CURVE FITTING BY LINEAR SMOOTHING

nonlinear regression problem : $y_i = f(x_i) + \varepsilon_i$

A) Assume $f(x)$ is linear and zero mean

$y_i = \beta x_i + \varepsilon_i$

and we want to estimate y^* at some new point x^*

show that with one predictor and linear f, $\hat{y}^*$ may be expressed as a linear smoother

x_i $p \times 1$

$$E(y) = x\hat{\beta} = x_i'(X'X)^{-1}X'y$$

$$E(y_i) = x_i^{*\prime}\hat{\beta} = x_i^{*\prime}(X'X)^{-1}X'y$$

$$(X'X)^{-1} = \left(\sum_i^n x_i x_i'\right)^{-1}$$

$$X'y = \sum_i^n x_i y_i$$

$$\left(\sum_i^n x_i x_i'\right)^{-1}\left(\sum_i^n x_i y_i\right) = \sum_i^n w_i \cdot y_i$$

linear combination of past y_i

⇒ the smoother is a linear combination of past y_i in particular is an average of the effect coming from different observations.

comparing with alternative approach:

(K-nearest-neighbour smoothing)

$$w_K(x_i, x^*) = \begin{cases} 1/K & x_i \text{ one of the closest sample points to } x^* \\ 0 & \text{otherwise} \end{cases}$$

in the first case all the other observations are playing a role, while in this case only the closest

⇒ approximation will be smoother if compared to the previous case.

will have results that more accommodate the surrounding points.

B) Kernel function: $K(x)$ smooth function satisfying:

- $\int_{\mathbb{R}} K(x)\,dx = 1$
- $\int_{\mathbb{R}} x\,K(x)\,dx = 0$
- $\int_{\mathbb{R}} x^2\,K(x)\,dx > 0$

ex: 1) uniform kernel:

$$K(x) = \frac{1}{2}\mathbb{1}(x) \quad , \quad \mathbb{1}(x) = \begin{cases} 1 & -1 \le |x| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

area under x considered same distance $\bar{x}$ here

2) Gaussian kernel:

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

⇒ used as weighting functions for fitting local models

which other obs should be more relevant?

derive the regression in that area

$$w(x_i, x^*) = \frac{1}{h} K\left(\frac{x_i - x^*}{h}\right) \Rightarrow$$ weighting function

h → BANDWIDTH

⇒ una sorta di tecnica che serve per ricordare

↳ you can use this kernel to get the normalized weights (determ which of the other variables are more relevant)

(→ you are using them "in a linear smoother")

⇒ KERNEL REGRESSION

We want to chose h such that gives the smallest possible prediction error.

see notebook for implementation and results

CROSS VALIDATION

A) write code that would split the data in two parts and use it accordingly to the mse criterion.

see jupiter notebook and python code for implementation and results.

B) Now repeat a similar experiment but with data simulated from the following functions:

B.1): "wiggly" function with small noise observations

B.2): "wiggly" function with high noise observations

B.3): "smooth" function with small noise observations

B.4): "smooth" function with high noise observations

see notebook and python code

D) leave one out remove

Note: The process we followed to find a good value for h has some drawbacks:

- loss of information by splitting the dataset

⇒ **alternative:** leave one out cross validation

$$LOOCV = \sum^{n} (y_i - \hat{y}_i^{(-i)})^2$$

$\hat{y}_i^{(-i)}$ → predictive value of y_i obtained by omitting the i-th pair (x_i, y_i) and fitting the model to the remaining dataset of size N-1

→ would do it $\forall i$ → N times

seems like we would have to compute $\forall h$ N models and then average the errors

⇓

Shortcut: $\hat{y} = Hy \triangleq X\hat{\beta} = \underbrace{X(X'X)^{-1}X'}_{H} y$

and considering that matrix one can write: (FOR ANY LINEAR SMOOTHER)

$$LOOCV = \sum^{n} \left(\frac{y_i - \hat{y}_i}{(1 - H_{ii})} \right)^2$$

LOCAL POLYNOMIAL REGRESSION

We can see kernel regression as a "locally constant" estimator obtained from locally weighted least squares.

$$\hat{f}(x) = a = \underset{R}{\operatorname{argmin}} \sum_{i=1}^{n} w_i (y_i - a)^2$$ ← $a(x)$ → minimizing MSE in x

w_i: normalized

↳ if $w_i = \frac{1}{n} \Rightarrow \hat{f}(x) = \bar{y}$

↳ if $w_i \propto w(x, x_i) = \frac{1}{h} k\left(\frac{x_i - x}{h}\right)$ ⇒ we get the kernel estimation.

w_i: unnormalized

weight depends on proximity

A) Generalization of locally constant regression ⇒ local polynomial regression.

$\hat{f}(x_i) = g_x(x_i \mid \hat{a})$ ⇒ each x is responsible for x_i in its neighborhood $= R_{x_i} \cdot \hat{a}$

x = target point

u = points in its neighborhood

$$g_x(u, a) = a_0 + \sum_{k=1}^{D} a_j (u - x)^k$$

quelli che vanno stimati

⇒ to be estimated using weighted least squares

$$\hat{a} = \underset{R^{D+1}}{\operatorname{argmin}} \sum w_i (y_i - g_x(x_i, a))^2$$

uno quelli che minimizzano LSE

Define R_x ($n \times D$) =

$$\begin{bmatrix} 1 & (x_1 - x)^1 & (x_1 - x)^2 & \cdots & (x_1 - x)^D \\ 1 & \vdots & \vdots & & \vdots \\ \vdots & & & & \\ 1 & (x_n - x)^1 & (x_n - x)^2 & & (x_1 - x)^D \end{bmatrix}$$

let's first find $\hat{a}$ that minimizes the sum of the errors:

$$\hat{a} = \underset{\mathbb{R}^{D+1}}{\operatorname{argmin}} \sum^{n} \tilde{w}_i (y_i - g_x(x_i|a))^2$$

($\tilde{w}_i$: normalized weights)

$$= \underset{\mathbb{R}^{D+1}}{\operatorname{argmin}} (y - Rx\underline{a})' \operatorname{diag}(\tilde{w}_1 \dots)(y - Rxa)$$

$$\hookrightarrow = y'\tilde{W}y - 2a'Rx'\tilde{W}y + a'Rx'\tilde{W}Rxa$$

$$\frac{\partial}{\partial a} = -2Rx'\tilde{W}y + 2a'Rx'\tilde{W}Rx = 0$$

$$\Rightarrow \hat{a} = (Rx'\tilde{W}Rx)^{-1}Rx'\tilde{W}y$$

B) Note that when $D=1$:

$$g_x(u,a) = \hat{a}_0 + \hat{a}_1(u-x) \quad \Rightarrow \text{we are fitting two terms: } a_0 \text{ and } a_1$$

so that $\begin{bmatrix} \hat{a}_0 \\ \hat{a}_1 \end{bmatrix} = (Rx'\tilde{W}Rx)^{-1}Rx'\tilde{W}y$

Rx': $\begin{bmatrix} 1 & \dots & 1 \\ (x_1-x) & \dots & (x_n-x) \end{bmatrix}$

$\tilde{W}$: $\begin{bmatrix} w_1 & & 0 \\ & \ddots & \\ 0 & & w_n \end{bmatrix}$

More specifically, we are asked to code for $\hat{f}(x)$ at the target point.

we have: $\hat{f}(x) = g_x(u=x, \hat{a}) = \hat{a}_0$

you end up using only the intercept, but actually the point of the local regression is to fit all the vector $\hat{a}$:

so that:

$$\hat{f}(x) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \hat{a}_0 \\ \hat{a}_1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \end{bmatrix} \left(\begin{bmatrix} 1 & \cdots & 1 \\ (x_1 - x) & \cdots & (x_n - x) \end{bmatrix} \begin{bmatrix} w_1 & & \\ & \ddots & \\ & & w_n \end{bmatrix} \begin{bmatrix} 1 & (x_1 - x) \\ \vdots & \vdots \\ 1 & (x_n - x) \end{bmatrix} \right)^{-1}$$

$$\begin{pmatrix} 1 & \cdots & 1 \\ (x_1 - x) & \cdots & (x_n - x) \end{pmatrix} \begin{pmatrix} w_1 & & \\ & \ddots & \\ & & w_n \end{pmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \end{bmatrix} \times \left(\underbrace{\begin{bmatrix} w_1 & \cdots & & w_n \\ w_1(x_1 - x) & \cdots & & w_n(x_n - x) \end{bmatrix}}_{2 \times n} \underbrace{\begin{bmatrix} 1 & (x_1 - x) \\ \vdots & \vdots \\ 1 & (x_n - x) \end{bmatrix}}_{n \times 2} \right)^{-1}$$

$$\begin{pmatrix} 1 & \cdots & 1 \\ (x_1 - x) & \cdots & (x_n - x) \end{pmatrix} \begin{pmatrix} w_1 & & \\ & \ddots & \\ & & w_n \end{pmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$$= [1\,0] \begin{pmatrix} \sum w_i & \sum w_i (x_i - x) \\ \sum w_i (x_i - x) & \sum w_i (x_i - x)^2 \end{pmatrix}^{-1} \begin{pmatrix} w_1 \dots & w_n \\ w_1 (x_1 - x) \dots & w_n (x_n - x) \end{pmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

(2×2 · (+1) · 2×n · n×1)

formula of inverse for 2×2 matrix

$$= [1\,0] \times \frac{1}{\sum w_i \sum w_i (x_i - x)^2 - \left(\sum w_i (x_i - x)\right)^2}$$

$$\times \begin{bmatrix} \sum w_i (x_i - x)^2 & -\sum w_i (x_i - x) \\ \sum w_i (x_i - x) & \sum w_i \end{bmatrix} \times \begin{bmatrix} \sum y_i w_i \\ \sum y_i w_i (x_i - x) \end{bmatrix}$$

(2×2 · 2×1)

$$= \frac{\sum w_i (x_i - x)^2 \sum y_i w_i - \sum w_i (x_i - x) \sum y_i w_i (x_i - x)}{\sum w_i \sum w_i (x_i - x)^2 - \left(\sum w_i (x_i - x)\right)^2}$$

$\left(\sum w_i (x_i - x)\right)^2 = \sum w_i (x_i - x) \sum w_i (x_i - x)$

$$= \frac{\sum w_i y_i \left(\sum w_i (x_i - x)^2 - (x_i - x) \sum w_i (x_i - x)\right)}{\sum w_i \left(\sum w_i (x_i - x)^2 - (x_i - x) \sum w_i (x_i - x)\right)}$$

(S_2: $\sum w_i (x_i - x)^2$; S_1: $\sum w_i (x_i - x)$)

$$= \frac{\sum_i^n K\left(\frac{x - x_i}{h}\right) \left(S_2(x) - (x_i - x) S_1(x)\right) \cdot y_i}{\sum_i^n K\left(\frac{x - x_i}{h}\right) \left(S_2(x) - (x_i - x) S_1(x)\right)}$$

c) Assuming that the residuals (left out from the interview models) have constant variance σ^2

⇒ derive mean and variance of the sampling distribution for the local polynomial estimate $\hat{f}(x)$ at some arbitrary point x

Again we have: $\boxed{\hat{f}(x_i) = g_x(x_i|\hat{a}) = R_{xi}\hat{a}}$

So that: • $E(\hat{f}(x_i)) = E(g_x(x_i|\hat{a})) = E(R_{xi}\hat{a})$

$$= R_{xi}\, E(\hat{a})$$

$$= R_{xi}\, E\left[(R_x^T \tilde{W} R_x)^{-1} R_x' \tilde{W} y\right]$$

$$= R_{xi}(R_x' \tilde{W} R_x)^{-1} R_x' \tilde{W}\, E(y)$$

$$\Rightarrow E(\hat{f}(x)) = \frac{\sum w_i(x) y_i}{\sum w_i(x)} = \tilde{w}^T y = \tilde{W}^T f(x)$$

• $V(\hat{f}(x_i)) = VAR(g_x(x_i|\hat{a})) = VAR(R_{xi}\hat{a})$

$$= R_{xi}\, VAR\left[(R_x' \tilde{W} R_x)^{-1} R_x' \tilde{W} y\right] R_{xi}'$$

$$= R_{xi}(R_x' \tilde{W} R_x)^{-1} R_x \tilde{W}\, V(y)\, \tilde{W}' R_x (R_x' \tilde{W} R_x)^{-1} R_{xi}'$$

$$= \sigma^2 R_{xi}(R_x' \tilde{W} R_x)^{-1} R_x \tilde{W} \tilde{W}' R_x (R_x' \tilde{W} R_x)^{-1} R_{xi}'$$

$$\Rightarrow V(\hat{f}(x)) = \sigma^2 \tilde{W}^T I \tilde{W} = \sigma^2 \|\tilde{w}\|_2^2 \quad \leq \sigma^2$$

D) $r = y - \hat{y} = y - Hy$ → projection matrix ("smoothing matrix")

$$\hat{\sigma}^2 = \frac{\|r\|_2^2}{n - 2\mathrm{tr}(H) + \mathrm{tr}(H'H)}$$

$$\Rightarrow E(\hat{\sigma}^2) = E\left(\frac{\|r\|_2^2}{n - 2\mathrm{tr}(H) + \mathrm{tr}(H'H)}\right)$$

$$= E\left(\frac{(y-Hy)^T(y-Hy)}{n - 2\mathrm{tr}(H) + \mathrm{tr}(H'H)}\right)$$

$$= E\left(\frac{y'y - 2y'Hy + y'H'Hy}{n - 2\mathrm{tr}(H) + \mathrm{tr}(H'H)}\right)$$

$$= \frac{E(y'y) - 2E(y'Hy) + E(y'H'Hy)}{n - 2\mathrm{tr}(H) + \mathrm{tr}(H'H)}$$

Hint: $E(x'Qx) = \mathrm{tr}(Q\Sigma) + \mu'\mu$, $E(y) = \mu$

$$= \frac{\mathrm{tr}(\Sigma) + \mu'\mu - 2\mathrm{tr}(H\Sigma) - 2\mu'H\mu + \mathrm{tr}(H'H\Sigma) + 2\mu'H'H\mu}{n - 2\mathrm{tr}(H) + \mathrm{tr}(H'H)}$$

poggia in $I\sigma^2 = \Sigma$

$$= \frac{\mathrm{tr}(\sigma^2 I) + \mu'\mu - 2\mathrm{tr}(H\sigma^2 I) - 2\mu'H\mu + \mathrm{tr}(H'H\sigma^2 I) + 2\mu'H'H\mu}{n - 2\mathrm{tr}(H) + \mathrm{tr}(H'H)}$$

plugging in $E(y) = \mu = f(x)$

$$= \frac{\operatorname{tr}(\sigma^2 I) + f(x)'f(x) - 2\operatorname{tr}(H\sigma^2 I) - 2f(x)'Hf(x) + \operatorname{tr}(H'H\sigma^2 I) + 2f(x)'H'Hf(x)}{n - 2\operatorname{tr}(H) + \operatorname{tr}(H'H)}$$

$$= \frac{\sigma^2[n - 2\operatorname{tr}(H) + \operatorname{tr}(H'H)] + f(x)'[I - 2H + H'H]f(x)}{n - 2\operatorname{tr}(H) + \operatorname{tr}(H'H)}$$

$$\approx \sigma^2 + \frac{f(x)'[(I-H)'(I-H)]f(x)}{n - 2\operatorname{tr}(H) + \operatorname{tr}(H'H)}$$

bias of the estimate

5) Write code to fit <u>local linear estimator</u> ⇒ D = 1

Apply this strategy to the dataset "utilities"

y = average daily gas bill in a given month

x = average temperature

with:
- gaussian kernel
- LOO CV method to chose h

see python code and jupyter notebook

7) see jupyter notebook

The assumption of homoskedasticity does not look reasonable. The variance seem to decrease as the value on the x axis increase.

In class, we discussed about an alternative approach could be assuming $y = f(x) + e$

$e = g(x)$

9) see jupyter notebook

GAUSSIAN PROCESSES

gaussian process: collection of random variables

$\{f(x) : x \in X\}$

such that, for any finite collection of inputs $x_1 \dots x_N \in X$,

the random vector

$[f(x_1) \dots , f(x_N)]^T$ has a multivariate normal distribution

to ∞ dimensional space

$f \sim GP(m, C)$ for some mean function $m : X \to \mathbb{R}$

and some covariance function $C : X \times X \to \mathbb{R}^+$

and we have: $E(f(x)) = m(x_1)$

$cov(f(x_1), f(x_2)) = C(x_1, x_2) \quad \forall x_1, x_2$

↳ typically covariance function are those that are such that covariance decay when points are more far away.

⇒ high covariance when x_1 and x_2 are close to each other

A) MATERN CLASS of COVARIANCE FUNCTIONS:

⇒ class of covariance functions used in applications of multivariate statistical analysis on metric spaces (⇒ spatial statistics)

According to this class of covariance functions, covariance only depends on distance between points (⇒ stationary covariance).

general formula:

$$C_\nu(d) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\,\frac{d}{\rho}\right)^\nu \cdot K_\nu\left(\sqrt{2\nu}\,\frac{d}{\rho}\right)$$

K_ν: modified Bessel function of the second kind

Γ: gamma function

ρ, ν: positive parameters.

⇓

special case: SQUARED EXPONENTIAL

$$C_{SE}(x_1, x_2) = \tau_1^2 \exp\left\{-\frac{1}{2}\left(\frac{d(x_1,x_2)}{b}\right)^2\right\} + \tau_2^2\,\delta(x_1,x_2)$$

$$= \tau_1^2 \exp\left\{-\frac{1}{2}\left(\frac{\|x_1 - x_2\|_2}{b}\right)^2\right\} + \tau_2^2\,\delta(x_1,x_2)$$

τ_1^2, b, τ_2^2: hyperparameters

$\delta(x_1,x_2)$: Kronecker delta function

⇒ note: CSE usually generates functions that are "too smooth"

EXAMPLE: $\mathcal{X} = [0,1]$ ⇒ write a function that simulates a mean zero gaussian process on $[0,1]$

using CSE

now: ① write function that takes points in $\mathcal{X}$ (which now is simply the unit interval) and hyperparameters and returns the "value of the random process at each point".

⇒ usando covariance ecc cosa mi restituisce f?

and "plots" the gaussian process.

Recall that has a multivariate distribution normal (the vector)
→ the marginals will also be normal and maybe that is what I can plot?

⇒ mean and variance are enough to define $f(x)$

But ≠ plot: I need to simulate

Comment ② try out different hyperparameters (start with small)

③ change the covariance function

Note that:

model 1: $X \sim GP(0, C_\Theta)$

Θ = squared exponential

$\Theta = (b, \tau_1^2, \tau_2^2)$

$$C_\Theta(x_i, x_j) = \tau_1^2 \cdot \exp\left\{-\frac{1}{2b}\left(d(x_i, x_2)^2\right)\right\} + \tau_2^2\, \delta(x_i, x_j)$$

equivalent to

model 2: $\tilde{\Theta} = (b, \tau_1^2, 0)$

$$\tilde{X} \sim GP(0, C_{\tilde{\Theta}})$$

$$Z \sim N(0, \tau_z^2 I)$$

$$X = \tilde{X} + Z$$

Consider a different function for constructing the covariance:

$$C_{MS2}(x_1, x_2) = \gamma_1^2 \left\{ 1 + \frac{\sqrt{5}d}{b} + \frac{5d^2}{3b^2} \right\} \exp\left(-\frac{\sqrt{5}d}{b}\right) + \gamma_2^2 \delta(x_1, x_2)$$

with $d = |x_1 - x_2|$

B) Assume you observe the value of a Gaussian process:

$f \sim GP(m, C)$ at points $x_1 \dots x_N$

⇒ what is the conditional distribution of the value of the process at some new point x^*?

$P(f(x^*) \mid f(x_1) \dots f(x_N))$

(notation: $c(x_i, x_j) = c_{ij}$)

Note that we have (given the definition of GP)

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \\ f(x^*) \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \\ f(x^*) \end{bmatrix} \sim N\left(\begin{matrix} 0 \\ 0 \end{matrix} , \begin{matrix} \boxed{C(X,X)} & \overset{n\times 1}{C(X,X_*)} \\ \overset{1\times n}{C(X_*,X)} & C(x_*^*, X_*) \end{matrix} \right)$$

variance that the model imposes on the marginal of $y_1 \dots y_n$

$\Rightarrow p(y_1 \dots y_n) \sim N(0, k(X,X))$

and we want to look for the distribution of $f_* \mid y_1 \dots y_n$

Now recall a general result for normal multivariate:

when we have:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{matrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{matrix} \right)$$

we can state:

$$x_2 \mid x_1 \sim N\left(\mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(x_1 - \mu_1),\ \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right)$$

Applying this result to our case, we have:

$$f_{x^*|y_1 \dots y_n} \sim N(\text{mean}, \text{variance})$$

$$\text{mean} = \underset{1 \times n}{c(x^*, x)} \cdot (c(x,x))^{-1} y$$

$$\text{variance} = c(x^*, x^*) - c(x^*, x)\left[c(x,x)\right]^{-1} c(x, x^*)$$

c) We want to prove the following lemma:

if:

1) $y|\theta \sim MVN(R\theta, \Sigma)$ $\Rightarrow p(\theta|y) \sim NVN$

2) $\theta \sim NVN(m, V)$

with R, Σ, m, V constants

proof. we have:

$$p(\theta|y) = p(y|\theta) \cdot p(\theta)$$

$$\propto \exp\left\{-\frac{1}{2}\left[(y-R\theta)'\Sigma^{-1}(y-R\theta) + (\theta-m)'V^{-1}(\theta-m)\right]\right\}$$

$$= \exp\left\{-\frac{1}{2}\left[y'\Sigma^{-1}y - 2\theta'R'\Sigma^{-1}y - \theta'R'\Sigma^{-1}R\theta + \theta'V^{-1}\theta - 2m'V^{-1}\theta + m'V^{-1}m\right]\right\}$$

Add and subtract $2m'R'\Sigma^{-1}R\theta$

$$= \exp\{-\tfrac{1}{2}[y'\Sigma^{-1}y - 2\theta'R'\Sigma^{-1}y - \theta'R'\Sigma^{-1}R\theta + \theta'V^{-1}\theta - 2m'V^{-1}\theta + m'V^{-1}m - 2m'R'\Sigma^{-1}R\theta + 2m'R'\Sigma^{-1}R\theta]\}$$

$$= \exp\{-\tfrac{1}{2}[y'\Sigma^{-1}y - 2\theta'R'\Sigma^{-1}y - \theta'R'\Sigma^{-1}R\theta + \theta'V^{-1}\theta - 2m'V^{-1}\theta + m'V^{-1}m - 2m'R'\Sigma^{-1}R\theta + m'R'\Sigma^{-1}R\theta - \theta'R'\Sigma^{-1}Rm]\}$$

Add and subtract $y'\Sigma^{-1}Rm$

$$= \exp\{-\tfrac{1}{2}[y'\Sigma^{-1}y - 2\theta'R'\Sigma^{-1}y - \theta'R'\Sigma^{-1}R\theta + \theta'V^{-1}\theta - 2m'V^{-1}\theta + m'V^{-1}m - 2m'R'\Sigma^{-1}R\theta + m'R'\Sigma^{-1}R\theta - \theta'R'\Sigma^{-1}Rm - 2m'R'\Sigma^{-1}y + y'\Sigma^{-1}Rm + m'R'\Sigma^{-1}y]\}$$

Add and subtract $m'(R'\Sigma^{-1}R)m$

$$= \exp\{-\tfrac{1}{2}[\theta'V^{-1}\theta + \theta'R'\Sigma^{-1}R\theta - 2m'V^{-1}\theta - 2m'R'\Sigma^{-1}R\theta - y'\Sigma^{-1}R\theta + y'\Sigma^{-1}Rm + m'R'\Sigma^{-1}R\theta - \theta'R'\Sigma^{-1}y + \theta'R'\Sigma^{-1}Rm + m'R'\Sigma^{-1}y + y'\Sigma^{-1}y - 2m'R'\Sigma^{-1}y + m'R'\Sigma^{-1}Rm - m'R'\Sigma^{-1}Rm]\}$$

Combine piece together:

$$\propto \exp\{-\tfrac{1}{2}[\theta'(V^{-1}+R'\Sigma^{-1}R)\theta - 2m'(V^{-1}+R'\Sigma^{-1}R)\theta - y'\Sigma^{-1}R\theta + y'\Sigma^{-1}Rm + m'R'\Sigma^{-1}R\theta - \theta'R'\Sigma^{-1}y + \theta'R'\Sigma^{-1}Rm + m'R'\Sigma^{-1}y + y'\Sigma^{-1}y - 2m'R'\Sigma^{-1}y + m'R'\Sigma^{-1}Rm - m'R'\Sigma^{-1}Rm + m'R'\Sigma^{-1}Rm]\}$$

$$\propto \exp\{-\frac{1}{2}[(\theta-m)'(V^{-1}+R'\Sigma^{-1}R)(\theta-m)+(y-Rm)'\Sigma^{-1}R(\theta-m)$$
$$-(\theta-m)'R'\Sigma^{-1}(y-Rm)+(y-Rm)'\Sigma^{-1}(y-Rm)]\}$$

$$\propto \exp\{-\frac{1}{2}\left[\begin{pmatrix}\theta-m\\ y-Rm\end{pmatrix}'\begin{bmatrix}V^{-1}+R'\Sigma^{-1}R & -R'\Sigma^{-1}\\ -\Sigma^{-1}R & \Sigma^{-1}\end{bmatrix}\begin{pmatrix}\theta-m\\ y-Rm\end{pmatrix}\right]\}$$

$$\Rightarrow p(\theta,y) \propto \exp\{-\frac{1}{2}(\quad)\}$$

$\Rightarrow$ kernel of a multivariate normal with:

- mean $= \begin{bmatrix} m \\ Rm \end{bmatrix}$
- variance $= \begin{bmatrix} V^{-1}+R'\Sigma^{-1}R & -R'\Sigma^{-1} \\ -\Sigma^{-1}R & \Sigma^{-1} \end{bmatrix}^{-1}$

Alternatively, we could also note:

$$\begin{matrix} y \\ \theta \end{matrix} = \begin{bmatrix} R \\ I \end{bmatrix}\theta + \begin{bmatrix} I \\ 0 \end{bmatrix}\varepsilon$$

which is an affine transformation of a multivariate normal
$\Rightarrow$ from previous exercise we know this is still a multivariate normal.

NON PARAMETRIC REGRESSION & GAUSSIAN SMOOTHING

model: $y_i = f(x_i) + \varepsilon_i$, $\varepsilon_i \sim N(0, \sigma^2)$

$f \sim GP(0, C)$

A)

we observed $x_1 \dots x_n$ → want to make inference on $f(x_1) \dots f(x_n)$

⇒ what is the posterior?

$p(f|y) \propto p(y|f) \cdot p(f)$ — since is for a finite vector of observations it's just a multivariate normal

$\propto \exp\{-\frac{1}{2}(y-f(x))'(I\sigma^2)^{-1}(y-f(x)) - \frac{1}{2}(f(x)-0)'C^{-1}(f(x)-0)\}$

$\propto \exp\{-\frac{1}{2}[f(x)'\sigma^2 I f(x) + f'(x)'C^{-1}f(x) - f(x)'(I\sigma^2)^{-1}y]$

⇒ $f|y \sim MVN(\text{mean}, \text{variance})$

variance: $(\sigma^{-1}I)^{-1} + C^{-1})^{-1} = \left(\frac{I}{\sigma^2} + C^{-1}\right)^{-1}$

mean: $(\sigma^2 I)^{-1} + C^{-1})^{-1} \cdot (\sigma^2 I)^{-1} y = C(C+\sigma^2 I)^{-1}$

B) Now we are interested in the predictive distribution:

note that since $f(x_1) \dots f(x_n) f(x^*)$ is a GP we have:

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \\ f(x^*) \end{bmatrix} \sim N\left(\begin{matrix} 0 \\ \\ 0 \end{matrix}, \begin{matrix} (C(X,X) + \sigma^2 I) & C(X, X^*) \\ \\ C(X^*, X) & C(X^*, X^*) \end{matrix} \right)$$

Since because of well known rules of multivariate normals, we have

$\underline{y} \sim N(0, \sigma^2 I + C(X,X))$ → see ex D)

Now using again the rule recalled in the exercise above, we have:

$f^* | y \sim MVN$ (mean, variance)

with: $\text{mean} = C(X^*, X)\left(C(X,X) + \sigma^2 I \right)^{-1} y$

$$\text{var} = C(X^*, X^*) - C(X^*, X)\left[C(X,X) + \sigma^2 I \right]^{-1} C(X, X^*)$$

note that it can be written as $\sum_i^n w_i y_i$

⇒ linear smoother

C) see jupiter notebook

D) Now assume $C = \text{Matern}(5/2)$

Derive the marginal of $y_1 \dots y_n$.

Recall well known result:

If $p(a|b) = N(a|Ab, S)$
$p(b) = N(b|\mu, \Sigma)$

$\Rightarrow p(a) = \int p(a|b)\, db = N(A\mu, A\Sigma A' + S)$

So that in our case we have:

$$p(y) = N(0, (C + \sigma^2 I))$$

Moreover, recall that the matern $(5/2)$ has the following structure:

$(\sigma^2 I) +$ (underbrace) "included in the marginal variance"

$$\begin{bmatrix} C_{11} & C_{21} & \\ C_{12} & \ddots & \\ & & \ddots \end{bmatrix}$$

$C_{ii} = \tau_1^2 + \tau_2^2$

$\Rightarrow$ marginal variance will be $\sigma^2 + \tau_1^2 + \tau_2^2$

$$C_{ij} = \tau_1^2 \cdot \left(1 + \frac{\sqrt{5}\, d_{ij}}{b} + \frac{5 d_{ij}^2}{3b^2}\right) \cdot \exp\left(-\frac{\sqrt{5}\, d_{ij}}{b}\right)$$

E) we are asked to evaluate the log of the marginal likelihood setting $\gamma_i^2 = 0$.

in D) we derived: $p(y) = \mathcal{N}(0, (C + \partial^2 I))$.

Hence we have:

$$\log(p(y)) = \log\left(|2\pi (C+\partial^2 I)|^{-\frac{1}{2}} \cdot \exp\left\{ -\frac{1}{2} y' (C+\partial^2 I)^{-1} y \right\} \right)$$

$$= -\frac{1}{2} \log\left(|2\pi (C+\partial^2 I)| \right) - \frac{1}{2} y' (C+\partial^2 I)^{-1} y$$

$$= -\frac{n}{2} \log(2\pi) - \frac{1}{2} \log(|C+\partial^2 I|) - \frac{1}{2} y' (C+\partial^2 I)^{-1} y$$

For implementation see jupyther notebook and python code