Assignment 1 - MVP's

Lindi Li
3460570

Gewei Cao
3461232

December 2022

§ 1 Intrudction

blabla

§ 2 Reproducing Kernel Hilbert Space

§ 2.1 Hilbert Space

Recall that an inner product $< a, b >$ can be

- 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.

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

- $\| f \| \geq 0$ and $\| f \| = 0$ iff $f = 0$;

- $\| f + g \| \leq \| f \| + \| g \|$;

- $\| \alpha f \| = | \alpha | \| f \|$

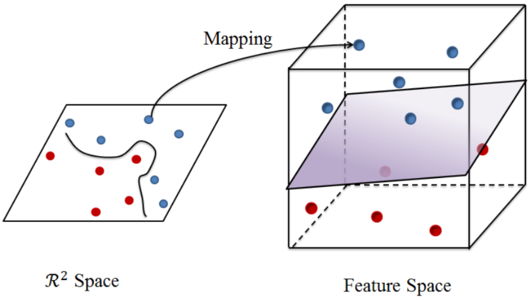
Now we can define a Hilbert space $\mathcal{H}$ an inner product space that is complete and separable with respect to the norm defined by inner product. An example of a defined norm in Hilbert space (i.e, the space L_2 of square integrable functions) can be

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

§ 2.2 Introduction to Kernel

2.2.1 Feature map

The motivation of kernel method is simple. Imagine there are some blue dots and red dots on a vector space $\mathbb{R}^n$ and we want to separate them by color. As it shows 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 feature map works:



Let's now use a simple example to illustrate the idea of 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, 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 $\phi(x)$:

$$\begin{aligned} y &= \phi(x)^\top w \\ &= \begin{bmatrix} \phi_1(x) \\ \vdots \\ \phi_m(x) \end{bmatrix} w \\ &= \begin{bmatrix} \phi_1(x) \cdots \phi_m(x) \end{bmatrix} w \end{aligned} \tag{2}$$

Y and Φ in generalization is defined by:

$$Y = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}^\top \tag{3}$$

$$\begin{aligned} \Phi &= \begin{bmatrix} \phi(x_1) & \cdots & \phi(x_n) \end{bmatrix}^\top \\ &= \begin{bmatrix} \phi_1(x_1) & \cdots & \phi_m(x_1) \\ \vdots & \vdots & \vdots \\ \phi_1(x_n) & \cdots & \phi_m(x_n) \end{bmatrix} \end{aligned} \tag{4}$$

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

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

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

$$w* = (\Phi^\top \Phi + \lambda I)^(-1)\Phi^\top Y \tag{5}$$

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

$$\begin{aligned} y_w * (x) &= \phi(x)^\top w* \\ &= \phi(x)^\top (\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top Y \\ &= \underbrace{\phi(x)^\top \Phi^\top}_{1 \times n} (\underbrace{\Phi\Phi^\top}_{n \times n} + \lambda I)^{-1} Y \end{aligned} \tag{6}$$

$$\text{using that}(\Phi^\top\Phi + \lambda I)^{-1}\Phi^\top = \Phi^\top(\Phi\Phi^\top + \lambda I)^{-1}$$

2.2.2 Kernel Method

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

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

$$[\phi(x)^\top\Phi^\top]_j = \phi(x)^\top\phi(x_j) = K(x, x_j) \tag{7}$$

This is simply the intuition of using the kernel method. 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 $k(x, y) =< \phi(x), \phi(y) >$. For example, the Gaussian kernel is

$$k(x_i, x_j) = e^{\frac{-\|x_i - x_j\|}{\sigma^2}}, \tag{8}$$

Gaussian kernel meaning the similarity between two points where $\| x_i - x_j \|$ is the Euclidean distance between x_i and x_j, and $\sigma^2 \in \mathcal{R}^+$ is the bandwidth of the kernel function, and it satisfies the following properties:
for $k : \chi \times \chi \to \mathcal{R}$ 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 $c'Kc \geq 0$.

Recall the simple example above, instead of computing the inner product of $< \phi(x), \phi(y) >$, we can define a corresponding kernel function $K(\mathbf{x}, \mathbf{y}) =< \mathbf{x}, \mathbf{y} >^2$. It can be easily proofed that $K(\mathbf{x}, \mathbf{y})$ is the same as $< \phi(\mathbf{x}), \phi(\mathbf{y}) >$:

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

§ 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:

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

To be more intuitively, a Reproducing Kernel Hilbert Space is a Hilbert space $\mathcal{H}$ with a reproducing kernel whose span is dense in $\mathcal{H}$. Equivalently, a 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 a practical word, however, it is extremely difficult to study many popular kernels since their Hilbert spaces is known to be infinite-dimensional in almost every case. Especially for the purpose of machine learning, we usually prefer solve an optimization problem in a finite-dimensinal space.

This is where the representer therom useful. It contributes to simplify the regularized risk-minimization problem by reducing the infinite-dimensional space to three-dimensional vector of optimal coefficients, and provide provisions for kernels in training data in machine learning.

To illustrate the representer theroem, we first set a paired observations $(x_1, y_1), \cdots, (x_n, y_n)$ to be either classification or regression.

- Classification: $L_y(f(x_1), \cdots, f(x_n)) = \sum_{i=1}^{n} PPPP_{y_i f(x_i) \leq 0}$
- Regression: $L_y(f(x_1), \cdots, f(x_n)) = \sum_{i=1}^{n} (y_i - f(x_i))^2$

In the RKHS $\mathcal{H}$, we can find the function f^* satisfying:

$$\mathcal{J}(f^*) = \underset{f \in \mathcal{H}}{min} \mathcal{J}(f), \tag{9}$$

where

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

Note that Ω is non-decreasing 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 theroem–Kernel ridge regression

In the simplest form of machine learning, in order to predict x, the algorithm collects the samples in the training set χ that are similar to x, and then take 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 weight 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.

From the similarity-based point of view, 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 $y = f(x)$ and assume that in a space of functions, there exists a function that can estimate $y = f(x)$ well. The target function could be represented by

$$f(x) = \sum_{i=1}^{N} w_i k(x, x_i), \tag{10}$$

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 addtional assumption that a smoother curve with less oscillations is prefered. We then utilize regularization to simplify the function and satisfy the additional assumption 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:

$$f^* = \underset{f \in \mathcal{H}}{argmin} \left(\sum_{i=1}^{n} (y_i - < f, \phi(x_i) >_{\mathcal{H}})^2 + \lambda \parallel f \parallel_{\mathcal{H}}^2 \right) \tag{11}$$

According to representer theorem, the solution to such a optimization problem if we know that f is a linear combination of feature space mappings of points: $f = \sum_{i=1}^{n} w_i \phi(x_i)$. We can then rewrite the kernel ridge regression as:

$$\sum_{i=1}^{n} (y_i - < f, \phi(x_i) >_{\mathcal{H}})^2 + \lambda \parallel f \parallel_{\mathcal{H}}^2 \tag{12}$$

$$= \parallel y_i - Kw \parallel^2 + \lambda w^\top K w. \tag{13}$$

By differentiation and setting the above equation to zero, we get:

$$w^* = (K + \lambda I_n)^{-1} y. \tag{14}$$

§ 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 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 a M dimensional dependent variable $\mathbf{y}$, and a $M \times N$ dimensional independent variable $\mathbf{X}$, 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, $\mathbf{K}$ is a $M \times M$ matrix.

The assumption of Gaussian Process is as following:

For a given vector $\mathbf{y}$, and its corresponding data $\mathbf{X}$, where vector $\boldsymbol{y} \in \mathbb{R}^M$ and $\boldsymbol{X}$ is $M \times N$ matrix. In addition, for $\mathbf{y}$ and $\mathbf{X}$ 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 $\mathbf{Z}$ and predicted value $f^*(\boldsymbol{z}) \in \mathbb{R}^n$, where $\boldsymbol{z} = (\boldsymbol{z_1}, \boldsymbol{z_2}, \boldsymbol{z_3}, ..., \boldsymbol{z_n})^T$.

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

$$\begin{bmatrix} f^*(\boldsymbol{z}) \\ \boldsymbol{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_{f^*(\boldsymbol{z})} \\ \mu_{\boldsymbol{y}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}_{Z,Z} & \boldsymbol{K}_{Z,X} \\ \boldsymbol{K}_{X,Z} & \hat{\boldsymbol{K}}_{X,X} \end{bmatrix}\right) \tag{15}$$

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

Then given data $\mathbf{y}$, $\mathbf{X}$ and $\mathbf{Z}$, according to the conditional distributions of the multivariate normal distribution[1], we have the posterior distribution

$$f^*(\boldsymbol{z})|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z} \sim \mathcal{N}(\mu_{f^*(\boldsymbol{z})} + \boldsymbol{K}_{Z,X}\hat{\boldsymbol{K}}_{X,X}^{-1}(\boldsymbol{y} - \mu_{\boldsymbol{y}}), \boldsymbol{K}_{Z,Z} - \boldsymbol{K}_{Z,X}\hat{\boldsymbol{K}}_{X,X}^{-1}\boldsymbol{K}_{X,Z}) \tag{16}$$

§ 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 arbitorary 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($\mathbf{x}$) follows a multivariate normal distribution which is:

$$f(\boldsymbol{x}) \sim \mathcal{N}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}^T)) \tag{17}$$

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

This is the prior distribution of our function $f(x)$, here we have a set of infinitely 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 $\epsilon \sim \mathcal{N}(\boldsymbol{0}, \Sigma^{\epsilon})$, 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 kernel matrix to denote variance-covariance matrix because 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. 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 formula of conditional distributioin in MVN as mentioned in 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 maturties x, we can obtain the posterior mean and variance function in 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, that 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 research paper's equation (5).

§ 4 Empirical study

blabla