Gaussian Process

Chunpai Wang

March 08, 2017

There are two equivalent view of Gaussian Process Regression, weight space view and function-space view. From weight-space view, Gaussian Process Regression is just kernelized Bayesian regression. Unlike classical learning problem, Bayesian algorithms do not attempt to identify "best fit" models of the data. Instead, they compute a posterior distribution over models. From function-space view, one can think of a Gaussian process as defining a distribution over functions, and inference taking place directly in the space of functions.

In this note, it will be helpful to review the **Schur complement, matrix inversion lemma, and some properties of multivariate Gaussian**, since we will use those to derive the kernel trick on Gaussian Process Regression. Section 1 and 2 cover the weight-space view.

1 Bayesian Linear Regression

Let $S = \{(\mathbf{x}^{(1)}, y^{(1)}), \cdots, (\mathbf{x}^{(n)}, y^{(n)})$ be a training set of i.i.d. example from some unknown distribution. The standard probabilistic interpretation of linear regression states that

$$y^{(i)} = \theta^\top \mathbf{x}^{(i)} + \epsilon^{(i)}, \qquad i = 1, ..., n \tag{1}$$

where $\epsilon^{(i)}$ are i.i.d. "noise" variables with independent $\mathcal{N}(0, \sigma^2)$ distributions. It follows that $y^{(i)} - \theta^\top x^{(i)} \sim \mathcal{N}(0, \sigma^2)$, or equivalently,

$$P(y^{(i)}|\mathbf{x}^{(i)}, \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^\top \mathbf{x}^{(i)})^2}{2\sigma^2}\right) \tag{2}$$

and

$$P(\mathbf{y}|X, \theta) = \prod_{i=1}^{n} P(y^{(i)}|\mathbf{x}^{(i)}, \theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^\top \mathbf{x}^{(i)})^2}{2\sigma^2}\right) \tag{3}$$

$$= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}|\mathbf{y} - X^\top \mathbf{w}|^2\right) \tag{4}$$

$$= \mathcal{N}(X^\top \mathbf{w}, \sigma^2 I) \tag{5}$$

Now, we denote

$$X = \begin{bmatrix} -(\mathbf{x}^{(1)})^\top - \\ -(\mathbf{x}^{(2)})^\top - \\ \vdots \\ -(\mathbf{x}^{(m)})^\top - \end{bmatrix} \in \mathbb{R}^{m \times n} \qquad \mathbf{y} = \begin{bmatrix} (y^{(1)}) \\ (y^{(2)}) \\ \vdots \\ (y^{(m)}) \end{bmatrix} \in \mathbb{R}^m \qquad \epsilon = \begin{bmatrix} \epsilon^{(1)} \\ \epsilon^{(2)} \\ \vdots \\ \epsilon^{(m)} \end{bmatrix} \tag{6}$$

In Bayesian linear regression, we assume that a prior distribution over parameters is provided;

$$\theta \sim \mathcal{N}(0, \Sigma_p) \tag{7}$$

a typical choice, for instance, is $\theta \in \mathcal{N}(0, \tau^2 I)$. Using Bayes's rule, we obtain the parameter posterior.

$$p(\theta|X, \mathbf{y}) = \frac{p(\mathbf{y}|X, \theta)p(\theta)}{p(\mathbf{y}|X)} = \frac{p(\mathbf{y}|X, \theta)p(\theta)}{\int_{\theta'} p(\mathbf{y}|X, \theta')p(\theta')d\theta'} = \frac{p(\theta)\prod_{i=1}^{m} p(y^{(i)}|x^{(i)}, \theta)}{\int_{\theta'} p(\theta')\prod_{i=1}^{m} p(y^{(i)}|x^{(i)}, \theta')d\theta'} \tag{8}$$

Since only the marginal likelihood does not involve the parameter θ, we have

$$p(\theta|X, \mathbf{y}) \propto \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y} - X^\top \mathbf{w})^\top (\mathbf{y} - X^\top \mathbf{w})\right) \cdot \exp\left(-\frac{1}{2}\theta^\top \Sigma_p^{-1} \theta\right) \tag{9}$$

Then, we can use the "completing the square" to obtain

$$p(\theta|X, \mathbf{y}) \propto \exp\left(-\frac{1}{2}(\theta - \bar{\theta})^\top \left(\frac{1}{\sigma^2} X X^\top + \Sigma_p^{-1}\right)(\theta - \bar{\theta})\right) \tag{10}$$

where $\bar{\theta} = \sigma^{-2}(\sigma^{2-}XX^\top + \Sigma_p^{-1})^{-1}X\mathbf{y}$, and we can

$$p(\theta|X,\mathbf{y}) \sim \mathcal{N}(\bar{\theta} = \frac{1}{\sigma^2}A^{-1}X\mathbf{y}, A^{-1}) \tag{11}$$

where $A = \sigma^{-2}XX^\top + \Sigma_p^{-1}$.

Assuming the same noise model on testing points as on our training points, the "output" of Bayesian linear regression on a new test point $\hat{\mathbf{x}}$ is not just a single guess $\hat{y}$, but rather an entire probability distribution over possible outputs, known as the **posterior predictive distribution**

$$p(\hat{y}|\hat{\mathbf{x}}, X, \mathbf{y}) = \int_\theta p(\hat{y}|\hat{\mathbf{x}}, \theta)p(\theta|X,\mathbf{y})d\theta \tag{12}$$

The RHS means it averages over all possible parameter values, weighted by their posterior probability. This is contrast to non-Bayesian schemes, where a single parameter is typically chosen by some criterion.

For many types of models, the integrals are difficult to compute, and hence, we often resort to approximation, such as MAP estimation. However, in the Bayesian linear regression, the integrals actually are tractable due to some properties of multivariate Gaussians. In particular, for Bayesian linear regression, one can show that

$$\theta|S \sim \mathcal{N}\left(\frac{1}{\sigma^2}A^{-1}X^\top\mathbf{y}, A^{-1}\right) \tag{13}$$

and

$$\hat{y}|\hat{\mathbf{x}}, X, \mathbf{y} \sim \mathcal{N}\left(\frac{1}{\sigma^2}\hat{\mathbf{x}}^\top A^{-1}X^\top\mathbf{y}, \hat{\mathbf{x}}A^{-1}\hat{\mathbf{x}}\right) \tag{14}$$

where $A = \frac{1}{\sigma^2}X^\top X + \Sigma_p^{-1}$.

The posterior distribution over the test output $\hat{y}$ for a test input $\hat{\mathbf{x}}$ is a Gaussian distribution, this distribution reflects the uncertainty in our predictions $\hat{y} = \theta^\top\hat{\mathbf{x}} + \hat{\epsilon}$, arising from both the randomness in ϵ and the uncertainty in our choice of parameters θ. In contrast, classical probabilistic linear regression models estimate parameter θ directly from the training data but provide no estimate of how reliable these learned parameters may be.

2 Weight-Space view of Gaussian Process Regression

We can use a set of basis functions to project the inputs into some higher dimensional space, and then apply the linear model in that space instead of directly on the input themselves. For now, we assume that the basis functions are given.

Denote by $\phi(\mathbf{x})$ the mapping from D-dimension to N dimensional feature space, $\Phi(\mathbf{x})$ be the aggregation of columns $\phi(\mathbf{x})$ for all cases in the training set. Now the model is

$$\mathbf{y} = \phi(\mathbf{x})^\top\theta \tag{15}$$

where the vector of parameters now have length N. Now the predictive distribution becomes

$$p(\hat{\mathbf{y}}|\hat{\mathbf{x}}, X, \mathbf{y}) \sim \mathcal{N}\left(\frac{1}{\sigma^2}\phi(\mathbf{x})^\top A^{-1}\Phi(X)\mathbf{y}, \phi(\hat{\mathbf{x}})^\top A^{-1}\phi(\hat{\mathbf{x}})\right) \tag{16}$$

where $A = \sigma^{-2}\Phi(X)\Phi(X)^\top + \Sigma_p^{-1}$.

To make predictions using this equation we need to invert the matrix A of size $N \times N$ which may not be convenient if N is large. However, if let $K = \Phi(X)^\top\Sigma_p\Phi(X)$, note that

$$\begin{aligned}
\frac{1}{\sigma^2}A^{-1}\Phi(X) &= \frac{1}{\sigma^2}A^{-1}\Phi(X)(K+\sigma^2 I)(K+\sigma^2 I)^{-1} && (17)\\
&= A^{-1}\left[\sigma^{-2}\Phi(X)(K+\sigma^2 I)\right](K+\sigma^2 I)^{-1} && (18)\\
&= A^{-1}\left[\sigma^{-2}\Phi(X)(\Phi(X)^\top\Sigma_p\Phi(X)+\sigma^2 I)\right](K+\sigma^2 I)^{-1} && (19)\\
&= A^{-1}\left[\sigma^{-2}\Phi(X)\Phi(X)^\top\Sigma_p\Phi(X)+\Phi(X)\right](K+\sigma^2 I)^{-1} && (20)\\
&= A^{-1}\left[\sigma^{-2}\Phi(X)\Phi(X)^\top\Sigma_p\Phi(X)+\Sigma_p^{-1}\Sigma_p\Phi(X)\right](K+\sigma^2 I)^{-1} && (21)\\
&= A^{-1}\left[\left(\sigma^{-2}\Phi(X)\Phi(X)^\top+\Sigma_p^{-1}\right)\Sigma_p\Phi(X)\right](K+\sigma^2 I)^{-1} && (22)\\
&= A^{-1}\left[A\Sigma_p\Phi(X)\right](K+\sigma^2 I)^{-1} && (23)\\
&= \Sigma_p\Phi(X)(K+\sigma^2 I)^{-1} && (24)
\end{aligned}$$

That is

$$\frac{1}{\sigma^2}\phi(\mathbf{x})^\top A^{-1}\Phi(X)\mathbf{y} = \phi(\hat{\mathbf{x}})\Sigma_p\Phi(K+\sigma^2 I)^{-1}\mathbf{y} \tag{25}$$

We can also use the matrix inversion lemma to rewrite the variance and we can get

$$p(\hat{\mathbf{y}}|\hat{\mathbf{x}}, X, \mathbf{y}) \sim \mathcal{N}\left(\phi(\hat{\mathbf{x}})\Sigma_p\Phi(K+\sigma^2 I)^{-1}\mathbf{y}, \phi(\hat{\mathbf{x}})\Sigma_p\phi(\hat{\mathbf{x}}) - \phi(\hat{\mathbf{x}})^\top\Sigma_p\Phi(K+\sigma^2 I)^{-1}\Phi(X)^\top\Sigma_p\phi(\hat{\mathbf{x}})\right) \tag{26}$$

3 Covariance Matrix, Covariance Function, and Kernel Function

3.1 Kernelized Ridge Regression

Now assume we have training dataset (X, Y)

$$Y = \beta^\top X + \epsilon \tag{27}$$

where $\beta \in \mathbb{R}^p, X \in \mathbb{R}^{p\times n}, \epsilon \in \mathbb{R}^{1\times n}$, that is n observations and r dimensions. The optimal β is

$$\hat{\beta} = (XX^\top)^{-1}XY^\top \tag{28}$$

Now the kernelized ridge regression can be formulated as

$$\begin{aligned} &\frac{1}{c}\|\beta\|_2^2 + \sum_i \xi_i^2 \\ s.t. \quad &y_i = \beta^\top x_i + \xi_i \quad \forall i = 1, ..., n \end{aligned} \tag{29}$$

The Lagrangian is

$$L(\beta, \xi, \alpha) = \frac{1}{c}\|\beta\|_2^2 + \sum_i \xi_i^2 + \sum_i \alpha_i(y_i - \beta^\top x_i - \xi_i) \tag{30}$$

Now, we can show that our predict value on query instance of $f(x)$ as $\hat{y} = \hat{\beta}^\top x$, which can be also expressed as

$$\hat{y} = b^\top X^\top x \tag{31}$$

where X is the observation, and x is the query or testing instance. We can leverage the SVD to decompose the observation matrix X

$$\begin{aligned} \hat{\beta} &= (XX^\top)^{-1}XY^\top && (32)\\ &= (U\Lambda V^\top V\Lambda U^\top)^{-1}XY^\top && (33)\\ &= (U\Lambda^2 U^\top)^{-1}XY^\top && (34)\\ &= U\underbrace{\Lambda^{-2}U^\top XY^\top}_{\in\mathbb{R}^{r\times 1}} \quad \text{since } U \text{ is orthonormal, that is } U = U^{-1} = U^\top && (35)\\ &= U \cdot a && (36)\\ &= X \cdot b && (37)\\ &= U\underbrace{SV^\top b}_{=a} && (38) \end{aligned}$$

Once we can express $\hat{y} = b^\top X^\top x$, we can see that that there are n inner product between column of X and query instance x, which can use the kernel trick. In addition, the mean square error can be expressed as

$$\begin{aligned} MSE &= (Y - \beta^\top X)(Y - \beta^\top X)^\top && (39)\\ &= (Y - b^\top X^\top X)(Y - b^\top X^\top X)^\top && (40)\\ & && (41) \end{aligned}$$

Taking the derivative and setting it to zero:

$$\frac{\partial MSE}{\partial b} = -2X^\top XY^\top + 2X^\top XX^\top Xb = 0 \tag{42}$$

We can replace the inner product $X^\top X$ as the kernel matrix K, then we have

$$KY^\top = KKb \tag{43}$$

Then we can get the optimal solution

$$\hat{b} = K^{-1}Y^\top + \hat{b_0} \tag{44}$$

where b_0 lies in null space of KX. Now we can make a prediction in the dual using the kernel trick

$$\hat{y} = \hat{y}(x) = \hat{\beta}^\top x = \hat{b}^\top X^\top x = Y^\top K^{-1}X^\top x = Y^\top K^{-1}k \tag{45}$$

where $k = X^\top x$.

3.2 Kernel and Convolution

4 Function-space View of Gaussian Process

Definition 1. A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

Gaussian processes are the extension of the multivariate Gaussians to infinite-sized collections of real-valued variables. A Gaussian process is completed specified by its mean function and covariance function. We define mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ of a real process $f(\mathbf{x})$ as

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})] \tag{46}$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \tag{47}$$

and we will write the Gaussian process as

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{48}$$