Gaussian Process 02 Weight Space

Chen Gong

14 December 2019

Gaussian Process 在这里我们主要讲解的是 Gaussian Process Regression。我们从 Bayesian Linear Regression 的角度来引出的 Gaussian Progress Regression。

1 Recall Bayesian Linear Regression

首先，我们需要回顾一下 Bayesian Linear Regression。

1. 首先对于一个贝叶斯线性回归来说，参数符合的分布 $p(w|Data) = \mathcal{N}(w|\mu_w, \Sigma_w)$。其中，$\mu_w = \sigma^{-2}A^{-1}X^TY$，$\Sigma_w = A^{-1}$，其中，$A = \sigma^{-2}X^TX + \Sigma_p^{-1}$。从这一步我们就成功的得到了在已知 Data 的情况下，未知参数的分布形式。

2. 在给定一个新的未知数向量 X^* 的情况下，我们可以首先利用 noise-free 形式：$f(x) = w^Tx = x^Tw$，然后再求得 noise 形式：$y = f(x) + \epsilon$，而 $\epsilon \sim \mathcal{N}(0, \sigma^2)$。来获得我们想要的 prediction 值。这样，我们就可以得到：

$$p(f(x^*)|Data, x^*) \sim \mathcal{N}(x^{*T}\mu_w, x^{*T}\Sigma_w x^*) \tag{1}$$

$$p(y^*|Data, x^*) \sim \mathcal{N}(x^{*T}\mu_w, x^{*T}\Sigma_w x^* + \sigma^2) \tag{2}$$

但是，问题马上就来了，因为很多时候，我们不能仅仅使用线性分类的方法来解决问题。现实生活中有许多非线性的问题来待我们求解。而一种经常使用的方法，也就是将数据投影到高维空间中来将非线性问题，转换成一个高维空间中的线性可分问题。或者是使用 Bayesian Logistics Regression 来进行分类。如果，是将数据投影到高维空间中的话，我们很自然的就想到了 Kernel Bayesian Linear Regression。

那么这个非线性转换可以被我们写成：If $\phi : x \mapsto z$，$x \in \mathbb{R}^p$，$x \in \mathbb{R}^p$，$z \in \mathbb{R}^q$，$z = \phi(x)$。

2 非线性转换后的表达

数据集被我们描述为：$X = (x_1, x_2, \cdots, x_N)^T, Y = (y_1, y_2, \cdots, y_N)^T$。根据之前我们得到的 Bayesian Linear Regression 结果，我们代入可以得到：

$$p(f(x^*)|X, Y, x^*) \sim \mathcal{N}(x^{*T}(\sigma^2 A^{-1}X^TY), x^{*T}A^{-1}x^*) \tag{3}$$

而其中，$A = \sigma^{-2}X^TX + \Sigma_p^{-1}$，If $\phi : x \mapsto z$，$x \in \mathbb{R}^p$，$z \in \mathbb{R}^q$，$z = \phi(x)(q > p)$。这里的 ϕ 是一个非线性转换。我们定义：$\Phi = (\phi(x_1), \phi(x_2), \cdots, \phi(x_N))^T_{N \times q}$。

转换之后为：$f(x) = \phi(x)^T w$。那么，

$$p(f(x^*)|X, Y, x^*) \sim \mathcal{N}(\sigma^{-2}\phi(x^*)^T(A^{-1}\Phi(X)^T Y), \phi(x^*)^T A^{-1}\phi(x^*)) \tag{4}$$

而其中，$A = \sigma^{-2}\Phi(X)^T\Phi(X) + \Sigma_p^{-1}$。但是，很快我们又将面临一个新的问题，也就是 A^{-1} 应该如何计算呢？这里我们需要使用到一个公式为，Woodbury Formula 公式（在矩阵含有一定结构时加速矩阵求逆）：

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1} \tag{5}$$

也可以通过如下变换的形式计算 A^{-1}：

$$\begin{aligned} A &= \sigma^{-2}\Phi(X)^T\Phi(X) + \Sigma_p^{-1} \\ A\Sigma_p &= \sigma^{-2}\Phi(X)^T\Phi(X)\Sigma_p + I \\ A\Sigma_p\Phi(X)^T &= \sigma^{-2}\Phi(X)^T\Phi(X)\Sigma_p\Phi(X)^T + \Phi(X)^T = \sigma^{-2}\Phi(X)^T(K + \sigma^2 I) \\ \sigma^{-2}A^{-1}\Phi(X)^T &= \Sigma_p\Phi(X)^T(K + \sigma^2 I)^{-1} \end{aligned} \tag{6}$$

然后，两边同乘一个 $\phi(x^*)^T$ 和 Y 就可以得到：

$$\sigma^{-2}\phi(x^*)^T A^{-1}\Phi(X)^T Y = \phi(x^*)^T\Sigma_p\Phi(X)^T(K + \sigma^2 I)^{-1}Y \tag{7}$$

而这个 $\sigma^{-2}\phi(x^*)^T A^{-1}\Phi(X)^T Y$ 正好就是 $p(f(x^*)|X, Y, x^*)$ 的期望（通过贝叶斯回归类比出来的正态分布中的期望）。而这里的 $\Sigma_p = p(w)$ 是一个先验 $\sim \mathcal{N}(0, \Sigma_p)$，而 σ^2 为先验分布的噪声，x^* 是一个 new input，而$K = \Phi\Sigma_p\Phi^T$。所以，使用类似的方法我们可以得到，$p(f(x^*)|X, Y, x^*)$'s Covarience 为：$\phi(x^*)^T\Sigma_p\phi(x^*) - \phi(x^*)^T\Sigma_p\Phi(X)^T(K + \sigma^2 I)^{-1}\Phi(X)\Sigma_p\phi(x^*)$。所以：

$$\begin{aligned} p(f(x^*)|X, Y, x^*) \sim &\mathcal{N}(\phi(x^*)\Sigma_p\Phi(X)^T(K + \sigma^2 I)^{-1}Y, \\ &\phi(x^*)^T\Sigma_p\phi(x^*) - \phi(x^*)^T\Sigma_p\Phi(X)^T(K + \sigma^2 I)^{-1}\Phi(X)\Sigma_p\phi(x^*)) \end{aligned} \tag{8}$$

而大家注意观察一下，下面几个等式：

$$\phi(x^*)^T\Sigma_p\Phi^T \qquad \phi(x^*)^T\Sigma_p\phi(x^*) \qquad \Phi\Sigma_p\Phi^T \qquad \Phi\Sigma_p\phi(x^*) \tag{9}$$

这里的 $\Phi_{N\times q}$ 表示的是经过变换之后的数据矩阵：

$$\Phi_{N\times q} = (\phi(x_1), \phi(x_2), \cdots, \phi(x_N))^T_{N\times q} \tag{10}$$

所以大家想一想就知道了，公式 (9) 中的四个公式实际上是一个东西，而 $\Phi(X)$ 只不过是将多个向量拼接在了一起而已。而 $K(x, x') = \phi(x)^T\Sigma_p\phi(x')$，$x, x'$ 是两个不一样的样本，矩阵展开以后，形式都是一样的。那么下一个问题就是 $K(x, x')$ 是否可以表达为一个 Kernel Function 的形式？那么，相关的探究就变得有趣了。

3 Kernel Trick

因为 Σ_p 是一个 positive define matrix，并且它也是 symmetry 的。所以，令 $\Sigma_p = (\Sigma_p^{\frac{1}{2}})^2$。那么，我们可以做如下的推导：

$$\begin{aligned} K(x, x') =& \phi(x)^T \Sigma_p^{\frac{1}{2}} \Sigma_p^{\frac{1}{2}} \phi(x') \\ =& (\Sigma_p^{\frac{1}{2}} \phi(x))^T \cdot \Sigma_p^{\frac{1}{2}} \phi(x) \\ =& \varphi(x)^T \varphi(x') \end{aligned} \tag{11}$$

其中，$\varphi(x) = \Sigma_p^{\frac{1}{2}} \phi(x)$。那么，我们利用 Kernel Trick 可以有效的避免求 $\phi(X)$，而是直接通过 $K(x, x')$ 中包含的高维空间的转化。而Bayesian Linear Regression + Kernel Trick 中就蕴含了一个 Non-Linear Transformation inner product。我们就可以将这个转换定义到一个核空间中，避免了直接来求这个复杂的转化。这也就是 Kernel Trick。

Gaussian Process Regression ，可以从两种视角去解释，而这两种视角可以得到 equal result：

1. Weight-Space view，也就是我们这一小节所讲的东西。指的就是那两个等式，$f(x) = \phi(x)^T w$ 和 $y = f(x) + \epsilon$。在这里我们的研究对象就是 w，假设 w 的先验，需要求得 w 的后验，所以是从 Weight-Space 的角度分析的。

2. Function-Space view，我们将 $f(x)$ 看成是一个随机变量，$f(x) \sim GP(m(x), K(x, x'))$。这个我们会在后面的小节中进行详细的描述，大家就可以看到 GP 的思想在其中的运用了。

而有一句话对 GPR 的总结，非常的有意思，Gaussian Process Regress is the extension of Bayesian Linear Regression with kernel trick. 仔细想一想就知道了，我们把逻辑思路理一下，我们想用贝叶斯练习回归来解决非线性的问题，所以我们需要把输入空间投射到一个高维空间中，低维空间中的线性不可分问题将可以转化为高维空间中的线性可分问题。那么，我们就需要一个转换函数来完成这个工作，但是这个转换函数怎么求？有可能会很难求，而且维度很高。那么，我们就不求了，直接使用核技巧，也就是两个向量的内积等于一个核函数的值就可以了。这大概就是本节中 Weight-Space View 的一个主线的思路。