

Gaussian Process

Kaiwen Zhou

Contents

1	Contextualization: Why do we need the kernel trick?	1
1.1	Feature Map Regression . . . . .	1
1.2	The Kernel Trick . . . . .	2
2	Topic: Gaussian Process	3
3	Topic: Kernel Design	3
3.1	Valid Kernels . . . . .	3
3.2	Feature Map Kernel: Connection to Stochastic Processes/Fields . . . . .	4
3.3	Application: Futures Curve Modeling . . . . .	4
3.4	Stationary Kernels . . . . .	5
4	Bayesian Learning With GPs	6
5	Model Selection and Hyperparameter Optimization	6
6	Topic: Links to SVD/PCA	7
6.1	Mercer’s Theorem . . . . .	7
6.2	The Hilbert Space of the Kernel . . . . .	7
6.3	Karhunen-Loève (KL) Expansion . . . . .	8
6.4	The Spectral Theorem . . . . .	8
6.5	Posterior Prediction In the Mercer Basis . . . . .	9

Readings

In addition to the lecture notes, the following are required readings:

- Chapters 2, 4 Rasmussen (2003)

Optional readings:

- Sheffield (2007): Gaussian free fields for mathematicians
- An excellent intro lecture on Kernel design by Nicolas Durrande can be found in the Sheffield ML summer school on Gaussian Processes

1 Contextualization: Why do we need the kernel trick?

1.1 Feature Map Regression

Supervised Learning and Conditional Expectation:

The general framework of supervised learning aims to learn a functional relationship  $y = f(\mathbf{x})$  by optimizing a loss function  $L(\mathbf{Y}, \mathbf{X}) = \sum_i \ell(y_i, \mathbf{x}_i)$  of the training data  $\mathbf{Y}, \mathbf{X} = (y_i, \mathbf{x}_i)$ . For quadratic loss, this is the same as estimating the conditional expectation of  $y$  given  $\mathbf{x}$  :

$$\mathbb{E}[y \mid \mathbf{x}] = \min_f |y - f(\mathbf{x})|^2$$

As the above optimization is over all functions, the choice of model amounts to a choice of parametrizing of a set of functions  $\{f(x)\}$ , and most ML models fall into the class of **additive models** where the function  $f(\mathbf{x})$  can be represented as a sum of base learners:

$$f(\mathbf{x}) = \sum_i f_i(\mathbf{x})$$

For example, OLS, Boosting, Random Forests, most neural nets, Fourier Transofrms, etc, all fall within this class of additive models.

Construct the Feature Map Regression:

Specifically, the feature map regression approach discussed in a previous lecture is an additive model of the form:

$$f(\mathbf{x}) := \sum_i^{N_F} \phi_i(\mathbf{x})w_i, \quad \text{or} \quad f(\mathbf{X}) := \Phi(\mathbf{X})\mathbf{w} \tag{1}$$

where the design matrix for model with  $N_F$  feature maps and a training set of  $N$  samples  $(y_i, \mathbf{x}_i)_{i=1}^N$  and is defined as:

$$\Phi(\mathbf{X}) := \begin{bmatrix} \phi_1(x_1) & \cdots & \phi_{N_F}(x_1) \\ \vdots & & \vdots \\ \phi_1(x_N) & \cdots & \phi_{N_F}(x_N) \end{bmatrix} \in \mathbb{R}^{N \times N_F}$$

[Equation \(1\)](#) is a regression model since learners  $\phi_i$  are fixed, then so is the design matrix for a given training set  $\mathbf{Y}, \mathbf{X}$ . Therefore, fitting the model simply amounts to solving for the unknown weights using the OLS formula:

$$\mathbf{w}^* = \min_{\mathbf{w}} \|\mathbf{y} - \Phi(\mathbf{X})\mathbf{w}\|^2 = \left( \Phi(\mathbf{X})^\top \Phi(\mathbf{X}) \right)^{-1} \Phi(\mathbf{X})^\top \mathbf{y} \quad (2)$$

Feature map regressions are powerful models because for suitable choices of  $\phi_i$ , one can approximate most well behaved function. For example, it turns out that Fourier Series can approximate any square-integrable function in finite domain  $L_2$  sense. Similarly, the Stone-Weierstrass theorem states that the polynomial feature maps can approximate any continuous function in  $L_1$  sense (pointwise). In other words, there exist feature map regression models that are **universal**. So why bother with more complex models?

### Curse of Dimensionality of Feature Map Regression:

Universality comes with a cost. The more flexible a model is, the easier it would overfit. To avoid such overfitting [Equation \(2\)](#) must be regularized, for example, by adding a shrinkage term:

$$\mathbf{w}^* = \left( \Phi(\mathbf{X})^\top \Phi(\mathbf{X}) + \sigma^2 \mathbf{I} \right)^{-1} \Phi(\mathbf{X})^\top \mathbf{y} \quad (3)$$

However, even if we have regularized, a much bigger issue of [Equation \(3\)](#) is that it requires the inversion of a  $N_F \times N_F$  matrix  $\mathbf{A} = \Phi(\mathbf{X})^\top \Phi(\mathbf{X}) + \sigma^2 \mathbf{I}$ . How big is  $N_F$ ? We actually never know! For example, if our feature map model is the Fourier Transform, this would be equivalent to asking how many Fourier coefficient do we need to fit a given function to an arbitrary accuracy. In order to be able to fit any continuous and square integrable function, we would actually need to include all Fourier components, or equivalently  $N_F \rightarrow \infty$ ! As a result, we would have to invert a  $\infty \times \infty$  matrix! That's impossible, and this is why we need the **kernel trick** as a work around.

## 1.2 The Kernel Trick

A major simplification occurs if rather than learning the weights  $\mathbf{w}$  as in [Equation \(3\)](#), we instead focus on the predictions given by the linear model. Suppose we train the model on  $(\mathbf{y}, \mathbf{X})$  and we wish to predict  $\mathbf{y}_* = (y_{*,j})_{j=1}^M$  given some test inputs  $\mathbf{X}_*$ . By substituting the optimal weights  $\mathbf{w}^*$  from [Equation \(3\)](#) we have:

$$\begin{aligned} \mathbf{y}_* &= \Phi(\mathbf{X}_*) \mathbf{w}^* = \Phi(\mathbf{X}_*) \left( \Phi(\mathbf{X})^\top \Phi(\mathbf{X}) + \sigma^2 \mathbf{I} \right)^{-1} \Phi(\mathbf{X})^\top \mathbf{y} \\ &= \Phi(\mathbf{X}_*) \sigma^{-2} \Phi(\mathbf{X})^\top \left( \Phi(\mathbf{X}) \sigma^{-2} \Phi(\mathbf{X})^\top + \mathbf{I} \right)^{-1} \mathbf{y} \\ &:= K(\mathbf{X}_*, \mathbf{X}) [K(\mathbf{X}, \mathbf{X}) + \mathbf{I}]^{-1} \mathbf{y} \end{aligned}$$

where the kernel matrix (here we only show  $K(\mathbf{X}_*, \mathbf{X})$ ) is

$$K(\mathbf{X}_*, \mathbf{X}) := \Phi(\mathbf{X}_*) \Phi(\mathbf{X})^\top = \begin{bmatrix} K(\mathbf{x}_{*,1}, \mathbf{x}_1) & \cdots & K(\mathbf{x}_{*,1}, \mathbf{x}_N) \\ \vdots & & \vdots \\ K(\mathbf{x}_{*,M}, \mathbf{x}_1) & \cdots & K(\mathbf{x}_{*,M}, \mathbf{x}_N) \end{bmatrix} \in \mathbb{R}^{M \times N}$$

(the  $\sigma^{-2}$  factor was absorbed in the definition of the design matrix)

The gist here is to realize that even though the design matrix was possibly infinite dimensional,  $\Phi(\mathbf{X}) \in \mathbb{R}^{N \times \{N_F \rightarrow \infty\}}$ , this prediction problem only depends on the kernel matrix  $K(\mathbf{X}_*, \mathbf{X}) \in \mathbb{R}^{M \times N}$ , which is finite dimensional and only depends on the number of training and test samples,  $N, M$ . It does not even depend on the dimensionality  $D$  of the features  $\mathbf{X} \in \mathbb{R}^{N \times D}$ . Therefore, to do prediction in feature map regression, one does not even need to fit for the optimal parameters  $\mathbf{w}$ ! The only parameter is the choice of regularization,  $\sigma$ .

We will come back shortly to the question of what constitutes a valid Kernel.

### Bayesian View:

From the definition in terms of the design matrix, we see that when evaluated on the train/test dataset, all the kernel matrices  $K(\mathbf{X}_*, \mathbf{X}_*)$ ,  $K(\mathbf{X}_*, \mathbf{X})$ ,  $K(\mathbf{X}, \mathbf{X}_*)$  and  $K(\mathbf{X}, \mathbf{X})$  are all positive semidefinite. As a result, this suggests that there is also a probabilistic description of the kernel-trick procedure. Indeed, the feature map prediction formula can also be derived within a Bayesian framework as the conditional expectation  $\mathbb{E}[\mathbf{y}_* | \mathbf{X}_*, \mathbf{y}, \mathbf{X}]$  of the following probabilistic model:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} | \mathbf{X}, \mathbf{X}_* \sim \mathcal{N} \left( \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I} & K(\mathbf{X}, \mathbf{X}_*) \\ K(\mathbf{X}_*, \mathbf{X}) & K(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix} \right) \quad (4)$$

Using the formulas for conditional mean and covariance, then if  $\mathbf{X}_* \neq \mathbf{X}$  we have:

$$\begin{aligned} \mathbb{E}[\mathbf{y}_* | \mathbf{X}_*, \mathbf{y}, \mathbf{X}] &= K(\mathbf{X}_*, \mathbf{X}) [K(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}]^{-1} \mathbf{y} \\ \text{Cov}[\mathbf{y}_* | \mathbf{X}_*, \mathbf{y}, \mathbf{X}] &= K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}_*, \mathbf{X}) [K(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}]^{-1} K(\mathbf{X}, \mathbf{X}_*) \end{aligned} \quad (5)$$

If on the other hand  $\mathbf{X}_* = \mathbf{X}$  and  $\sigma = 0$ , then  $\mathbb{E}[\mathbf{y}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}] = \mathbf{y}$  and  $\text{Cov}[\mathbf{y}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}] = 0$ . In this case, there is no uncertainty of the posterior predictions as they are assumed to equal the prior observations,  $\mathbb{E}[\mathbf{y}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}] = \mathbf{y}$ . This is exactly why, we shall see later, the posterior path goes through the data points in the train set.

Note that the posterior conditional expectation formula in the first equation above is the same as the prediction formula for feature map regression after the kernel trick is applied.

## 2 Topic: Gaussian Process

Gaussian Fields are a multi-dimensional generalization of Gaussian Processes. However in the ML community the two names are used interchangeably. As a reminder, a Gaussian Process is a familiar concept in stochastic theory:

**Definition 2.1. (Gaussian Fields)** A continuous real (stochastic) function  $Y_{\mathbf{x}} := Y(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}$  is a Gaussian Field if and only if for any set of points  $(x_i)_{i=1}^k$ , the  $k$ -dimensional variable

$$\mathbf{Y}_{\mathbf{x}_1, \dots, \mathbf{x}_k} := (Y_{\mathbf{x}_1}, \dots, Y_{\mathbf{x}_k})$$

is a multivariate Gaussian random variable. Equivalently, every linear combination  $\mathbf{w} \cdot \mathbf{Y}_{\mathbf{x}_1, \dots, \mathbf{x}_k}$  has a univariate Gaussian distribution.

**Definition 2.2. (Gaussian Process)** A time continuous stochastic process  $\{X_t, t \in T\}$  is a Gaussian Process if and only if for every set of indices  $(t_i)_{i=1}^k$  in the index set  $T$  the  $k$ -dimensional variable

$$\mathbf{X}_{t_1, \dots, t_k} := (X_{t_1}, \dots, X_{t_k})$$

is a multivariate Gaussian random variable. Equivalently, every linear combination  $\mathbf{w} \cdot \mathbf{X}_{t_1, \dots, t_k}$  has a univariate Gaussian distribution.

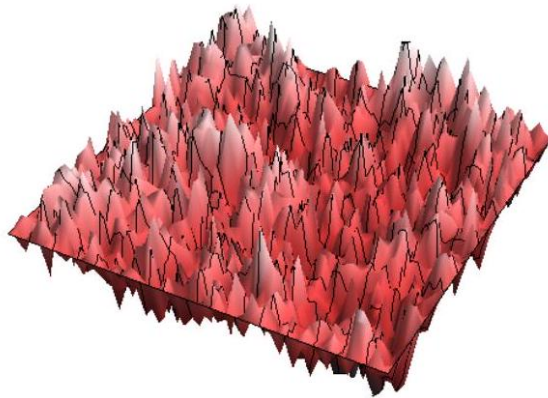


Figure 1: A 2D Gaussian Field.

## 3 Topic: Kernel Design

What is a Kernel? It is clear that Equation (4) is a Gaussian Field due to the fact that the kernel matrices entering into the equation are all positive semidefinite. This motivates the following

**Definition 3.1. (Kernel, Mercer Conditions)** A kernel is a continuous function  $K : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$  with the following properties (Mercer Conditions):

1.  $K$  is symmetric:  $K(\mathbf{x}, \mathbf{y}) = K(\mathbf{y}, \mathbf{x})$
2.  $K$  is positive semidefinite, i.e. for any finite set of points  $(x_i)_{i=1}^n$  we have

$$\sum_{i=1}^n \sum_{j=1}^n K(\mathbf{x}_i, \mathbf{x}_j) c_i c_j \geq 0$$

for all choices of real numbers  $(c_i)_{i=1}^n$ .

### 3.1 Valid Kernels

Which Symmetric Functions are Valid Kernels? Given any symmetric real function  $K(\mathbf{x}, \mathbf{y})$  it turns out it is not generally easy to show that the function defines a valid Kernel. We will elaborate on this later. However, it turns out that there are several easy ways to generate new kernels from old as follows.

**Proposition 3.2. (Valid Kernel Properties)**

- (a) **Product of Valid Kernels is a Valid Kernel:** Given two valid kernels  $K_1$  and  $K_2$ , the product  $K = K_1 K_2$  is a valid kernel.
- (b) **Sum of Valid Kernels is a Valid Kernel:** Given two valid kernels  $K_1$  and  $K_2$ , the sum  $K = K_1 + K_2$  is a valid kernel.
- (c) **Corollary - Mixture of Valid Kernels is a Valid Kernel:** Given two valid kernels  $K_1$  and  $K_2$  and  $a \geq 0, b \geq 0$ , then  $K = aK_1 + bK_2$  is a valid kernel for  $a, b \geq 0$ . One often also assumes  $a + b = 1$ .
- (d) **Composition Kernel:** If  $K_1 : \mathbb{R}^{D \times D} \rightarrow \mathbb{R}$  is a valid Kernel and  $\mathbf{f} : \mathbb{R}^N \rightarrow \mathbb{R}^D$  is any vector function, then  $K(\mathbf{x}, \mathbf{y}) := K_1(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{y}))$  is also a valid Kernel.
- (e) **Corollary - Periodic Kernel from a given kernel** If  $\mathbf{f}$  is a periodic function, then  $K$  above would be a periodic Kernel generating paths with the same periodicity as  $\mathbf{f}$ .
- (f) **Tensor Product/Sum Kernel:** Suppose  $K_1 : \mathbb{R}^{D_1 \times D_1} \rightarrow \mathbb{R}$  and  $K_2 : \mathbb{R}^{D_2 \times D_2} \rightarrow \mathbb{R}$  are valid kernels. Suppose also  $\mathbf{z} := (\mathbf{x}; \mathbf{y}) \in \mathbb{R}^{D_1 + D_2}$  where  $\mathbf{x} \in \mathbb{R}_1^{D_1}$  and  $\mathbf{y} \in \mathbb{R}_2^{D_2}$ . Then both  $K(\mathbf{z}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{x}) + K_2(\mathbf{y}, \mathbf{y})$  and  $K(\mathbf{z}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{x})K_2(\mathbf{y}, \mathbf{y})$  are valid kernels.

**Proposition 3.3. (Linear (Dot Product) Kernel)**  $K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$  is a valid kernel for any  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$  (why?).

Even though technically  $K(\mathbf{x}, \mathbf{y})$  is a quadratic function, this is called a linear Kernel, because the posterior expectation and covariance formulas Equation (5) for this Kernel would correspond to those of linear regression.

**Proposition 3.4. (Feature Map Kernel)**  $K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})\phi(\mathbf{y})$  is a valid kernel for any  $\phi(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}$ .

One way to show that this is a valid kernel directly from the definition and (66). Alternatively, the scalar Linear Kernel  $K(t, s) = ts$  is seen as a valid Kernel for scalar  $t, s \in \mathbb{R}$ . The result then follows by realizing that the Feature Map kernel is a Composition Kernel of the scalar Linear Kernel.

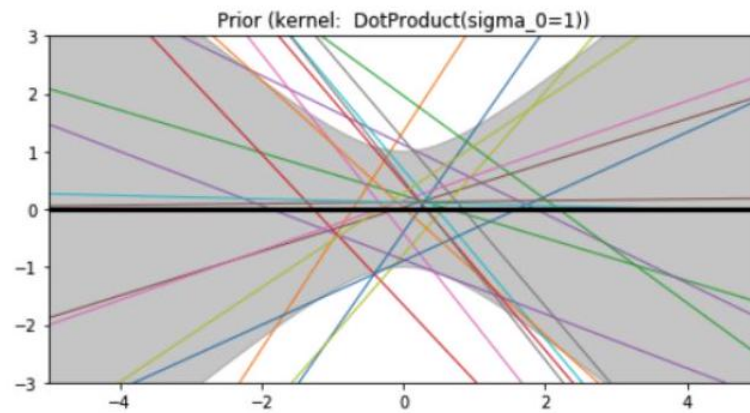


Figure 2: Prior paths from the Linear (Dot Product) Kernel mixed with a Constant Variance Kernel,  $K(x, y) = \sigma^2 + xy$ .

### 3.2 Feature Map Kernel: Connection to Stochastic Processes/Fields

#### 1-D Features: The Kernel of a Stochastic Process

**Theorem 3.5.** For a one-dimensional square-integrable zero-mean continuous process  $Y_t$  over a probability space  $(\Omega, \mathcal{F}, \mathbb{P})$  indexed over the closed interval  $[a, b]$  the following function  $K_Y(s, t)$  is a valid Kernel (show this!):

$$K_{YY}(s, t) = \mathbb{E}[Y_s Y_t]$$

In time-series analysis the above Kernel is known as the level-Autocovariance Function (ACF). Square-integrability is needed to ensure that the total variance  $V = \int_a^b \mathbb{E}[Y_t^2] dt = \int_a^b K_{YY}(t, t) dt$  of the process is finite.

**Corollary 3.6.** Given a square-integrable  $\Pi_t = \int_a^t \Delta_s Y_s ds$  the following is a valid Kernel:

$$K_{\Pi\Pi}(t, t') = \mathbb{E}[\Pi_t \Pi_{t'}] = \mathbb{E}\left[\int_a^t \Delta_s Y_s ds \int_a^{t'} \Delta_{s'} Y_{s'} ds'\right]$$

The above theorem and corollary allow us to construct kernels corresponding to a any specified stochastic process.

**Example 3.7. (White-Noise Kernel)** The Kernel of  $dW_t \sim N(0, dt)$  is

$$K_{noise}(s, t) = dt \mathbb{1}_{s,t} \quad \text{where} \quad \mathbb{1}_{s,t} = \begin{cases} 1 & \text{if } s = t \\ 0 & \text{if } s \neq t \end{cases}$$

**Example 3.8. (Brownian Kernel)** The Kernel of  $X_t = \int_0^t dW_s$  is:

$$K_{BM}(s, t) = \min(t, s).$$

Note that for time-varying vol process  $dX_t = \sigma(t)dW_t$  the Kernel is  $K(t, t') = \min\left(\int_0^t \sigma^2(s) ds, \int_0^{t'} \sigma^2(s') ds'\right) = \min(QVS(t), QVS(t'))$  where  $QVS(t) = \mathbb{E}[X_t^2] = \mathbb{E}\left[\int_0^t dX_s^2\right] = \int_0^t \sigma^2(s) ds$  is the quadratic variance swap of  $X_t$ .

**Example 3.9. (Bachelier Local-Vol Kernel)** The Kernel of the Bachelier Local-Vol process  $dX_t = \sigma^2(X_t, t) dW_t$  is:

$$K_{LV}(s, t) = \mathbb{E}[X_s X_t] = \min(QVS(t), QVS(t'))$$

where  $QVS(t) = \mathbb{E}[X_t^2] = \mathbb{E}\left[\int_0^t dX_s^2\right] = \mathbb{E}\left[\int_0^t \sigma^2(X_s, s) ds\right]$ .

**Example 3.10. (Ornstein-Uhlenbeck Kernel)** For sufficiently large  $t, s \gg a = 0$ , the Kernel of the OU process  $dx_t = -\alpha dt + \sigma dW_t$  is:

$$K_{OU}(s, t) = \frac{\sigma^2}{2\alpha} e^{-\alpha|t-s|} \quad (6)$$

#### N-D Features: The Kernel of a Stochastic Process

For a  $N$ -dimensional square-integrable zero-mean continuous field  $Y_{\mathbf{x}}$  over a probability space  $(\Omega, \mathcal{F}, \mathbb{P})$  indexed over some compact space  $C$  (for example  $C = [0, 1]^N \subset \mathbb{R}^N$ ), then the following function  $K_{YY}(\mathbf{x}, \mathbf{x}')$  is a valid Kernel:

$$K_{YY}(\mathbf{x}, \mathbf{x}') = \mathbb{E}[Y_{\mathbf{x}} Y_{\mathbf{x}'}]$$

This would be a higher-dimensional generalization of the ACF. Square-integrability is needed again to ensure that the total variance  $V = \int_C \mathbb{E}[Y_{\mathbf{x}}^2] d\mathbf{x} = \int_C K_{YY}(\mathbf{x}, \mathbf{x}) d\mathbf{x}$  of the process is finite.

**Example 3.11. (N-D Brownian Motion Kernel)** Suppose  $\mathbf{Y}_t \in \mathbb{R}^N$  be the level of  $N$ -dimensional correlated Brownian motion over a finite  $t$ -domain (to ensure square-integrability):

$$\mathbf{Y}_t = \int_0^t d\mathbf{W}_s, \quad \mathbb{E}[dW_{t,i} dW_{t,i}] = C_{ij} dt$$

Define  $x := (t, i)$ ,  $t \in [0, 1], i \in 1, \dots, N$  as well as the r.v.  $Z_{\mathbf{x}} := Y_{t,i}$  we have:

$$K_{NdBM}(\mathbf{x}, \mathbf{x}') = \mathbb{E}[Z_{\mathbf{x}} Z_{\mathbf{x}'}] = \mathbb{E}[Y_{t,i} Y_{t',i'}] = \min(t, t') C_{ii'}$$

### 3.3 Application: Futures Curve Modeling

Simply replace the asset index  $i$ , with  $\tau = T - t$  of a continuously rolled constant time-to-maturity strategy. As there is a continuum of  $\tau$  we have  $\mathbf{Y}_t = (Y_{t,\tau})_{\tau} \in \mathbb{R}_+^{\mathbb{R}}$ . Ignoring the carry drift, we are then modeling:

$$\mathbf{Y}_t = \int_0^t d\mathbf{W}_s, \quad \mathbb{E}[dW_{t,\tau} dW_{t,\tau'}] = C(\tau, \tau') dt$$

Define  $x := (t, \tau)$ ,  $t \in [0, 1], \tau \in \mathbb{R}_+$  as well as the r.v.  $Z_{\mathbf{x}} := Y_{t,\tau}$  we have:

$$K(\mathbf{x}, \mathbf{x}') = \mathbb{E}[Z_{\mathbf{x}} Z_{\mathbf{x}'}] = \mathbb{E}[Y_{t,\tau} Y_{t',\tau'}] = \min(t, t') C(\tau, \tau')$$

The same construction generalizes to higher dimensional features instead of  $\tau$  where in a similar manner the Kernel would be the tensor product of a time component  $\min(t, t')$  and a spatial/feature component.



### 3.4 Stationary Kernels

In time-series analysis, stationary processes are such that their ACF is time-homogeneous. A generalization in  $N - D$  is the following:

**Definition 3.12. (Stationary Kernel)** A Kernel is stationary if

$$K(\mathbf{x}, \mathbf{y}) = K(\mathbf{x} - \mathbf{y})$$

**Corollary 3.13.** Due to the fact that a Kernel is a symmetric function, if a Kernel is stationary then

$$K(\mathbf{x}, \mathbf{y}) = K(\boldsymbol{\tau}) = K(-\boldsymbol{\tau}) = K(|\boldsymbol{\tau}|), \quad \boldsymbol{\tau} := \mathbf{x} - \mathbf{y}$$

**Theorem 3.14. (Bochner's Theorem)** A function  $K : \mathbb{R}^D \rightarrow \mathbb{R}$  is a stationary kernel, i.e. the ACF of a stationary square-integrable field, if and only if it can be represented as:

$$K(\boldsymbol{\tau}) = \int_{\mathbb{R}^D} e^{2\pi i \boldsymbol{\omega} \cdot \boldsymbol{\tau}} d\mu(\boldsymbol{\omega})$$

where  $\mu$  is a unique positive measure.

For the case of  $\tau \in \mathbb{R}$ , Bochner's Theorem is also known as the following:

**Theorem 3.15. (Wiener-Khinchine Theorem)** For every stationary continuous-time square-integrable stochastic process  $x_t$  there exists a monotone function  $F(f)$  such that the level-autocovariance of the process  $K_{XX}(\tau) = \mathbb{E}[X_t X_{t-\tau}]$  can be written as:

$$K_{XX}(\tau) = \int_{-\infty}^{\infty} e^{2\pi i f \tau} dF(f)$$

In the  $N$ -D case we write  $d\mu(\boldsymbol{\omega}) = S(\boldsymbol{\omega})d\boldsymbol{\omega}$ , then  $S(\boldsymbol{\omega})$  is called the spectral density of the Kernel/process. Essentially this is the Fourier transform of the Kernel:

$$S(\boldsymbol{\omega}) = \int e^{-2\pi i \boldsymbol{\omega} \cdot \boldsymbol{\tau}} K(\boldsymbol{\tau}) d\boldsymbol{\tau}$$

Note that  $K(0) = \int S(\boldsymbol{\omega})d\boldsymbol{\omega}$  must be finite for square-integrable field (why?), which is why without loss of generality the process

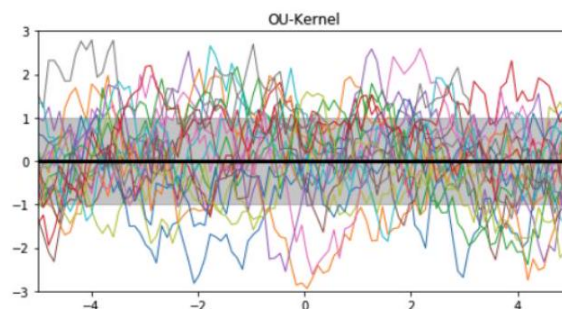
can be rescaled so that  $\int d\mu(\boldsymbol{\omega}) = \int S(\boldsymbol{\omega})d\boldsymbol{\omega} = 1$ .

Square-integrability is therefore required in order to ensure that  $d\mu$  is a proper measure.

#### Stationary Kernels: Intuition

Because the Kernel (or variance-covariance)  $K(\mathbf{x}, \mathbf{y})$  only depends on the distance  $|\mathbf{x} - \mathbf{y}|$  between  $\mathbf{x}$  and  $\mathbf{y}$ , but not on the actual positions  $\mathbf{x}, \mathbf{y}$ , the local variability of the prior paths/surfaces generated by stationary Kernels is the same throughout the entire domain. Non-stationary kernels, on the other hand, generate spatially varying local variance/covariance of the prior paths/surfaces.

For example, the linear kernel is not stationary, as is clearly also visible in Figure 2. An example of a stationary Kernel is that of the OU process (11) whose prior paths can be seen below:



**Figure 3:** Prior paths from OU Kernel (or  $\nu = 1/2$  Matern Kernel with scale parameter  $\ell$ ),  $K(x, y) = \exp(-\tau/\ell)$ .

#### Example 3.16. (Examples of Stationary Kernels)

- OU Kernel in Equation (6).
- RBF Kernel:  $K_{RBF}(\mathbf{x}, \mathbf{x}') = K_{RBF}(\tau) = \exp(-\tau^2/2\ell^2)$ , where  $\tau = \sqrt{\tau} = \sqrt{|\mathbf{x} - \mathbf{x}'|}$  where  $\ell > 0$  is the scale hyperparameter of the Kernel. Why is this a valid Kernel?
- Matern Kernel:  $K_{Matern}(\tau) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\tau}{\ell}\right)^\nu K_\nu\left(\frac{\sqrt{2\nu}\tau}{\ell}\right)$  where  $\nu, \ell > 0$  are the hyperparameters of the Kernel. It turns out that as  $\nu \rightarrow \infty$  the Matern Kernel becomes the RBF Kernel  $K_{Matern} \rightarrow \exp(-\tau^2/2\ell^2)$  while when  $\nu = 1/2$  it becomes the OU Kernel  $K_{Matern}|_{\nu=1/2} = \exp(-\tau/\ell)$
- $\gamma$ -exponential Kernel:  $K_{\gamma\text{-Exp}}(\tau) = \exp(-(\tau/\ell)^\gamma)$ ,  $0 < \gamma \leq 2$
- Rational-Quadratic Kernel:  $K_{RQ}(\tau) = \left(1 + \frac{\tau^2}{2\alpha\ell^2}\right)^{-\alpha}$
- Anisotropic stationary Kernels: Simply set  $\tau^2 = (\mathbf{x} - \mathbf{x}')^\top \mathbf{S}^{-1} (\mathbf{x} - \mathbf{x}')$  where  $\mathbf{S}$  is a positive-definite matrix.
- Anisotropic factor models: In the above we simply specify the  $D \times D$  matrix  $\mathbf{S}$  by its dimensionally reduced factor structure  $\mathbf{S} = \mathbf{V}\mathbf{V}^\top + \mathbf{R}$  where  $\mathbf{V}$  is a  $D \times k$  matrix specifying the loadings of the  $k$  factors and  $\mathbf{R}$  is a diagonal positive definite matrix specifying the idiosyncratic variances of each of the coordinates of  $\mathbf{x}$

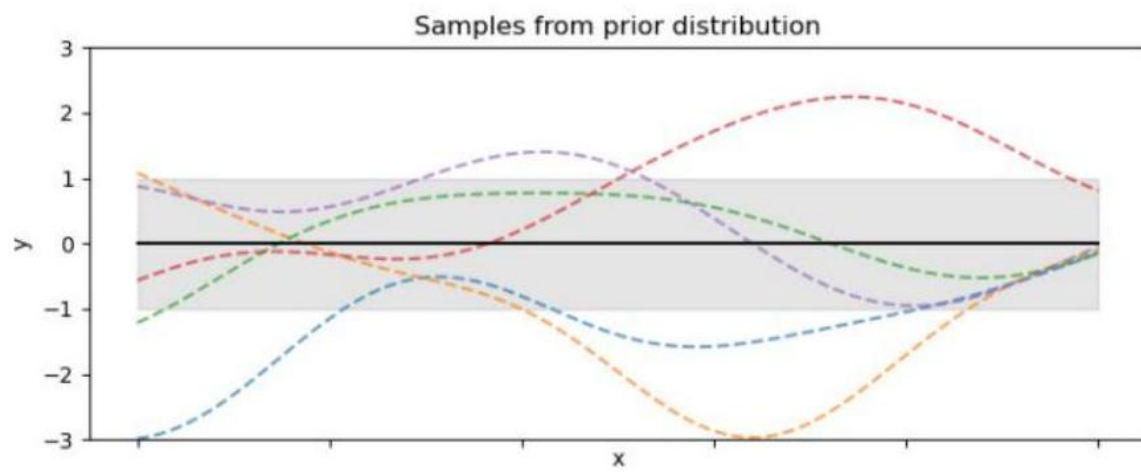


Figure 4: Sample paths from the prior of the RBF Kernel  $K(x_1, x_2) = \exp(-|x_1 - x_2|^2 / \sigma^2)$

## 4 Bayesian Learning With GPs

### Generating From The Prior

Generating from the prior amounts to choosing a set of test grid points  $\mathbf{X}_* = (\mathbf{x}_{*,i})_{i=1}^M$  and generating different realizations of  $\mathbf{y}_* = (y_{*,i})_{i=1}^M$  by drawing from the following multivariate Gaussian (prior) distribution

$$\mathbf{y}_* | \mathbf{X}_* \sim N(0, K(\mathbf{X}_*, \mathbf{X}_*))$$

### Training and Generating From The Posterior

Generating from the posterior amounts to choosing a set of test grid points  $\mathbf{X}_* = (\mathbf{x}_{*,i})_{i=1}^M$  and generating different realizations of  $\mathbf{y}_* = (y_{*,i})_{i=1}^M$  by drawing from the posterior multivariate Gaussian distribution in (24)

$$\mathbf{y}_* | \mathbf{X}_*, \mathbf{y}, \mathbf{X} \sim N(\mathbb{E}[\mathbf{y}_* | \mathbf{X}_*, \mathbf{y}, \mathbf{X}], \text{Cov}[\mathbf{y}_* | \mathbf{X}_*, \mathbf{y}, \mathbf{X}])$$

where the conditional mean and covariance are as in (16).

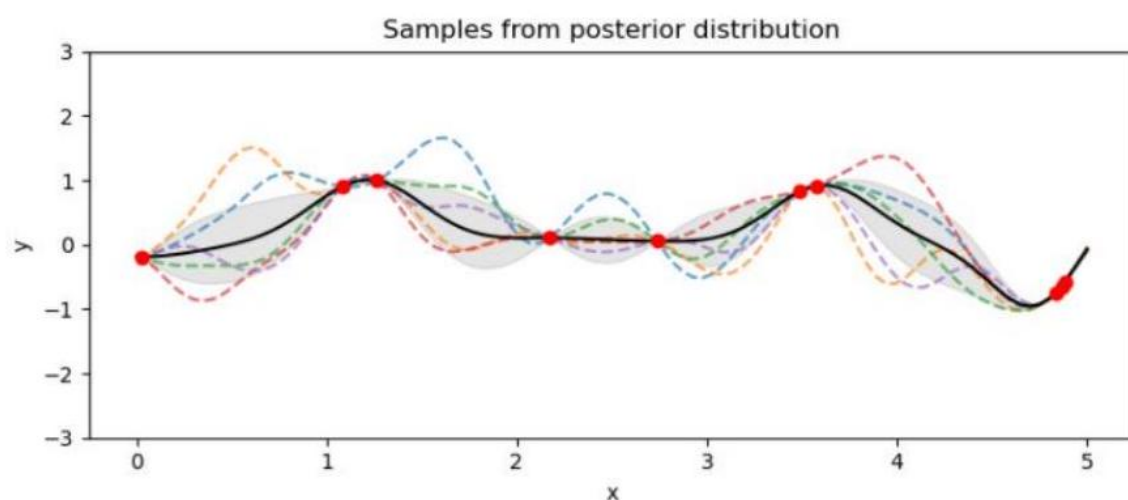


Figure 5: Sample paths from the posterior of the RBF Kernel  $K(x_1, x_2) = \exp(-|x_1 - x_2|^2 / \sigma^2)$  based on a training set of only 9 samples.

### The Shape Of The Generated Path

Why is the generated path so smooth? Shouldn't it be bouncing up and down since it's sampling from a Gaussian distribution?

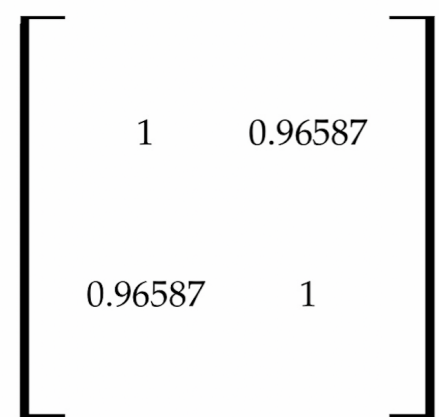
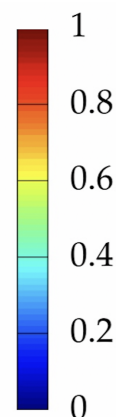
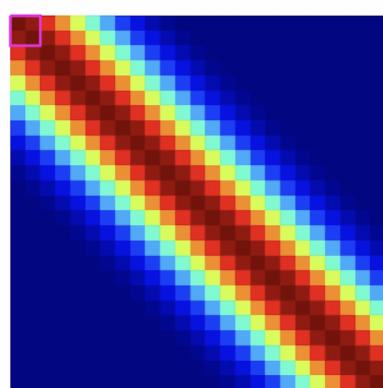
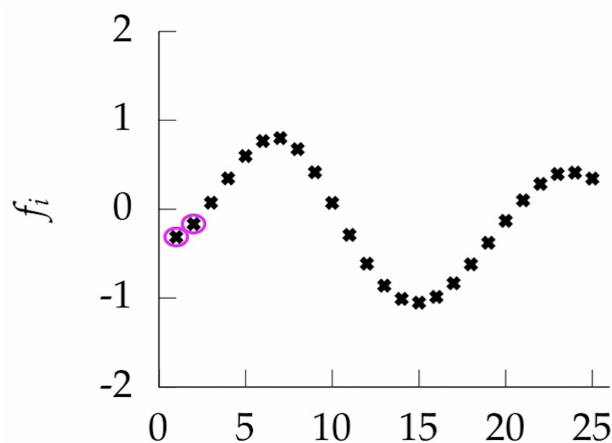


Figure 6: A 25 dimensional multivariate Gaussian R.V. sample from the corresponding RBF kernel. Here  $f_i \equiv y_i$ .

Figure 7: Color map showing correlations between dimensions.

Figure 8: The zoomed in correlation of the first two samples.

In the above example, we generated a path  $\{x_i, y_i\}_{i=1}^{25}$  using the (Gaussian) prior distribution (directly treat  $\mathbf{x}$  as test data) where its covariance matrix is specified by  $\mathbf{x} = (1, 2, \dots, 25)$  and the RBF kernel. The generated  $\mathbf{y} \in \mathbb{R}^{25}$ , in Figure 6 is a single sample from this distribution. We can see from Figure 7 that adjacent samples are heavily correlated, whereas distant samples are not, and this is exactly why the sample path generated from this RBF prior is relatively smooth instead of bouncing up and down.

## 5 Model Selection and Hyperparameter Optimization

### What is GPR Model Selection?

As we noted earlier, the training of a Gaussian Process [Equation \(5\)](#) for a fixed Kernel does not involve fitting any parameters. In other words, we have solved the  $L_2$  optimization problem for the predictions  $y_*$  conditional on  $X_*$  and training data  $y, X$  in a single optimization step.

However, in many cases the Kernel itself may depend, generally in a non-linear fashion, on a set of hyperparameters  $\theta$ ,  $K(x, x') = K(x, x'; \theta)$ . For example, the  $\ell$  parameter in the RBF Kernel is a hyperparameter. If  $\ell$  is too small, the bandwidth of the Kernel may be too narrow and the model overfits; if  $\ell$  is too large, the bandwidth of the Kernel may be too wide and the model underfits. The optimal  $\ell$  balances the bias-variance tradeoff.

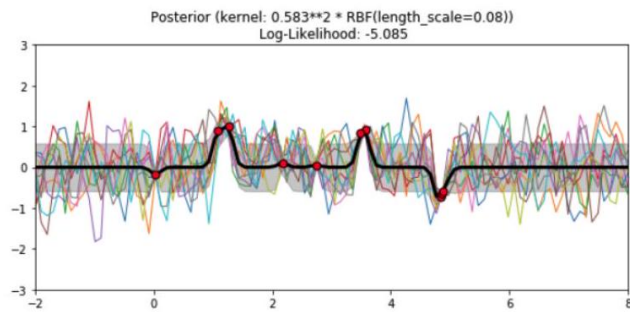


Figure 9: An RBF fit with narrow bandwidth.

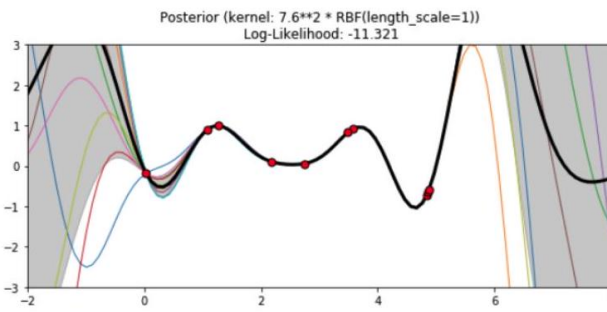


Figure 10: An RBF fit with wide bandwidth.

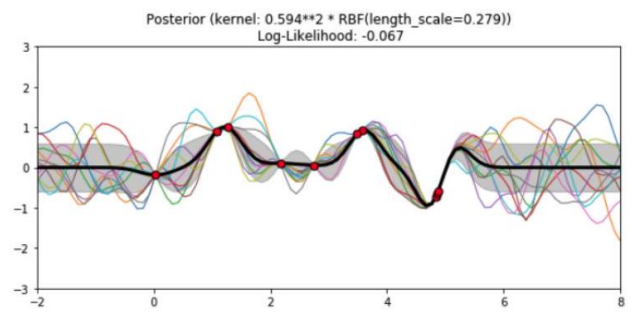


Figure 11: The optimal bandwidth RBF fit.

One way to find  $\theta_{\text{opt}}$  would be through grid search and cross-validation. However in bayesian modelling there is typically insufficient training data to adopt this approach. What are the alternatives?

### Optimizing the Likelihood of the Data

Within the Bayesian framework, hyperparameter optimization amounts to optimizing the likelihood of the data as a function of the hyperparameters:

$$L(\theta | y, X) = \log p(y | X; \theta) = -\frac{1}{2} y' (K(X, X; \theta) + \sigma^2 I)^{-1} y - \frac{1}{2} \log |K(X, X; \theta) + \sigma^2 I| - \frac{n}{2} \log 2\pi$$

In general (63) is non-trivial both because of the non-linear dependence in  $\theta$  and because of the fact that at gradient optimization step the inverse of  $K(X, X; \theta) + \sigma^2 I$  would have to be computed.

Fortunately, in many problems one does not require many hyperparameters to optimize over. Quite often, using a simple Kernel such as the RBF kernel suffices in getting good results. However, **hyperparameter optimization is in fact one of the biggest bottlenecks of the GPR approach**, especially in the limit of large amounts of data. On the other hand, Bayesian approach is precisely most useful in the opposite limit of small amounts of data which is especially relevant to finance.

## 6 Topic: Links to SVD/PCA

### 6.1 Mercer's Theorem

In a previous section we found that a sufficient condition for a Kernel to be valid is to express it as a covariance function, or alternatively a quadratic outer product of a set of feature maps,  $K(x, x') = \Phi(x)\Phi(x')^T = \sum_{\alpha} \phi_{\alpha}(x)\phi_{\alpha}(x')$  in which case the second Mercer condition follows directly. This is great, however, under what condition can we decompose a Kernel as an outer product of feature maps? The answer is given by the Mercer's theorem.

**Theorem 6.1. (Mercer's Theorem, 1909)** Every valid Kernel in 1-D, i.e. a symmetric function  $K(s, t), s, t \in [a, b]$  satisfying the Mercer condition [Definition 3.1](#), can be expressed as:

$$K(s, t) = \sum_{i=1}^{\infty} \lambda_i e_i(s) e_i(t)$$

where  $\{\lambda_i \geq 0, e_i(t)\}_{i=1}^{\infty}$  are the solutions of the following *Friedholm Integral Eigenvalue equation*:

$$\int_a^b K_Y(s, t) e_k(s) ds = \lambda_k e_k(t)$$

and  $\{e_i(t)\}_{i=1}^{\infty}$  are orthonormal w.r.t. the Hilbert norm  $\langle e_i, e_j \rangle_{\mathcal{H}} := \int_a^b e_i(t) e_j(t) dt = \delta_{ij}$ .

The feature maps  $e_i(t)$  are the functional analogue of PCA factors of the Kernel. Indeed, they are orthonormal w.r.t. the Hilbert norm  $\langle \cdot, \cdot \rangle_{\mathcal{H}}$  and so they form a complete basis of functions which are linear superspositions of the factors. What can we say about such functions?

### 6.2 The Hilbert Space of the Kernel

Consider the functions generated by this Mercer orthonormal basis associated with a given Kernel:

$$f(t) = \sum_{i=0}^{\infty} \alpha_i e_i(t) \tag{7}$$

It is easy to check that these functions form a real vector space embodied with an inner product  $\langle \cdot \rangle = \langle \cdot, \cdot \rangle_{\mathcal{H}}$ . In other words these functions form a Hilbert Space  $\mathcal{H}$ . What type of functions are in  $\mathcal{H}$ ?

For example, if each of the basis functions  $\{e_i(t)\}_{i=1}^{\infty}$  were differentiable it is clear that if one were to consider functions in [Equation \(7\)](#) generated by finitely many of the basis vectors, then the result would be differentiable as well.

As differentiable functions have zero quadratic variation, so would this imply that the Hilbert space  $H$  only consists of deterministic functions? It turns out the answer is no precisely because the expansions in [Equation \(7\)](#) constitute of countably infinite series of superspositions of  $\{e_i(t)\}$ .

In fact, if the Kernel is a covariance of a given square-integrable stochastic process, then it turns out that there exists a sequence of functions in  $\mathcal{H}$  which converges in  $L_2$ -sense to any given path of the process.



### 6.3 Karhunen-Loève (KL) Expansion

Suppose  $K(t, s) = \mathbb{E}[Y_t Y_s]$  for a zero-mean square-integrable process over a probability space  $(\Omega, \mathcal{F}, \mathbb{P})$  indexed over the closed interval  $[a, b]$ . Then  $K(t, s)$  is a Mercer Kernel and can be written as  $K(s, t) = \sum_{i=1}^{\infty} \lambda_i e_i(s) e_i(t)$  for some  $\{\lambda_i \geq 0, e_i(t)\}_{i=1}^{\infty}$ . Then we have the following theorem

**Theorem 6.2. (Karhunen-Loève (KL) Expansion)** Any given path  $Y_t$  of the process can be written as:

$$Y_t =_{L^2} \sum_{k=1}^{\infty} \sqrt{\lambda_k} \xi_k e_k(t) = \sum_{k=1}^{\infty} Y_{k,t}$$

where  $\xi_k$  are each independently drawn with zero mean and unit variance and can be found by projecting the path onto the  $k$ -th basis:

$$\xi_k = \frac{1}{\sqrt{\lambda_k}} \int_a^b Y_t e_k(t) dt$$

The KL expansion can be thought of as a functional generalization of PCA where  $Y_{k,t}$  is the projection of  $Y_t$  on to the  $k$ -th PCA component  $e_k(t)$  of the Kernel  $K_Y(s, t)$ . In fact, similar to PCA the KL expansion can be thought of as the optimal MSE minimizer as follows

**Theorem 6.3. (The KL Expansion is MSE-Optimal)** For any order  $K \geq 1$ , the first-  $K$  terms in the KL expansion optimize the MSE as follows:

$$\mathbb{E} \left| Y_t - \sum_{k=1}^K \sqrt{\lambda_k} \xi_k e_k(t) \right|^2 \leq \mathbb{E} \left| Y_t - \sum_{k=1}^K \eta_k \phi_k(t) \right|^2$$

where  $\{\phi_k(t)\}_{k=1}^K$  is any other set of orthonormal functions.

**Example 6.4. (KL Basis For Brownian Motion)** If the underlying process is  $Y_t = W_t = \int_0^t dW_s$  then  $K_Y(t, s) = \min(t, s)$  and the KL basis is known analytically. Assuming  $[a, b] = [0, 1]$ :

$$e_k(t) = \sqrt{(2) \sin \left( \left( k - \frac{1}{2} \right) \pi t \right)}, \quad \lambda_k = \frac{1}{\left( k - \frac{1}{2} \right)^2 \pi^2}$$

and therefore each Brownian Path admits the following Wiener representation:

$$W_t = \sqrt{2} \sum_{k=1}^{\infty} \xi_k \frac{\sin \left( \left( k - \frac{1}{2} \right) \pi t \right)}{\left( k - \frac{1}{2} \right) \pi}$$

Note that

- The scaled process  $\sqrt{c}W_{t/c}$  in (53) is a Brownian motion on  $[0, c]$ .
- The KL eigenfunctions in (51) for Brownian Motion are in fact Fourier components with a suitable phase.
- What distinguishes Brownian paths  $W_t$  from all other paths is that the variance of each random Fourier mode of  $W_t$  decays quadratically with  $k$  and in fact equals  $1/\lambda_k = 1/(k - 1/2)^2 \pi^2$ .
- The projection formula (49) for the case of Brownian Motion reduces to a suitable inverse Fourier transform (iFFT). The random coefficients  $\xi_k$  of a path  $Y_t$  are the (random) coordinates of  $Y_t$  in the PCA basis of  $K_Y(t, s)$  and (49) can be thought of as the inverse- KL (iKL) transform with respect to the basis of  $K_Y$ .

#### KL Expansion as MSE Minimization

As a PCA basis of  $K_Y$ , the KL basis in (7) also minimizes the MSE when approximating any given realized path  $Y_t$  generated by the process in the following sense. Given any orthonormal basis of  $\{\phi_k\}_{k=1}^{\infty}$  of  $L^2$ , the truncated KL basis provides the best MSE approximation of  $Y_t$  in the sense that

$$\mathbb{E} \left| Y_t - \sum_{k=1}^K \sqrt{\lambda_k} \xi_k e_k(t) \right|^2 \leq \mathbb{E} \left| Y_t - \sum_{k=1}^K \eta_k \phi_k(t) \right|^2$$

for all  $K \geq 1$ .

**Example 6.5. (A Different Representation of Brownian Paths)** An alternative representation of Brownian Motion also due to Wiener is:

$$W_t = \xi_0 t + \sqrt{2} \sum_{k=1}^{\infty} \xi_k \frac{\sin \pi k t}{\pi k} \quad (8)$$

However Equation (8) is not MSE-optimal. In fact the linear in  $t$  term is not even orthogonal to the Fourier ones. In other words (55) does not decompose  $W_t$  into uncorrelated sources of risk and is not MSE optimal at any order.

### 6.4 The Spectral Theorem

So far, Mercer's Theorem involved Kernels  $K(t, t'), t, t' \in [a, b]$ , which led to the KL expansion of paths of Gaussian Processes. Do similar results exist for (stochastic) functions of higher dimensional features  $\mathbf{x} \in V$  and more general vector fields  $V$ ? For example if  $V = \mathbb{R}^D$  then instead of continuous stochastic paths corresponding to Gaussian Processes, one would be considering stochastic continuous surfaces corresponding to Gaussian Fields.

It turns out historically the generalizations were in fact derived by Hilbert prior to Mercer's 1909 work in a series of his papers in 1904-1910 and later were fully developed by von Neumann and Stone in the 1920s. All such results are known as Spectral Theorem (s). For an overview, see steen72spectralhist.

Let's work backwards and state one of the Spectral Theorems which is perhaps the closest generalization of Mercer's Theorem:



**Theorem 6.6. (Spectral Theorem: Compact Self-Adjoint Operators)** Every compact self-adjoint operator  $K : \mathcal{H} \rightarrow \mathcal{H}$ , where  $\mathcal{H}$  is some Hilbert space of functions  $\phi : V \rightarrow \mathbb{R}$ , has a countable real basis  $(\lambda_i \in \mathbb{R}, e_i(x))_{i=1}^{\infty}, x \in V$  such that

$$K(x, x') = \sum_{i=0}^{\infty} \lambda_i e_i(x) e_i(x')$$

with  $\lambda_i \rightarrow 0$ . If  $K$  is positive semidefinite, i.e. if  $\langle \phi, K\phi \rangle_{\mathcal{H}} \geq 0$  for all  $\phi \in \mathcal{H}$  then the  $\lambda_i$  are non-negative.

To gain intuition on the previous theorem, we need to clarify what a compact, self-adjoint and positive semidefinite operator is. These are all concepts in infinite dimensional functional calculus generalizing the analogous concepts in finite dimensional linear algebra.

### Hilbert-Schmidt Operators

Following Feldman's notes on the matter, similar to results in finite dimensional linear algebra where a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  can be thought of as a linear operator  $\mathbf{A} : \mathbb{R}^n \rightarrow \mathbb{R}^m$  mapping a column vector  $\mathbf{u} \in \mathbb{R}^n$  to a row vector  $\mathbf{v} = \mathbf{A}\mathbf{u} \in \mathbb{R}^m$  by left-multiplication, given two measurable spaces  $(X, \mu), (Y, \nu)$  one can think of a real-valued function of two variables  $K(x, y), x \in X, y \in Y$  as the following linear operator from the Hilbert Space of square-integrable functions  $\mathcal{H}_Y = L_2(Y, \nu)$  of  $Y$  to that of  $\mathcal{H}_X = L_2(X, \mu)$  of  $X$ :

$$(Kf)(x) := \int_Y K(x, y) f(y) d\nu(y) = \langle \cdot, Kf \rangle_{\mathcal{H}}$$

Or  $K : \mathcal{H}_Y \rightarrow \mathcal{H}_X$  is simply the linear convolution operator w.r.t. the second variable of  $K$ .

Similarly to the finite-dimensional case where  $\mathbf{A}$  could alternatively be considered to act on the row vectors by right-multiplication to produce column vectors, one defines the adjoint of  $K$  as  $K^\dagger : \mathcal{H}_X \rightarrow \mathcal{H}_Y$  as:

$$(K^\dagger g)(y) := \int_X g(x) K(x, y) d\mu(x) := \langle K^\dagger g, \cdot \rangle_{\mathcal{H}}$$

An operator  $K$  is self-adjoint if  $K^\dagger = K$ . It is clear that if  $K(x, x') = K(x', x)$  is a symmetric real-valued function of two variables  $x, x' \in (X, \mu)$  then  $K$  is also self-adjoint.

### Compact Operators on $L_2$ Hilbert Spaces

**Definition 6.7.** A linear operator  $C : \mathcal{H}_1 \rightarrow \mathcal{H}_2$  is compact if for every bounded sequence  $(f_i(x))_{i=1}^{\infty} \in \mathcal{H}_1$  there is a subsequence of  $((Cf_i)(x))_{i=1}^{\infty} \in \mathcal{H}_2$  which is bounded.

**Proposition 6.8.** If  $\mathcal{H}_1$  and  $\mathcal{H}_2$  are Hilbert spaces with inner products  $\langle \cdot, \cdot \rangle_1$  and  $\langle \cdot, \cdot \rangle_2$  respectively, then an operator  $C : \mathcal{H}_1 \rightarrow \mathcal{H}_2$  is compact if and only if  $C$  is completely continuous, i.e.  $\|Cf_n - Cf\|_2^2 \rightarrow 0$  whenever  $\langle g, f_n \rangle_1 \rightarrow \langle g, f \rangle_1$  for all  $g$  in  $\mathcal{H}_1$ . Such operators are also called continuous with respect to the weak topology.

### Relevance to ML and Gaussian Processes

**Example 6.9.** Let a function  $C : V_1 \times V_2 \rightarrow \mathbb{R}$  be a continuous real-valued two-variable function whose domain  $V_1 \times V_2$  is a compact vector space (for example  $V_i = [0, 1]^{D_i}$ ). Then when viewed as a linear operator  $C : \mathcal{H}_{L_2(V_2, \nu)} \rightarrow \mathcal{H}_{L_2(V_1, \mu)}$  from the Hilbert spaces of  $L_2$ -integrable functions on  $V_2$  and to that on  $V_1$ ,  $C$  is a compact operator.

**Corollary 6.10.** A symmetric real-valued function  $K(\mathbf{x}, \mathbf{y})$  defined on compact domains of  $\mathbb{R}^D$  is a self-adjoint compact operator in the Hilbert space of square-integrable functions on the same domain, and as a result has a countably discrete spectrum so that:

$$K(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\infty} \lambda_i e_i(\mathbf{x}) e_i(\mathbf{y})$$

where  $\langle e_i, e_j \rangle = \delta_{ij}$ . If in addition  $K$  is a positive semidefinite operator, e.g.  $\langle g, Kf \rangle \geq 0$  for all  $f, g \in \mathcal{H}$ , then the spectrum is non-negative  $\lambda_i \geq 0$ .

Once we establish a countably-infinite non-negative spectrum of  $K$ , all the Karhunen-Loève Expansion results also follow.

In particular, if (59) holds where  $K(x, x') = E[Y_{\mathbf{x}} Y_{\mathbf{x}'}]$  is the covariance kernel of any  $L_2$  integrable continuous stochastic field, then the basis  $e_i$  can approximate optimally any realization of  $Y_{\mathbf{x}}$  to arbitrary finite accuracy.

$$\mathbb{E} \left| Y_{\mathbf{x}} - \sum_{i=1}^L \sqrt{\lambda_i} \xi_i e_i(\mathbf{x}) \right|^2 \leq \mathbb{E} \left| Y_{\mathbf{x}} - \sum_{i=1}^L \eta_i \phi_i(\mathbf{x}) \right|^2$$

## 6.5 Posterior Prediction In the Mercer Basis

Suppose we know how to diagonalize a Kernel so we know its spectral basis (59). Then the posterior expectation formula (16) can be written as (show this!):

$$\mathbb{E}[y_* | \mathbf{X}_*, \mathbf{y}, \mathbf{X}] = K(\mathbf{X}_*, \mathbf{X}) [K(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}]^{-1} \mathbf{y} = \sum_k \frac{\lambda_k}{\lambda_k + \sigma^2} \mathbf{P}_k(\mathbf{X}_*, \mathbf{X}) \mathbf{y}$$

where the matrices  $\mathbf{P}_k$  in (62) are defined as:

$$\mathbf{P}(\mathbf{X}_*, \mathbf{X}) := \begin{bmatrix} e_k(\mathbf{x}_{*,1}) e_k(\mathbf{x}_1) & \cdots & e_k(\mathbf{x}_{*,1}) e_k(\mathbf{x}_n) \\ \vdots & & \vdots \\ e_k(\mathbf{x}_{*,m}) e_k(\mathbf{x}_1) & \cdots & e_k(\mathbf{x}_{*,m}) e_k(\mathbf{x}_n) \end{bmatrix}$$

and are projection matrices  $P_k(\mathbf{x}, \mathbf{y}) = e_k(\mathbf{x}) e_k(\mathbf{y})$  evaluated on the train-test data.

Question: how does the posterior covariance look like in the Mercer basis?

### Computing the Mercer Basis

Typically, computing the Mercer Basis involves solving the Friedholm equation (7):

$$\int_V K(\mathbf{x}, \mathbf{y}) e_k(\mathbf{y}) d\mu(V) = \lambda_k e_k(\mathbf{x})$$

which is typically not analytically tractable even if the kernel  $K(\mathbf{x}, \mathbf{y})$  is. Therefore, for most general kernels, the equation is not known functionally. However, for any train-test data, one can always diagonalize the kernel  $K(\mathbf{X}_*, \mathbf{X})$  matrix and each eigenspace contribution of this matrix would be a restriction of (63) to the data.

### References

- [1] Rasmussen, et al. (2006). Gaussian Processes In Machine Learning.
- [2] Sheffield, S. (2007). "Gaussian Free Fields for Mathematicians". In: 139 (3-4), pp. 521-541.