University of Bonn
Research Module in
Econometrics and Statistics

TITLE OF YOUR PAPER

January 6, 2023

Lindi Li 3460570
Gewei Cao 3461232

Contents

Abstract

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

1 Introduction

1.1 Highlighting

LaTeXallows you to highlight text in various ways: **bold**, *italics*, with SMALL CAPS or `as a coding font`.[1]

1.2 Citing

Citing in LaTeXis easy. You could easier cite with the text flow like this “Referring to.
Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

[1] This command adds a footnote to your text.

2 Reproducing Kernel Hilbert Space

2.1 Hilbert Space

The theory of Hilbert space was initiated by David Hilbert. **?** provides a detailed explanation of Hilbert space $\mathcal{H}$. It is defined as an inner product space that is complete and separable with respect to the norm defined by the inner product. An example of a defined norm in Hilbert space (i.e, the space L_2 of square-integrable functions) can be

$$\parallel f \parallel = \left(\int_a^b f^2(t)dx \right)^{\frac{1}{2}}.$$

A normed space is a vector space N on which a norm is defined. A non-negative function $\parallel \cdot \parallel$ is a norm if and only if $\forall f, g \in N$ and $\alpha \in \mathbb{R}$:

- $\parallel f \parallel \geq 0$ and $\parallel f \parallel = 0$ iff $f = 0$;
- $\parallel f + g \parallel \leq \parallel f \parallel + \parallel g \parallel$;
- $\parallel \alpha f \parallel = \mid \alpha \mid \parallel f \parallel$

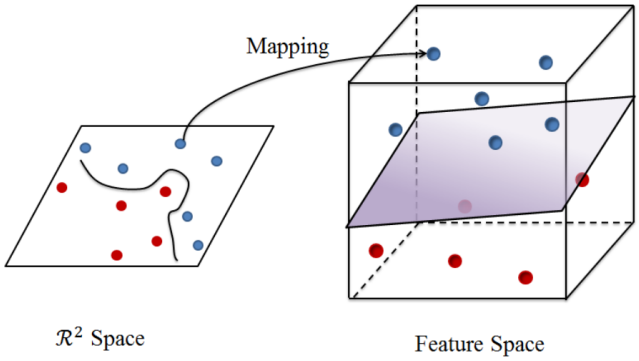
Examples of inner product $< a, b >$ in a Hilbert space are

- a usual dot product: $< a, b > = a'b = \sum_i a_i b_i$.
- a kernel product: $< a, b > = k(a, b) = \psi(a)'\psi(b)$, where $\psi(a)$ may have infinite dimensions.

2.2 Introduction to Kernel

2.2.1 Feature map

The motivation of the kernel method is simple. Imagine there are some blue dots and red dots on a vector space $\mathcal{R}^2$ and we want to separate them by color. As it is shown in the left-hand side figure, it is difficult to divide them through a straight line. However, we may be able to separate them easily by mapping each dot into a high-dimension feature space. The figure below shows how the feature map works:



Let's now use a simple example to illustrate the idea of a feature map. we set two vectors $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} y_1 & y_2 \end{bmatrix}$ in a two-dimension space. Tow functions $\phi(x)$ and $\phi(y)$ are defined as:

$$\phi(x) = \begin{bmatrix} x_1x_1 & x_1x_2 & x_2x_1 & x_2x_2 \end{bmatrix}$$

$$\phi(y) = \begin{bmatrix} y_1y_1 & y_1y_2 & y_2y_1 & y_2y_2 \end{bmatrix}$$

We are now successfully mapping them into a four-dimension feature space. To write the above example in a general form of linear regression, we first set an equation where $\phi(\cdot) \in \mathcal{R}^m$ and $\phi(x)$ is defined as the mapping function. Note that we assume there is a linear relation between y and $\phi(x)$:

$$\begin{aligned} y &= \phi(x)^\top w \\ &= \begin{bmatrix} \phi_1(x) \cdots \phi_m(x) \end{bmatrix} w \end{aligned} \tag{1}$$

Y and Φ in generalization is defined by:

$$Y = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}^\top \tag{2}$$

$$\begin{aligned} \Phi &= \begin{bmatrix} \phi(x_1) & \cdots & \phi(x_n) \end{bmatrix}^\top \\ &= \begin{bmatrix} \phi_1(x_1) & \cdots & \phi_m(x_1) \\ \vdots & \vdots & \vdots \\ \phi_1(x_n) & \cdots & \phi_m(x_n) \end{bmatrix} \end{aligned} \tag{3}$$

Recall the regularized risk minimization problem of ridge regression. In this case, it can be re-written as:

$$\begin{aligned} w* &= \underset{w}{argmin} \sum_{i=1}^{n} (y_i - \phi(x_i)^\top w)^2 + \lambda \parallel w \parallel_2^2 \\ &= \underset{w}{argmin} \parallel Y - \Phi w \parallel_2^2 + \lambda \parallel w \parallel_2^2 \end{aligned}$$

The least-square solution can also be re-defined by:

$$w* = (\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top Y \tag{4}$$

Then we replace $w*$ with $(\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top$ in $y = \phi(x)^\top w$, we get:

$$\begin{aligned} y_w*(x) &= \phi(x)^\top w* \\ &= \phi(x)^\top (\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top Y \\ &= \underbrace{\phi(x)^\top \Phi^\top}_{1 \times n} (\underbrace{\Phi \Phi^\top}_{n \times n} + \lambda I)^{-1} Y \\ &\text{using that} (\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top = \Phi^\top (\Phi \Phi^\top + \lambda I)^{-1} \end{aligned} \tag{5}$$

2.2.2 Kernel Method

In most cases, it is surprisingly difficult to know and calculate the feature function after mapping. We want to avoid computing $\phi(x)$ in an explicit way, especially when m is very large. Therefore, we define a kernel function:

$$[\Phi \Phi^\top]_{i,j} = \phi(x_i)^\top \phi(x_j) = K(x_i, x_j)$$

$$[\phi(x)^\top \Phi^\top]_j = \phi(x)^\top \phi(x_j) = K(x, x_j) \tag{6}$$

This is simply the intuition of using the kernel method. For example, the Gaussian kernel is

$$k(x_i, x_j) = e^{\frac{-\|x_i - x_j\|}{\sigma^2}},$$

Gaussian kernel meaning the similarity between two points where $\parallel x_i - x_j \parallel$ is the Euclidean distance between x_i and x_j, and $\sigma^2 \in \mathbb{R}^+$ is the bandwidth of the kernel function. As shown

in **?**, it has the following properties:
for $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel if

- k is symmetric: $k(x, y) = k(y, x)$.
- k is positive semi-definite, meaning that $\sum_i \sum_j \alpha_i \alpha_j k(x_i, x_j) \geq 0, \forall \alpha_i, \alpha_j \in \mathbb{R}, x \in \mathbb{R}^{\mathbb{D}}, \mathbb{D} \in \mathbb{Z}^+$.
- We define the corresponding kernel matrix as the matrix K with entries $k_{ij} = k(x_i, x_j)$, the sencond property of k is equivalent to saying that $\mathbf{a}'K\mathbf{a} \geq 0$.

Now we can define a function k: $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel if and only if there exists a Hilbert space $\mathcal{H}$ and a map $\phi : \mathcal{X} \to \mathcal{H}$ such that $k(x, y) = \langle \phi(x), \phi(y) \rangle$.
Recall the simple example above, instead of computing the inner product of $\langle \phi(x), \phi(y) \rangle$, we can define a corresponding kernel function $K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle^2$. It can be easily proofed that $K(\mathbf{x}, \mathbf{y})$ is the same as $\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$:

$$\begin{aligned} K(\mathbf{x}, \mathbf{y}) &= < \mathbf{x}, \mathbf{y} >^2 \\ &= (x_1 y_1 + x_2 y_2)^2 \\ &= x_1^2 y_1^2 + 2 x_1 y_2 x_2 y_1 + x_2^2 y_2^2 \\ &= \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle \end{aligned}$$

2.3 Reproducing Kernel Hilbert Space

Consider a Hilbert space $\mathcal{H}$ full of real-valued functions from $\mathcal{X}$ to $\mathbb{R}$, and a mapping $\Phi : \mathcal{X} \to \mathbb{R}^{\mathcal{X}}$ defined as $x \to \Phi(x) = k_x = k(\cdot, x)$. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a reproducing kernel of $\mathcal{H}$, and $\mathcal{H}$ is a reproducing kernel Hilbert space, if:

- $\forall x \in \mathcal{X}$, $k(\cdot, x) \in \mathcal{H}$,
- $\forall x \in \mathcal{X}$, $f \in \mathcal{H}$, $(\cdot), k(\cdot, x) \rangle_{\mathcal{H}} = f(x)$, which is the reproducing property.

To be more intuitive, a Reproducing Kernel Hilbert Space is a Hilbert space $\mathcal{H}$ with a reproducing kernel whose span is dense in $\mathcal{H}$. Equivalently, an RKHS can be defined as a Hilbert space of valid functions with all evaluation functionals bounded and linear.

2.4 Representer Theorem

In the previous section, we have learned that there is always a pair of $(\mathcal{X}, k)$ as a Hilbert space or a subset of that space whenever the input domain $\mathcal{X}$ exists. Such a fact means that we are able to study the various data structures in Hilbert spaces. In the practical world, however, it is extremely difficult to study many popular kernels since their Hilbert spaces are known to be infinite-dimensional in almost every case. Especially for the purpose of machine learning, we usually prefer to solve an optimization problem in a finite-dimensional space.

This is where the representer theorem is useful. It contributes to simplifying the regularized risk-minimization problem by reducing the infinite-dimensional space to a finite-dimensional vector of optimal coefficients and provides provisions for kernels in training data in machine learning.

Suppose we are given a nonempty set $\mathcal{X}$, a positive definite real-valued kernel k on $\mathcal{X} \times \mathcal{X}$, a training sample $(x_1, y_1), \ldots, (x_m, y_m) \in \mathcal{X} \times \mathbb{R}$, a strictly monotonically increasing real-valued function f on $[0, \infty[$. As explained in **?**, we can find the function f^* in the RKHS $\mathcal{H}$ satisfying:

$$\mathcal{J}(f^*) = \underset{f \in \mathcal{H}}{min} \mathcal{J}(f),$$

where

$$\mathcal{J}(f) = L_y(f(x_1), \cdots, f(x_n)) + \Omega(\| f \|_{\mathcal{H}}^2).$$

Note that Ω is a non-decreasing regularizer and y is a vector of y_i.

The representer theorem is that the solution to $\underset{f \in \mathcal{H}}{min}[L_y(f(x_1), \cdots, f(x_n)) + \Omega(\| f \|_{\mathcal{H}}^2)]$ can be written in a simpler version, which takes the form $f^* = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot)$. If Ω is strictly increasing, all solutions apply to this form.

2.5 Example Using Representer Theorem–Kernel Ridge Regression

In the simplest form of machine learning, in order to predict x, the algorithm collects the samples in the training set $\mathcal{X}$ that are similar to x, and then takes the weighted value of these samples as the predict value of x. Here comes the questions:

- How to measure the similarity between samples?
- How to weigh the value of each sample?

In general, the higher the similarity of the sample to our point of interest x, the more the sampling weights. To evaluate the similarity between two observations, a kernel is defined as a function of two input patterns $k(x_i, x_j)$, mapping onto a real-valued output. **?** has proven that the advantage of using such a kernel as a similarity measure is that it allows us to construct algorithms in dot product spaces.

The use of kernels for regression can be described in two stages. We set $y_i \in \mathbb{R}$ as dependent variable, and x_i as a $1 \times D$ vector $x_i \in \mathbb{R}^D$ of covariate values. Assume that (y_i, x_i) where $i = 1, \ldots, N$ is i.i.d. We first define a target function $y = g(x)$ and assume that in a space of functions, there exists a function that can estimate $y = g(x)$ well. The target function can be represented by

$$\begin{aligned} g(\cdot) &= \langle \mathbf{w}, \Phi(x) \rangle \\ &= \sum_{i=1}^{N} w_i k(\cdot, x_i). \end{aligned} \tag{7}$$

We can solve this linear regression problem by least square $\underset{w \in \mathbb{R}^d}{minimize} \frac{1}{n} \sum_{i=1}^{n} (y_i - \langle \phi(x_i), w \rangle)^2$ in the feature space with a mapping $\phi : \mathcal{X} \to \mathbb{R}^d$. Luckily, the representer theorem already tells us that the least squared problem always has a solution of the form $w^* = \sum_{j=1}^{n} \alpha_j \phi(x_j)$. We then plug it into the objective function and use the reproducing property of RKHS, and get:

$$\underset{w \in \mathbb{R}^d}{minimize} \frac{1}{n} \sum_{i=1}^{n} (y_i - \langle \phi(x_i), w \rangle^2) \Leftrightarrow$$

$$\underset{\alpha \in \mathbb{R}^n}{minimize} \frac{1}{n} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{n} \alpha_j \langle \phi(x_i), \phi(x_j) \rangle)^2$$

$$\underset{\alpha \in \mathbb{R}^n}{minimize} \frac{1}{n} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{n} \alpha_j k(x_i, x_j))^2$$

in matrix notation:

$$\underset{\alpha \in \mathbb{R}^n}{minimize} \frac{1}{n} \parallel Y - K\alpha \parallel^2$$

A perfect solution to estimate N parameters for N observations would be choosing $\hat{w} = K^{-1}y$. However, such a fit could result in extremely high variance. Therefore, we impose

an additional assumption that a smoother curve with fewer oscillations is preferred. We then utilize regularization to simplify the function and satisfy the additional assumption by adding a penalty term λ in the second stage.

Use features of $\phi(x_i)$ in the place of x_i, a ridge regression with kernel based can be defined:

$$w^* = \underset{w \in \mathcal{H}}{argmin} \left(\sum_{i=1}^{n} (y_i - \langle w, \phi(x_i) \rangle_{\mathcal{H}})^2 + \lambda \parallel w \parallel_{\mathcal{H}}^2 \right) \tag{8}$$

Remember the corresponding kernel matrix as the matrix K with entries $k_{ij} = k(x_i, x_j)$ is equivalent to saying that $\mathbf{a}'K\mathbf{a} \geq 0$, we can then rewrite the kernel ridge regression as:

$$\sum_{i=1}^{n} (y_i - \langle w, \phi(x_i) \rangle_{\mathcal{H}})^2 + \lambda \parallel w \parallel_{\mathcal{H}}^2$$

$$\Leftrightarrow \parallel y_i - K\alpha \parallel^2 + \lambda \mathbf{a}' K \mathbf{a}. \tag{9}$$

By differentiation and setting the equation (9) to zero, we get:

$$\alpha^* = (K + \lambda I_n)^{-1} y. \tag{10}$$

2.6 Example of Solving the Kernel Function

In this paper, we study in the RKHS $\mathcal{H} = \mathcal{H}_{\omega,\delta}$ consisting of differentiable functions $h : [0, \infty) \to \mathbb{R}$ of the form $h(x) = \int_0^x h'(t)dt$ with continuous derivatives, $h'(x) = h'(0) + \int_0^x h''(t)dt$ for integrable h'', and with finite norm

$$\langle h, h \rangle = \parallel h \parallel_{\omega,\delta} = \left(\int_0^{\infty} (\delta h'(x)^2 + (1-\delta) h''(x)^2) \omega(x) dx \right)^{\frac{1}{2}} \tag{11}$$

for some measurable weight function $\omega : [0, \infty) \to [1, \infty)$ and shape parameter $\delta \in (0, 1)$. With additional assumption in the research paper's appendix A.2, we can extend it to the case $\delta \in \{0, 1\}$.

The Lemma 3 assumes that for any fixed $y \geq 0$, exists a solution ϕ of the linear differential equation

$$\delta \phi \omega - (1-\delta)(\phi' \omega)' = 1_{[0,y]} \tag{12}$$

and for $\psi \in \mathcal{H}_{\omega,\delta}, \psi(x) = \int_0^x \phi(t)dt$, then for $h \in \mathcal{H}_{\omega,\delta}$ with $h'(x) = 0$ for $x > n$ for some finite n, we can write

$$\langle \psi, h \rangle_{\omega,\delta} = \int_0^\infty (\delta \psi'(x)h'(x) + (1-\delta)\psi''h''(x))\omega(x)dx \tag{13}$$

according to the definition for any $h \in \mathcal{H}_{\omega,\delta}$. The assumption of sloution exists and Lemma 4 could give us that $\langle \psi, h \rangle = h(y)$ which implies that $k(\cdot, y) = \psi$ by the reproducing property. Then $k(x, y) = \psi(x)$, and remind that $\psi(x) = \int_0^x \phi(t)dt$, then we can find the form of $k(x, y)$ if we know the form of ϕ, and we could solve ϕ by giving different value of δ and ω. Below is an example of how to solve the research paper's equation 8.
In the research paper, the weight function $\omega(x) = e^{\alpha x}$, if $\alpha = 0, \delta = 1$, then $\phi = 1_{[0,y]}$, and $k(x, y) = \psi(x) = \int_0^x 1_{[0,y]}dt = min\{x, y\}$.

3 Gaussian Process and Bayesian Perspective

After estimating the discount function $g(\boldsymbol{x})$, we want to do inference with this function, and because of the nonparametric estimation, we cannot calculate the statistical distribution of parameters, hence here we use Gaussian Process to estimate the distribution of discount function and construct a confidence interval.

3.1 Definition

We have data $\mathbf{D} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^M$, and assume that mean of y is 0.

Task: find the distribution of $f^*(x)$.

Assume that the true form of prediction function is: $y_i = f(\boldsymbol{x_i}) + \epsilon_i$ and $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$. Here we have an M dimensional dependent variable **y**, and a $M \times N$ dimensional independent variable **X**, where M is the number of observations, and N is the dimension of x, i.e. $\boldsymbol{x_i} \in \mathbb{R}^N$. The function $f(\boldsymbol{x_i}) : \mathbb{R}^N \to \mathbb{R}$ takes vector $\boldsymbol{x_i} \in \mathbb{R}^N$. Let $\boldsymbol{K}_{X,X} = k(\boldsymbol{x}, \boldsymbol{x}^T)$ which is the matrix of $k(\boldsymbol{x_i}, \boldsymbol{x_j})$. Thus, **K** is a $M \times M$ matrix.

The assumption of the Gaussian Process is as follows:

For a given vector **y**, and its corresponding data **X**, where vector $\boldsymbol{y} \in \mathbb{R}^M$ and $\boldsymbol{X}$ is $M \times N$ matrix. In addition, for **y** and **X** data, the error term $\epsilon \sim \mathcal{N}(\mathbf{0}, \Sigma^{\epsilon})$, and $\Sigma^{\epsilon} = diag(\sigma_1^2, \sigma_2^2, \sigma_3^2, ..., \sigma_M^2)$. Meanwhile we have arbitrary $n \times N$ matrix **Z** and predicted value $f^*(\boldsymbol{z}) \in \mathbb{R}^n$, where $\boldsymbol{z} = (\boldsymbol{z_1}, \boldsymbol{z_2}, \boldsymbol{z_3}, \boldsymbol{.....}, \boldsymbol{z_n})^T$.

Then we assume **y** and $f^*(\boldsymbol{z})$ follow a $(M+n)$ multivariate normal distribution(MVN):

$$\begin{bmatrix} f^*(\boldsymbol{z}) \\ \boldsymbol{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_{f^*(\boldsymbol{z})} \\ \mu_{\boldsymbol{y}} \end{bmatrix} , \begin{bmatrix} \boldsymbol{K}_{Z,Z} & \boldsymbol{K}_{Z,X} \\ \boldsymbol{K}_{X,Z} & \hat{\boldsymbol{K}}_{X,X} \end{bmatrix} \right) \tag{14}$$

where $\hat{\boldsymbol{K}}_{X,X} = \boldsymbol{K}_{X,X} + \Sigma^{\epsilon}$.

Then given data **y**, **X** and **Z**, according to the conditional distributions of the multivariate normal distribution[2], we have the posterior distribution

$$f^*(\boldsymbol{z})|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z} \sim \mathcal{N}(\mu_{f^*(\boldsymbol{z})} + \boldsymbol{K}_{Z,X}\hat{\boldsymbol{K}}_{X,X}^{-1}(\boldsymbol{y} - \mu_{\boldsymbol{y}}), \boldsymbol{K}_{Z,Z} - \boldsymbol{K}_{Z,X}\hat{\boldsymbol{K}}_{X,X}^{-1}\boldsymbol{K}_{X,Z}) \tag{15}$$

3.2 Intuition behind Gaussian Process

The idea behind this process is that, assume our interested function is $f(x)$, $f(\boldsymbol{x}) : \mathbb{R}^N \to \mathbb{R}$, and we have an arbitrary vector of independent variable $\boldsymbol{x} = (\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_M})^T$, and for each $\boldsymbol{x_i}, i = 1, 2, ..., M, x_i \in \mathbb{R}^N$, then we can obtain a series of $f(\boldsymbol{x}) = (f(\boldsymbol{x_1}), f(\boldsymbol{x_2}), f(\boldsymbol{x_3}), ..., f(\boldsymbol{x_M}))^T$ We assume that the series of f(**x**) follows a multivariate normal distribution which is:

$$f(\boldsymbol{x}) \sim \mathcal{N}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}^T)) \tag{16}$$

This is the prior distribution of our function $f(x)$, here we have a set of infinite functions that follow this distribution, their mean is the function $\mu(\boldsymbol{x_i})$, and the variance of them is $k(\boldsymbol{x_i}, \boldsymbol{x_i}^T)$. This makes the distribution of $f(\boldsymbol{x})$ to be called Gaussian Process (GP). Note that if we add a noise term $\epsilon \sim \mathcal{N}(\mathbf{0}, \Sigma^{\epsilon})$, then our prior distribution of $y = f(\boldsymbol{x}) + \epsilon \sim \mathcal{N}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}^T) + \Sigma^{\epsilon})$ is also a Gaussian Process. Here we use the kernel matrix to de-

[2]https://statproofbook.github.io/P/mvn-cond

note the variance-covariance matrix because the kernel value represents how near two data points in the space are, with this property we can obtain a smooth function.

Remind that our goal is to estimate the distribution of $f(\boldsymbol{x}^*)$ given observed training data set $D = \{\boldsymbol{x_i}, y_i\}_{i=1}^M$ and test data set $\{\boldsymbol{x_j^*}\}_{j=1}^n$. Firstly we compare our nonparametric case to a parametric case. In a parametric case, assume the parameter θ determines the form of $f_\theta(\cdot)$, according to the Bayesian rule, $p(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{x}, \boldsymbol{y}) = \int_\theta p(\boldsymbol{y}^*, \theta|\boldsymbol{x}^*, \boldsymbol{x}, \boldsymbol{y})d\theta = \int_\theta p(\boldsymbol{y}^*|\theta, \boldsymbol{x}^*)p(\theta|\boldsymbol{x}, \boldsymbol{y})d\theta$, where $\boldsymbol{y}^*$ is the prediction of given data $\boldsymbol{x}^*$, and its form of model is determmined by paramater θ. The estimated θ value is determined by training data D. This is to say that we update our parameter θ by given D, and use $p(\theta|\boldsymbol{x}, \boldsymbol{y})$ as a new prior probability, and based on this to predict posterior of $\boldsymbol{y}^*$.

Therefore, back to our GP nonparametric case, θ could be substituted by function $f(\cdot)$. One can show that the joint distribution of $(f(\boldsymbol{x}^*), \boldsymbol{y})^T$ follows a multivariate normal distribution as in the definition before, because of the assumption of GP and the property of MVN. With the joint distribution, we want to find posterior probability: $p(f(\boldsymbol{x}^*)|\boldsymbol{x}^*, \boldsymbol{x}, \boldsymbol{y}) = \int p(f(\boldsymbol{x}^*)|f, \boldsymbol{x}^*)p(f|\boldsymbol{x}, \boldsymbol{y})df$, where $p(f|\boldsymbol{x}, \boldsymbol{y})$ is the posterior of $f(\cdot)$ given D, and is regarded as prior when estimating $p(f(\boldsymbol{x}^*)|\boldsymbol{x}^*, D)$, this process is called Bayesian updating. Fortunately, we do not need to take any integral in GP, because the posterior of $f(\boldsymbol{x}^*)$ could be calculated by the formula of conditional distribution in MVN as mentioned in the former section.

3.3 Gaussian Process in research paper

In this research paper, authors assumed discount function $g(\boldsymbol{z})$ given a vector of different maturties $\boldsymbol{z} = (z_1, z_2, ..., z_n)$ follows a MVN distribution $\mathcal{N}(m(\boldsymbol{z}), k(\boldsymbol{z}, \boldsymbol{z}^T))$. Then this is a Gaussian Process, and by Bayesian updating for given price P, corresponding cash flow matrix C, and time to maturities x, we can obtain the posterior mean and variance function in the research paper's equation (12) and equation (13). Therefore, the variance function of MVN could give us the confidence interval of $g(z)$ i.e. for each maturity time z we calculate $k^{post}(z, z)$ as its normal variance, which could help us to evaluate the precision of our prediction. Furthermore, with the posterior distribution of $g(\boldsymbol{x})$, it is implied that the coupon

bond price $Cg(\boldsymbol{x}) \sim \mathcal{N}(Cm^{post}(\boldsymbol{x}), Ck^{post}(\boldsymbol{x}, \boldsymbol{x}^T)C^T)$.

Note that authors assume the variance covariance matrix of error term $\Sigma^{\epsilon} = diag(\sigma_1^2, \sigma_2^2, \sigma_3^2, ..., \sigma_M^2)$ where diagnoal elements all satisify $\omega_i = \frac{\lambda}{\sigma_i^2}$, this implies that we give a higher weight for a bond price which has less noise. In addition, we assume that the prior mean function is constant $m(x) = 1$ which assumes no time value of money. With these assumptions, the posterior mean function coincides with estimated $\hat{g}(\boldsymbol{x})$ as in the research paper's equation (5).

4 Research Design

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5 Conclusion

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Appendix

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.