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 $\alpha \in \mathbb{R}$:

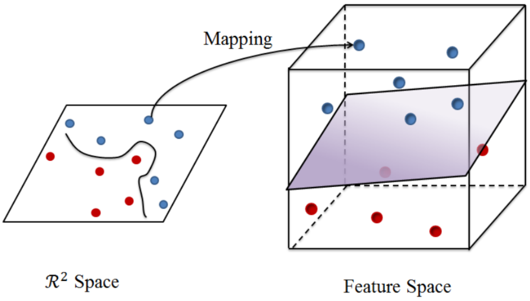
- $\| f \| \geq 0$ and $\| f \| = 0$ iff $f = 0$;
- $\| f + g \| \leq \| f \| + \| g \|$;
- $\| \alpha f \| = | \alpha | \| f \|$

Now we can define a Hilbert space 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 Kernel Method

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 mapping works:



Let's now use a simple example to illustrate the idea of kernel method. 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:

$$\phi(x) = \begin{bmatrix} x_1 x_1 & x_1 x_2 & x_2 x_1 & x_2 x_2 \end{bmatrix}$$

$$\phi(y) = \begin{bmatrix} y_1 y_1 & y_1 y_2 & y_2 y_1 & y_2 y_2 \end{bmatrix}$$

We now successfully mapping them into a four-dimension feature space. Therefore, 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 $<\phi(\mathbf{x}), \phi(\mathbf{y})>$:

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

To write the above example in a general form, we first set an equation where $\phi(\cdot) \in \mathbb{R}^m$ and $\phi(x)$ is defined as the mapping function:

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

Y and Φ in generalization is defined by:

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

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

In most of the cases, we want to avoid computing $\phi(x)$ in a explicit way, especially when m is very large. Therefore, we define a kernel function:

$$[\Phi\Phi^(top)]_{i,j} = \phi(x_i)^(top)\phi(x_j) = K(x_i, x_j) \tag{5}$$

$\phi(x_i)^t op\phi(x_j)$ is usually very difficult to compute, and thus we use $\mathbb{K}(x_i, x_j)$ to replace. 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. For example, the Gaussian kernel is

$$k(x_i, x_j) = e^{\frac{\|x_i - x_j\|}{\sigma^2}}, \tag{6}$$

where $\| x_i - x_j \|$ is the Euclidean distance between x_i and x_j, and $\sigma^2 \in \mathbb{R}^+$ is the bandwidth of the kernel function.
We now define that $k : \chi \times \chi \rightarrow \mathbb{R}$ is a kernel if

- k is symmetric: $k(x, y) = k(y, x)$.
- k is positive semi-definite, meaning that $\sum_i \sum_j \alpha_i \alpha_j k(x_i, x_j) \geq 0, \forall \alpha_i, \alpha_j \in \mathbb{R}, x \in \mathbb{R}^{\mathbb{D}}, D \in \mathbb{Z}^+$.
- 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 $c'Kc \geq 0$.

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

$$f(x) = \sum_{i=1}^{N} w_i k(x, x_i), \tag{7}$$

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. To achieve this purpose, Hilbert space and reproducing kernel Hilbert space will be introduced below.

§ 3 Gaussian Process and Bayesian Perspective

§ 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):

$$\begin{bmatrix} f^*(\boldsymbol{z}) \\ \boldsymbol{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_{f^*(\boldsymbol{z})} \\ \mu_{\boldsymbol{y}} \end{bmatrix} , \begin{bmatrix} \boldsymbol{K}_{Z,Z} & \boldsymbol{K}_{Z,X} \\ \boldsymbol{K}_{X,Z} & \hat{\boldsymbol{K}}_{X,X} \end{bmatrix} \right) \tag{8}$$

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

$$f^*(\boldsymbol{z})|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z} \sim \mathcal{N}(\mu_{f^*(\boldsymbol{z})} + \boldsymbol{K}_{Z,X}\hat{\boldsymbol{K}}_{X,X}^{-1}(\boldsymbol{y} - \mu_{\boldsymbol{y}}), \boldsymbol{K}_{Z,Z} - \boldsymbol{K}_{Z,X}\hat{\boldsymbol{K}}_{X,X}^{-1}\boldsymbol{K}_{X,Z}) \tag{9}$$

§ 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:

$$f(\boldsymbol{x}) \sim \mathcal{N}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}^T)) \tag{10}$$

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 $\epsilon \sim \mathcal{N}(\mathbf{0}, \Sigma^\epsilon)$, then our prior distribution of

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

$y = f(\boldsymbol{x}) + \epsilon \sim \mathcal{N}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}^T) + \Sigma^{\epsilon})$ 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, becaus 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 our object paper, for given price data P,

§ 4 Empirical study

blabla