CS281B/Stat241B: Advanced Topics in Learning & Decision Making

Gaussian Processes and Reproducing Kernels

Lecturer: Michael I. Jordan *Scribe: Mark A. Paskin*

1 Gaussian Processes and RKHS

Let $Z(x)$ be a Gaussian Process and consider the function

$$Y(x) = Z(x) + \epsilon$$

where ϵ is a noise term with zero mean. Define $f(x)$ as follows:

$$f(x) \quad \stackrel{\triangle}{=} \quad E\left[Z(x) \,|\, y_1, y_2, \ldots y_N\right]$$

where $\{(x_i, y_i = Y(x_i))\}_{i=1}^N$ is a training data set. Then from last lecture, we know

$$f(x) \quad = \quad k^T K^{-1} y$$

where $K_{ij} = EY(x_i)Y(x_j)$ is the covariance between training inputs y_i and y_j and $k_i = EY(x)Y(x_i)$ is the covariance between the training input y_i and the new point, $Y(x)$.

In general, we can specify the Gaussian Process by choosing a covariance function $k(x, x')$, so $k(x_i, x_j) \stackrel{\triangle}{=} K_{ij}$ and $k(x, x_i) \stackrel{\triangle}{=} k_i$. If we do this, then we can write

$$\begin{aligned} f(x) &= k^T K^{-1} y \\ &= \sum_i \beta_i k(x, x_i) \end{aligned}$$

where $\beta = K^{-1}y$. This should look familiar: $f(x)$ is in an RKHS with reproducing kernel $k(x, x')$.

Exercises.

1. Prove that $k(x, x')$ is positive semidefinite.
2. Prove that $k(x, x')$ is a reproducing kernel for an appropriate choice of inner product.

2 Gaussian Processes and Regularization

Recall that the log posterior of a Gaussian Process is given by

$$J = \underbrace{\frac{\beta}{2}(y - z)^T(y - z)}_{\text{likelihood}} + \underbrace{\frac{1}{2} z^T K^{-1} z}_{\text{prior}}$$

where

$$z = \begin{bmatrix} Z(x_1) \\ Z(x_2) \\ \vdots \\ Z(x_N) \end{bmatrix} \text{ and } y = \begin{bmatrix} Y(x_1) \\ Y(x_2) \\ \vdots \\ Y(x_N) \end{bmatrix}$$

We know that the optimal solution z must lie in the span of kernels evaluated at the data points:

$$z = K\alpha$$

That is,

$$z(x_i) = \sum_j \alpha_j k(x_i, x_j)$$

This implies that

$$\begin{aligned} J &= \frac{\beta}{2}(y - K\alpha)^T(y - K\alpha) + \frac{1}{2}\alpha^T K K^{-1} K\alpha \\ &= \underbrace{\frac{\beta}{2}(y - K\alpha)^T(y - K\alpha)}_{\text{loss function}} + \underbrace{\frac{1}{2}\alpha^T K\alpha}_{||z||_{\mathcal{H}}, \text{ the RKHS norm of } z} \end{aligned}$$

Thus, the log posterior has the same form as a RKHS-norm penalized regularization problem. Thus, by choosing a covariance kernel $k(x, x')$, we are implementing an RKHS-based regularization procedure.

3 Gaussian Processes and Bayesian Linear Regression

We can also invert this line of argument—every Gaussian process has an interpretation as a Bayesian linear regression. Let $k(x, x')$ be a covariance function. By Mercer's Theorem, we can write

$$k(x, x') = \sum_i \lambda_i \phi_i(x)\phi_i(x')$$

where $\{\phi_i(\cdot)\}$ are the eigenfunctions of $k(\cdot, \cdot)$.

Recall Bayesian Linear Regression:

$$y(x) = \theta^T \phi(x) + \epsilon$$

where our prior specifies that $\theta \sim N(0, \Sigma)$ and $\epsilon \sim N(0, R)$.

Now consider a particular choice of Σ as the infinite diagonal matrix $\Sigma = \text{diag}(\lambda_1, \lambda_2, \ldots)$, and $\phi(x)$ as the infinite vector $\phi(x) = (\phi_1(x), \phi_2(x), \ldots)^T$.

The Bayesian Linear Regression framework implies that

$$\text{Cov}(y(x), y(x')) = \phi(x)^T \Sigma \phi(x') + R$$

With our particular choices, we see that this becomes the eigenfunction decomposition of our kernel function $k(\cdot, \cdot)$. Thus, this special version of Bayesian Linear Regression is equivalent to the Gaussian Process formulation. Unlike the GP formulation, however, the BLR formulation is intractable computationally, since it requires inverting infinite matrices.

4 Choosing Kernels

Now that we can interpret kernel functions as covariance functions for Gaussian Processes, we have a better intuition for designing them. Below are some examples of kernel functions that can be understood from this viewpoint:

- **Gaussian RBF Kernels.** Captures the notion that "close is relevant":

$$k(x, x') = \theta_0 + \theta_1 \exp\left\{-\frac{1}{2}\sum_i \frac{(x_i - x'_i)^2}{\lambda_i}\right\}$$

- **Linear Regression Kernels.** Captures linear trends:

$$k(x, x') = \theta_0 + \theta_1 \sum_i x_i x'_i$$

- **Periodic Kernels.** Captures periodic similarity:

$$k(x, x') = \theta \exp\left\{-\frac{1}{2}\sum_i \frac{\sin \frac{\pi}{\gamma_i}(x_i - x'_i)^2}{\lambda_i}\right\}$$

- **Independent Marginal Variance Kernels.** This kernel has marginal variance which is independent of x:

$$k(x, x') = \theta \prod_i \left(\frac{z\psi_i(x)\psi_i(x')}{\psi_i^2(x) + \psi_i^2(x')}\right) \exp\left\{-\frac{1}{2}\sum_i \frac{(x_i - x'_i)^2}{\psi_i^2(x) + \psi_i^2(x')}\right\}$$

  The product term is required to make this kernel positive semidefinite.

- **Warping Kernels.** If $k(\cdot, \cdot)$ is a kernel, then so is

$$\hat{k}(x, x') = k(u(x), u(x'))$$

  where $u(\cdot)$ is an arbitrary function.

More kernels can be made from these by sums and products (which are still kernels).

The θ in all of these kernels are parameters. Cross-validation can be used to fit them. Alternatively, a maximum-likelihood estimate can be found by optimizing the Gaussian Process likelihood function. One can even be Bayesian and endow θ with hyperparameters, although this is computationally expensive.

The log likelihood of a Gaussian Process is given by

$$\ell = -\frac{1}{2}\log\det K - \frac{1}{2}y^T K^{-1} y + C$$

We can maximize it by gradient ascent:

$$\frac{\partial \ell}{\partial \theta} = -\frac{1}{2}\text{trace}\left(K^{-1}\frac{\partial K}{\partial \theta}\right) + \frac{1}{2}y^T K^{-1}\frac{\partial K}{\partial \theta}K^{-1}y$$

There are some tricks one can employ to avoid inverting K, which is $N \times N$. For example, we can approximate using the "randomized trace trick". If $x \sim N(0, I)$, then

$$E[\text{trace}(x^T A x)] = \text{trace}(AE[xx^T]) = \text{trace}(A)$$

Thus, we can approximate trace(A) by sampling.

5 Equivalent Kernels

Recall (from the first lectures!) that in Kernel Regression, our function estimates took the form

$$\hat{f}(x) = \sum_i y_i k(x, x_i)$$

where $k(x, x')$ was some "local" function that determined the degree of similarity assumed between points x and x' with respect to their function values. In the Gaussian Process framework, our estimates take the form

$$f(x) = k^T K^{-1} y$$

which can be thought of as a sum over the y_i. Let $b(x, x_i) = (K^{-1}k)_i$. Then,

$$\hat{f}(x) = \sum_i y_i b(x, x_i)$$

This is similar to the expression for Kernel Regression. $b(\cdot, \cdot)$ is called the **dual** or **equivalent kernel** for $k(\cdot, \cdot)$. Interestingly, while $k(\cdot, \cdot)$ can be wildly non-local, e.g., the cone-function, its corresponding equivalent kernel is generally a local function; i.e, $b(\cdot, \cdot)$ goes to zero quickly as its arguments diverge.

In fact, the notion of equivalent kernels is general, and can be used to cleanly define the difference between parametric and nonparametric estimators. If the equivalent kernel corresponding to a particular estimator becomes more local as the number of training points increases, then the estimator is non-parametric; if the neighborhood of the equivalent kernel remains constant as the number of training points increases, then it is parametric.