Take Home Exam 6 — Computational Finance

Mezzetti Anita 307498

3)

a) Kernel functions

→ return · co-variance matrix
· gradient vector (vector with partial derivatives of kernel with the respect to each of its hyperparameters)

Sol

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

$$\frac{dk}{d\sigma_0} = 2\sigma_0 \, e^{-\frac{1}{2}\frac{\|x_n - x_m\|^2}{e^2}}$$

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

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

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

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

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

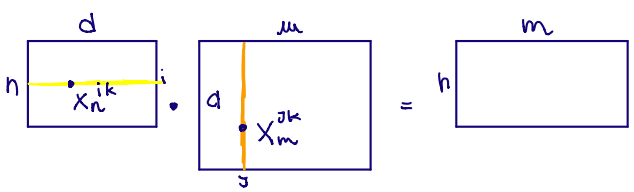
x_n $n \times d$
x_m $m \times d$
c $\in \mathbb{R}$

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

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

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

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



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

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

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

$i = 1:n$
$j = 1:m$

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

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

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

return $k(x_n, x_m)$, $\frac{dk}{d\sigma_0}$, $\frac{dk}{d\sigma_1}$, $\frac{dk}{dc}$

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

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

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

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

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

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

To make things easier, let us rename the most used parts

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

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

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

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

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

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

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
0 $\Rightarrow$ " 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_type : one of the three kernels of a)
- theta0 : initial parameters } for optimization
- bound_theta : bounds parameters }
- 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 $K_y = k + \sigma^2 I \qquad \alpha = K_y^{-1} y$

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

where θ is the vector of all hyperparameters of the model

We follow the algorithm in slide 34:

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

So, let us proceed for steps:

1. We want to maximize the marginal likelihood:

$$\max_\theta \log p(y|X,\theta) = \min_\theta -\log p(y|X,\theta)$$

We use min because MATLAB optimizers compute the min
⇒ 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_type = 'squaredexponential', the parameters of k are σ and ℓ

⇒ $\theta_k = [\sigma_0, \ell]$ θ_k : parameter of the kernel

However, these are not all the parameter 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 find.

σ^2 comes from the variance of y: $\quad Cov(y) = k(x,X) + \sigma^2 \mathbb{1}$

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

- θ_K: depends from the kernel
- σ: indipendent from the kernel

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

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

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

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

  For each parameter θ_j in θ_K, we can use $\frac{\partial}{\partial \theta_j} \log(y|X,\theta)$ that, in turns, 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 is found using the MATLAB function `diff`

Therefore we proceed as following in the `fit GPR` function for part ④:

- we create the vector θ `theta` of our parameter, where

$$\theta = [\theta_K, \sigma^2]$$
$$\text{theta} = [\text{theta\_K}, \text{sigma2}]$$

- We extend the initial conditions and the bounds 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 θ_K, σ^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_minimize` function → returns `[f, g]`

f: objective function
g: gradient vector (as many elements as hyperparameters)

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 (θ_k): use the function $\frac{\partial}{\partial \theta_j} \log(y|X,\theta)$.

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

  We define it in the function derivative_f_theta

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

⇒ we obtain the gradient vector

- call the optimizer fmincon passing to_minimize, θ_0 and the bounds.
  returns theta_opt

(2)(3) Steps:
- theta = theta_opt
- Find k using the optimal parameters

(note that, in this case, the idea of not calculation dK when it is not necessary, avoid us some useless steps)

- Define $K_y = K + \sigma^2 I$ ⇒ in code `ky = k + sigma2(end)*I`
- Define L = chol(Ky)
  ↳ factorize Ky in R such that $K_y = L' * L$
- Find α and ν as indicated

(4) Note that $K_* = k(\bar{X}^*, X)$, $K_{**} = k(X^*, X^*)$ ⇒ find m^* and $\bar{K}$
m_star K_bar

We return m_star, K_bar, theta_opt and the max_lik

$\log p(y|X,\theta)$ calculated at theta_opt
maximum likelihood.

c) We use the code of Exercise sheet 12 to test the accuracy of our code.
The main part 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 tree different kernels and understant which is the most suitable one:
the best kernel is the one which leads to the maxima marginal likelihood.