6.435: Bayesian Modeling and Inference **Lecture Date: April 12, 2022**

Gaussian Processes for Machine Learning

Lecturer: Logan Blaine & Teng Gao *Scribe: Rahul Kasar & Ali Ramadhan*

1 Introduction

This lecture is the first on Gaussian processes but continues our discussion of nonparametric Bayesian methods. Compared to other methods discussed previously, the focus of the reading was how to apply Gaussian processes (GPs) to prediction problems and machine learning (ML). The benefit of GPs for ML is that we can consider an infinite set of possible functions to model the data and make predictions. As we observe more data, the GP's posterior predictive distribution gets updated to reflect the new information. The reading also introduces a covariance structure through the GP's kernel that allows us to model the data flexibly using a wide variety of kernels to make predictions. GPs also easily provide estimates of the uncertainties in its predictions.

The reading by Rasmussen and Williams (2005) introduces GPs using two fundamentally equivalent views, the weight space view and the function space view. The weight space view learns parameters for a class of functions and is introduced by showcasing how it can be applied to fit a Bayesian linear model to some data. The function space view imposes a prior on all possible functions via a choice of GP kernel and is showcased by fitting a GP to some data points. The reading shows that the weight space view with a specific basis function is equivalent to the function space view with appropriate kernel.

2 Weight Space View

The goal of weight space view is that we are interested in making inferences on the conditional distribution of the target variable y given the inputs x. There is no attempt made to try to model the true distribution of the inputs. The weight space view has benefits in that it is easily understood by non-practitioners and easy to implement. The reading starts by analyzing a Bayesian linear model. It then shows that we can increase the expressiveness of the Bayesian linear model by projecting the inputs into a higher dimensional feature space and apply the linear model in this higher feature space.

2.1 Bayesian Linear Model

In a Bayesian linear model we model the input/target relationship as

$$y = X^T w + \epsilon \quad \text{where} \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2 I), \quad w \sim \mathcal{N}(0, \Sigma_p) \tag{1}$$

We have set a Gaussian prior on the weight vector with zero mean and covariance matrix Σ_p. The motivation for this prior is that it makes the model analytically tractable. σ_n^2 is the variance of the noise in measurements of y values. The posterior distribution of the weights is given as

$$w|X, y \sim \mathcal{N}(\sigma_n^{-2} A^{-1} X y, A^{-1}) \tag{2}$$

We are ultimately interested in making predictions for new data. To compute the predictive distribution we average over all possible parameters values weighted by their posterior probability (above). We then make predictions for new observations by drawing from this distribution. Figure 2.1 in the reading shows the prior, likelihood, posterior, and predictive distribution generated from the weight space view. The predictive distribution is

$$f_*|x_*, X, y \sim \mathcal{N}\left(\overline{w} x_*^T, x_*^T A^{-1} x_*\right) \tag{3}$$

where $\overline{w} = \sigma_n^2 A^{-1} X y$

As mentioned, a drawback to the linear model is that lacks flexibility to model complicated relationships between the input and target.

2.1.1 Bayesian Linear Model using Basis Functions

An idea to overcome the shortcomings of the rigidness of the linear model to project the inputs into higher dimensional space using basis function ϕ and then performing a Bayesian linear regression on the transformed features. The predictive distribution then becomes a complicated function of ϕ and the user has to make decisions on how to chose $\phi(X)$. Additionally, the predictive distribution as a function of $\phi(X)$ is computationally very difficult as the A matrix requires to be inverted.

The paper introduces kernels to replace basis functions. We can reformulate the predictive distribution using kernels and in the process remove ϕ and A from the distribution. This is useful when it is easier to compute the kernel than computing the transformations of the feature itself. The kernel is also thought of as the covariance function which is explained later on.

3 Function Space View

In the function space view, a GP is used to describe a distribution over functions. Once fit, the GP's posterior provides a distribution over functions that fit the data. The function space view places a prior on all functions allowed by the Gaussian Process kernel. GPs derive their name from the formal definition that a GP is a collection of random variables, any finite number of which have a joint Gaussian distribution.

3.1 Gaussian processes in the function space view

A GP describing a real process $f(x)$ is completely specified by its mean function $m(x)$ and covariance function $k(x, x')$

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \tag{4}$$
$$m(x) = \mathbb{E}[f(x)] \tag{5}$$
$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))] \tag{6}$$

Fitting a GP to data then involves placing a prior on m and k then conditioning the joint Gaussian prior distribution on the observations to obtain a posterior predictive distribution over the functions that describe the data. The reading assumes $m = 0$ for simplicity. The conditioning (or fitting) computationally involves inverting an $n \times n$ square matrix whose where n is the number of training data points. So the computational cost scales the same as matrix inversion, which is nominally $\mathcal{O}(n^3)$ but in practice may be faster due to the endless software and hardware developments made to speed up matrix inversion.

With this definition, it can be shown that the weight space view is equivalent to a function space view defined as a GP

$$f(x) \sim \mathcal{GP}\left(0, \phi^T(x)\Sigma_p\phi(x')\right) \tag{7}$$

For example, the Bayesian Linear model is equivalent to a functional space representation with a linear covariance function.

3.2 Covariance kernel priors for Gaussian processes

Our prior belief about the data can inform the choice of prior covariance kernel $k(x, x')$. For example, the squared exponential kernel represents a prior on all infinitely differentiable functions

$$k(x, x') = \sigma^2 \exp\left\{-\frac{1}{2}\left(\frac{|x_p - x_q|}{\ell}\right)^2\right\} \tag{8}$$

where σ is the scale factor and ℓ is the length scale of the kernel. Kernels can be added and multiplied to create more sophisticated kernels.

Note that we are able to model the covariance between the outputs as a function of the inputs. For the squared exponential function, when inputs are very close in space, the covariance approaches σ^2 and decreases as the distance gets larger. GPs for ML exploit this covariance structure in order to make predictions.

3.3 Making prediction with noise-free observations

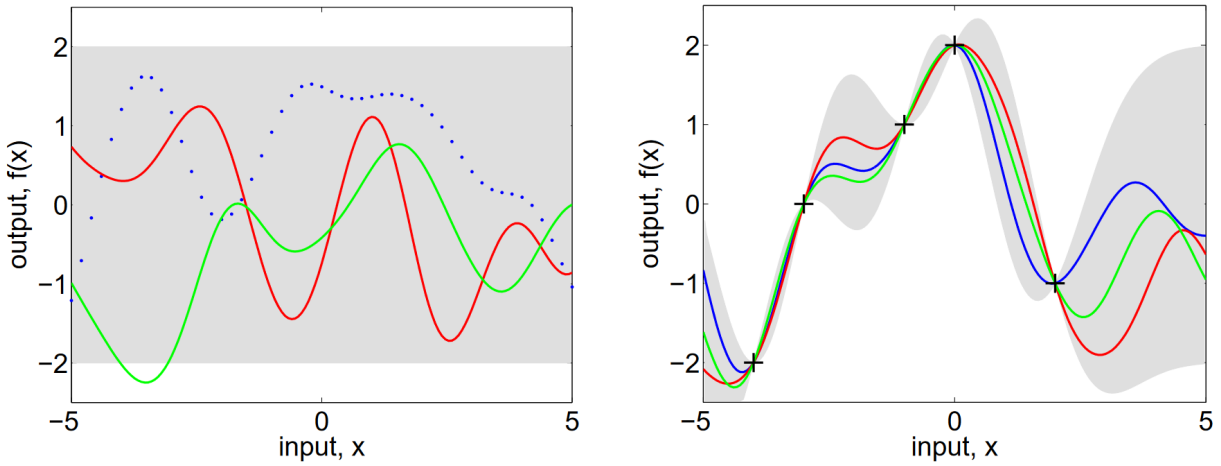
Once a prior kernel has been picked, the GP can be fitted/conditioned on data rather easily. Assuming there is no measurement noise in the data, the joint distribution over the training outputs f and the test outputs f_* according to the prior is

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right) \tag{9}$$

which leads to a noise-free predictive distribution

$$f_*|X_*, X, f \sim \mathcal{N}\left(K(X_*, X)K(X, X)^{-1}f, K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*)\right) \tag{10}$$

Figure 1 shows an example of fitting a GP to some data using the noise-free assumption.



(a), prior (b), posterior

Figure 1: (Left) Three samples from a GP prior using a squared exponential kernel. The red and green curves are plotted to show that samples from a GP are infinite dimensional and infinitely smooth while the green dots show that GPs are made tractable by sampling at a finite number of points (bringing it down to a finite dimensional object). (Right) After conditioning on some data (the black + signs) sampling from the GP posterior produces functions that go through our data (assuming zero measurement noise) and interpolate smoothly in between leading to natural confidence intervals (gray shading). This is taken from figure 2.2 of Rasmussen and Williams (2005).

3.4 Making predictions with noisy observations

When you do not have access to values of $f(x)$ themselves, but only noisy observations of the form $y = f(x)+\varepsilon$ where ε is additive independent identically distributed Gaussian noise with variance σ_n^2, the prior then becomes $\mathrm{Cov}(y_p, y_q) = k(x_p, x_q) + \sigma_n^2 \delta_{pq}$ or $\mathrm{Cov}(y) = K(X, X) + \sigma_n^2 I$ where δ_{pq} is the Kronecker delta function. Conditioning on the data, the joint distribution of the observed target values and the function values at the test locations under the prior becomes

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right) \tag{11}$$

with the noisy predictive distribution $f_*|X, y, X_*$ being given by equations (2.22)–(2.24) of Rasmussen and Williams (2005).

References

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. The MIT Press.