)/\ell^2) \tag{20}$$

$$\mathrm{LIN}(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}' \tag{21}$$

$$\mathrm{N}(\mathbf{x}, \mathbf{x}') = \delta(\mathbf{x} - \mathbf{x}') \tag{22}$$

$$\mathrm{C}(\mathbf{x}, \mathbf{x}') = 1 \tag{23}$$

- Construct a kernel by composing primitive kernels p with $\{+, \times\}$ using exhaustive search, optimise MLL
  - eg. $k = \lambda_1\mathrm{RBF} \times \lambda_2\mathrm{PER} + \lambda_3\mathrm{LIN}$



[1] Duvenaud et al. Structure discovery in nonparametric regression through compositional kernel search, 2013

Constructing kernels piece-by-piece II

Neural Kernel Network[2]

- Setup a massive kernel that sums over all primitives and takes their products
- Learn the combination weights

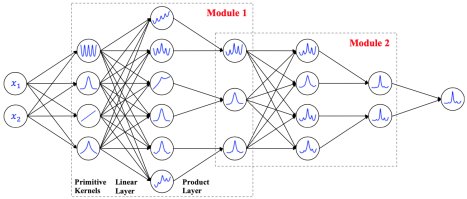
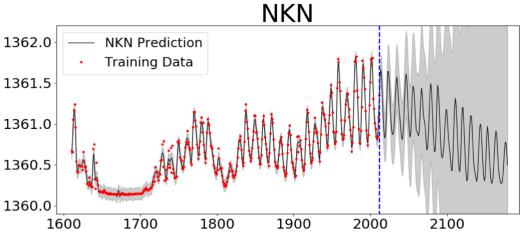


Figure 2. Neural Kernel Network: each module consists of a **Linear** layer and a **Product** layer. NKN is based on compositional rules for kernels, thus every individual unit itself represents a kernel.



[2]Sun et al, Differentiable Compositional Kernel Learning for Gaussian Processes, 2018

Deep Kernel Learning[3]

- Use neural network as feature extractor $\mathrm{NN}_\theta : \mathcal{X} \mapsto \mathbb{R}^P$,

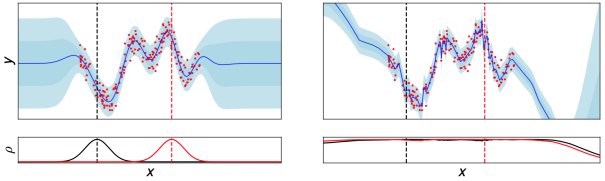
$$f(x) \sim \mathcal{GP}\Big(0, k\big(\mathrm{NN}_\theta(x), \mathrm{NN}_\theta(x')\big)\Big), \tag{24}$$

- We optimise against MLL

$$\max_\theta \quad \log p(\mathbf{y}; \theta, \sigma_{\mathrm{obs}}) = -\frac{1}{2}\mathbf{y}^T(K_\theta + \sigma_{\mathrm{obs}}^2)^{-1}\mathbf{y} - \frac{1}{2}\log\det(K_\theta + \sigma_{\mathrm{obs}}^2). \tag{25}$$

  - The gradient $\nabla_\theta \log p(\mathbf{y}; \theta, \sigma_{\mathrm{obs}})$ is computed by autodiff (PyTorch)

- Can lead to pathological results



(a) SE kernel (b) Exact DKL kernel

[3] Wilson et al, Deep Kernel Learning, 2015

Agenda for today

1. Recap
2. What is a kernel? Which kernel to choose?
3. Structured kernels
4. Spectral kernels
5. Non-stationary kernels

Structured kernels

- What if we need to regress non-vectorial inputs, such as images, strings or graphs
  - Classify an pixel image $x \in \mathbb{R}^{W \times H \times 3}$ of width W, height H and C color channels into classes
  - Predict solubility of molecular graph $x = (V, E)$ with vertices V and edges E
  - Classify a gene sequence $x = x_1 x_2 \cdots x_N$ with $x_i \in \{A, C, G, T\}$ (eg. ACGGCTTGTGTA)
- Universal principle
  1. Extract P features $\phi(x) \in \mathbb{R}^P$
  2. Compare with ordinary kernels $k(x, x') := k(\phi(x), \phi(x'))$
- Example 1: k-mers
  - A k-mer s is a contiguous sub-string of length k. Let $\phi_s(x)$ denote the count of s in x. Then the kernel is

$$k(x, x') = \sum_s w_s \phi_s(x) \phi_s(x') = \langle \sqrt{\mathbf{w}} \boldsymbol{\phi}(\mathbf{x}), \sqrt{\mathbf{w}} \boldsymbol{\phi}(\mathbf{x}) \rangle \tag{26}$$

    with w_s weighting different k-mers, eg by length. For instance, $\phi_{s=GT}(\text{"}ACGGCTTGTGTA\text{"}) = 2$
  - Exhaustive enumeration of up to millions of features
  - With graphs we extract all sub-graphs of size k, with images all sub-patches of size $k \times k$

- What if we need to regress non-vectorial inputs, such as images, strings or graphs
  - Classify an pixel image $x \in \mathbb{R}^{W \times H \times 3}$ of width W, height H and C color channels into classes
  - Predict solubility of molecular graph $x = (V, E)$ with vertices V and edges E
  - Classify a gene sequence $x = x_1 x_2 \cdots x_N$ with $x_i \in \{A, C, G, T\}$ (eg. ACGGCTTGTGTA)
- Universal principle
  1. Extract P features $\phi(x) \in \mathbb{R}^P$
  2. Compare with ordinary kernels $k(x, x') := k(\phi(x), \phi(x'))$
- Example 1: k-mers
  - A k-mer s is a contiguous sub-string of length k. Let $\phi_s(x)$ denote the count of s in x. Then the kernel is

$$k(x, x') = \sum_s w_s \phi_s(x) \phi_s(x') = \langle \sqrt{\mathbf{w}} \boldsymbol{\phi}(\mathbf{x}), \sqrt{\mathbf{w}} \boldsymbol{\phi}(\mathbf{x}) \rangle \tag{26}$$

    with w_s weighting different k-mers, eg by length. For instance, $\phi_{s=GT}(\text{"}ACGGCTTGTGTA\text{"}) = 2$
  - Exhaustive enumeration of up to millions of features
  - With graphs we extract all sub-graphs of size k, with images all sub-patches of size $k \times k$

Example 2: Fingerprints

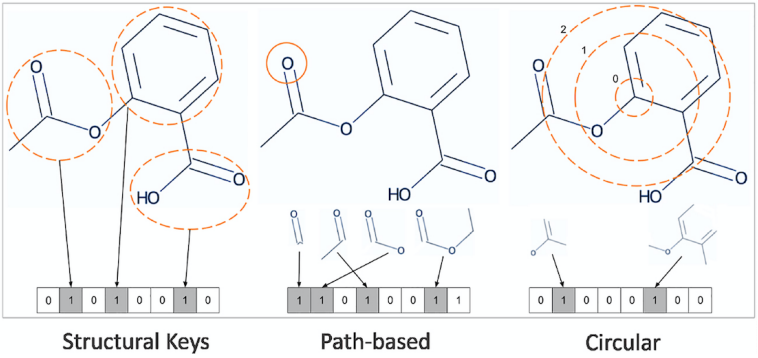


Figure from Baptista, Correia, Pereira, Rocha. Evaluating molecular representations in machine learning models for drug response prediction and interpretability. J of Int Bioinf 2022

Gaussian processes for images[4]

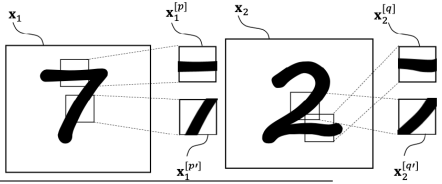
- Assume grayscale image $\mathbf{x} \in \mathbb{R}^{W \times H}$ of size (W, H) pixels with class y
- We define a Gaussian process on (k, k) size patches $\mathbf{x}^{[p]} \in \mathbb{R}^{k \times k}$

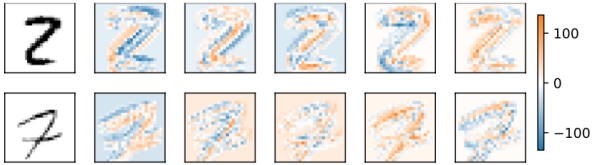
$$g \sim \mathcal{GP}(0, k(\mathbf{x}^{[p]}, \mathbf{x}'^{[p]})) \tag{27}$$

$$f = \sum_p g(\mathbf{x}^{[p]}) \tag{28}$$

$$\Rightarrow f \sim \mathcal{GP}\Big(0, \underbrace{\sum_{p,p'} k(\mathbf{x}^{[p]}, \mathbf{x}'^{[p']})}_{k(\mathbf{x}, \mathbf{x}')}\Big) \tag{29}$$

- The kernel compares all patch responses across all positions
- Represents a non-linear discrete convolution





[4] van der Wilk et al, Convolutional Gaussian Processes 2017

Agenda for today

1. Recap
2. What is a kernel? Which kernel to choose?
3. Structured kernels
4. Spectral kernels
5. Non-stationary kernels

Fourier transforms

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

$$S(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \omega} dx \tag{30}$$

where

  - i is the imaginary number with $i^2 = -1$ and $i^0 = 1$
  - $\omega \in \mathbb{R}$ is a frequency

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

$$f(x) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i x \omega} d\omega \tag{31}$$

- Euler's formula helps compute Fouriers in practise

$$e^{ix} = \underbrace{\cos x}_{\text{real}} + \underbrace{i \sin x}_{\text{imaginary}} \tag{32}$$

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

- Hence

$$\exp(2\pi i x \omega) = \cos(2\pi x \omega) + i \sin(2\pi x \omega) \tag{33}$$
$$\exp(-2\pi i x \omega) = \cos(2\pi x \omega) - i \sin(2\pi x \omega) \tag{34}$$

Fourier transforms

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

$$S(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \omega} dx \tag{30}$$

where

  - i is the imaginary number with $i^2 = -1$ and $i^0 = 1$
  - $\omega \in \mathbb{R}$ is a frequency

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

$$f(x) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i x \omega} d\omega \tag{31}$$

- Euler's formula helps compute Fouriers in practise

$$e^{ix} = \underbrace{\cos x}_{\text{real}} + \underbrace{i \sin x}_{\text{imaginary}} \tag{32}$$

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

- Hence

$$\exp(2\pi i x \omega) = \cos(2\pi x \omega) + i \sin(2\pi x \omega) \tag{33}$$

$$\exp(-2\pi i x \omega) = \cos(2\pi x \omega) - i \sin(2\pi x \omega) \tag{34}$$

Fourier transforms

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

$$S(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \omega} dx \tag{30}$$

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

$$f(x) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i x \omega} d\omega \tag{31}$$

- Euler's formula helps compute Fouriers in practise

$$e^{ix} = \underbrace{\cos x}_{\text{real}} + \underbrace{i \sin x}_{\text{imaginary}} \tag{32}$$

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

$$\exp(2\pi i x \omega) = \cos(2\pi x \omega) + i \sin(2\pi x \omega) \tag{33}$$

$$\exp(-2\pi i x \omega) = \cos(2\pi x \omega) - i \sin(2\pi x \omega) \tag{34}$$

Fourier transforms

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

$$S(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \omega} dx \tag{30}$$

where

  - i is the imaginary number with $i^2 = -1$ and $i^0 = 1$
  - $\omega \in \mathbb{R}$ is a frequency

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

$$f(x) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i x \omega} d\omega \tag{31}$$

- Euler's formula helps compute Fouriers in practise

$$e^{ix} = \underbrace{\cos x}_{\text{real}} + \underbrace{i \sin x}_{\text{imaginary}} \tag{32}$$

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

- Hence

$$\exp(2\pi i x \omega) = \cos(2\pi x \omega) + i \sin(2\pi x \omega) \tag{33}$$
$$\exp(-2\pi i x \omega) = \cos(2\pi x \omega) - i \sin(2\pi x \omega) \tag{34}$$

Fourier duals

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

Result (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

$$S(\omega) = \int_{-\infty}^{\infty} K(\tau) e^{-2\pi i \omega^T \tau} d\tau \quad \text{(FT)}$$

$$K(\tau) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i \omega^T \tau} d\omega. \quad \text{(IFT)}$$

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 $S(\omega)$, it's IFT is a kernel
   - If we change S to S', the kernel also changes from K to K'
   - $\Rightarrow$ Inter-domain kernel learning

Fourier duals

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

Result (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

$$S(\omega) = \int_{-\infty}^{\infty} K(\tau) e^{-2\pi i \omega^T \tau} d\tau \quad \text{(FT)}$$

$$K(\tau) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i \omega^T \tau} d\omega. \quad \text{(IFT)}$$

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 $S(\omega)$, it's IFT is a kernel
   - If we change S to S', the kernel also changes from K to K'
   - $\Rightarrow$ Inter-domain kernel learning

Fourier duals

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

Result (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

$$S(\omega) = \int_{-\infty}^{\infty} K(\tau) e^{-2\pi i \omega^T \tau} d\tau \quad \text{(FT)}$$

$$K(\tau) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i \omega^T \tau} d\omega. \quad \text{(IFT)}$$

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 $S(\omega)$, it's IFT is a kernel
   - If we change S to S', the kernel also changes from K to K'
   - $\Rightarrow$ Inter-domain kernel learning

Fourier duals

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

Result (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

$$S(\omega) = \int_{-\infty}^{\infty} K(\tau) e^{-2\pi i \omega^T \tau} d\tau \quad \text{(FT)}$$

$$K(\tau) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i \omega^T \tau} d\omega. \quad \text{(IFT)}$$

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 $S(\omega)$, it's IFT is a kernel
   - If we change S to S', the kernel also changes from K to K'
   - $\Rightarrow$ Inter-domain kernel learning

Kernel sinusoid representation

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

$$K(\tau) = \int_{-\infty}^{\infty} S(\omega) e^{2\pi i \tau \omega} d\omega \tag{35}$$

$$= \int_{-\infty}^{\infty} S(\omega) \cos(2\pi\tau\omega) d\omega + \int_{-\infty}^{\infty} iS(\omega) \sin(2\pi\tau\omega) d\omega \tag{36}$$

$$= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) + \int_{-\infty}^{0} iS(\omega) \sin(2\pi\tau\omega) d\omega + \int_{0}^{\infty} iS(\omega) \sin(2\pi\tau\omega) d\omega \tag{37}$$

$$= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) + \int_{0}^{\infty} iS(-\omega) \sin(2\pi\tau(-\omega)) d\omega + \int_{0}^{\infty} iS(\omega) \sin(2\pi\tau\omega) d\omega \tag{38}$$

$$= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) + \int_{0}^{\infty} -iS(\omega) \sin(2\pi\tau\omega) d\omega + \int_{0}^{\infty} iS(\omega) \sin(2\pi\tau\omega) d\omega \tag{39}$$

$$= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) \tag{40}$$

- Hence, all real-valued 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 formula $e^{\pm ix} = \cos x \pm i \sin x$
- Sine identity $\sin(-x) = -\sin(x)$
- Then we can solve the inverse Fourier as

$$
\begin{aligned}
K(\tau) &= \int_{-\infty}^{\infty} S(\omega) e^{2\pi i \tau \omega} d\omega && (35)\\
&= \int_{-\infty}^{\infty} S(\omega) \cos(2\pi\tau\omega) d\omega + \int_{-\infty}^{\infty} iS(\omega) \sin(2\pi\tau\omega) d\omega && (36)\\
&= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) + \int_{-\infty}^{0} iS(\omega) \sin(2\pi\tau\omega) d\omega + \int_{0}^{\infty} iS(\omega) \sin(2\pi\tau\omega) d\omega && (37)\\
&= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) + \int_{0}^{\infty} iS(-\omega) \sin(2\pi\tau(-\omega)) d\omega + \int_{0}^{\infty} iS(\omega) \sin(2\pi\tau\omega) d\omega && (38)\\
&= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) + \int_{0}^{\infty} -iS(\omega) \sin(2\pi\tau\omega) d\omega + \int_{0}^{\infty} iS(\omega) \sin(2\pi\tau\omega) d\omega && (39)\\
&= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) && (40)
\end{aligned}
$$

- Hence, all real-valued 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 formula $e^{\pm ix} = \cos x \pm i \sin x$
- Sine identity $\sin(-x) = -\sin(x)$
- Then we can solve the inverse Fourier as

$$
\begin{aligned}
K(\tau) &= \int_{-\infty}^{\infty} S(\omega) e^{2\pi i \tau \omega} d\omega && (35)\\
&= \int_{-\infty}^{\infty} S(\omega) \cos(2\pi\tau\omega) d\omega + \int_{-\infty}^{\infty} iS(\omega) \sin(2\pi\tau\omega) d\omega && (36)\\
&= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) + \int_{-\infty}^{0} iS(\omega) \sin(2\pi\tau\omega) d\omega + \int_{0}^{\infty} iS(\omega) \sin(2\pi\tau\omega) d\omega && (37)\\
&= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) + \int_{0}^{\infty} iS(-\omega) \sin(2\pi\tau(-\omega)) d\omega + \int_{0}^{\infty} iS(\omega) \sin(2\pi\tau\omega) d\omega && (38)\\
&= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) + \int_{0}^{\infty} -iS(\omega) \sin(2\pi\tau\omega) d\omega + \int_{0}^{\infty} iS(\omega) \sin(2\pi\tau\omega) d\omega && (39)\\
&= \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) && (40)
\end{aligned}
$$

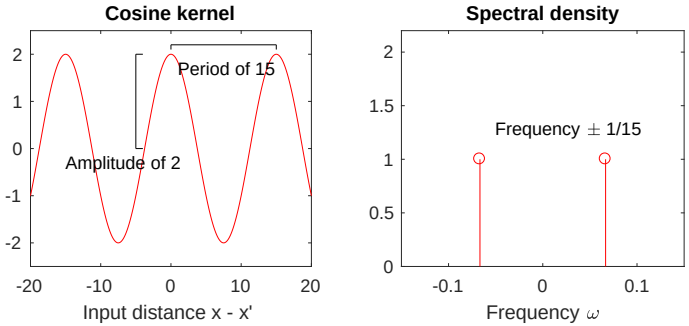
- Hence, all real-valued stationary kernels are $S(\omega)$-weighted combinations of sinusoids $\cos(2\pi\tau\omega)$

Kernel sinusoid representation

- Our new general **stationary** kernel definition

$$K(\tau) = \mathbb{E}_{S(\omega)} \cos(2\pi\tau\omega) \tag{41}$$

- 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

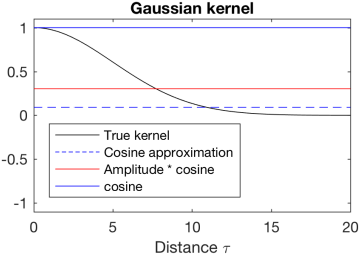
- Gaussian kernel $K_{SE}(\tau) = \exp(-\tau^2/\ell^2)$ fourier representation

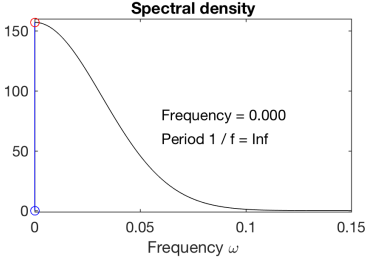
$$S_{SE}(\omega) = \int_{-\infty}^{\infty} K_{SE}(\tau) e^{-2\pi i \omega^T \tau} d\tau \tag{42}$$

$$= 2\pi\ell^2 \exp(-2\pi^2\ell^2\omega^2) \tag{43}$$

$$K_{SE}(\tau) = \int_0^{\infty} \underbrace{S_{SE}(\omega)}_{\text{amplitudes}} \cdot \underbrace{\cos(2\pi\tau\omega)}_{\text{sinusoids}} d\omega \tag{44}$$

$$\approx \sum_{\omega} S_{SE}(\omega) \cdot \cos(2\pi\tau\omega) \tag{45}$$





Gaussian kernel sinusoids

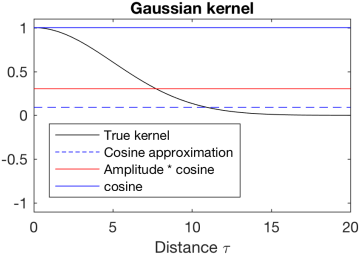
- Gaussian kernel $K_{SE}(\tau) = \exp(-\tau^2/\ell^2)$ fourier representation

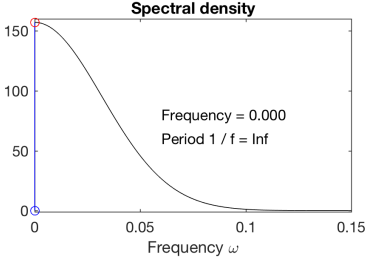
$$S_{SE}(\omega) = \int_{-\infty}^{\infty} K_{SE}(\tau) e^{-2\pi i \omega^T \tau} d\tau \tag{46}$$

$$= 2\pi\ell^2 \exp(-2\pi^2\ell^2\omega^2) \tag{47}$$

$$K_{SE}(\tau) = \int_0^{\infty} \underbrace{S_{SE}(\omega)}_{\text{amplitudes}} \cdot \underbrace{\cos(2\pi\tau\omega)}_{\text{sinusoids}} d\omega \tag{48}$$

$$\approx \sum_{\omega} S_{SE}(\omega) \cdot \cos(2\pi\tau\omega) \tag{49}$$





Gaussian kernel sinusoids

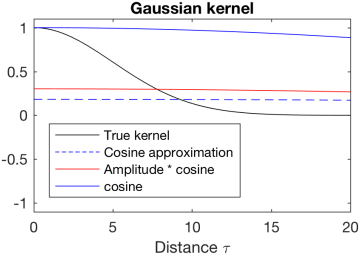
- Gaussian kernel $K_{SE}(\tau) = \exp(-\tau^2/\ell^2)$ fourier representation

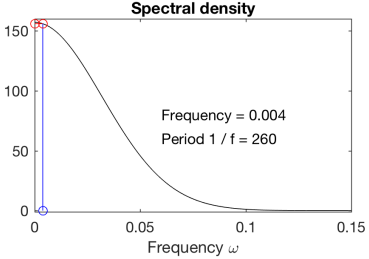
$$S_{SE}(\omega) = \int_{-\infty}^{\infty} K_{SE}(\tau) e^{-2\pi i \omega^T \tau} d\tau \tag{46}$$

$$= 2\pi\ell^2 \exp(-2\pi^2\ell^2\omega^2) \tag{47}$$

$$K_{SE}(\tau) = \int_0^{\infty} \underbrace{S_{SE}(\omega)}_{\text{amplitudes}} \cdot \underbrace{\cos(2\pi\tau\omega)}_{\text{sinusoids}} d\omega \tag{48}$$

$$\approx \sum_{\omega} S_{SE}(\omega) \cdot \cos(2\pi\tau\omega) \tag{49}$$





Gaussian kernel sinusoids

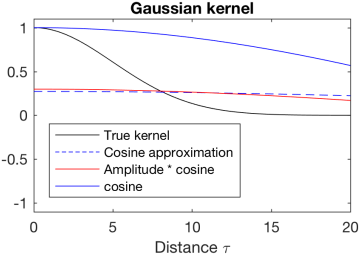
- Gaussian kernel $K_{SE}(\tau) = \exp(-\tau^2/\ell^2)$ fourier representation

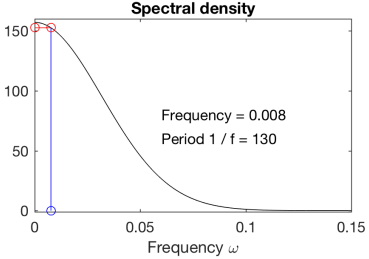
$$S_{SE}(\omega) = \int_{-\infty}^{\infty} K_{SE}(\tau) e^{-2\pi i \omega^T \tau} d\tau \tag{46}$$

$$= 2\pi\ell^2 \exp(-2\pi^2\ell^2\omega^2) \tag{47}$$

$$K_{SE}(\tau) = \int_0^{\infty} \underbrace{S_{SE}(\omega)}_{\text{amplitudes}} \cdot \underbrace{\cos(2\pi\tau\omega)}_{\text{sinusoids}} d\omega \tag{48}$$

$$\approx \sum_{\omega} S_{SE}(\omega) \cdot \cos(2\pi\tau\omega) \tag{49}$$





Gaussian kernel sinusoids

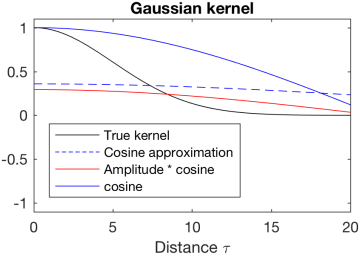
- Gaussian kernel $K_{SE}(\tau) = \exp(-\tau^2/\ell^2)$ fourier representation

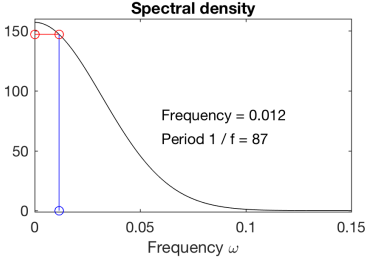
$$S_{SE}(\omega) = \int_{-\infty}^{\infty} K_{SE}(\tau) e^{-2\pi i \omega^T \tau} d\tau \tag{46}$$

$$= 2\pi\ell^2 \exp(-2\pi^2 \ell^2 \omega^2) \tag{47}$$

$$K_{SE}(\tau) = \int_0^{\infty} \underbrace{S_{SE}(\omega)}_{\text{amplitudes}} \cdot \underbrace{\cos(2\pi\tau\omega)}_{\text{sinusoids}} d\omega \tag{48}$$

$$\approx \sum_{\omega} S_{SE}(\omega) \cdot \cos(2\pi\tau\omega) \tag{49}$$





Gaussian kernel sinusoids

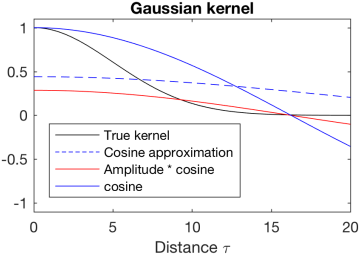
- Gaussian kernel $K_{SE}(\tau) = \exp(-\tau^2/\ell^2)$ fourier representation

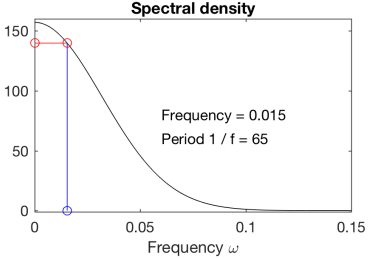
$$S_{SE}(\omega) = \int_{-\infty}^{\infty} K_{SE}(\tau) e^{-2\pi i \omega^T \tau} d\tau \tag{46}$$

$$= 2\pi\ell^2 \exp(-2\pi^2 \ell^2 \omega^2) \tag{47}$$

$$K_{SE}(\tau) = \int_0^{\infty} \underbrace{S_{SE}(\omega)}_{\text{amplitudes}} \cdot \underbrace{\cos(2\pi\tau\omega)}_{\text{sinusoids}} d\omega \tag{48}$$

$$\approx \sum_{\omega} S_{SE}(\omega) \cdot \cos(2\pi\tau\omega) \tag{49}$$





Gaussian kernel sinusoids

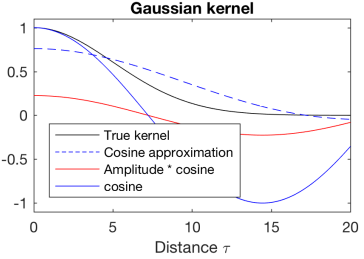
- Gaussian kernel $K_{SE}(\tau) = \exp(-\tau^2/\ell^2)$ fourier representation

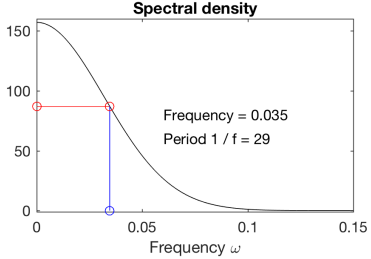
$$S_{SE}(\omega) = \int_{-\infty}^{\infty} K_{SE}(\tau) e^{-2\pi i \omega^T \tau} d\tau \tag{46}$$

$$= 2\pi\ell^2 \exp(-2\pi^2 \ell^2 \omega^2) \tag{47}$$

$$K_{SE}(\tau) = \int_0^{\infty} \underbrace{S_{SE}(\omega)}_{\text{amplitudes}} \cdot \underbrace{\cos(2\pi\tau\omega)}_{\text{sinusoids}} d\omega \tag{48}$$

$$\approx \sum_{\omega} S_{SE}(\omega) \cdot \cos(2\pi\tau\omega) \tag{49}$$





Gaussian kernel sinusoids

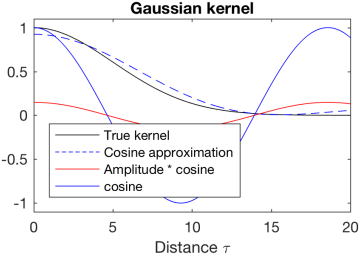
- Gaussian kernel $K_{SE}(\tau) = \exp(-\tau^2/\ell^2)$ fourier representation

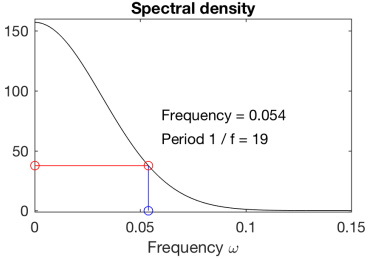
$$S_{SE}(\omega) = \int_{-\infty}^{\infty} K_{SE}(\tau) e^{-2\pi i \omega^T \tau} d\tau \tag{46}$$

$$= 2\pi\ell^2 \exp(-2\pi^2 \ell^2 \omega^2) \tag{47}$$

$$K_{SE}(\tau) = \int_0^{\infty} \underbrace{S_{SE}(\omega)}_{\text{amplitudes}} \cdot \underbrace{\cos(2\pi\tau\omega)}_{\text{sinusoids}} d\omega \tag{48}$$

$$\approx \sum_{\omega} S_{SE}(\omega) \cdot \cos(2\pi\tau\omega) \tag{49}$$





Gaussian kernel sinusoids

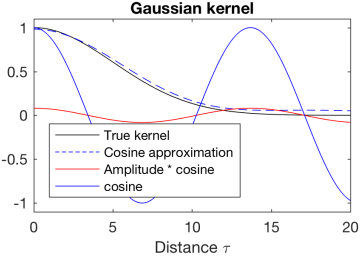
- Gaussian kernel $K_{SE}(\tau) = \exp(-\tau^2/\ell^2)$ fourier representation

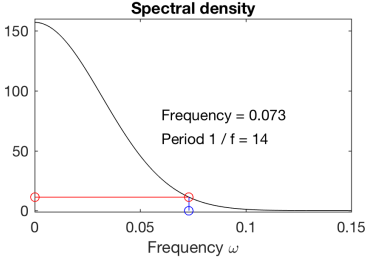
$$S_{SE}(\omega) = \int_{-\infty}^{\infty} K_{SE}(\tau) e^{-2\pi i \omega^T \tau} d\tau \tag{46}$$

$$= 2\pi\ell^2 \exp(-2\pi^2 \ell^2 \omega^2) \tag{47}$$

$$K_{SE}(\tau) = \int_0^{\infty} \underbrace{S_{SE}(\omega)}_{\text{amplitudes}} \cdot \underbrace{\cos(2\pi\tau\omega)}_{\text{sinusoids}} d\omega \tag{48}$$

$$\approx \sum_{\omega} S_{SE}(\omega) \cdot \cos(2\pi\tau\omega) \tag{49}$$





Gaussian kernel sinusoids

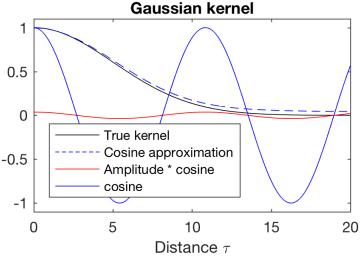
- Gaussian kernel $K_{SE}(\tau) = \exp(-\tau^2/\ell^2)$ fourier representation

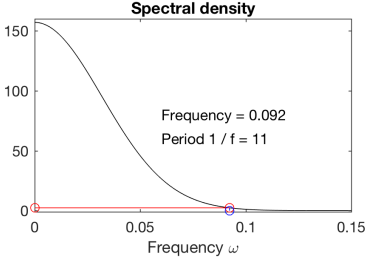
$$S_{SE}(\omega) = \int_{-\infty}^{\infty} K_{SE}(\tau) e^{-2\pi i \omega^T \tau} d\tau \tag{46}$$

$$= 2\pi\ell^2 \exp(-2\pi^2 \ell^2 \omega^2) \tag{47}$$

$$K_{SE}(\tau) = \int_0^{\infty} \underbrace{S_{SE}(\omega)}_{\text{amplitudes}} \cdot \underbrace{\cos(2\pi\tau\omega)}_{\text{sinusoids}} d\omega \tag{48}$$

$$\approx \sum_{\omega} S_{SE}(\omega) \cdot \cos(2\pi\tau\omega) \tag{49}$$





Gaussian kernel sinusoids

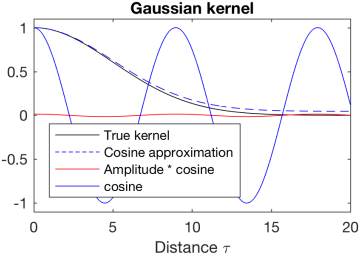
- Gaussian kernel $K_{SE}(\tau) = \exp(-\tau^2/\ell^2)$ fourier representation

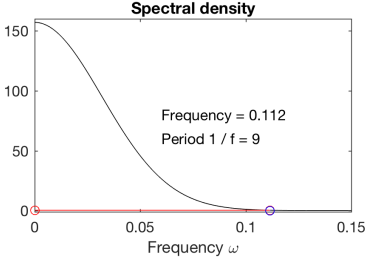
$$S_{SE}(\omega) = \int_{-\infty}^{\infty} K_{SE}(\tau) e^{-2\pi i \omega^T \tau} d\tau \tag{46}$$

$$= 2\pi\ell^2 \exp(-2\pi^2\ell^2\omega^2) \tag{47}$$

$$K_{SE}(\tau) = \int_0^{\infty} \underbrace{S_{SE}(\omega)}_{\text{amplitudes}} \cdot \underbrace{\cos(2\pi\tau\omega)}_{\text{sinusoids}} d\omega \tag{48}$$

$$\approx \sum_{\omega} S_{SE}(\omega) \cdot \cos(2\pi\tau\omega) \tag{49}$$





Gaussian kernel sinusoids

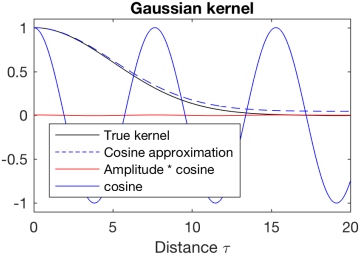
- Gaussian kernel $K_{SE}(\tau) = \exp(-\tau^2/\ell^2)$ fourier representation

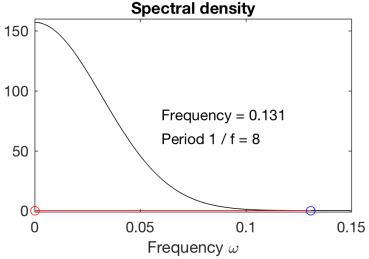
$$S_{SE}(\omega) = \int_{-\infty}^{\infty} K_{SE}(\tau) e^{-2\pi i \omega^T \tau} d\tau \tag{46}$$

$$= 2\pi\ell^2 \exp(-2\pi^2 \ell^2 \omega^2) \tag{47}$$

$$K_{SE}(\tau) = \int_0^{\infty} \underbrace{S_{SE}(\omega)}_{\text{amplitudes}} \cdot \underbrace{\cos(2\pi\tau\omega)}_{\text{sinusoids}} d\omega \tag{48}$$

$$\approx \sum_{\omega} S_{SE}(\omega) \cdot \cos(2\pi\tau\omega) \tag{49}$$





Gaussian kernel sinusoids

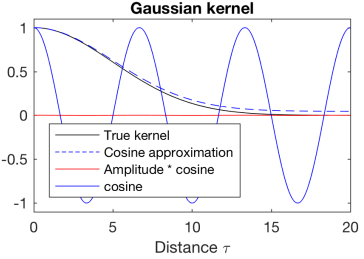
- Gaussian kernel $K_{SE}(\tau) = \exp(-\tau^2/\ell^2)$ fourier representation

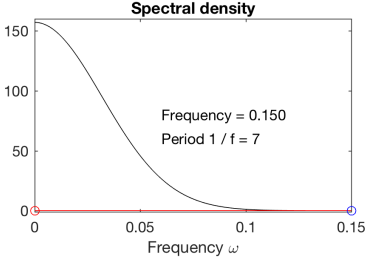
$$S_{SE}(\omega) = \int_{-\infty}^{\infty} K_{SE}(\tau) e^{-2\pi i \omega^T \tau} d\tau \tag{46}$$

$$= 2\pi\ell^2 \exp(-2\pi^2\ell^2\omega^2) \tag{47}$$

$$K_{SE}(\tau) = \int_0^{\infty} \underbrace{S_{SE}(\omega)}_{\text{amplitudes}} \cdot \underbrace{\cos(2\pi\tau\omega)}_{\text{sinusoids}} d\omega \tag{48}$$

$$\approx \sum_{\omega} S_{SE}(\omega) \cdot \cos(2\pi\tau\omega) \tag{49}$$



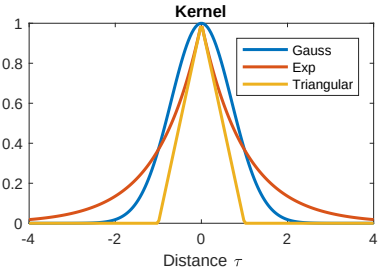


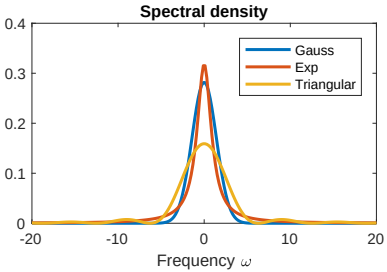
Some spectral densities

$$K_{gauss}(\tau) = \exp(-\frac{\tau^2}{\ell^2}) \qquad S_{gauss}(\omega) = \frac{\sqrt{\ell}}{2\sqrt{\pi}} \exp(-\ell\omega^2/4) \tag{50}$$

$$K_{exp}(\tau) = \exp(-|\tau|/\ell) \qquad S_{exp}(\omega) = 1/(\pi/\ell + \pi\ell\omega^2) \tag{51}$$

$$K_{tri}(\tau) = 0.5(1 - |\tau|)_+ \qquad S_{tri}(\omega) = (1 - \cos\omega)/(\pi\omega^2) \tag{52}$$





- Can we construct new kernels from custom spectral densities?

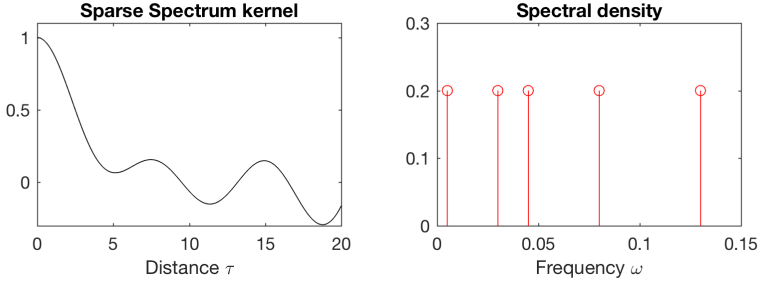
Sparse Spectrum (SS) kernel

- Define Q real frequencies $(\omega_1, \ldots, \omega_Q)^T \in \mathbb{R}^Q$ with Fourier dual[5]

$$S(\omega) := \frac{1}{Q} \sum_{i=1}^{Q} \delta(\omega = \omega_i) \tag{53}$$

$$\Rightarrow K(\tau) = \frac{1}{Q} \sum_{i=1}^{Q} \cos(2\pi\tau\omega_i) \tag{54}$$

- Highly regular covariance, prone to overfitting



[5] Lazaro-Gredilla et al (JMLR 2010) Sparse spectrum gaussian process regression

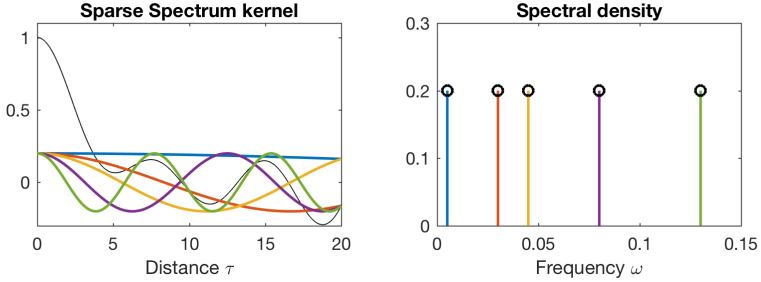
Sparse Spectrum (SS) kernel

- Define Q real frequencies $(\omega_1, \ldots, \omega_Q)^T \in \mathbb{R}^Q$ with Fourier dual[5]

$$S(\omega) := \frac{1}{Q} \sum_{i=1}^{Q} \delta(\omega = \omega_i) \tag{53}$$

$$\Rightarrow K(\tau) = \frac{1}{Q} \sum_{i=1}^{Q} \cos(2\pi\tau\omega_i) \tag{54}$$

- Highly regular covariance, prone to overfitting



[5] Lazaro-Gredilla et al (JMLR 2010) Sparse spectrum gaussian process regression

Wilson: Spectral Mixture (SM) kernel

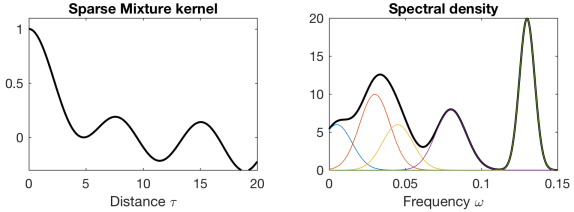
- Define mixture of Q Gaussians $\{a_i \mathcal{N}(\mu_i, \sigma_i^2)\}_{i=1}^{Q}$ [6]

$$S(\omega) := \sum_{i=1}^{Q} a_i \mathcal{N}(\omega | \mu_i, \sigma_i^2) \tag{55}$$

$$\Rightarrow K(\tau) = \int_{-\infty}^{\infty} S(\omega) \cos(2\pi\tau\omega) d\omega \tag{56}$$

$$= \sum_{i=1}^{Q} a_i \underbrace{\exp(-2\pi^2 \sigma_i^2 \tau^2)}_{\text{smooth decay}} \underbrace{\cos(2\pi\tau\mu_i)}_{\text{periodic}} \tag{57}$$

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



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

Wilson: Spectral Mixture (SM) kernel

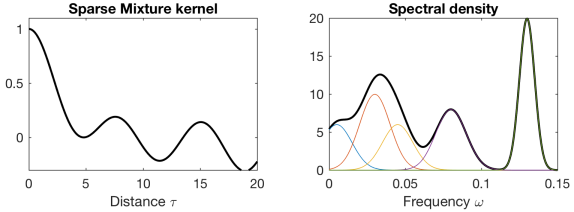
- Define mixture of Q Gaussians $\{a_i \mathcal{N}(\mu_i, \sigma_i^2)\}_{i=1}^{Q}$ [6]

$$S(\omega) := \sum_{i=1}^{Q} a_i \mathcal{N}(\omega | \mu_i, \sigma_i^2) \tag{55}$$

$$\Rightarrow K(\tau) = \int_{-\infty}^{\infty} S(\omega) \cos(2\pi\tau\omega) d\omega \tag{56}$$

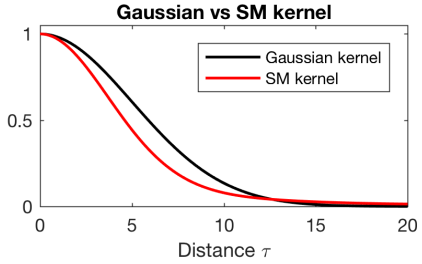
$$= \sum_{i=1}^{Q} a_i \underbrace{\exp(-2\pi^2 \sigma_i^2 \tau^2)}_{\text{smooth decay}} \underbrace{\cos(2\pi\tau\mu_i)}_{\text{periodic}} \tag{57}$$

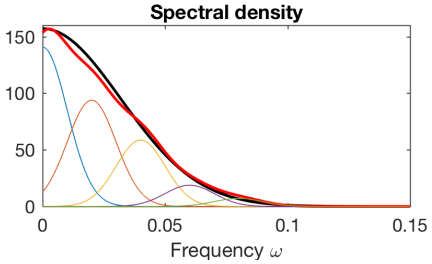
- Dense in the set of stationary kernels $\Rightarrow$ can generate **any** real stationary kernel



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

Wilson: Spectral Mixture (SM) kernel



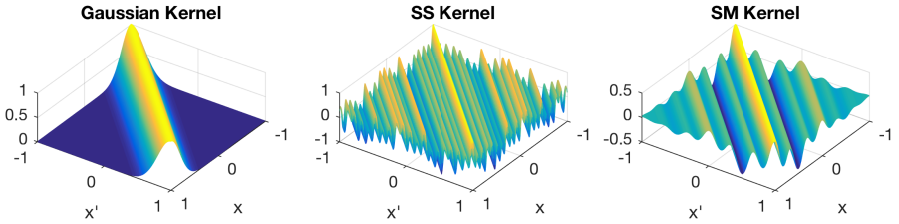


- Approximate gaussian kernel with SM kernel with $Q = 5$ components, i.e.

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

for certain a_i, μ_i, σ_i

Spectral kernels

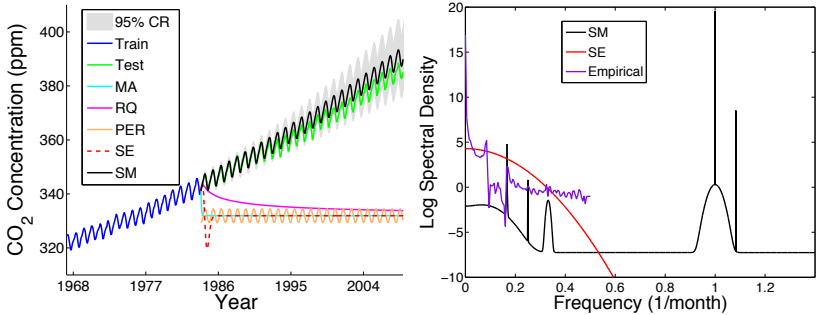


- Image from Remes, Heinonen, Kaski: Non-stationary spectral kernels, NIPS'17

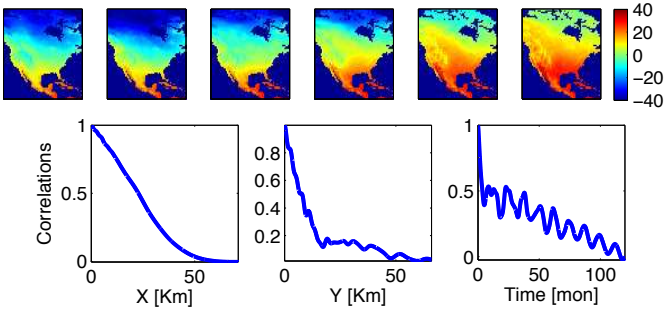
- Optimize $3Q$ hyperparameters $\theta = \{a_i, \mu_i, \sigma_i\}_{i=1}^{Q}$ of kernel $K_\theta(x - x') = \sum_{i=1}^{Q} a_i \exp(-2\pi^2 \sigma_i{}^2 \tau^2) \cos(2\pi\tau\mu_i)$ by maximizing

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

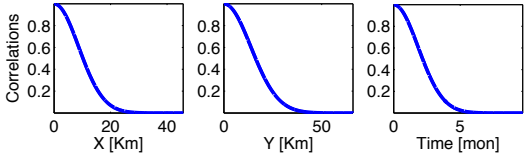
- After kernel is fixed, predictions have closed form
- Instable optimisation



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

Agenda for today

1. Recap
2. What is a kernel? Which kernel to choose?
3. Structured kernels
4. Spectral kernels
5. Non-stationary kernels

More flexible assumptions

- Standard GP regression

$$y(\mathbf{x}) = f(\mathbf{x}) + \varepsilon \tag{58}$$
$$\varepsilon \sim \mathcal{N}(0, \sigma^2) \tag{59}$$
$$f(\mathbf{x}) \sim \mathcal{GP}(0, k_\ell(\mathbf{x}, \mathbf{x}')) \tag{60}$$

  - Global noise variance
  - Global kernel function
- Heteroscedastic GPs: What if noise depends on inputs? Covered in last lecture
- Non-stationary GPs: What if function dynamics depends on inputs?

More flexible assumptions

- Standard GP regression

$$y(\mathbf{x}) = f(\mathbf{x}) + \varepsilon \tag{58}$$
$$\varepsilon \sim \mathcal{N}(0, \sigma^2) \tag{59}$$
$$f(\mathbf{x}) \sim \mathcal{GP}(0, k_\ell(\mathbf{x}, \mathbf{x}')) \tag{60}$$

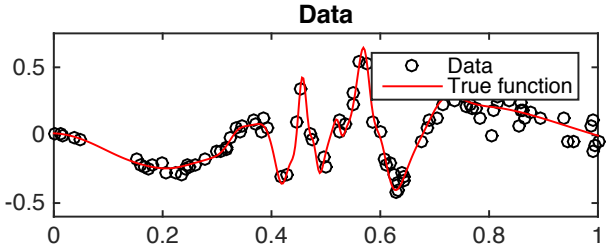
  - Global noise variance
  - Global kernel function
- Heteroscedastic GPs: What if noise depends on inputs? Covered in last lecture
- Non-stationary GPs: What if function dynamics depends on inputs?

- Stationary kernels are translation-invariant:

$$K(x, x') = K(x + a, x' + a) \tag{61}$$
$$K(x, x') = K(x - x') \tag{62}$$

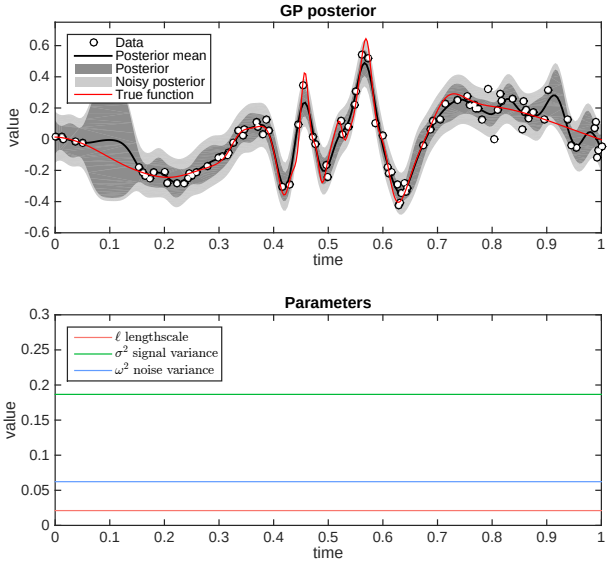
  for any a
  - Stationary kernels are function of vector distance $x - x'$
  - For instance if input variable is 'age' in years, then a stationary kernel has property $K(1, 2) = K(80, 81)$
  - Strange to assume that 1 and 2 year olds are as similar to each other as 80 and 81 year olds
- Non-stationary 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



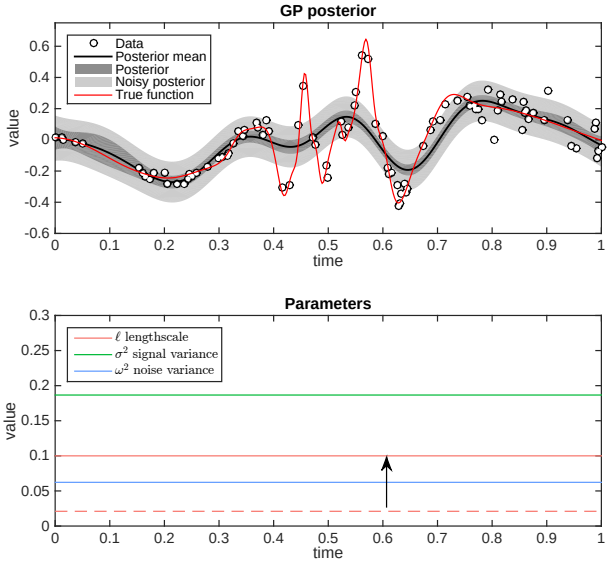
- Simple dataset

Problem with stationary functions



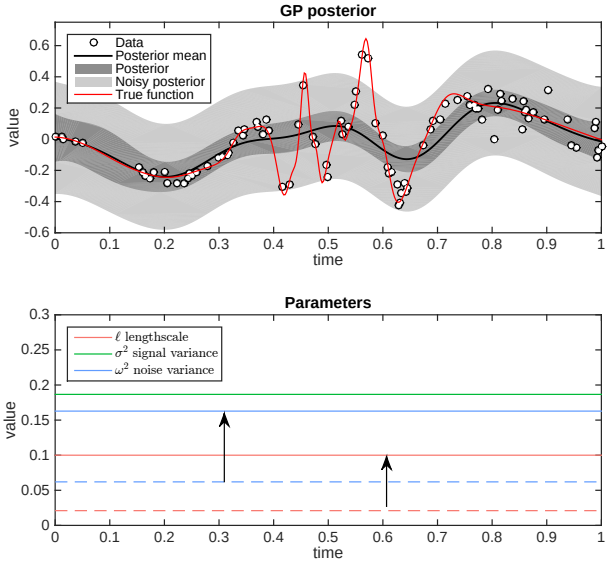
- Optimal Gaussian process fit
- Bad fit in the beginning

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

- The Gaussian kernel has a fixed, global lengthscale

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\ell^2}\right) \tag{63}$$

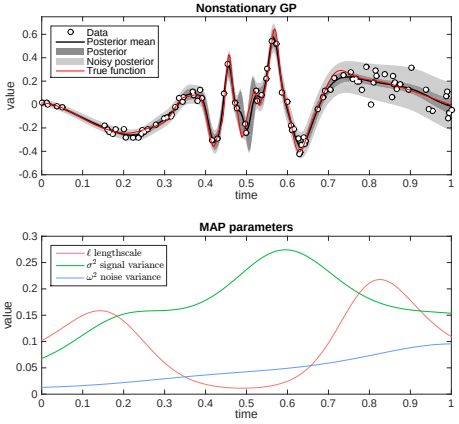
  - Equally smooth functions everywhere
- The non-stationary Gaussian kernel ('Gibbs kernel') admits a lengthscale function $\ell(x)$

$$K(x, x') = \underbrace{\sqrt{\frac{2\ell(x)\ell(x')}{\ell(x)^2 + \ell(x')^2}}}_{\text{normalizer}} \exp\left(-\frac{(x - x')^2}{\ell(x)^2 + \ell(x')^2}\right) \tag{64}$$

- The multivariate Gibbs kernel, where $\Sigma_i := \Sigma(\mathbf{x}_i) \in \mathbb{R}^{D \times D}$

$$K(\mathbf{x}_i, \mathbf{x}_j) = |\Sigma_i|^{1/4}|\Sigma_j|^{1/4}|(\Sigma_i + \Sigma_j)/2|^{-1/2} \exp\left(-(\mathbf{x}_i - \mathbf{x}_j)^T((\Sigma_i + \Sigma_j)/2)^{-1}(\mathbf{x}_i - \mathbf{x}_j)\right) \tag{65}$$

Non-stationary solution[7]



- Function process

$$y(x) = f(x) + \varepsilon(x) \tag{66}$$
$$f(x) \sim \mathcal{GP}(0, \sigma(x)\sigma(x')K_{\ell(.)}(x, x')) \tag{67}$$
$$\varepsilon(x) \sim \mathcal{N}(0, \omega(x)^2) \tag{68}$$

- Parameter processes

$$\ell(x) \sim \mathcal{GP}(\mu_\ell, K_\ell(x, x')) \tag{69}$$
$$\sigma(x) \sim \mathcal{GP}(\mu_\sigma, K_\sigma(x, x')) \tag{70}$$
$$\omega(x) \sim \mathcal{GP}(\mu_\omega, K_\omega(x, x')) \tag{71}$$

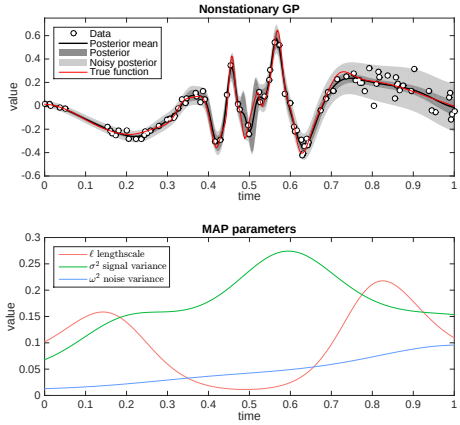
- Kernel

$$K(x, x') = \sqrt{\frac{2\ell(x)\ell(x')}{\ell(x)^2 + \ell(x')^2}} \exp\left(-\frac{(x - x')^2}{\ell(x)^2 + \ell(x')^2}\right) \tag{72}$$

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

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

Non-stationary inference



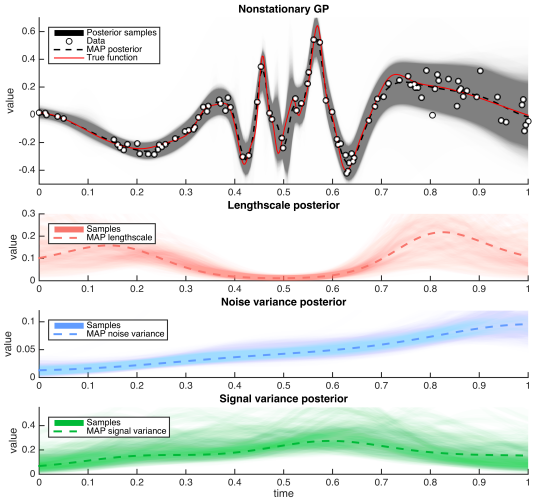
- Marginal joint likelihood

$$\mathcal{L} = p(\mathbf{y}, \boldsymbol{\ell}, \boldsymbol{\omega}, \boldsymbol{\sigma}) = p(\mathbf{y}|\boldsymbol{\ell}, \boldsymbol{\omega}, \boldsymbol{\sigma}) p(\boldsymbol{\ell}) p(\boldsymbol{\sigma}) p(\boldsymbol{\omega}) \tag{73}$$

$$= \mathcal{N}(\mathbf{y}|\mathbf{0}, \boldsymbol{\sigma}\boldsymbol{\sigma}^T \circ K_{\boldsymbol{\ell}} + diag(\boldsymbol{\omega})) \mathcal{N}(\boldsymbol{\ell}|\mu_\ell, K_\ell) \mathcal{N}(\boldsymbol{\sigma}|\mu_\sigma, K_\sigma) \mathcal{N}(\boldsymbol{\omega}|\mu_\omega, K_\omega) \tag{74}$$

- 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[8]

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

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

Summary

- The kernel choice dictates the GP function space
- GP performance depends **heavily** on how well the kernel matches data
- ARD-Gaussian kernel is a convenient 'default' kernel that **interpolates** well
  - Simple, stationary, efficient
- Non-stationary Gaussian kernel can learn **adaptive** interpolations
  - Smoothly evolving functions
- Spectral kernels can **extrapolate** repeating patterns
  - Can learn arbitrary periodic patterns, but can we trust them?
- Compositional and deep kernels search or enumerate for structure or features in inputs
  - Very flexible, prone to overfitting