

3)

a) Kernel functions

→ return co-variance matrix

gradient vector (vector with partial derivatives of kernel with respect to each of its hyperparameters)

Sol

• SE:

$$k(x_n, x_m) = \sigma_0^2 e^{-\frac{1}{2} \frac{\|x_n - x_m\|^2}{e^2}}$$

$$\frac{dk}{d\sigma_0} = 2\sigma_0 e^{-\frac{1}{2} \frac{\|x_n - x_m\|^2}{e^2}} \frac{-\frac{1}{2} \frac{\|x_n - x_m\|^2}{e^2}}{e^2} \frac{d}{de} \left(\frac{1}{e^2} \right)$$

$$\frac{dk}{de} = \sigma_0^2 e^{-\frac{1}{2} \frac{\|x_n - x_m\|^2}{e^2}} \frac{d}{de} \left(-\frac{1}{2} \frac{\|x_n - x_m\|^2}{e^2} \right) = -\frac{\|x_n - x_m\|^2}{2} \sigma_0^2 e^{-\frac{1}{2} \frac{\|x_n - x_m\|^2}{e^2}} (-2) \frac{1}{e^3}$$

$$= \|x_n - x_m\|^2 \cdot \frac{\sigma_0^2}{e^3} e^{-\frac{1}{2} \frac{\|x_n - x_m\|^2}{e^2}}$$

return $k(x_n, x_m)$, $\frac{dk}{d\sigma_0}$, $\frac{dk}{de}$

• Linear: $k(x_n, x_m) = \sigma_0^2 + \sigma_1^2 (x_n - c)^T (x_m - c)$

 $x_n \in \mathbb{R}^{n \times d}$
 $x_m \in \mathbb{R}^{m \times d}$
 $c \in \mathbb{R}^d$

$$\frac{dk}{\sigma_0} = 2\sigma_0 \cdot \text{ones}(n, m)$$

$$\frac{dk}{\sigma_1} = 2\sigma_1 (x_n - c)^T (x_m - c)$$

$$\frac{dk}{c} = \sigma_1^2 \frac{d}{dc} (x_n - c)(x_m - c)^T$$

$x_n \in \mathbb{R}^{n \times d}$
 $x_m \in \mathbb{R}^{m \times d}$

$$\begin{matrix} & d \\ n & \cdot \\ & x_n^{ik} \end{matrix} \cdot \begin{matrix} m \\ d \\ \cdot \\ x_m^{jk} \end{matrix} = \begin{matrix} m \\ n \end{matrix}$$

Considering one row: $x_n \in \mathbb{R}^{1 \times d}$ $x_n \cdot x_m^T \in \mathbb{R}$
 $x_m \in \mathbb{R}^{d \times 1}$

Let us call $A = (x_n - c)(x_m - c)^T$, we need $\frac{\partial A}{\partial c}$

$$A_{ij} = (x_n - c)_i \cdot (x_m - c)_j^T = \sum_{k=1}^d (x_n^{ik} - c)(x_m^{jk} - c)$$

$$= \sum_{k=1}^d (x_n^{ik} x_m^{jk} - c x_m^{jk} - c x_n^{ik} + c^2)$$

$$\frac{\partial A_{ij}}{\partial c} = \sum_{k=1}^d - (x_m^{jk} + x_n^{ik}) + 2c$$

$$\Rightarrow \frac{\partial k_{ij}}{\partial c} = \sigma_1^2 \left[- \left(\sum_{k=1}^d (x_m^{jk} + x_n^{ik}) \right) + 2dc \right] \quad i=1 \dots n \quad j=1 \dots m$$

return $k(x_n, x_m)$, $\frac{dk}{d\sigma_0}$, $\frac{dk}{de}$, $\frac{dk}{dp}$

Periodic: $k(x_n, x_m) = \sigma_0^2 \exp \left\{ -\frac{\omega}{e^2} \sin^2 \left(\frac{\pi \|x_n - x_m\|}{p} \right) \right\}$ $p = \text{periods}$

$$\frac{dk}{d\sigma} = 2\sigma_0 \exp \left\{ -\frac{\omega}{e^2} \sin^2 \left(\frac{\pi \|x_n - x_m\|}{p} \right) \right\}$$

$$\frac{dk}{de} = \sigma_0^2 \left(\frac{4}{e^3} \sin^2 \left(\frac{\pi \|x_n - x_m\|}{p} \right) \right) \exp \left\{ -\frac{\omega}{e^2} \sin^2 \left(\frac{\pi \|x_n - x_m\|}{p} \right) \right\}$$

$$\frac{dk}{dp} = \sigma_0^2 \left(-\frac{\omega}{e^2} \right) \frac{d}{dp} \left[\sin^2 \frac{\pi \|x_n - x_m\|}{p} \right] \exp \left\{ -\frac{\omega}{e^2} \sin^2 \left(\frac{\pi \|x_n - x_m\|}{p} \right) \right\}$$

$$= -2 \frac{\sigma_0^2}{e^2} \cdot 2 \cdot \sin \left(\frac{\pi \|x_n - x_m\|}{p} \right) \cdot \pi \|x_n - x_m\| \left(-\frac{1}{p^2} \right) \exp \left\{ -\frac{\omega}{e^2} \sin^2 \left(\frac{\pi \|x_n - x_m\|}{p} \right) \right\}$$

$$= -4 \left(\frac{\sigma_0}{e \cdot p} \right)^2 \pi \|x_n - x_m\| \cdot \sin \left(\frac{\pi \|x_n - x_m\|}{p} \right) \cdot \exp \left\{ -\frac{\omega}{e^2} \sin^2 \left(\frac{\pi \|x_n - x_m\|}{p} \right) \right\}$$

To make things easier, let ω reuse the most used parts

$$\text{sin-fun} = \sin \left(\frac{\pi \|x_n - x_m\|}{p} \right)$$

$$\text{exp-fun} = \exp \left\{ -\frac{\omega}{e^2} (\text{sin-fun})^2 \right\}$$

$$\rightarrow k(x_n, x_m) = \sigma_0^2 \cdot \text{exp-fun}$$

$$\frac{dk}{d\sigma} = 2\sigma_0 \cdot \text{exp-fun}$$

$$\frac{dk}{de} = \sigma_0^2 \left(\frac{4}{e^3} (\text{sin-fun})^2 \right) \text{exp-fun}$$

$$\frac{dk}{dp} = -4 \left(\frac{\sigma_0}{e \cdot p} \right)^2 \pi \|x_n - x_m\| \cdot \text{sin-fun} \cdot \text{exp-fun}$$

return $k(x_n, x_m)$, $\frac{dk}{d\sigma_0}$, $\frac{dk}{de}$, $\frac{dk}{dp}$

Important note: To make things easier for us in point b), we do not always return both k and its derivative. We add an input `return_k` and if `return_k == 1` \Rightarrow return k
 $\circ \Rightarrow$ "gradient vector"

This does not impact the structure of the functions or the solutions. It is only a way to make outputs easier to use.

b) Write the function fitGPR. The inputs are:

- X : training input
- y : " output
- $K\text{-type}$: one of the three kernels of a)
- θ_0 : initial parameters
- bound_theta : bounds parameters } for optimization
- X_{star} : test points

The marginal likelihood is given by:

$$\log p(y|X, \theta) = -\frac{y^T (K + \sigma_n^2 I)^{-1} y}{2} - \frac{\log \det(K + \sigma_n^2 I)}{2} - \frac{N \log(2\pi)}{2} \quad (1)$$

Denote $K_y = K + \sigma^2 I$ $\alpha = K_y^{-1} y$

$$\frac{\partial}{\partial \theta_j} \log(y|X, \theta) = -\frac{1}{2} \text{tr}((\alpha \alpha^T - K_y^{-1}) \frac{\partial K_y}{\partial \theta_j}) \quad (2)$$

where θ is the vector of all hyperparameters of the model

We follow the algorithm in slide 3A:

- ① Estimate $\hat{\theta}$ (optimal hyperparameters) by maximizing the marginal likelihood.
- ② Set $\theta = \hat{\theta}$ and obtain $K_y = K + \sigma^2 I$ and $L = \text{cholesky}(K_y)$
- ③ Obtain $\alpha = L^T / (L \backslash y)$ and $U = L \backslash K_*$
- ④ Return the posterior mean m_* and the covariance matrix \bar{K} given by
 $m_* = K_* \alpha$
 $\bar{K} = K_{**} - U^T U$

So, let us proceed for steps:

- ① We want to maximize the marginal likelihood:

$$\max_{\theta} \log p(y|X, \theta) = \max_{\theta} \left(-\log p(y|X, \theta) \right)$$

We use \max because MATLAB optimizers compute the min
 \Rightarrow to the optimizer we pass the function $f = -\log p(y|X, \theta)$.

Which are the parameters? They depend on what kernel function we pass to fitGPR. For example, if we suppose $K\text{-type} = \text{'squaredexponential'}$, the parameters of K are σ and ℓ

$$\Rightarrow \theta_k = [\sigma_0, \ell] \quad \theta_K : \text{parameters of the kernel}$$

However, these are not all the parameters we need to consider: if we look at $\log p(y|X, \theta)$, we see that also $\sigma^2 = \sigma_n^2$ is a parameter that must be found.

σ^2 comes from the variance of y : $\text{Cov}(y) = k(x, x) + \sigma^2 I$

\Rightarrow The set of parameters to be found is $\theta = [\theta_K, \sigma]$

\downarrow ↴ independent from the kernel
depends from the kernel

To optimize f we need to pass to the optimizer also a gradient vector:

$$\text{if } \theta = [\sigma_0, e] \Rightarrow g = \left[\frac{\partial f}{\partial \sigma_0}, \frac{\partial f}{\partial e}, \frac{\partial f}{\partial \sigma^2} \right]$$

In general $g = \left[\underbrace{\frac{\partial f}{\partial \theta_K}}_{\text{a vector}}, \frac{\partial f}{\partial \sigma^2} \right]$

- $\frac{\partial f}{\partial \theta_K} = \frac{\partial}{\partial \theta_K} (-\log p(y|X, \theta))$ ← This function is given in the slides

For each parameter θ_j in θ_K , we can use $\frac{\partial}{\partial \theta_j} \log(p(y|X, \theta))$ that, in turn, depends on $\frac{\partial K}{\partial \theta_j}$ ← output of point a)

To find $\frac{\partial f}{\partial \sigma^2}$ we create a function, called derivative-f-sigma2, which returns $\frac{\partial f}{\partial \sigma^2}$ as a function of σ^2 . The derivative can be found using the MATLAB function diff, but it is really slow

Edited.
Please see *

Therefore we proceed as following in the fit GPR function for part ①:

- we create the vector θ of our parameter, where

$$\theta = [\theta_K, \sigma^2]$$

$$\text{theta} = [\text{theta_K}, \text{sigma2}]$$

- We extend the initial condition and the bound of θ_K , in order to include σ^2 .
- Calculate K and dk using one of the 3 different functions of point a), according to k -type.
- We create the function to_minimise function which returns f and g . This is the method we will have to give to fmincon.

to_minimise is a function of $\theta_K, \sigma^2, X, Y, k$ -type

for each type we use a different gradient vector.

- ↳
- define initial value for σ^2 and its bounds
 - adjust bounds for fmincon
 - definition of the to_minimise function → returns $[f, g]$

f : objective function

\mathbf{g} : gradient vector (as many elements as hyper-parameters)

Steps into the function:

- define $\log p$ and, consequently, f

↳ see $\log p(y|X, \theta)$ -

- define the gradient vector:

as anticipated, we need to work separately for the parameters from K and σ^2

* parameters of $K(\theta_K)$: use the function $\frac{\partial}{\partial \theta_K} \log(y|X, \theta)$.

Remark: we are minimizing f , not $\log p$! \Rightarrow also here we work with $f \Rightarrow \frac{\partial f}{\partial \theta_K}$

We define it in the function derivative_f_theta

* For σ^2 we proceed differently: we use the MATLAB function diff applied to f .

\Rightarrow we obtain the gradient vector

- call the optimizer fmincon passing to minimize, θ_0 and the bounds.
return θ_{opt}

②③ Steps :

- $\theta_{\text{opt}} = \theta_{\text{theta-opt}}$
- Find K using the optimal parameters

(note that, in this case, the idea of not calculating cK when it is not necessary, avoids us some useless steps)

- Define $K_y = K + \sigma^2 \mathbb{I} \Rightarrow$ in code $K_y = K + \underbrace{\text{theta}(\text{end})}_{\sigma^2} \mathbb{I}$
- Define $L = \text{chol}(K_y)$
↳ factorize K_y in R such that $K_y = L' * L$
- Find α and v as indicated

④ Note that $K_* = K(\bar{x}^*, x) \Rightarrow$ find m^* and \bar{K}
 $K_{**} = K(x^*, x^*)$ \Rightarrow m_{star} K -line

We return m_{star} , K -line, θ_{opt} and the max_lk

$\log p(y|X, \theta)$ calculated at θ_{opt} ↪
maximum likelihood.

- c) We use the code of Exercise sheet 12 to test the accuracy of our code.
 The main part of this Take Home Exam is the previous one, the creation of `fitGPR.m`, while this one is needed to test the accuracy.

Therefore, we create the X_{train} , y_{train} and X_{test} sets, we initialize the parameters initial values and their bounds and we call `fitGPR`.
 y_{test} is the mean output of this function.

After that, we also call the matlab function `fitgpr` in order to make a comparison with our results.

This comparison is made implementing the MAE and the AAE.

Note that as comment, our code includes also what is needed to use the three different kernels and understand which is the most suitable one:
 the best kernel is the one which leads to the maxima marginal likelihood.

- ✳ When we set X and y to high sizes, solve diff for symbolic matrices is computationally too expensive

Therefore, we try to solve numerically $\frac{\partial f}{\partial \sigma^2}$:

$$f = \frac{y^T (K + \sigma_n^2 I)^{-1} y}{2} + \frac{\log \det(K + \sigma_n^2 I)}{2} + \frac{N \log(2\pi)}{2}$$

$$\frac{\partial f}{\partial \sigma^2} = \frac{1}{2} \left[y^T \left[\frac{\partial}{\partial \sigma^2} [(K + \sigma_n^2 I)^{-1}] \right] y + \frac{1}{\det(K + \sigma_n^2 I)} \frac{\partial}{\partial \sigma^2} [\det(K + \sigma_n^2 I)] \right]$$

$\frac{\partial A^{-1}}{\partial x} = -A^{-1} \frac{\partial A}{\partial x} A^{-1}$
✳

$\frac{\partial \det A}{\partial x} = \det A \cdot \text{tr}(A^{-1} \frac{\partial A}{\partial x})$

$$= \frac{1}{2} \left[y^T \cdot \text{der_inv_k} \cdot y + \frac{1}{\cancel{\det(K + \sigma_n^2 I)}} \cdot \cancel{\det(K + \sigma_n^2 I)} \cdot \text{tr}((K + \sigma_n^2 I)^{-1}) \right]$$

$$= \frac{1}{2} \left[y^T \cdot \text{der_inv_k} \cdot y + \text{tr}((K + \sigma_n^2 I)^{-1}) \right]$$

\downarrow
 $= -(K + \sigma_n^2 I)^{-1} I (K + \sigma_n^2 I)^{-1}$

- ✳ Ref: "The Matrix Cookbook", Paterson and Paderson, 2005