Gaussian Process

Kaiwen Zhou

Contents

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 relationshp $y = f(\mathbf{x})$ 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:

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

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 ϕ_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:

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

Feature map regressions are powerful models because for suitable choices of ϕ_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:

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

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 $\mathbf{A} = \mathbf{\Phi}(\mathbf{X})^\top \mathbf{\Phi}(\mathbf{X}) + \sigma^2 \mathbf{I}$. 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 $N_F \to \infty$! 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:

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

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

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

(the σ^{-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, $\mathbf{\Phi}(\mathbf{X}) \in \mathbb{R}^{N \times \{N_F \to \infty\}}$, 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, σ.

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 $\mathbb{E}[\mathbf{y}_* \mid \mathbf{X}_*, \mathbf{y}, \mathbf{X}]$ of the following probabilistic model:

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

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

$$\begin{aligned}
\mathbb{E}[\mathbf{y}_* \mid \mathbf{X}_*, \mathbf{y}, \mathbf{X}] &= K(\mathbf{X}_*, \mathbf{X})\left[K(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}\right]^{-1} \mathbf{y} \\
\mathrm{Cov}[\mathbf{y}_* \mid \mathbf{X}_*, \mathbf{y}, \mathbf{X}] &= K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}_*, \mathbf{X})\left[K(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}\right]^{-1} K(\mathbf{X}, \mathbf{X}_*)
\end{aligned} \tag{5}$$

If on the other hand $\mathbf{X}_* = \mathbf{X}$ and $\sigma = 0$, then $\mathbb{E}[\mathbf{y}_* \mid \mathbf{X}, \mathbf{y}, \mathbf{X}] = \mathbf{y}$ and $\mathrm{Cov}[\mathbf{y}_* \mid \mathbf{X}, \mathbf{y}, \mathbf{X}] = 0$. In this case, there is no uncertainty of the posterior predictions as they are assumed to equal the prior observations, $\mathbb{E}[\mathbf{y}_* \mid \mathbf{X}, \mathbf{y}, \mathbf{X}] = y$.

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 $Y_{\mathrm{x}} := Y(\mathrm{x}) : \mathbb{R}^D \to R$ is a Gaussian Field if and only if for any set of points $(x_i)_{i=1}^k$, the $k-$ dimensional variable*

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

is a multivariate Gaussian random variable. Equivalently, every linear combination $\mathbf{w} \cdot \mathbf{Y}_{\mathbf{x}_1,\ldots,\mathbf{x}_k}$ 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 for every set of indices $(t_i)_{k=1}^k$ in the index set T the k-dimensional variable*

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

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

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)*** *A kernel is a continuous function $K : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ with the following properties (Mercer Conditions):*

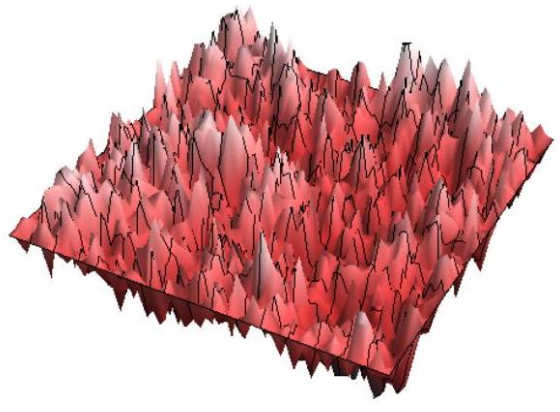

Figure 1: A 2D Gaussian Field.

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 $K_1 : \mathbb{R}^{D \times D} \to \mathbb{R}$ is a valid Kernel and $\mathbf{f} : \mathbb{R}^N \to \mathbb{R}^D$ 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 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 $K_1 : \mathbb{R}^{D_1 \times D_1} \to \mathbb{R}$ and $K_2 : \mathbb{R}^{D_2 \times D_2} \to \mathbb{R}$ are valid kernels. Suppose also $\mathbf{z} := (\mathbf{x}; \mathbf{y}) \in \mathbb{R}^{D_1 + D_2}$ where $\mathbf{x} \in \mathbb{R}_1^D$ and $\mathbf{y} \in \mathbb{R}_2^D$. 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?).*

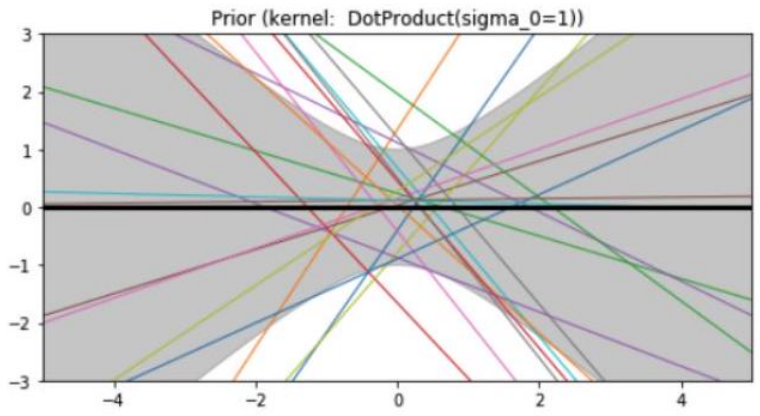

Figure 2: Prior paths from the Linear (Dot Product) Kernel mixed with a Constant Variance Kernel, $K(x, y) = \sigma^2 + xy$.

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 $\phi(\mathbf{x}) : \mathbb{R}^D \to R$.*

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 $t, s \in R$. The result then follows by realizing that the Feature Map kernel is a Composition Kernel of the scalar Linear Kernel.

3.2 Feature Map Kernel: Connection to Stochastic Processes/Fields

1-D Features: The Kernel of a Stochastic Process

For a one-dimensional square-intergrable 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 $V = \int_a^b \mathbb{E}\left[Y_t^2\right] dt = \int_a^b K_{YY}(t,t)dt$ of the process is finite.

Corollary

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

$$K_{\Pi\Pi}\left(t, t^\top\right) = \mathbb{E}\left[\Pi_t \Pi_t^\top\right] = \mathbb{E}\left[\int_a^t \Delta_s Y_s ds \int_a^{t^\top} \Delta_s^\top Y_s^\top ds^\top\right]$$

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

Example 3.5. ***(White-Noise Kernel)*** *The Kernel of* $dW_t \sim N(0, dt)$ *is*

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

Example 3.6. ***(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* $K\left(t, t^\top\right) = \min\left(\int_0^t \sigma^2(s)ds, \int_0^{t^\top} \sigma^2\left(s^\top\right) ds^\top\right) = \min\left(QVS(t), QVS\left(t^\top\right)\right)$ *where* $QVS(t) = \mathbb{E}\left[X_t^2\right] = \mathbb{E}\left[\int_0^t dX_s^2\right] = \int_0^t \sigma^2(s)ds$ *is the quadratic variance swap of* X_t*.*

Example 3.7. ***(Bachelier Local-Vol Kernel)*** *The Kernel of the Bachelier Local-Vol process* $dX_t = \sigma^2\left(X_t, t\right) dW_t$ *is:*

$$K_{LV}(s,t) = \mathbb{E}\left[X_s X_t\right] = \min\left(QVS(t), QVS\left(t^\top\right)\right)$$

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

Example 3.8. ***(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:*

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

N-D Features: The Kernel of a Stochastic Process

For a N-dimensional square-intergrable 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 $K_{YY}\left(\mathbf{x}, \mathbf{x}'\right)$ is a valid Kernel:

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

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

Example 3.9. ***(N-D Brownian Motion Kernel)*** *Suppose* $\mathbf{Y}_t \in R^N$ *be the level of* N*-dimensional correlated Brownian motion over a finite* t *- domain (to ensure square-integrability):*

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

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

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

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 τ we have $\mathbf{Y}_t = \left(Y_{t,\tau}\right)_\tau \in \mathbb{R}_+^{\mathbb{R}}$. Ignoring the carry drift, we are then modeling:

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

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

$Z_{\mathrm{x}} := Y_{t,\tau}$ we have:

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

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

3.4 Stationary Kernels

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

Definition 3.10. ***(Stationary Kernel)*** *A Kernel is stationary if*

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

Corollary 3.11. *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 3.12. ***(Bochner's Theorem)*** *A function $K : \mathbb{R}^D \to \mathbb{R}$ is a stationary kernel, i.e. the ACF of a stationary square-integrable field, if and only if it can be represented as:*

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

where μ is a unique positive measure.

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

Theorem 3.13. ***(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 $d\mu(\omega) = S(\omega)d\omega$, then $S(\omega)$ 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 $K(\mathbf{0}) = \int S(\boldsymbol{\omega})d\boldsymbol{\omega}$ 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 requred in order to ensure that $d\mu$ is a proper measure.

Stationary Kernels: Intuition

Because the Kernel (or variance-covariance) $K(\mathrm{x}, \mathrm{y})$ only depends on the distance $|\mathrm{x} - \mathrm{y}|$ between x and y, but not on the actual positions x, 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:

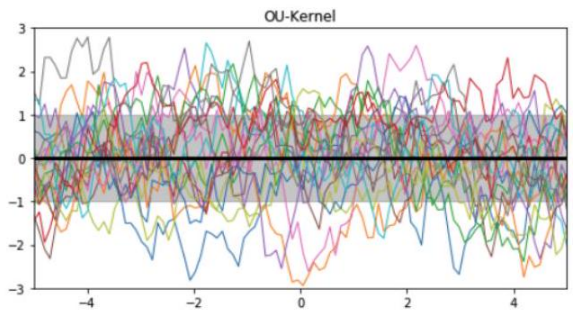

Figure 3: Prior paths from OU Kernel (or $\nu = 1/2$ Matern Kernel with scale parameter ℓ), $K(x, y) = \exp(-\tau/\ell)$.

Example 3.14. ***(Examples of Stationary Kernels)***

- *OU Kernel in Equation (6).*
- *RBF Kernel:* $K_{RBF}\left(\mathrm{x}, \mathrm{x}^{\top}\right) = K_{RBF}(\tau) = \exp\left(-\tau^2/2\ell^2\right)$*, where* $\tau = \sqrt{\tau} = \sqrt{|\mathrm{x} - \mathrm{x}^{prime}|}$ *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* $\nu \to \infty$ *the Matern Kernel becomes the RBF Kernel* $K_{Matern} \to \exp\left(-\tau^2/2\ell^2\right)$ *while when* $\nu = 1/2$ *it becomes the OU Kernel* $K_{Matern}\,|_{\nu=1/2} = \exp(-\tau/\ell)$
- γ*-exponential Kernel:* $K_{\gamma-\mathrm{Exp}}(\tau) = \exp\left(-(\tau/\ell)^{\gamma}\right), 0 < \gamma \leq 2$
- *Rational-Quadratic Kernel:* $K_{RQ}(\tau) = \left(1 + \frac{\tau^2}{2\alpha\ell^2}\right)^{-\alpha}$
- *Anisotropic stationary Kernels: Simply set* $\tau^2 = \left(\mathrm{x} - \mathrm{x}^{prime}\right)^{\top} \mathbf{S}^{-1} \left(\mathrm{x} - \mathrm{x}^{prime}\right)$ *where* $\mathbf{S}$ *is a positive-definite matrix.*
- *Anisotropic factor models: In the above we simply specify the* $D \times D$ *matrix* 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* $\mathbf{x}$

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

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

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)

$$\mathbf{y}_* \mid \mathbf{X}_*, \mathbf{y}, \mathbf{X} \sim N\left(\mathbb{E}\left[\mathbf{y}_* \mid \mathbf{X}_*, \mathbf{y}, \mathbf{X}\right], \mathrm{Cov}\left[\mathbf{y}_* \mid \mathbf{X}_*, \mathbf{y}, \mathbf{X}\right]\right)$$

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

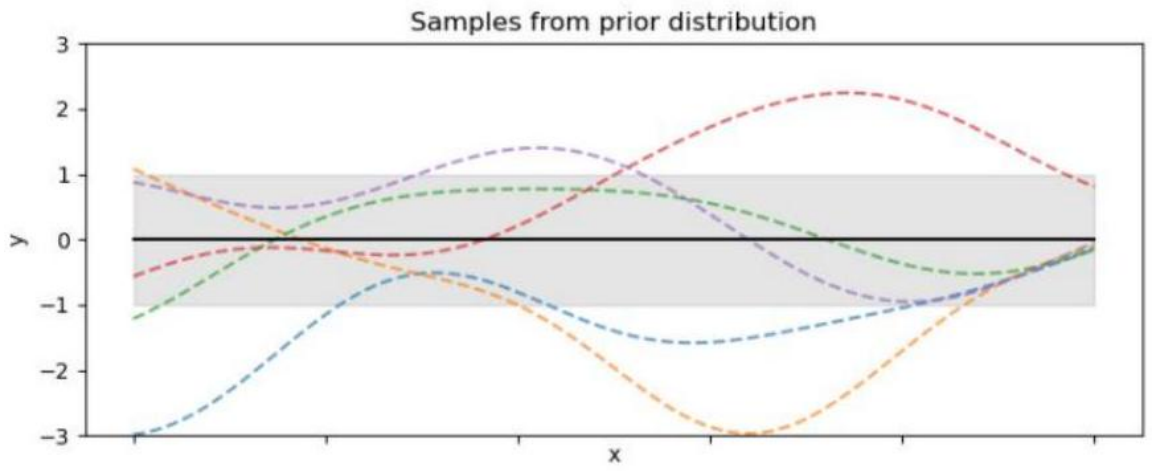

Figure 4: Sample paths from the prior of the RBF Kernel $K\left(x_1, x_2\right) = \exp\left(-\left|x_1 - x_2\right|^2 / \sigma^2\right)$

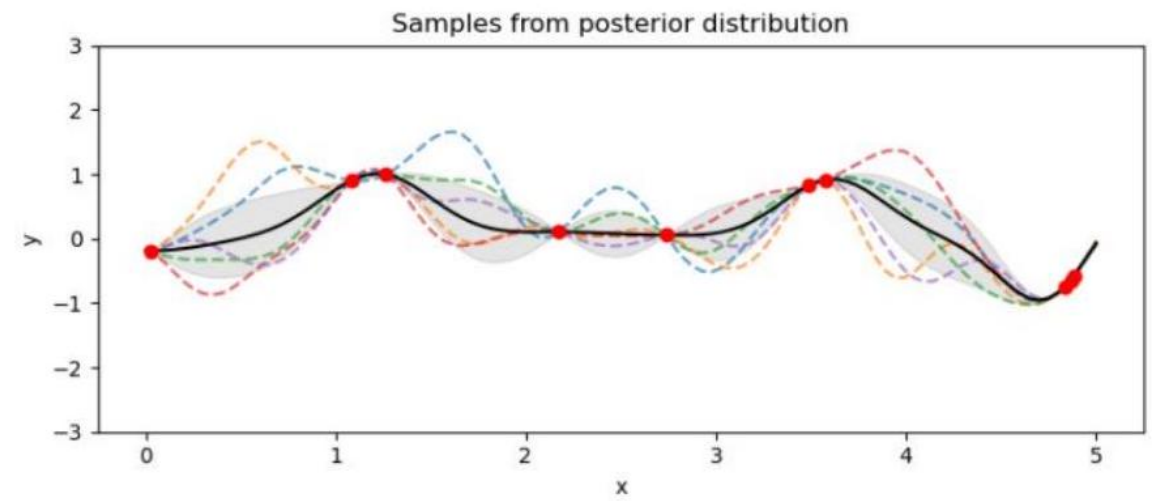

Figure 5: Sample paths from the posterior of the RBF Kernel $K\left(x_1, x_2\right) = \exp\left(-\left|x_1 - x_2\right|^2 / \sigma^2\right)$ based on a training set of only 9 samples.

5 Model Selection and Hyperparameter Optimization

What is GPR Model Selection? I

As we noted earlier, the training of a Gaussian Process (16) for a fixed Kernel does not involve fitting any parameters. In other words, we have solved the L_2 optimization problem for the predictions $\mathbf{y}_*$ conditional on $\mathbf{X}_*$ and training data $\mathbf{y}, \mathbf{X}$ 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 θ, $K\left(\mathrm{x}, \mathrm{x}'\right) = K\left(\mathrm{x}, \mathrm{x}'; \theta\right)$. For example, the ℓ parameter in the RBF Kernel is a hyperparameter.

What is GPR Model Selection? II

If ℓ is too small, the bandwidth of the Kernel may be too narrow and the model overfits:

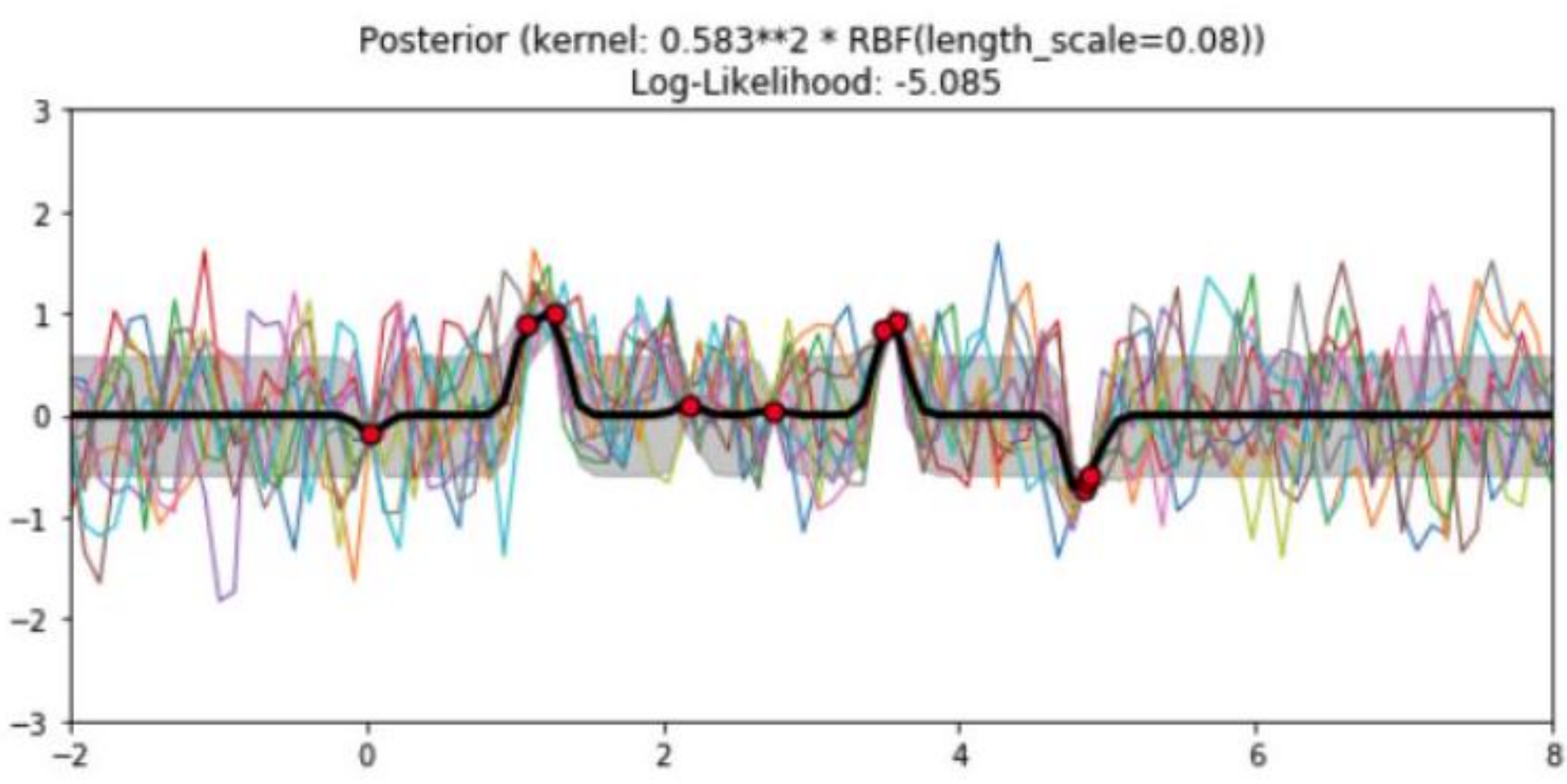

Figure 6: An RBF fit with narrow bandwidth.

What is GPR Model Selection? III

If ℓ is too large, the bandwidth of the Kernel may be too wide and the model underfits:

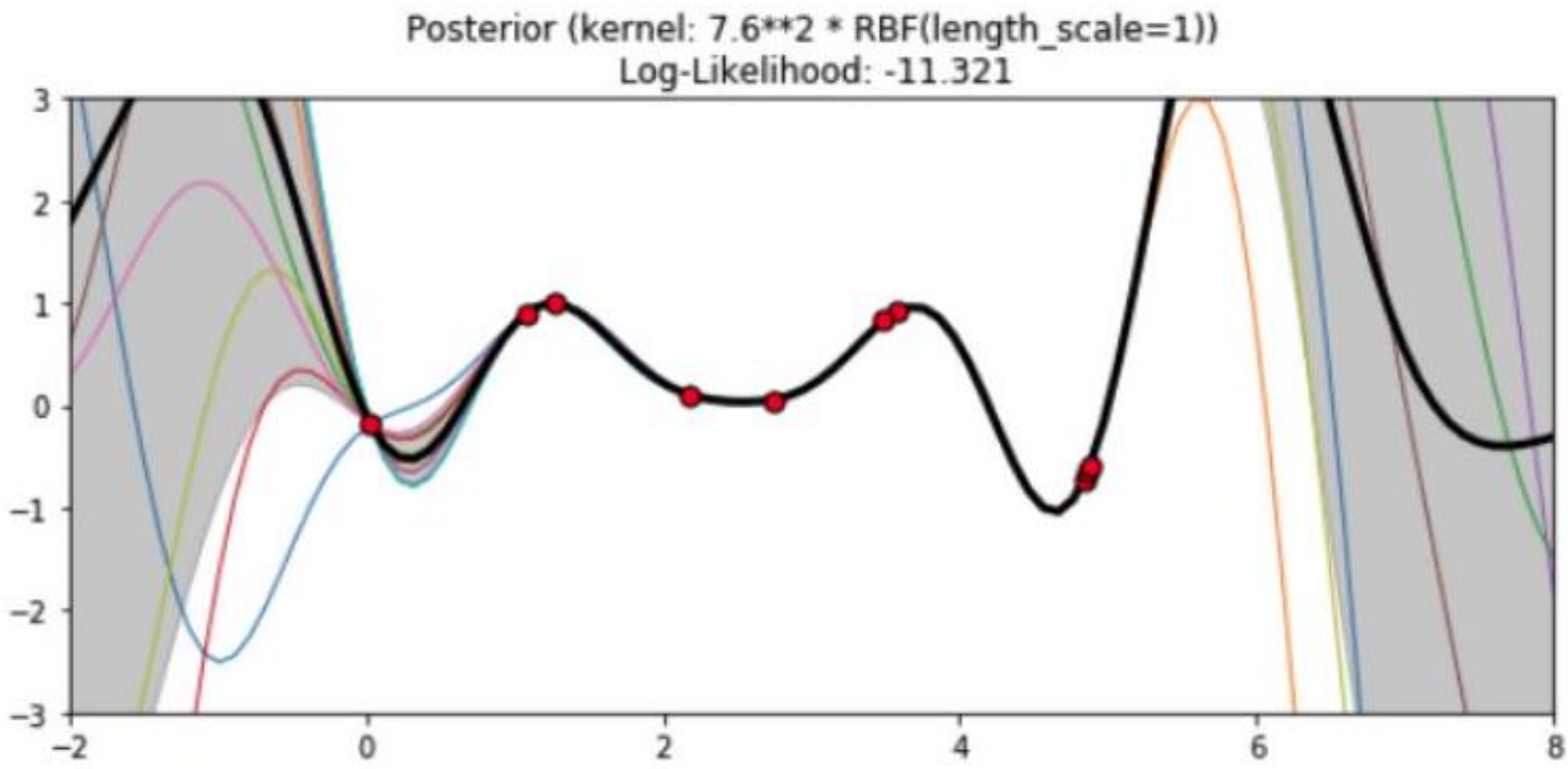

Figure 7: An RBF fit with wide bandwidth.

What is GPR Model Selection? IV

The optimal ℓ balances the bias-variance tradeoff.

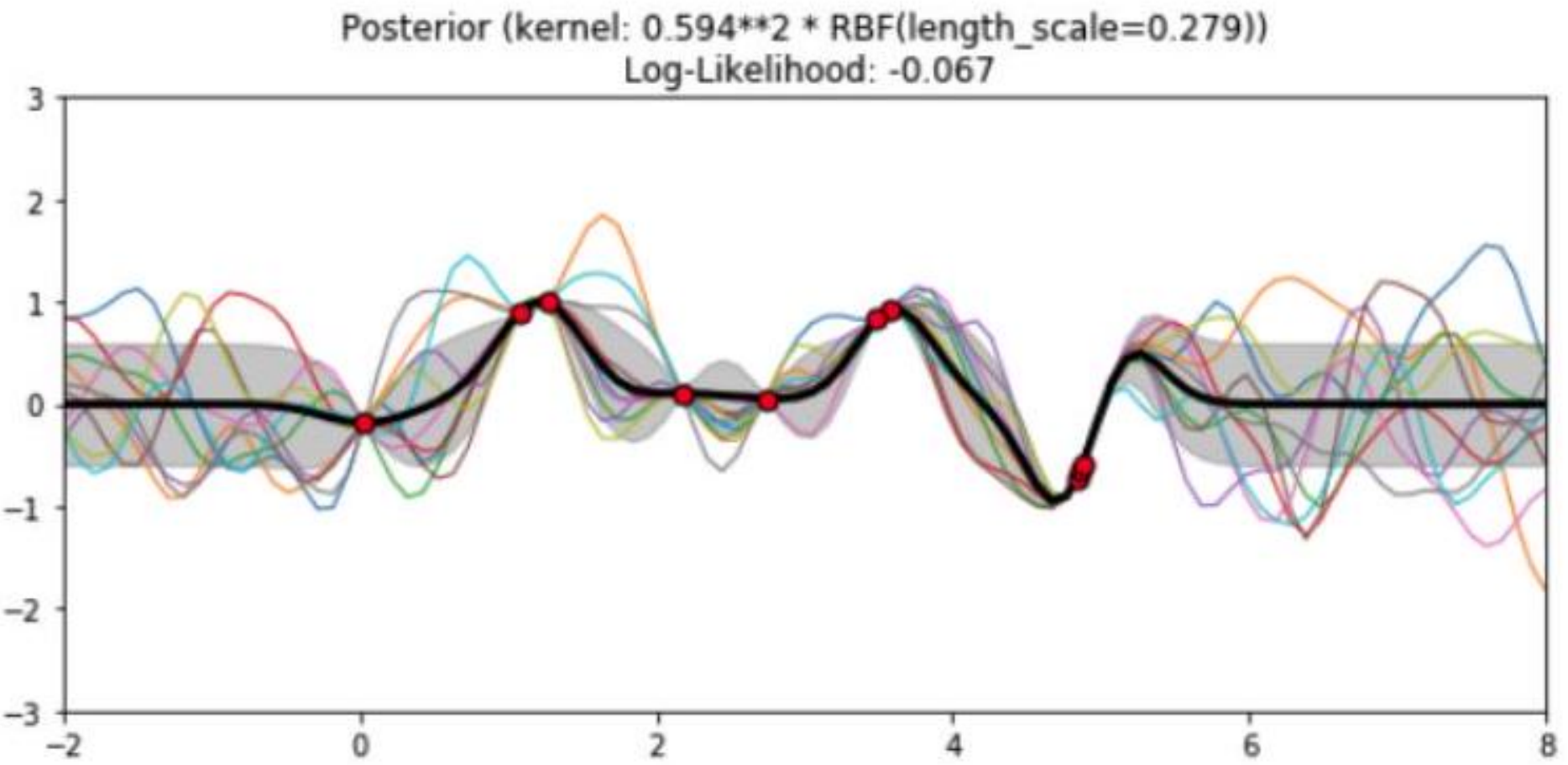

Figure 8: The optimal bandwidth RBF fit.

One way to find θ_{opt} would be through grid search and crossvalidation. However in bayesian modelling there is typically insufficient training data to adopt this approach. What are the alternatives?

Optimizing the Likelihood of the Data I

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

$$
\begin{aligned}
L(\theta \mid \mathbf{y}, \mathbf{X}) &= \log p(\mathbf{y} \mid \mathbf{X}; \boldsymbol{\theta}) \\
&= -\frac{1}{2}\mathbf{y}' \left(K(\mathbf{X}, \mathbf{X}; \boldsymbol{\theta}) + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{y} \\
&- \frac{1}{2}\log\left|\left(K(\mathbf{X}, \mathbf{X}; \boldsymbol{\theta}) + \sigma^2 \mathbf{I}\right)\right| - \frac{n}{2}\log 2\pi
\end{aligned}
$$

Optimizing the Likelihood of the Data II

In general (63) is non-trivial both because of the non-linear dependence in θ and because of the fact that at gradient optimization step the inverse of $K(\mathbf{X}, \mathbf{X}; \boldsymbol{\theta}) + \sigma^2 \mathbf{I}$ 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 one 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 oposite limit of small amounts of data which is especially relevant to finance.

Links to SVD/PCA: Mercer's Theorem, Karhunen-Loève Expansion And Their Generalizations

Mercer's Theorem I

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, $K\left(\mathrm{x}, \mathrm{x}'\right) = \mathbf{\Phi}(\mathrm{x})\mathbf{\Phi}\left(\mathrm{x}'\right)' = \sum_\alpha \phi_\alpha(\mathrm{x})\phi_\alpha\left(\mathrm{x}'\right)$ in which case the second Mercer condition follows directly.

Under what condition does the reverse hold, i.e. that we can decompose a Kernel as an outer product of feature maps? The following theorem shows that the Mercer

Mercer's Theorem II

Theorem (Merce, '1909)

Every valid Kernel in $1-D$, i.e. a symmetric function

$K(s,t), s,t \in [a,b]$ satisfying the Mercer condition (66), 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 wrt the Hilbert norm

$\langle e_i, e_j \rangle_{\mathcal{H}} := \int_a^b e_i(t) e_j(t) dt = \delta_{ij}$.

Mercer's Theorem III

The feature maps $e_i(t)$ are the functional analogue of PCA factors of the Kernel. Indeed, they are orthonormal wrt the Hilbert norm

$\langle , \rangle_{\mathcal{H}}$ and so they form a complete basis of functions which are linear superspositions of the factors. What can we say about such functions?

The Hilbert Space of the Kernel I

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

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

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

The Hilbert Space of the Kernel II

What type of functions are in 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 (47) 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?

The Hilbert Space of the Kernel III

It turns out the answer is no precisely because the expansions in

(47) constitue of countably infnite series of superpositions 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 H which converges in L_2-sense to any given path of the process.

Karhunen-Loève (KL) Expansion I

Suppose $K(t,s) = \mathbb{E}[Y_t Y_s]$ for a zero-mean square-integrable process over a probability space $(\Omega, F, \mathbf{P})$ 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

Karhunen-Loève (KL) Expansion II

Theorem (Karhunen-Loève (KL) Expansion)

Any given path Y_t of the process can be written as:

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

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

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

Karhunen-Loève (KL) Expansion III

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

Karhunen-Loève (KL) Expansion IV

Theorem (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.

KL Basis For Brownian Motion I

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 $[a,b] = [0,1]$:

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

and therefore each Brownian Path admits the following Wiener representation:

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

KL Basis For Brownian Motion II

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 Browninan Motion reduces to a suitable inverse Fourier transform (iFFT). The random coefficients ξ_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: A Different Representation of Brownian Paths

An alternative representation of Brownian Motion also due to Wiener is:

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

However (55) 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.

The Spectral Theorem I

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 exists for (stochastic) functions of higher dimensional features $\mathrm{x} \in V$ 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.

The Spectral Theorem II

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

Theorem (Spectral Thm: Compact Self-Adjoint Operators)

Every compact self-adjoint operator $K : \mathcal{H} \to R$, where $\mathcal{H}$ is some Hilbert space of functions $\phi : V \to \mathbb{R}$, 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 $\lambda_i \to 0$. 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 λ_i are non-negative.

The Spectral Theorem III

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 I

Following Feldman's notes on the matter, similar to results in finite dimemsional linear algebra where a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ can be thought of as a linear operator $\mathbf{A} : \mathbb{R}^n \to \mathbb{R}^m$ mapping a column vector $\mathbf{u} \in \mathbb{R}^n$ to a row vector $\mathbf{v} = \mathbf{Au} \in \mathbb{R}^m$ 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 $K : \mathcal{H}_Y \to \mathcal{H}_X$ is simply the linear convolution operator wrt the second variable of K.

Hilbert-Schmidt Operators II

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 $K^\dagger : \mathcal{H}_X \to \mathcal{H}_Y$ as:

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

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 I

Definition

A linear operator $C : \mathcal{H}_1 \to \mathcal{H}_2$ 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

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

Relevance to ML and Gaussian Processes I

Example

Let a function $C : V_1 \times V_2 \to \mathbb{R}$ 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 $C : \mathcal{H}_{L_2(V_2,\nu)} \to \mathcal{H}_{L_2(V_1,\mu)}$ from the Hilbert spaces of L_2-integrable functions on V_2 and to that on V_1, C is a compact operator.

Relevance to ML and Gaussian Processes II

Corollary

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

$$K(\mathbf{x},\mathbf{y}) = L_2 \sum_{i=1}^{\infty} \lambda_i e_i(\mathbf{x}) e_i(\mathbf{y})$$

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 $\lambda_i \geq 0$.

Relevance to ML and Gaussian Processes III

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

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

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

Posterior Prediction In the Mercer Basis I

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!):

$$\begin{aligned}\mathbb{E}[\mathbf{y}_* \mid \mathbf{X}_*, \mathbf{y}, \mathbf{X}] &= K(\mathbf{X}_*, \mathbf{X})\left[K(\mathbf{X},\mathbf{X}) + \sigma^2\mathbf{I}\right]^{-1}\mathbf{y} \\ &= \sum_k \frac{\lambda_k}{\lambda_k + \sigma^2}\mathbf{P}_k(\mathbf{X}_*, \mathbf{X})\,\mathbf{y}\end{aligned}$$

Posterior Prediction In the Mercer Basis II

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

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

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 $K\left(\mathbf{X}_*, \mathbf{X}\right)$ matrix and each eigenspace contribution of this matrix would be a restriction of (63) to the data.

6 Topic: EM Algorithm in GMMs

To estimate the GMM, we will want to maximize the log-likelihood function

$$\log p(\mathbf{x} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \right\} \tag{7}$$

First-order condition for $\boldsymbol{\mu}_k$:

Taking the partial derivative of Equation (7) with respect to $\boldsymbol{\mu}_k$, we obtain the first order condition

$$0 = \sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_j \pi_j \mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}}_{\gamma(z_{nk})} \boldsymbol{\Sigma}_k^{-1}\left(\mathbf{x}_n - \boldsymbol{\mu}_k\right) \quad \Longrightarrow \quad \boldsymbol{\Sigma}_k^{-1} \cdot \left(\sum_{n=1}^{N} \gamma\left(z_{nk}\right) \mathbf{x}_n\right) = \boldsymbol{\Sigma}_k^{-1} \cdot \left(\sum_{n=1}^{N} \gamma\left(z_{nk}\right)\right) \boldsymbol{\mu}_k \quad \Longrightarrow \quad \boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma\left(z_{nk}\right) \mathbf{x}_n$$

where $N_k := \sum_{n=1}^{N} \gamma\left(z_{nk}\right)$.

Interpretation:

- N_k is the "effective number of points" assigned to the k-th cluster; and
- The mean $\boldsymbol{\mu}_k$ of the k-th Gaussian component is a weighted mean of all of the points in the data set for which the weighting factor for the data point, $\mathbf{x}_n$, is given by the posterior probability $\gamma\left(z_{nk}\right)$ that component k was responsible for generating $\mathbf{x}_n$.

First-order condition for $\boldsymbol{\Sigma}_k$:

Taking the partial derivative of Equation (7) with respect to $\boldsymbol{\Sigma}_k$, it is easy to see that we obtain the first order condition

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma\left(z_{nk}\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_k\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_k\right)^{\top}$$

Note that this equation is of the same form as that of a single Gaussian trained on the data set, but with (a) each data point weighted by the corresponding posterior probability and (b) the denominator given by the effective number of points associated with the corresponding component.

First-order condition for π_k:

Let us maximize the log likelihood function with respect to π_k, subject to the constraint $\sum_{k=1}^{K} \pi_k = 1$. The resulting Lagrangian takes the form

$$L := \log p(\mathbf{x} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

where λ is a Lagrange multiplier. Taking the partial derivative of (14) with respect to π_k, we obtain the first order condition

$$0 = \frac{\partial L}{\partial \pi_k} = \lambda + \sum_{n=1}^{N} \frac{\mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_j \pi_j \mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)} = \lambda + \frac{1}{\pi_k} \sum_{n=1}^{N} \gamma\left(z_{nk}\right) \tag{8}$$

Multiply both sides of Equation (8) by π_k and then summing over k, we obtain

$$0 = \sum_{k=1}^{K} \left(\lambda \pi_k + \pi_k \cdot \frac{1}{\pi_k} \sum_{n=1}^{N} \gamma\left(z_{nk}\right) \right) = \lambda \left(\sum_{k=1}^{K} \pi_k \right) + \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma\left(z_{nk}\right) \xrightarrow[\sum_{k=1}^{K} z_{nk}=1]{\sum_{k=1}^{K} \pi_k = 1} \lambda + \sum_{n=1}^{N} 1 \quad \Longrightarrow \quad \lambda = -N$$

Inserting $\lambda = -N$ into Equation (8), we conclude that

$$\pi_k = \frac{N_k}{N}$$

Interpretation: The mixing coefficient for the k-th component is given by the average of the responsibilities which that component takes for explaining the data points.

In all, we have the first order necessary conditions are

$$\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{n=1}^{N}\gamma\left(z_{nk}\right)\mathbf{x}_n, \quad \boldsymbol{\Sigma}_k = \frac{1}{N_k}\sum_{n=1}^{N}\gamma\left(z_{nk}\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_k\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_k\right)^\top, \quad \pi_k = \frac{N_k}{N},$$

We note that this is not a closed-form solution for the parameters of the GMM. (Why?)

However, we can try to solve the problem using an iterative scheme as follows. This scheme is a version of the Expectation-Maximization (EM) algorithm applied to the GMM.

Algorithm 1 EM Algorithm for Gaussian Mixtures

procedure EM-GMMs($\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k, \mathbf{x}_n$)
  $\delta, tol = 1, 0.005$
  $\boldsymbol{\mu}_k \leftarrow \boldsymbol{\mu}_k^0$
  $\boldsymbol{\Sigma}_k \leftarrow \boldsymbol{\Sigma}_k^0$
  $\pi_k \leftarrow \pi_k^0$
  $L \leftarrow \sum_{n=1}^{N}\log\left\{\sum_{k=1}^{K}\pi_k^0\mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k^0, \boldsymbol{\Sigma}_k^0\right)\right\}$
  while $\delta \geq tol$ **do**
    E step: Evaluate the responsibilities using the current parameter values
    $\gamma\left(z_{nk}\right) \leftarrow \frac{\pi_k\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K}\pi_j\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)}$
    M step: Re-estimate the parameters using the current responsibilities
    $N_k \leftarrow \sum_{n=1}^{N}\gamma\left(z_{nk}\right)$
    $\boldsymbol{\mu}_k^{\text{new}} \leftarrow \frac{1}{N_k}\sum_{n=1}^{N}\gamma\left(z_{nk}\right)\mathbf{x}_n$
    $\boldsymbol{\Sigma}_k^{\text{new}} \leftarrow \frac{1}{N_k}\sum_{n=1}^{N}\gamma\left(z_{nk}\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^\top$
    $\pi_k^{\text{new}} \leftarrow \frac{N_k}{N}$
    $L^{\text{new}} \leftarrow \sum_{n=1}^{N}\log\left\{\sum_{k=1}^{K}\pi_k^{\text{new}}\mathcal{N}\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k^{\text{new}}, \boldsymbol{\Sigma}_k^{\text{new}}\right)\right\}$
    $\delta \leftarrow |L^{\text{new}} - L|$
    $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k, L \leftarrow \boldsymbol{\mu}_k^{\text{new}}, \boldsymbol{\Sigma}_k^{\text{new}}, \pi_k^{\text{new}}, L^{\text{new}}$
  return $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$

In general, The log likelihood function of GMMs are not convex, and hence there will be multiple local maxima. While EM is not guaranteed to find the largest of these, it is guaranteed to converge monotonically to a local optima. See Bishop (2006, Section 9.4) for a proof.

EM Algorithm in GMMs v.s. K-Means:

Remark 6.1.

1. In comparison, the EM algorithm for GMMs takes many more iterations to converge than standard K-means clustering. Each iteration of the EM algorithm is also significantly more expensive. To speed up convergence it is common to first perform K-means clustering to find suitable starting points, and then perform the EM algorithm from these.
2. Note that (a) the covariance matrices can be initialized to the sample covariances of the clusters found by K-means, and (b) the mixing coefficients can be set to the fractions of data points assigned to the respective clusters.

△

References

[1] Bishop, Christopher M. (2006). Pattern Recognition And Machine Learning. Springer. URL: https://www.microsoft.com/enus/research/uploads/prod/2006/01/Bishop-PatternRecognition-and-Machine-Learning-2006.pdf.