Lecture note: Gaussian processes

in HT2022 Advanced Probabilistic Machine Learning

Zheng Zhao[1]
Uppsala Universitet
26 September, 2022

Target audience: students with background, for example, machine learning, engineering, statistics, and physics.
Time: Around 3 hours.

1 Introduction

Let us start by considering an example function $X\colon [0,\infty) \to \mathbb{R}$ defined by

$$X(t) = a\,t + b, \tag{1}$$

where a and b are some real parameters. Now that if we let $a \sim \mathrm{N}(m_a, v_a)$ and $b \sim \mathrm{N}(m_b, v_b)$ be two independent Normal random variables instead of fixed reals, X then becomes a random variable as well. This X is the basic object that we have studied in the previous lecture on Bayesian linear regression. Furthermore, we can view X as a function-valued random variable, since whenever we draw a pair of samples from $\{a, b\}$ we obtain an affine function which is a sample from X; this is illustrated in Figure 1.

The random variable X in this example is a Gaussian process (GP), the definition of which we detail in the following section.

2 Gaussian processes

Indeed we can define GP as function-valued random variable following the heuristics in Introduction. Say, for example, if X takes value in a Hilbert space of functions H endowed with inner product $\langle \cdot, \cdot \rangle$, then we say that X is H-Gaussian if $\langle X, u \rangle$ is Gaussian in $\mathbb{R}$ for all $u \in H$, and you perhaps very often see "GP is an infinite-dimensional prior" in many textbooks/lecture notes. However, the precise definition of GP in this routine requires to introduce infinite-dimensional Gaussian measures (see, e.g., Kuo, 1975) which is not a prerequisite for the target audience of this lecture note. Therefore, we give a definition of GP in terms of finite-dimensional Gaussian distributions as follows, since we can view

[1]Contact: firstname.lastname@it.uu.se or https://zz.zabemon.com. To download the LaTeX source and companion codes of this lecture note, visit `https://github.com/spdes/gp`.

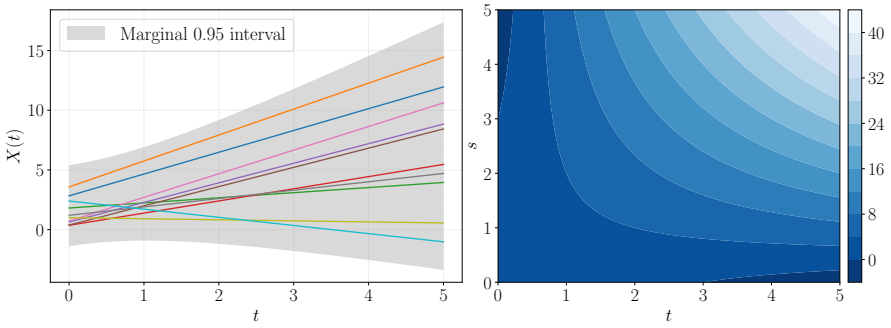

Figure 1: Left: samples (in different colours) of X in Equation (1) are affine functions, where we let $m_a = 1$, $v_a = 1$, $m_b = 2$, and $v_b = 3$. Right: the covariance function of X.

stochastic process as a collection of random variables (i.e., $X \coloneqq \{X(t) \colon t \geq 0\}$) as well.

Definition 1 (Gaussian process)**.** *A stochastic process $X \coloneqq \{X(t) \in \mathbb{R} \colon t \geq 0\}$ is said to be a Gaussian process on $[0, \infty)$, if for every $T \geq 1$ and reals $t_1 < t_2 < \cdots < t_T \in [0, \infty)$, the random variables $X(t_1), X(t_2), \ldots, X(t_T)$ are jointly Normal distributed (Karatzas and Shreve, 1991).*

Since the distribution of Normal random variables are determined by their means and covariances, we usually translate Definition 1 by a shorthand (commonly seen in the statistical machine learning community, see, e.g., Rasmussen and Williams, 2006)

$$X(t) \sim \mathrm{GP}\big(m(t), C(t, s)\big), \tag{2}$$

where m and C stand for the mean and covariance functions of X, respectively, and they are defined by

$$\begin{aligned} m(t) &\coloneqq \mathbb{E}[X(t)], \\ C(t, s) &\coloneqq \mathrm{Cov}[X(t), X(s)], \end{aligned}$$

for all t, s. The mean and covariance functions also determine the probabilistic properties of X, such as continuity, volatility, and stationarity.

Now let us use notation $\mathrm{N}(x_{1:T} \mid m_{1:T}, C_{1:T})$ to denote the (multivariate) Normal probability density function (PDF) of $X_{1:T} \coloneqq [X(t_1), X(t_2), \ldots, X(t_T)] \in \mathbb{R}^T$ with its mean vector $m_{1:T} \in \mathbb{R}^T$ and covariance matrix (also known as the Gram matrix or kernel matrix) $C_{1:T} \in \mathbb{R}^{T \times T}$. Definition 1 or Equation (2) *essentially* means that

$$\begin{aligned} X_{1:T} \sim \mathrm{N}\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_T \end{bmatrix} \;\middle|\; \begin{bmatrix} m(t_1) \\ m(t_2) \\ \vdots \\ m(t_T) \end{bmatrix}, \begin{bmatrix} C(t_1, t_1) & C(t_1, t_2) & \cdots & C(t_1, t_T) \\ C(t_2, t_1) & C(t_2, t_2) & \cdots & C(t_2, t_T) \\ \vdots & \vdots & \ddots & \vdots \\ C(t_T, t_1) & C(t_T, t_2) & \cdots & C(t_T, t_T) \end{bmatrix} \right) \\ \coloneqq \mathrm{N}(x_{1:T} \mid m_{1:T}, C_{1:T}). \end{aligned} \tag{3}$$

Please note the emphasised "essentially". This warns that one should not take Definition 1 and the multivariate Normal PDF representation of GP equivalent, because the PDF in Equation (3) is well-defined only if the covariance matrix is positive definite. There is a plenty of counterexamples whose GP covariance matrices are not positive definite. For instance, let $X_0 \sim \mathrm{N}(0,1)$ be a standard Gaussian and define $X(t) = X_0$ for all t. By Definition 1, this process X is a GP, however, the covariance matrix of X at any times $t_1, t_1, \ldots, t_T$ is an all-one matrix which is singular, hence, this GP X does not have a PDF alike Equation (3).

We can also define GP in terms of characteristic function. More precisely, we say that X is a GP if for any $T \geq 1$ and times $t_1, t_2, \ldots, t_T$, the joint characteristic function $\phi_{X_{1:T}}$ of $X_{1:T}$ at these times follows

$$\begin{aligned}\phi_{X_{1:T}}(z_{1:T}) &\coloneqq \mathbb{E}\big[\exp(\mathrm{i}\, z_{1:T}^{\mathsf{T}}\, X_{1:T})\big] \\ &= \exp\left(\mathrm{i}\sum_{k=1}^{T} m(t_k)\, z_k - \frac{1}{2}\sum_{j,k=1}^{T} C(t_j, t_k)\, z_j\, z_k\right) \\ &= \exp\Big(\mathrm{i}\, z_{1:T}^{\mathsf{T}}\, m_{1:T} - \frac{1}{2}\, C_{1:T}\, z_{1:T}\Big).\end{aligned} \tag{4}$$

The definition using the equation above is equivalent to Definition 1, moreover, the covariance matrix $C_{1:T}$ needs not to be positive definite. The characteristic function mainly is useful in analysis, for example, to show the distribution of a limit of random variables, we can conveniently show the limit of the characteristic function of the random variable in the sequence, thanks to the dominated convergence theorem.

2.1 Example GPs

For the sake of pedagogy, we exemplify and plot a few commonly seen GPs whose mean and covariance functions are given.

Example 2 (Affine GP)**.** *Recall the GP X defined in Equation* (1)*. By its definition, we find that its mean and covariance functions are*

$$\begin{aligned}m(t) &= m_a\, t + m_b, \\ C(t,s) &= \mathrm{Cov}[(a\, t + b), (a\, s + b)] \\ &= v_a\, t\, s + v_b - (t + s)\, m_a\, m_b.\end{aligned} \tag{5}$$

The samples and covariance matrix of this GP are plotted in Figure 1.

Example 3 ((Fractional) Brownian motion)**.** *Brownian motion (also known as Wiener process) is a fundamental stochastic process defined from independent Normal increments (see, e.g., Karatzas and Shreve, 1991, for the precise definition). Its definition implies that it is a GP following the mean and covariance functions given by*

$$\begin{aligned}m(t) &= 0, \\ C(t,s) &= \min(t,s).\end{aligned}$$

The sample path of a Brownian motion is rough, and the rough means that the path is almost surely non-differentiable and is Hölder continuous of degree less than 0.5*. To generalise Brownian motion with any degree of Hölder continuity, we use the fractional Brownian motion, the covariance function of which is given by*

$$C(t, s; r) = 0.5\,(|t|^{2\,r} + |s|^{2\,r} - |t - s|^{2\,r}),$$

where $r \in (0, 1)$ *is the parameter that controls the continuity. When* $r = 0.5$*, the fractional Brownian motion reduces to a standard Brownian motion. Samples and covariance matrices of the fractional Brownian motion with* $r = 0.5$ *and* 0.9 *are shown in Figure 2.*[2]

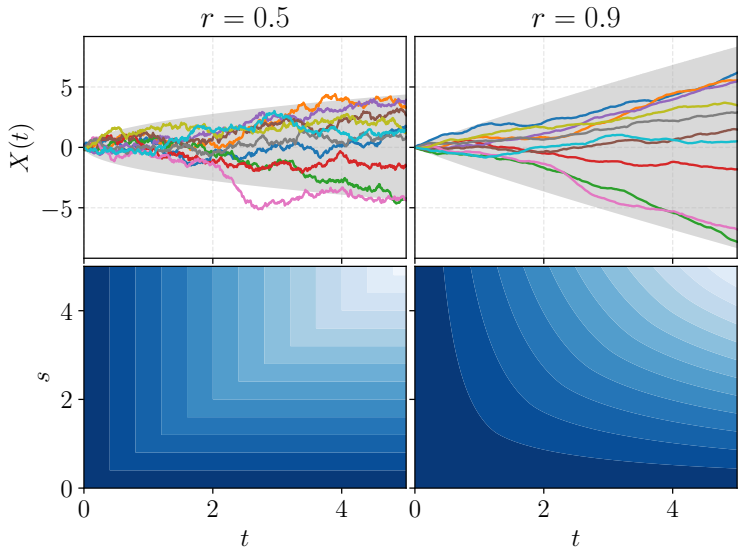

Figure 2: Samples (top) and covariance matrices (bottom) of the standard Brownian motion (left) and a fractional Brownian motion with $r = 0.9$ (right).

Example 4 (Ornstein–Uhlenbeck/Exponential GP)**.** *Exponential GPs are such that their covariance functions are of the form*

$$C(t, s; \ell, \sigma) = \sigma^2 \exp\Big(-\frac{|t - s|}{\ell}\Big), \tag{6}$$

where ℓ *and* σ *refer to the length scale and magnitude (scale) parameters, respectively. The exponential GP is also found under another name Ornstein–Uhlenbeck process in the stochastic calculus and quantitative finance communities, as it is the solution to a linear stochastic differential equation (see also Example 25).*

[2] Colourbars from now on are omitted for aesthetics, please compared to Figure 1 or generate the figure with colourbar by yourself from the companion codes.

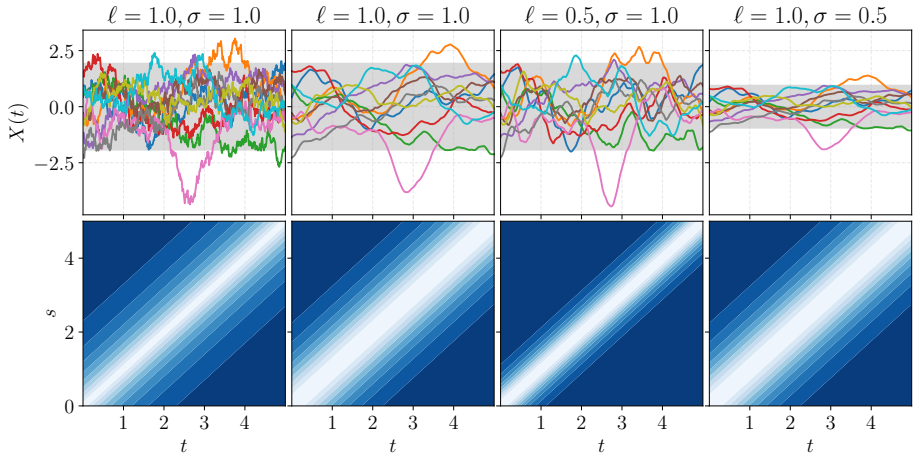

Figure 3: Samples (top) and covariance matrices (bottom) from Matérn GPs with $\nu = 1/2$ (first column) and $\nu = 3/2$ (other three columns).

Example 5 (Matérn GP)**.** *We can generalise the exponential GP with varying degree of smoothness by using the Whittle–Matérn covariance function*

$$C(t, s; \ell, \sigma, \nu) = \frac{\sigma^2 \, 2^{1-\nu}}{\Gamma(\nu)} \Big(\frac{\sqrt{2}\,\nu\,|t-s|}{\ell} \Big)^{\nu} \mathrm{K}_{\nu} \Big(\frac{\sqrt{2}\,\nu\,|t-s|}{\ell} \Big), \tag{7}$$

where Γ is the Gamma function, and K_ν is the modified Bessel function of the second kind. The parameter $\nu \in \{\frac{1}{2}, \frac{3}{2}, \ldots\}$ controls the smoothness, in the way that the GP is $\nu - \frac{1}{2}$ times differentiable. When $\nu = 0.5$, the Matérn covariance function reduces to Equation (6)*. The Matérn GP family has two distinctive parameters, that are, the length scale ℓ and magnitude (scale) σ. These two parameters, in a loose sense, control the "correlation/variability/volatility" of the GP in the horizontal and vertical directions. This is illustrated in Figure 3.*

Regarding the Matérn GPs, it is of interests to ask if we can let their parameters ℓ and σ be functions of time. This is crucial in real-world modelling, because the underlying processes for many models are non-stationary, for example, in geophysics (Higdon et al., 1999) and fluid mechanics (Monin and Yaglom, 1971). The answer to the question is false, since the Matérn covariance function is in general not positive definite if we replace the parameters by any arbitrary function (e.g., $t \mapsto \ell(t)$). To solve this problem, Paciorek and Schervish (2004) craft a generalisation of the Matérn covariance function so that one can transform any stationary covariance function into a non-stationary one. Zhao et al. (2021) show another way to construct non-stationary GPs from stochastic differential equations without explicitly using non-stationary covariance functions (see also Section 7.2).

Example 6 (Radial basis GP)**.** *Since the Normal PDF is positive definite, we*

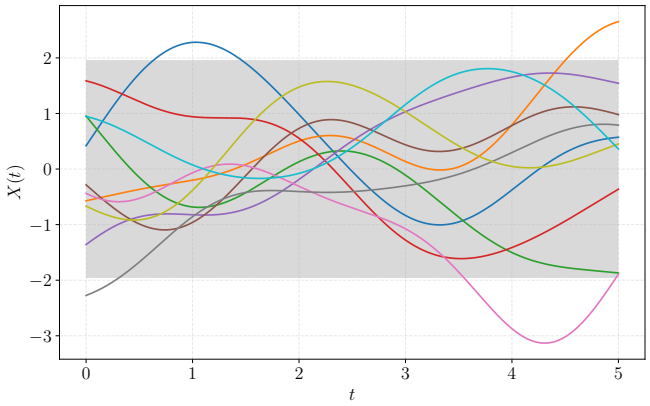

Figure 4: Samples from an RBF GP with $\ell = 1$ and $\sigma = 1$.

can borrow it as the covariance function of a GP too:

$$C(t, s; \ell, \sigma) = \sigma^2 \exp\left(-\frac{1}{2}\left(\frac{t-s}{\ell}\right)^2\right), \tag{8}$$

where parameters ℓ and σ are similar to that of the Matérn family. Note that this covariance function is infinitely smooth in its two arguments. This implies that the GP using this covariance function is infinitely smooth as well. In the literature, this type of covariance function is often referred as "squared exponential", but I personally do not concur with this name, because it is really not by taking the square of the exponential covariance function.

Many physicists argue against using the RBF covariance function in physical applications (see, e.g., Stein, 1999), as the RBF GP is unrealistically too smooth. For example, we should not use the RBF GP to model the diffusion dynamics of particle/molecular in microscopic scale, because motion of particles/molecular is inherently not smooth. On the other hand, the fractional Brownian motion or the Matérn GPs may be used in this case, thanks to their capability of tuning the path regularity.

The RBF GP may also give numerical problems in the covariance matrix. The condition number of the RBF covariance matrix scales badly in the number of data points T.

Example 7 (Sinusoidal GP)**.** *Analogously to the example in Introduction, we can define a sinusoidal GP by*

$$X(t) = a\sin(t) + b\cos(t), \tag{9}$$

where $a, b \sim \mathrm{N}(0, 1)$ are independent. The mean and covariance functions of this GP are

$$\begin{aligned} m(t) &= 0, \\ C(t, s) &= \cos(s - t). \end{aligned}$$

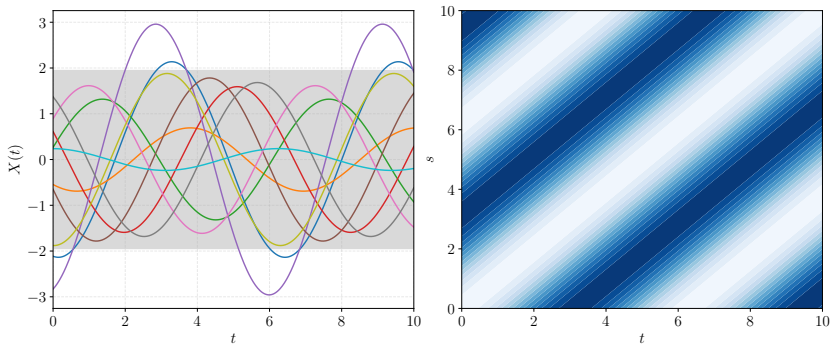

Figure 5: Samples (left) and covariance matrix (right) from a sinusoidal GP.

By the definition of this GP, we expect that the samples of it are sinusoidals with varying amplitude and phase. This is illustrated in Figure 5.

The examples above illustrate samples of GPs, yet, we have not learnt how to draw the GP samples. Drawing GP samples is trivial if we know the explicit construction of the GP, for example, in Equations (1) and (9). However, in practice we do not know the explicit construction of GP, but only a pair of mean and covariance functions of it. In the following section, we show a number of approaches to draw samples from a GP when its mean and covariance functions are given.

3 Draw GP samples

Suppose that we are given a GP

$$X(t) \sim \mathrm{GP}\big(m(t), C(t, t')\big)$$

with mean function m and covariance function C. We would like to draw samples of X evaluated at times $t_1, t_2, \ldots, t_T$. By the definition of GP, this is the same as with drawing samples from a collection of joint Normal random variables $X_{1:T} \coloneqq \begin{bmatrix} X(t_1) & X(t_2) & \cdots & X(t_T) \end{bmatrix} \in \mathbb{R}^T$ which follows a multivariate Normal PDF $\mathrm{N}(x_{1:T} \mid m_{1:T}, C_{1:T})$ (if the PDF exits). Of course in Python, you simply call `numpy.random.multivariate_normal`, but how would you actually implement this function?

3.1 Cholesky decomposition

Let $\xi_{1:T} \coloneqq \begin{bmatrix} \xi_1 & \xi_2 & \cdots & \xi_T \end{bmatrix} \in \mathbb{R}^T$ be a collection of T independent identically distributed standard Normal random variables (i.e., $\mathrm{N}(0, 1)$), and suppose that we know how to draw samples from them. Then we can leverage $\xi_{1:T}$ to draw $X_{1:T}$ by the Cholesky decomposition $\sqrt{C_{1:T}}$ of the covariance matrix $C_{1:T}$. More

precisely, the distribution of $X_{1:T}$ is the same as with

$$m_{1:T} + \sqrt{C_{1:T}}\,\xi_{1:T}, \tag{10}$$

where $\sqrt{C_{1:T}}$ is defined via $\sqrt{C_{1:T}}\,\sqrt{C_{1:T}}^{\mathsf{T}} = C_{1:T}$ not the elementwise square root of $C_{1:T}$. This is true because any linear transformation of Normal random variables are Normal too, and the mean and covariance of the random variable $m_{1:T} + \sqrt{C_{1:T}}\,\xi_{1:T}$ match $m_{1:T}$ and $C_{1:T}$, respectively. This algorithm is summarised as follows.

Algorithm 1: Draw a GP sample using Cholesky decomposition

Inputs: Times $t_1, t_2, \ldots, t_T$, mean function $t \mapsto m(t)$, and covariance function $t, s \mapsto C(t, s)$

1 Compute $m_{1:T}$ and $C_{1:T}$ at the times;
2 Compute Cholesky decomposition $\sqrt{C_{1:T}}$;
3 Draw $\xi_{1:T}$;
4 $X_{1:T} = m_{1:T} + \sqrt{C_{1:T}}\,\xi_{1:T}$;
5 **return** $X_{1:T}$

It is important to point out that we can understand this algorithm in terms of orthonormalisation (e.g., Gram–Schmidt process). To see this, let us first recall a basic result in linear algebra. Suppose that $\phi_1, \phi_2, \ldots, \phi_T$ are linearly independent basis functions, and define an inner product $\langle \cdot, \cdot \rangle$. Then we can transform these linearly independent basis functions into a system of orthonormal basis by the Gram matrix of ϕ's. Formally, the Gram matrix is

$$G \coloneqq \begin{bmatrix} \langle \phi_1, \phi_1 \rangle & \langle \phi_1, \phi_2 \rangle & \cdots & \langle \phi_1, \phi_T \rangle \\ \langle \phi_2, \phi_1 \rangle & \langle \phi_2, \phi_2 \rangle & \cdots & \langle \phi_2, \phi_T \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \phi_T, \phi_1 \rangle & \langle \phi_T, \phi_2 \rangle & \cdots & \langle \phi_T, \phi_T \rangle \end{bmatrix},$$

and the new functions $\psi_1, \psi_2, \ldots, \psi_T$ defined via

$$\psi_i \coloneqq \sum_{j=1}^{T} (\sqrt{G}^{-1})_{ij}\, \phi_j, \quad i = 1, 2, \ldots, T,$$

are orthonormal in terms of $\langle \cdot, \cdot \rangle$. Please prove this result as an exercise. If you prefer to use the vector notation, the result writes

$$\begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_T \end{bmatrix} = \sqrt{G}^{-1} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_T \end{bmatrix}.$$

To apply this result in our case, let us define the functions by $\phi_i(x) \coloneqq x_i - m_i$ which selects the i-th element of its argument $x \in \mathbb{R}^T$ then subtracts the corresponding mean component. It is clear that $\phi_1, \phi_2, \ldots, \phi_T$ are linearly independent. For any two (Normal) random variables Y and Z, let us define their inner product by $\langle Y, Z \rangle \coloneqq \mathbb{E}[Y\,Z]$, then $\langle \phi_i(X_{1:T}), \phi_j(X_{1:T}) \rangle = \mathbb{E}[(X_i - m_i)\,(X_j - m_j)] = C(t_i, t_j)$. Therefore, the Gram matrix is the same as with our covariance matrix. Let $\xi_i \coloneqq \psi_i(X_{1:T})$. Evidently, $\psi_1, \psi_2, \ldots, \psi_T$ are orthonormal in the inner product from the orthonormalisation process. Finally, we have

$$X_{1:T} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_T \end{bmatrix} + m_{1:T} = \sqrt{C_{1:T}} \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_T \end{bmatrix} + m_{1:T}$$

which equals to Equation (10). Essentially, we use the inverse of the Cholesky decomposition $\sqrt{C_{1:T}}^{-1}$ of the covariance matrix to whitten $X_{1:T}$. This is also equivalent to say that we use $\sqrt{C_{1:T}}$ to transform the whittened $\xi_{1:T}$ back to $X_{1:T}$.

It is also worth noting a numerical issue of the Cholesky approach. This approach is sensitive to the condition number of the covariance matrix. For instance, when the times $t_1, t_2, \ldots, t_T$ are densely located, the covariance matrix $C_{1:T}$ can be numerically ill-conditioned.

3.2 Spectral decomposition

We can generalise the Cholesky decomposition algorithm by eigendecompositions of the covariance matrix. Since $C_{1:T}$ is symmetric positive semi-definite, its eigenvalues are all real and non-negative hence, we can decomposite it by

$$C_{1:T} = U\,V\,U^{\mathsf{T}},$$

where columns in $U \in \mathbb{R}^{T \times T}$ are the (orthonormal) eigenvectors, and $V \in \mathbb{R}^{T \times T}$ is a diagonal matrix containing the corresponding eigenvalues. Then,

$$m_{1:T} + U\sqrt{V}\,\xi_{1:T}$$

has the same distribution as $X_{1:T}$. We can also equivalently write the equation above in another fashion if you prefer

$$m_{1:T} + \sum_{i=1}^{T} \sqrt{v_j}\,\xi_j\,u_j,$$

where $v_j \geq 0$ and $u_j \in \mathbb{R}^T$ are the j-th eigenvalue and eigenvector, respectively.

Compared to the Cholesky decomposition approach, this spectral decomposition allows $C_{1:T}$ to have zero eigenvalue(s).

3.3 Computational complexity

The computational complexity of Algorithm 1 is $O(T^3)$, because the dominating procedure in the algorithm is computing the Cholesky decomposition $\sqrt{C_{1:T}}$ which is (nearly) cubic in T. The spectral decomposition approach has a similar cubic complexity too. This means that the sampling algorithms presented above are computationally invalid if the number of times T is large, not to mention that it is hard to store a dense matrix $C_{1:T}$ of huge size $T \times T$ nevertheless. Therefore, in the following three sections, we show a few alternative sampling methods that suffer less from this computational problem.

3.4 Approximate sampling with Lánczos interation

Recall the spectral decomposition of $X_{1:T}$ formulated previously:

$$m_{1:T} + \sum_{i=1}^{T} \sqrt{v_j}\, \xi_j\, u_j.$$

If T is large, we may approximate the decomposition by a truncation

$$\widehat{X}_m \coloneqq m_{1:T} + \sum_{i=1}^{m} \sqrt{v_j}\, \xi_j\, u_j$$

for some small number $m \ll T$, so that we do not need to compute all the eigenvalues and eigenvectors. But, how do we find/select the number m, and how to select the eigenvalues $\{v_i\}_{1=1}^{m}$ in order to have a good approximation? A heuristic solution is to sort all the eigenvalues $\{v_i\}_{1=1}^{T}$ then select the m biggest ones. But to sort them, we need to compute all the eigenvalues which is the problem that we were trying to avoid.

We could find the m-most useful eigenvalues (and eigenvectors) by using the Lánczos algorithm (see, Golub and van Loan, 2013, Ch. 10). This algorithm forms a tridiagonal matrix $\widehat{V} \in \mathbb{R}^{m \times m}$ and another orthonormal matrix $\widehat{U} \in \mathbb{R}^{T \times m}$ in the way that

$$\widehat{V} = \widehat{U}^{\mathsf{T}}\, C_{1:T}\, \widehat{U}.$$

This algorithm is fast, since it can find the matrix $\widehat{U}$ and the non-zero elements of $\widehat{V}$ in one loop of size m. Furthermore, if v and u is a pair of eigenvalue and eigenvector of $\widehat{V}$, then v and $\widehat{U}\, u$ are an pair of eigenvalue and eigenvector of $C_{1:T}$ too. Therefore, we can use this algorithm to approximate the spectral decomposition of $C_{1:T}$ by a smaller decomposition problem of $\widehat{V}$. Since $\widehat{V}$ is tridiagonal, finding its eigenvalues and eigenvectors can be far cheaper than the original cubic complexity. For instance, Coakley and Rokhlin (2013) compute the eigenvalues in a subquadratic complexity. However, the downside of the Láczos approach is that it is picky for $C_{1:T}$, the matrix of which should have extremal eigenvalues.

In Figure 6, we show an example of applying the Lánczos algorithm to make (approximate) samples from a Matérn GP. From this figure, we find that the

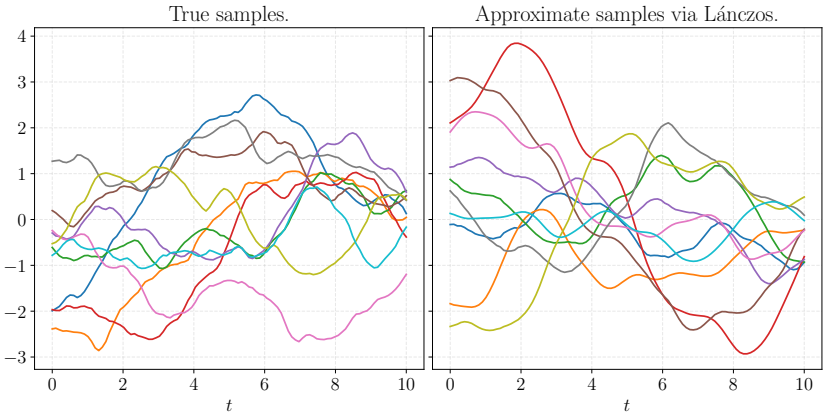

Figure 6: Approximate samples by using Lánczos iteration for a Matérn $\nu = 3/2$ GP. In this demonstration, we let $T = 100$ and $m = 20$.

samples produced by the Lánczos algorithm approximate the true samples to a reasonable extent. However, since we have discarded many eigenvalues, the path of the approximate sample appears smoother than that of the true sample.

For the sake of simplicity, we prefer not to detail more about the Lánczos algorithm, but we kindly refer the readers to, for example, the monograph by Golub and van Loan (2013) for more insights.

3.5 Kosambi–Karhunen–Loève expansion

The crux of the mentioned computational problem is that we were thinking the stochastic process as a collection of random variables, because sampling from *high-dimensional* vector space is challenging. To avoid this problem, we now need to rethink the stochastic process as a function-valued random variable. That is, instead of drawing sample of the process evaluated at finite times, we draw the sample path entirely as a function from an *infinite-dimensional* space. We introduce the Kosambi–Karhunen–Loève (KKL) expansion as follows.

Suppose that $X\colon \Omega \to L^2(D)$ is a square-integrable process (you can find the precise definition of "square-integrable L^2" in many probability theory textbooks) on some domain $D \subset [0, \infty)$, and that $\{\varphi_1, \varphi_2, \ldots\}$ is any set of orthonormal basis of $L^2(D)$, then we can expand the process in the way that

$$X(t) = m(t) + \sum_{i=1}^{\infty} \lambda_i \, \varphi_i(t),$$

for all $t \in D$, where every coefficient λ_i is random and is given by

$$\lambda_i = \langle X - m, \varphi_i \rangle_{L^2(D)}.$$

There are of course many orthonormal basis systems in the space. Since in our context we know the covariance function C of the process, we can choose

$\{\lambda_1, \lambda_2, \ldots\}$ and $\{\varphi_1, \varphi_2, \ldots\}$ from the eigenvalues and eigenfunctions of an integral operator K defined by

$$(Kf)(t) \coloneqq \int_D C(t,s)\, f(s)\, \mathrm{d}s, \quad f \in L^2(D).$$

Now let $\{v_i, \varphi_i\}_{i=1}^{\cdots}$ be the eigenvalues and eigenfunctions of K, then the KKL expansion is such that

$$X(t) = m(t) + \sum_{i=1}^{\infty} \sqrt{v_i}\, \xi_i\, \varphi_i(t),$$

where $\mathbb{E}[\xi_i] = 0$, $\mathrm{Var}[\xi_i] = 1$, and $\{\xi_1, \xi_2, \ldots\}$ are mutually uncorrelated (prove this as an exercise). If X is a GP, $\{\xi_1, \xi_2, \ldots\}$ are then independent standard Normal random variables. The covariance function is also such that $C(t,s) = \sum_{i=1}^{\infty} v_i\, \varphi_i(t)\, \varphi_j(s)$ by Mercer's theorem.

The KKL expansion implies that we can approximate samples of X by samples from independent Normal random variables which are easy to sample from. We summarise the algorithm in the following.

Algorithm 2: Approximate GP using KKL expansion

Inputs: Mean function $t \mapsto m(t)$, and covariance function $t, s \mapsto C(t,s)$, expansion order N.

1 Derive the eigenvalues and eigenfunctions of the oprator K, then order them and select the first N ones $\{v_i, \varphi_i\}_{i=1}^{N}$;
2 Draw independent Normal random variables $\{\xi_1, \xi_2, \ldots, \xi_N\}$;
3 **Function** $X(t)$*:*
4 | **return** $m(t) + \sum_{i=1}^{N} \sqrt{v_i}\, \xi_i\, \varphi_i(t)$
5 **end**
6 **return** X

The downside of the KKL approach is that we need to derive the eigenvalues and eigenfunctions of the integral operator from the given covariance function. This is analytically doable only for a few isolated cases, for example, the expansion of Brownian motion as shown in the following example. Otherwise, one has to approximate them numerically.

Example 8. *Let $C(t,s) = \min(t,s)$, and $D = [0,1]$, then the eigenfunctions and eigenvalues of the operator K are given by*

$$\varphi_i(t) = \sqrt{2} \sin\left(\left(i - \frac{1}{2}\right) \pi\, t\right),$$

$$v_i = \frac{1}{\left(i - \frac{1}{2}\right)^2 \pi^2},$$

for $i = 1, 2, \ldots$ and $t \in [0,1]$. See, for example, Lord et al. (2014, pp. 205) for the derivation. We illustrate an example of this expansion in Figure 7 with increasing order of truncation N.

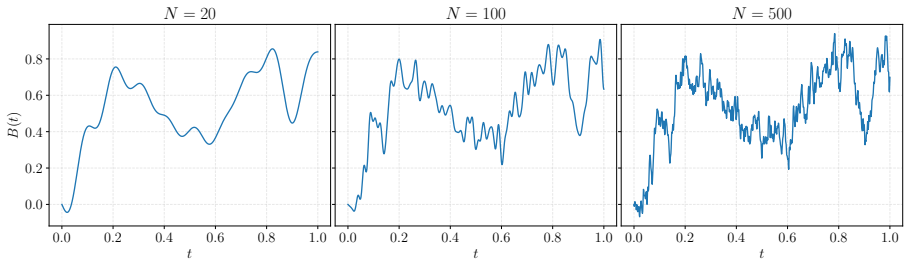

Figure 7: KKL expansion of a Brownian motion sample with increasing truncation order N. The approximate sample path is getting closer to the true one as N increases.

Remark 9. *Example 8 gives an KKL expansion of Brownian motion*

$$B(t) = \sum_{i=1}^{\infty} \frac{\sqrt{2}}{\left(i - \frac{1}{2}\right)\pi} \sin\left(\left(i - \frac{1}{2}\right)\pi\, t\right).$$

Now take a derivative of any summand above, we get

$$\sqrt{2}\cos\left(\left(i - \frac{1}{2}\right)\pi\, t\right).$$

Here you should note is that for the cosine component in the equation above, its amplitude is a constant $\sqrt{2}$ which is independent of the frequency index i. This says that the spectral density of the "derivative" process is white. Indeed, one can define the white noise process as the formal derivative of Brownian motion.

3.6 Circulant embedding for stationary GPs

The truncated spectral decomposition and KKL expansion methods are efficient indeed, however, we should note that these methods are approximate. If one wants to make exact GP samples, and the covariance function being symmetric positive definite is the only information that we know, then we cannot really do better than the Cholesky decomposition method. In order to develop a more efficient sampling method, either we know the explicit construction of the GP (e.g., Examples 2 and 7), or we know some additional information/structure of the covariance function/matrix . To elucidate this argument, let us consider an extreme example as follows.

Suppose that we have a GP with covariance function $C(t, s) = 1$. In addition to knowing that the covariance matrix is positive semi-definite, we also know that the covariance matrix is an all-one matrix. Therefore, to make a sample from this GP, we simply draw from a Normal random variable X_0 then let $X(t) = X_0$ for any query t. As another example, if the covariance matrix is diagonal, then the sample of the underlying GP is a collection of independent Normal random variables. In these two examples, we can draw GP samples in

one-shot because we leverage the structures of their covariance matrices. On the other hand, covariance functions in reality are much more complicated than these two extreme examples.

In what follows, we present an exact GP sampling method that applies to a wide family of GPs that are stationary (Wood and Chan, 1994; Chan and Wood, 1997). The stationary GPs cover a plenty of useful covariance functions, for instance, Matérn and radial basis functions which are commonly used by the GP practitioners. The essence of this method is the fact that the covariance matrix of any stationary GP has a circulant extension. Therefore, we can leverage this circulant matrix structure to form a decomposition of the covariance matrix by discrete Fourier transform which can be computed lighting fast. To explain this method, let us start by the definition of stationary GPs.

Definition 10 (Stationary process)**.** *A stochastic process $X\colon [0,\infty) \to \mathbb{R}$ is said to be stationary if for all $T \geq 1$ and reals $t_1 < t_2 < \ldots < t_T \in [0,\infty)$, the distribution of $\{X(t_1), X(t_2), \ldots, X(t_T)\}$ is the same with the distribution of $\{X(t_1+\tau), X(t_2+\tau), \ldots, X(t_T+\tau)\}$ for all translation τ.*

Definition 10 means that the finite-dimensional distribution of the process is invariant under translation. Imagine that you select a time window of fixed length, the distribution of the stationary process in the time window must not change if you shift the time window, and this result must hold for arbitrary window length and position.

Now consider zero-mean GPs. We say that the GP is stationary if its covariance function $C(t,s) = C(t+\tau, s+\tau)$ for all t, s, τ, viz., the covariance function depends on the time difference $t-s$ only. The covariance matrices of stationary GPs are closely related to symmetric Toeplitz matrices.

Definition 11 (Symmetric Toeplitz matrix)**.** *We say that a matrix $A \in \mathbb{R}^{T\times T}$ is symmetric Toeplitz of first column $\begin{bmatrix} a_0 & a_1 & \cdots & a_{T-1} \end{bmatrix} \in \mathbb{R}^T$, if the element of the matrix satisfies $A_{ij} = a_{|i-j|}$ for all indices $1 \leq i, j \leq T$. That is,*

$$A = \begin{bmatrix} a_0 & a_1 & \cdots & a_{T-2} & a_{T-1} \\ a_1 & a_0 & a_1 & \ddots & a_{T-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{T-2} & \ddots & a_1 & a_0 & a_1 \\ a_{T-1} & a_{T-2} & \cdots & a_1 & a_0 \end{bmatrix}.$$

Proposition 12. *Suppose that the times $t_1, t_2, \ldots, t_T$ are evenly placed, that is, $t_2 - t_1 = t_3 - t_2 = \ldots = t_T - t_{T-1}$. Then the covariance matrix $C_{1:T} \in \mathbb{R}^{T\times T}$ of any stationary GP at these times is symmetric Toeplitz of first column $\begin{bmatrix} C(t_1,t_1) & C(t_2,t_1) & \cdots & C(t_T,t_1) \end{bmatrix} \in \mathbb{R}^T$.*

Proof. This is concluded by definition. □

The proposition above says that the covariance matrix of stationary GP is completely determined by a vector which is its first column. This means that on

a computer we do not need to store the entire dense matrix but its first column vector only; For any algebra that operates on the covariance matrix, we may do algebras on its first column.

Next, we show that every symmetric Toeplitz matrix has a (minimal) circulant extension, the result of which we use to make GP samples.

Definition 13 (Circulant matrix)**.** *A matrix $A \in \mathbb{R}^{T\times T}$ is called circulant of first column $\begin{bmatrix} a_0 & a_1 & \cdots & a_{T-1}\end{bmatrix} \in \mathbb{R}^T$, if elements of the matrix are such that $A_{ij} = a_{i-j}$ for all $1 \le j \le i$ and $A_{ij} = a_{i-j+T}$ for all $i+1 \le j \le T$. That is, every column of the matrix is a one-step cyclic shift of its preceding column:*

$$A = \begin{bmatrix} a_0 & a_{T-1} & \cdots & a_2 & a_1 \\ a_1 & a_0 & \ddots & \vdots & a_2 \\ \vdots & a_1 & \ddots & \vdots & \vdots \\ a_{T-2} & \vdots & \ddots & a_0 & a_{T-1} \\ a_{T-1} & a_{T-2} & \cdots & a_1 & a_0 \end{bmatrix}$$

Circulant matrix is symmetric if its first column satisfies $a_{T-i} = a_i$ for $i = 1, 2, \ldots, T-1$.

It turns out that every symmetric Toeplitz matrix, hence, every stationary GP covariance matrix (with evenly placed times) has a symmetric circulant extension.

Proposition 14. *Every symmetric Toeplitz matrix $A \in \mathbb{R}^{T\times T}$ of first column $\begin{bmatrix} a_0, a_1, \ldots, a_{T-1}\end{bmatrix} \in \mathbb{R}^T$ has an extension matrix $\overline{A}$ that is symmetric circulant of first column $\begin{bmatrix} a_0, a_1, \ldots, a_{T-1}, a_{T-2}, \ldots, a_1\end{bmatrix} \in \mathbb{R}^{2T-2}$. The extension means an embedding that*

$$\overline{A} = \begin{bmatrix} A & B \\ B^{\mathsf{T}} & E \end{bmatrix}$$

for some matrices B and E. Furthermore, this $\overline{A}$ is the smallest circulant matrix that can embed A.

Example 15. *Consider a covariance function $C(t,s) = \exp(-|t-s|)$ and times $t_1 = 1$, $t_2 = 2$, and $t_3 = 3$. The covariance matrix $C_{1:3}$ reads*

$$C_{1:3} = \begin{bmatrix} 1 & \mathrm{e}^{-1} & \mathrm{e}^{-2} \\ \mathrm{e}^{-1} & 1 & \mathrm{e}^{-1} \\ \mathrm{e}^{-2} & \mathrm{e}^{-1} & 1 \end{bmatrix}$$

which is a symmetric Toeplitz matrix of first column $\begin{bmatrix} 1 & \mathrm{e}^{-1} & \mathrm{e}^{-2}\end{bmatrix}$. Its minimal circulant extension is

$$\begin{bmatrix} 1 & \mathrm{e}^{-1} & \mathrm{e}^{-2} & \mathrm{e}^{-1} \\ \mathrm{e}^{-1} & 1 & \mathrm{e}^{-1} & \mathrm{e}^{-2} \\ \mathrm{e}^{-2} & \mathrm{e}^{-1} & 1 & \mathrm{e}^{-1} \\ \mathrm{e}^{-1} & \mathrm{e}^{-2} & \mathrm{e}^{-1} & 1 \end{bmatrix}$$

of first column $\begin{bmatrix} 1 & \mathrm{e}^{-1} & \mathrm{e}^{-2} & \mathrm{e}^{-1}\end{bmatrix}$.

Remark 16. *Although the minimal circulant extension of any symmetric Toeplitz is symmetric, the extension might not be non-negative definite. This specifically depends on the covariance function at hand.*

Suppose that the minimal circulant $\overline{C}_{1:n} \in \mathbb{R}^{n \times n}$ of the covariance matrix $C_{1:T}$ is positive semi-definite, where $n = 2\,T - 2$. We can make samples according to $\overline{C}_{1:n}$ then slice the first T components of the samples to get the samples from that of $C_{1:T}$. This seems to be a fuss, since the size of the extension $\overline{C}_{1:n}$ is bigger than $C_{1:T}$. However, thanks to a remarkable Fourier representation of circulant matrix, it is indeed worth sampling from $\overline{C}_{1:n}$.

Proposition 17. *Any circulant matrix $A \in \mathbb{R}^{n \times n}$ of first column $a_{0:n-1} \coloneqq \begin{bmatrix} a_0 & a_1 & \cdots & a_{n-1} \end{bmatrix} \in \mathbb{R}^n$ has a Fourier representation*

$$A = F\,D\,F^\star,$$

where $F \in \mathbb{C}^{n \times n}$ is a Fourier matrix with element

$$F_{ij} \coloneqq \frac{1}{\sqrt{n}} \left(\mathrm{e}^{-2\,\pi\,\mathrm{i}\,/\,n}\right)^{(i-1)\,(j-1)},$$

and D is a diagonal matrix with diagonal $\sqrt{n}\,F^\star\,a_{0:n-1}$. The star superscript means conjugate transpose.

Remark 18. *The Fourier matrix is such that $\sqrt{n}\,F\,a$ and $\frac{1}{\sqrt{n}}\,F^\star\,a$ represent the discrete Fourier transform (DFT) and the inverse DFT for any vector a, respectively.*

The Fourier representation in the proposition above suggests that to sample according to a circulant matrix we can combine the inverse DFT of the first column, and samples from complex Normal random variables. This is detailed in the following proposition.

Proposition 19. *Let $\xi_{1:n} \in \mathbb{C}^n$ be a vector of standard complex Normal random variables, and suppose that the minimal circulant extension $\overline{C}_{1:n}$ of the covariance matrix $C_{1:T}$ is positive semi-definite. Let $X^1_{1:T}$ and $X^2_{1:T}$ be the first T components of*

$$\mathrm{Re}\big(F\,\sqrt{D}\,\xi_{1:n}\big) \quad \textit{and} \quad \mathrm{Im}\big(F\,\sqrt{D}\,\xi_{1:n}\big),$$

respectively, where the diagonal of D is $\sqrt{n}\,F^\star\,\overline{c}_{1:n}$, and $\overline{c}_{1:n}$ is the first column of $\overline{C}_{1:n}$. Then, $X^1_{1:T}$ and $X^2_{1:T}$ are independent identically distributed, and the distribution is $\mathrm{N}(0, C_{1:T})$.

To compute the DFT and inverse DFT in practice, we use the fast Fourier transform (FFT) algorithm. The computational complexity of FFT is $O(n \log n)$ which is significantly smaller than $O(T^3)$ when T is large. We summarise the sampling method in the following algorithm, and exemplify a result from the algorithm in Figure 8.

Exercise 20. *What if the times are not evenly placed? If this is true, the covariance matrix is not symmetric Toeplitz. How to remedy this problem?*

Algorithm 3: Sampling from stationary GP by minimal circulant embedding

Inputs: Evenly placed times $t_1, t_2, \ldots, t_T$, mean function $t \mapsto m(t)$, and covariance function $t, s \mapsto C(t, s)$

$n = 2\,T - 2$;
Compute $m_{1:T}$ at the times;
Compute the first column $c_{0:T-1}$ as per Proposition 12;
Compute the circulant first column extension $\bar{c}_{1:n} \in \mathbb{R}^n$ of $c_{0:T-1}$ as per Proposition 14;
$d_{1:n} = n * \texttt{ifft}(\bar{c}_{1:n})$;
Draw independent $\alpha, \beta \sim \mathrm{N}(0, I_n)$ and let $\xi_{1:n} = \alpha + \beta\,\mathrm{i}$;
Elementwise vector multiplication $z_{1:n} = \sqrt{d_{1:n}}\,\xi_{1:n}$;
Let $g_{1:T}$ be the first T components of $\texttt{fft}(z_{1:n}) \,/\, \sqrt{n}$;
return *Two indepedent samples* $m_{1:T} + \texttt{real}(g_{1:T})$ *and* $m_{1:T} + \texttt{imag}(g_{1:T})$

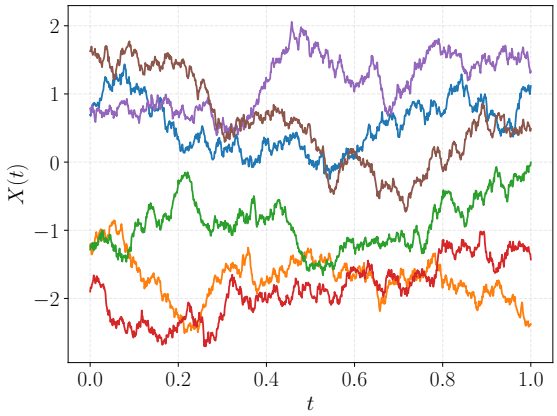

Figure 8: Samples from a GP with exponential covariance function ($\ell = 1$, $\sigma = 1$) by using Algorithm 3. Try run the code while setting T to be mega large, and compare with the Cholesky approach.

4 Regression

In this section we show how to solve a GP regression problem from the model of the form

$$\begin{aligned} X(t) &\sim \mathrm{GP}\big(m(t), C(t, t')\big), \\ Y_k &= a\,X(t_k) + b + \xi_k, \end{aligned} \tag{11}$$

where the random variable $Y_k \in \mathbb{R}$ represents the measurement of the GP X at time t_k through any affine transformation defined via $a \neq 0$, b, and measurement noise $\xi_k \sim \mathrm{N}(0, \Xi)$. Now suppose that we have a set of measurements/data

$y_{1:T} \coloneqq \{y_1, y_2, \ldots, y_T\}$ from $Y_{1:T}$ at times $t_1, t_2, \ldots, t_T$, the goal is to solve the posterior distribution of X conditionally on these measurements. More specifically, we would like to solve the posterior density

$$p(x_{1:T} \mid y_{1:T}) = \mathrm{N}(x_{1:T} \mid \alpha_{1:T}, \beta_{1:T})$$

of $X_{1:T}$ given $y_{1:T}$. Since the measurement random variables and the GP are jointly Normal distributed, the posterior distribution is Normal as well, with mean $\alpha_{1:T} \coloneqq \mathbb{E}[X_{1:T} \mid y_{1:T}]$ and covariance $\beta_{1:T} \coloneqq \mathrm{Cov}[X_{1:T} \mid y_{1:T}]$. To obtain the posterior mean and covariance, we recall the Gaussian identity that we learnt from the previous lecture on Bayesian linear regression.

Proposition 21 (Gaussian identity)**.** *Let $X_{1:T} \in \mathbb{R}^T$ and $Y_{1:T} \in \mathbb{R}^T$ be jointly Normal such that*

$$\begin{bmatrix} X_{1:T} \\ Y_{1:T} \end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix} x_{1:T} \\ y_{1:T} \end{bmatrix} \;\middle|\; \begin{bmatrix} \mathbb{E}[X_{1:T}] \\ \mathbb{E}[Y_{1:T}] \end{bmatrix}, \begin{bmatrix} \mathrm{Cov}[X_{1:T}] & \mathrm{Cov}[X_{1:T}, Y_{1:T}] \\ \mathrm{Cov}[Y_{1:T}, X_{1:T}] & \mathrm{Cov}[Y_{1:T}] \end{bmatrix} \right).$$

Then the distribution of $X_{1:T}$ conditioned on $Y_{1:T} = y_{1:T}$ is Normal with mean and covariance given by

$$\mathbb{E}[X_{1:T} \mid y_{1:T}] = \mathbb{E}[X_{1:T}] + \mathrm{Cov}[X_{1:T}, Y_{1:T}]\big(\mathrm{Cov}[Y_{1:T}]\big)^{-1}\big(y_{1:T} - \mathbb{E}[Y_{1:T}]\big)$$

and

$$\mathrm{Cov}[X_{1:T} \mid y_{1:T}] = \mathrm{Cov}[X_{1:T}] - \mathrm{Cov}[X_{1:T}, Y_{1:T}]\big(\mathrm{Cov}[Y_{1:T}]\big)^{-1}\mathrm{Cov}[Y_{1:T}, X_{1:T}],$$

respectively. Note that the posterior covariance $\mathrm{Cov}[X_{1:T} \mid y_{1:T}]$ is independent of the data values $y_{1:T}$.

Proof. See, for example, a note by Schön and Lindsten (2011). □

The GP regression model in Equation (11) gives the following results.

$$\begin{aligned} \mathbb{E}[X_{1:T}] &= m_{1:T}, \\ \mathbb{E}[Y_{1:T}] &= a\, m_{1:T} + b, \\ \mathrm{Cov}[X_{1:T}] &= C_{1:T}, \\ \mathrm{Cov}[Y_{1:T}] &= a^2\, C_{1:T} + \Xi\, I_T, \\ \mathrm{Cov}[X_{1:T}, Y_{1:T}] &= \mathrm{Cov}[X_{1:T}, a\, X_{1:T} + b + \xi_{1:T}] \\ &= a\, C_{1:T}. \end{aligned}$$

By substituting the equations above into Proposition 21, we obtain the GP regression posterior mean and covariance

$$\begin{aligned} G_{1:T} &\coloneqq a^2\, C_{1:T} + \Xi\, I_T, \\ \alpha_{1:T} &= m_{1:T} + a\, C_{1:T}\, G_{1:T}^{-1}\big(y_{1:T} - (a\, m_{1:T} + b)\big), \\ \beta_{1:T} &= C_{1:T} - a^2\, C_{1:T}\, G_{1:T}^{-1}\, C_{1:T}. \end{aligned} \tag{12}$$

We summarise the GP regression formulae in Algorithm 4.

Algorithm 4: GP regression on Equation (11)

Inputs: Model parameters a, b, and Ξ, times $t_1, t_2, \ldots, t_T$, data $y_{1:T}$, mean function $t \mapsto m(t)$, and covariance function $t, s \mapsto C(t, s)$

1 Compute the mean vector $m_{1:T}$ and covariance matrix $C_{1:T}$ at the times;

2 Compute the posterior mean $\alpha_{1:T}$ and covariance $\beta_{1:T}$ as per Equation (12);

3 **return** $\alpha_{1:T}$ *and* $\beta_{1:T}$

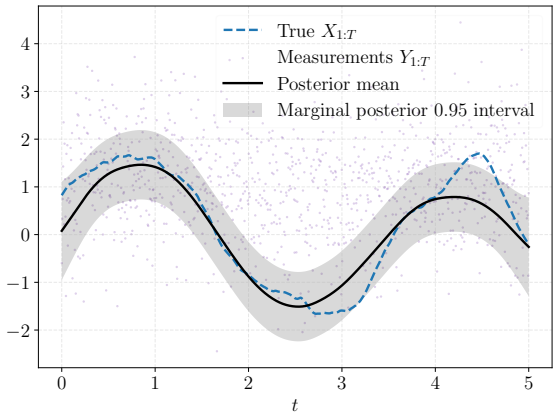

Figure 9: GP regression when $m(t) = \sin(2\,t)$ and C is a Matérn $\nu = 3/2$ covariance function with $\ell = 1$ and $\sigma = 1$. The model parameters are $a = 0.2$, $b = 1$, and $\Xi = 1$.

Exercise 22 (GP interpolation and extrapolation)**.** *The formulated GP regression is concerned with the posterior distribution at the data times $t_1, t_2, \ldots, t_T$ only. In reality, we might also want the posterior distribution at any (query) times $s_1, s_2, \ldots, s_N$ other than the data times. Please modify Equation* (12) *to achieve so.*

Note that Equation (12) requires to solve a matrix inversion $G_{1:T}^{-1}$. In practice, we do not compute this matrix inversion explicitly, but instead, we directly obtain the $G_{1:T}^{-1}$-vector or $G_{1:T}^{-1}$-matrix multiplication by solving a linear system. This implicit solution usually requires less computational cost and is more numerically stable than explicitly computing the matrix inversion. Please refer to the companion code to see how this is implemented. We in addition note that this step is the computational bottleneck of GP regression, as its computational complexity is $O(T^3)$.

In Figure 9, we plot an example of regression for a Matérn GP.

5 Maximum likelihood estimation of parameters

Let us recall the GP regression model

$$\begin{aligned} X(t) &\sim \mathrm{GP}\big(m(t;\theta), C(t,t';\theta)\big), \\ Y_k &= a\, X(t_k) + b + \xi_k, \end{aligned} \tag{13}$$

and in addition we assume that the GP has an unknown parameter θ that we want to estimate from the measurements $y_{1:T}$. For instance, in the Matérn case, we can denote $\theta = \begin{bmatrix}\theta_1 & \theta_2\end{bmatrix}$, and let $\ell = g(\theta_1)$ and $\sigma = g(\theta_2)$ for some positive bijection g. To estimate the parameter θ, we commonly use the maximum likelihood estimation (MLE) method. This method consists in maximising the marginal log likelihood

$$\begin{aligned} \log p(y_{1:T};\theta) &= \log \int p(y_{1:T} \mid x_{1:T})\, p(x_{1:T};\theta)\, \mathrm{d}x_{1:T} \\ &= \log \int \mathrm{N}(y_{1:T} \mid a\, x_{1:T} + b, \Xi\, I_T)\, \mathrm{N}(x_{1:T} \mid m_{1:T}, C_{1:T};\theta)\, \mathrm{d}x_{1:T} \\ &= -\frac{1}{2}\Big(\big(y_{1:T} - (a\, m_{1:T} + b)\big)^{\mathsf{T}}\, G_{1:T}^{-1}\, \big(y_{1:T} - (a\, m_{1:T} + b)\big) \\ &\qquad + \log\det(2\,\pi\, G_{1:T}) \Big), \end{aligned}$$

with respect to the parameter θ. Please note that now $m_{1:T}$ and $G_{1:T}$ depend on the parameter θ, hence, to find the optimal parameters we have to resort to non-linear optimisers (Nocedal and Wright, 2006) depending on the mean and covariance functions of the model. The optimisation usually requires the gradient $\partial \log p(y_{1:T};\theta) \,/\, \partial\theta$. Although you can derive the gradient by hand (this is an exercise), in practice we can compute the gradient efficiently and bug-lessly by using automatic differentiations, for instance, JAX (Python).

We give a demonstration as follows. Let us define a GP model with mean function $m(t;w) = w\sin(\pi\, t)$ and a Matérn $\nu = 3\,/\,2$ covariance function. Choose $w = 2$, and the length scale and magnitude of the covariance function be $\ell = \sigma = 1$. Since it is not possible to identify a and σ at the same time, we fix $a = 1$ and $b = 0$. To minimise the negative log likelihood, we use the L-BFGS-B optimiser. The result is plotted in Figure 10. In this figure, we find that the estimated parameters are close to the true ones.

Exercise 23. *Choose $m(t;w) = \sin(w\,\pi\, t)$ for some w, and a Matérn covariance function the same as in Figure 10. Try if you can simultaneously estimate ℓ, σ, and in particular, w, from the data. If not, find out why.*

6 Approximate GP regression for non-Gaussian likelihood

In reality many likelihood models (i.e., $p(y_{1:T} \mid x_{1:T})$) are not linear Gaussian like in Equation (11). For example, when use GP for classifications, we often

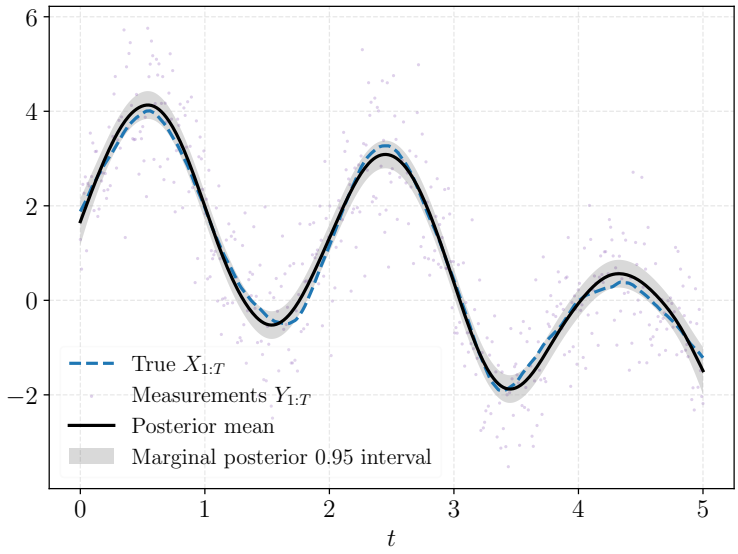

Figure 10: The estimated parameters are $w \approx 1.98$, $\sigma \approx 1.33$, and $\ell \approx 1.28$ which are close to the true values 2, 1, and 1, respectively.

choose a categorical distribution parametrised by the GP through a softmax function. It is hard to solve the posterior distribution of the GP in closed-form.

Textbook methods for approximating the GP posterior distribution include, for example, Laplace approximation and Markov chain Monte Carlo (MCMC). The essence of Laplace approximation is to approximate the GP posterior distribution by a Gaussian whose mean and covariance are given by the maximum a posteriori (MAP) solution, and the inverse Hessian of the MAP, respectively. The Laplace method is a fine example for pedagogy, as it is easy to implement. In the following example, we show a Laplace approximation to a GP regression problem with non-linear Poisson likelihood.

Example 24 (GP regression with Poisson likelihood)**.** *Consider a GP regression model defined by*

$$\begin{aligned} X(t) &\sim \mathrm{GP}\big(0, C(t,s)\big), \\ Y_k \mid X(t_k) &\sim \mathrm{Poisson}\big(y; \lambda(X(t_k))\big), \\ \lambda(x) &\coloneqq \tanh(x) + 1, \end{aligned}$$

where C is an exponential covariance function. The posterior distribution of X in this example is not available in closed-form. In Figure 11 we demonstrate a Laplace approximation to the posterior distribution.

The bottleneck for GP regression problems is the computation. As an example, when using the Laplace approximation, we need to take a matrix inversion of a Hessian matrix of size T. This computation is expensive when T is large.

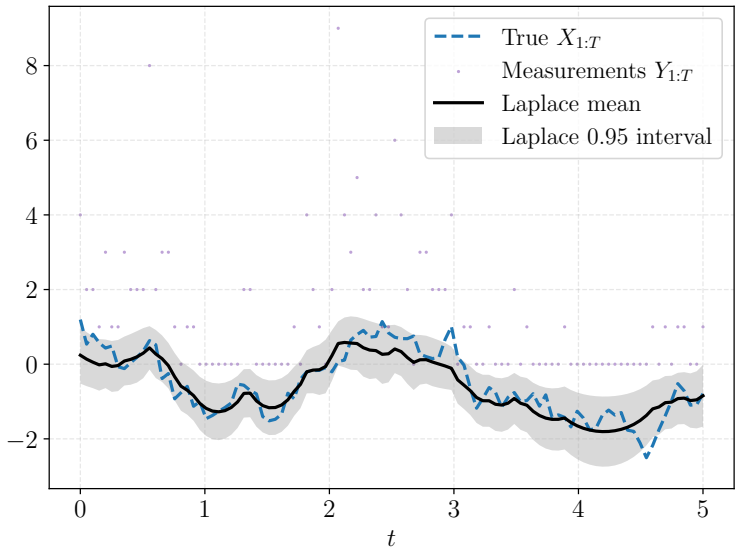

Figure 11: Laplace approximation to the GP regression model in Example 24.

As for the MCMC methods, many of them do not scale well with large T either, because the dimension of the sampling problem is determined by T.

We will expand this section in a future version of this lecture note.

Previously in Section 3 we have shown a few methods that can make samples from GP efficiently. These methods in principle may apply to the regression problem as well, but they may introduce approximations even for the linear Gaussian likelihood models, for example, the truncated KKL expansion. In the following section, we briefly sketch two alternative constructions of (Markov) GPs. These GPs can reduce the cubic complexity $O(T^3)$ to a linear one $O(T)$ (even sublinear $O(\log T)$ if implemented in parallel).

7 Alternative GP constructions for scalability

The definition of GP implies that the mean and covariance functions uniquely determine a GP. However, this does not mean that we have to explicitly assign the mean and covariance functions in order to construct a GP. As an example, Brownian motion is a GP by definition, but the definition itself does not explicitly involve its covariance function. Avoiding using the covariance function can be beneficial in computation for large datasets, since we do not need to deal with huge covariance matrices.

In what follows, we briefly recall two commonly used representations of GPs that are scalable for large number of data points. These GPs are called Markov Gaussian processes, and they are computationally efficient thanks to their Markov property. The Markov property is useful because to compute ex-

pectations that depend on the entire history, like $\mathbb{E}[X(t_k) \mid X(t_{k-1}), X(t_{k-2}), \ldots, X(t_0)]$, we only need to condition on the previous one $\mathbb{E}[X(t_k) \mid X(t_{k-1})]$.

7.1 Gauss–Markov random fields using precision matrices

Recall that the computation bottleneck of GP is storing and computing the covariance matrix $C_{1:T}$, and the inversion of $C_{1:T}$. The essence of the Gauss–Markov random field (GMRF) approach consists in representing GPs with the precision matrix instead of the covariance matrix (Rue and Held, 2005). The precision matrix is defined as the inverse of the covariance matrix and can be computed directly from a correlation graph (see, e.g., Rue and Held, 2005, Thm. 2.3). Moreover, thanks to the Markov property of GMRF, the precision matrices of GMRFs are tridiagonal which are sparse. Hence, the storage and algebraic operations (e.g., solving linear systems and spectral decompositions) on the precision matrix are computationally cheap.

7.2 Linear stochastic differential equations

Consider a linear stochastic differential equation (SDE)

$$\mathrm{d}X(t) = \mu\, X(t)\,\mathrm{d}t + \eta\,\mathrm{d}B(t), \quad X(t_0) = X_0, \tag{14}$$

where B is a Brownian motion. Solutions to the SDEs of the form in the equation above are continuous-time Markov GPs (Karatzas and Shreve, 1991, Sec. 5.6), and their means and covariance functions are implicitly defined by their SDE coefficients. For details of SDEs, we recommend a course that starts from October 2022 which introduces SDEs from a computational aspect, see `https://github.com/spdes/computational-sde-intro-lecture`.

Example 25. *The solution to the SDE*

$$\mathrm{d}X(t) = -\frac{1}{\ell}\, X(t)\,\mathrm{d}t + \sqrt{\frac{2}{\ell}}\,\sigma\,\mathrm{d}B(t), \quad X(t_0) \sim \mathrm{N}(0, \sigma^2),$$

is an exponential/Orstein–Uhlenbeck GP with mean $m(t) = 0$ and the exponential covariance function in Example 4.

There are a number of upsides for using the SDE constructions of GPs. The main upside is computation because we do not need to compute the covariance matrices in, for example, sampling and regression. To make a trajectory from SDE-GPs at times $t_1, t_2, \ldots, t_T$, we can conditionally make samples $X(t_k) \mid X(t_{k-1})$ sequentially in time for $k = 1, 2, \ldots, T$. This is an algorithm of complexity $O(T)$ which is significantly smaller than the cubic complexity from that of Cholesky decomposition. As for the regression, we can solve the GP posterior distribution of the model in Equation (11) by using a Kalman filter and smoother (KFS), and the result from the KFS is exactly the same as from computing Equation (12). Computing Equation (12) requires a cubic complexity, while the complexity of KFS is linear $O(T)$. Moreover, if the KFS

is implemented in parallel, the regression problem can be even done faster in logarithmic time (Corenflos et al., 2021). We refer the readers to Särkkä and Solin (2019) for more details and examples of SDE-GPs.

Another upside of the SDE-GP construction is that many real-world models (e.g., physics and finance) are inherently based on differential equations. For example, in fluid mechanics, we model the motion of fluid, such as velocity and vorticity, by differential equations, since the exact physical solution to the motion are rarely accessible but their infinitesimal dynamics are. When taking randomness into account, these equations become stochastic (partial) differential equations which form (Gaussian) random fields (Lord et al., 2014) under a few assumptions. See Lindgren et al. (2011) for how the SPDE Matérn GPs are applied in geostatistics.

The SDE-GP construction also allows for introducing non-stationarity in a "simple manner". As an example, suppose that one desires a Matérn GP such that its length scale and magnitude are some pre-given/learnable functions of time, for instance, $\ell(t) = \mathrm{NN}(t)$ for some (positive) neural networks. In order to do so, one has to invent an extension of the Matérn covariance function, because the original covariance function in Equation (7) will not be a valid covariance function if we just replace its ℓ and σ by the functions. On the other hand, one can just simply replace these parameters by the functions in the SDE construction of the Matérn GP, for example, in Example 25, and the covariance function is automatically valid by definition (see, Zhao, 2021).

8 Companion codes

You may find the implementations of the examples/algorithms of this lecture note at `https://github.com/spdes/gp`.

9 Changelog and errata

Working release

- Add algebras on covariance functions, and the continuity of GP.
- Expand the GP regression for non-linear non-Gaussian likelihood models.
- Add examples and materials for the parameter estimation biasedness and identifiability.

1 September 2022

Draft initiated.

References

Grace Chan and Andrew T. A. Wood. Algorithm AS 312: An algorithm for simulating stationary Gaussian random fields. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 46(1):171–181, 1997.

Ed S. Coakley and Valdimir Rokhlin. A fast divide-and-conquer algorithm for computing the spectra of real symmetric tridiagonal matrices. *Applied and Computational Harmonic Analysis*, 34(3):379–414, 2013.

Adrien Corenflos, Zheng Zhao, and Simo Särkkä. Gaussian process regression in logarithmic time. *arXiv preprint arXiv:2102.09964*, 2021.

Gene H. Golub and Charles F. van Loan. *Matrix computations*. The Johns Hopkins University Press, 4th edition, 2013.

Dave Higdon, Jenise Swall, and J. Kern. Non-stationary spatial modeling. *Bayesian Statistics*, 6(1):761–768, 1999.

Ioannis Karatzas and Steven E. Shreve. *Brownian motion and stochastic valculus*, volume 113 of *Graduate Texts in Mathematics*. Springer-Verlag New York, 2nd edition, 1991.

Hui-Hsiung Kuo. *Gaussian measures in Banach spaces*, volume 463 of *Lecture Notes in Mathematics*. Springer-Verlag Heidelberg, 1975.

Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011.

Gabriel J. Lord, Catherine E. Powell, and Tony Shardlow. *An introduction to computational stochastic PDEs*, volume 50 of *Cambridge Texts in Applied Mathematics*. Cambridge University Press, 2014.

Andrei S. Monin and Akiva M. Yaglom. *Statistical fluid methanics, Volumes I and II*. The MIT Press, 1971.

Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer-Verlag New York, 2nd edition, 2006.

Christopher J. Paciorek and Mark J. Schervish. Nonstationary covariance functions for Gaussian process regression. In *Advances in Neural Information Processing Systems 16*, pages 273–280. MIT Press, 2004.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. The MIT Press, 2006.

Håvard Rue and Leonhard Held. *Gaussian Markov random fields: theory and applications*. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, 2005.

Simo Särkkä and Arno Solin. *Applied stochastic differential equations*, volume 10 of *Institute of Mathematical Statistics Textbooks*. Cambridge University Press, 2019.

Thomas B. Schön and Fredrik Lindsten. Manipulating the multivariate Gaussian density. *Lecture note*, 2011.

Michael L. Stein. *Interpolation of spatial data*. Springer Series in Statistics. Springer, 1999.

Andrew T. A. Wood and Grace Chan. Simulation of stationary Gaussian processes in $[0, 1]^d$. *Journal of Computational and Graphical Statistics*, 3(4): 409–432, 1994.

Zheng Zhao. *State-space deep Gaussian processes with applications*. PhD thesis, Aalto University, 2021. URL `https://github.com/zgbkdlm/dissertation`.

Zheng Zhao, Muhammad Emzir, and Simo Särkkä. Deep state-space Gaussian processes. *Statistics and Computing*, 31(6):75, 2021.