Lecture 17: Modelling: Gaussian Processes and Boosting

In this lecture, we go over a few important modeling approaches to make the discussion complete.

Gaussian Processes (GP)

Let us consider our model to be a function that predicts output value y given a mathematical representation of our input as x:

$$y = f(\mathbf{x}) + \epsilon$$

some examples of this include:

$$\begin{aligned} y &= \mathbf{w}^\top \mathbf{x} + \epsilon && \text{(Linear model)} \\ y &= \mathbf{w}^\top \phi(\mathbf{x}) + \epsilon && \text{(Kernel ridge regression)} \end{aligned}$$

Assuming the noise to be Gaussian $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and each value y_i to be independently distributed, we obtain:

$$\begin{aligned} p(y|\mathbf{x}, \mathbf{w}) &= \prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \mathbf{w}) \\ &= \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - \mathbf{w}^\top \mathbf{x})^2}{2\sigma^2}\right) \\ &= \mathcal{N}(\mathbf{w}^\top X, \sigma^2 \mathbf{I}) \end{aligned}$$

Using a Bayesian approach, we specify a prior over the parameters such as the zero mean and a variance Σ_p i.e. $\mathbf{w} \sim \mathcal{N}(0, \Sigma_p)$. The posterior distribution of the weights given the prior is also a Gaussian because it is a multiplication of two Gaussian distributions:

$$\begin{aligned} p(\mathbf{w}|y, X) &\propto p(y|X, \mathbf{w}) \times p(\mathbf{w}) && \text{(Baye's theorem)} \\ &= \mathcal{N}(\bar{\mathbf{w}}, \Sigma_{\mathbf{w}}) \end{aligned}$$

where the mean $\bar{\mathbf{w}}$ and the covariance $\Sigma_{\mathbf{w}}$ can be computed given the data in X, y. To make predictions, for a test case, we average over all possible parameter values weighted by the probability of the model (a Gaussian posterior).

$$\begin{aligned} p(f_*|x_*, X, y) &= \int p(f_*|x_*, \mathbf{w}) p(\mathbf{w}|X, y) \, \mathrm{d}\mathbf{w} \\ &= \mathcal{N}(\bar{\mathbf{w}}_*, \Sigma_{\mathbf{w}_*}) \end{aligned}$$

Hopefully, this demonstrates the benefits of using a probabilistic model under the Bayesian approach. The idea of **Gaussian processes (GP)** takes this a step further and instead of considering the model for function f to be of a particular form, defines it to be a distribution over the values it can take. A GP is a collection of random variables with any finite number of variables jointly distributed in a Gaussian fashion. For a function f to be a GP means it can be fully specified using a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$ denoted using $f \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$. The kernel matrix allows us to indirectly specify a basis function expansion or feature presentation for x. This can be seen using a similar example as above by considering a class of functions $f = \phi(\mathbf{x})^\top \mathbf{w}$ with a Gaussian prior on the weights $\mathbf{w} \sim \mathcal{N}(0, \Sigma_p)$.

$$\begin{aligned} \mathbb{E}[f(\mathbf{x})] &= \phi(\mathbf{x})^\top \mathbb{E}[\mathbf{w}] = 0 \\ \mathbb{E}[f(\mathbf{x}), f(\mathbf{x}')] &= \phi(\mathbf{x})^\top \mathbb{E}[\mathbf{w}\mathbf{w}^\top] \phi(\mathbf{x}) \end{aligned}$$

which means, by specifying a kernel matrix, we completely determine the underlying basis function representation given by $\phi(\mathbf{x})$ due to Mercer's theorem. The choice of the kernel is important and can usually come from a) our expert knowledge about the input representations, b) model selection by considering the marginal distribution over the GP, and c) using cross-validation approaches. The most common choice for the kernel is a squared exponential kernel that specifies the hypothesis functions to be smooth within a length scale parameter ℓ such that responses of input points within that length scale are very similar.

$$k(\mathbf{x}, \mathbf{x}')) = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^\top M(\mathbf{x} - \mathbf{x}')\right) + \sigma_n^2 \delta_{\mathbf{x},\mathbf{x}'}$$

where $M = \texttt{diag}(\ell)^{-2}$ and σ_f^2, σ_n^2 are the signal and noise variance. For simplicity, assume $m(\mathbf{x}) = 0$ and define our GP as $\mathbf{y} \sim \mathcal{N}(0, \Sigma)$ where $\Sigma_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Using our GP assumption, we know that values from our model are jointly distributed in a Gaussian thus all we need is a mean function and a covariance. We can compute these using the kernel function specified as a prior. The mathematical derivation of how to predict a new test point is quite involved so we ignore the details and instead provide an intuitive explanation using samples from a distribution. Note that our prior is on the output values our hypotheses class of functions can take. We can think of this as a sampling from a box of functions each with equal probability at the beginning as our prior. When we collect data and compute a posterior, we will constrain the set of functions we can sample to a much smaller number thus decreasing the uncertainty around each prediction value. To specify a prior, we need to select a kernel and a mean function. Given this prior, we can draw samples using a multivariate normal distribution and look at what the possible set of functions looks like. This is shown in Figure 1 where the solid lines are different functions possible given the prior. Once we have collected some data from our process, we can update and compute the

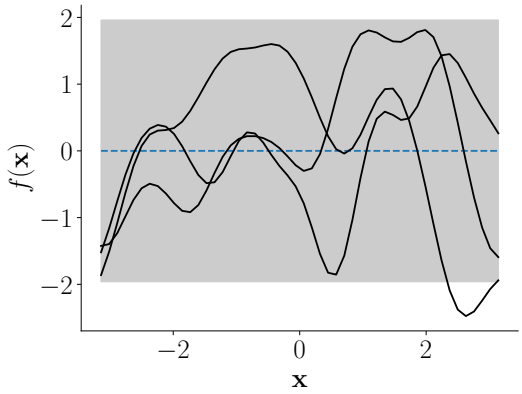

Figure 1: Samples from the GP prior of RBF Kernel and zero mean function

posterior distribution for sampling our new set of functions. These functions are shown in Figure 2 with the points we have collected plotted using the red 'x' mark. Notice that by simply collecting some data and updating our posterior, we have dramatically reduced uncertainty around different functions possible. But at the same time, we can also see how big a role our prior play in our fitting process. Given the smoothness property, we are only able to sample functions that are smooth up to a specific length scale along the x-axis that is optimized to maximize the likelihood of sampling the input data. In other words, if our original function has sharp peaks between the data we have

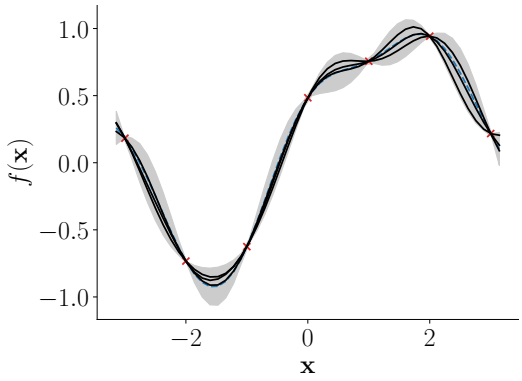

Figure 2: Samples from the GP posterior

trained our GP on, we would do a terrible job using the prior we used. This is one of the main limitations of GPs and is an active research area. Particularly for a physical system, assigning a prior provides us with a great opportunity to encode knowledge that we know about the system and verify if it has been correctly encoded.

Bayesian Additive Regression Trees (BART)

We have seen in GPs that one of the key ingredients for this model to work is the 'Kernel' that defines how two outputs at two different inputs are correlated. Sometimes we may not have explicit knowledge of what this Kernel is for scientific examples or hard to encode into a mathematical representation that a kernel takes. One alternative in such cases is to use a technique called Bayesian additive regression trees which are a collection of binary trees that fit multiple-step functions for predicting output given an input. To understand BART, we first need to understand what is a regression tree model. A regression tree model is simply a set of rules on the input variates that split the space spanned by the variates into different small regions where the output value has low variance. A collection of such regression tree models is called an additive regression tree wherein we sum the output from all the different trees to produce a final output. An regression tree model b number of terminal nodes with associated terminal values given by $\{\mu_1, \mu_2, \ldots, \mu_b\}$. An additive regression tree model will have T such trees each with a set of terminal node values given by $M_j, j \in \{1, 2, 3, \ldots, T\}$. The Bayesian version of the tree is achieved by specifying a regularization prior to the number of trees, the depths of each tree (i.e. how many split points we have), and the noise variance parameter σ. The BART model can be written as follows given a single regression tree as $g(\mathbf{x}, T_i, M_i)$ and the noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$:

$$\mathbb{E}[y|X] = \sum_{i=1}^{T} g(\mathbf{x}, T_i, M_i) + \epsilon$$

The actual algorithm involves fitting T number of trees in parallel for a total of N iterations and perturbing the tree in each iteration such that the prediction improves the residual after each iteration. The prediction of any test point is provided by averaging the sum of the prediction of all the T trees and an uncertainty value is provided by computing the variance among the prediction of different iterations after a specific number of burn-in iterations. We can consider BART to be a prior over step functions much like how a GP is a prior over continuous function and a theoretical result

says that in the limit of $T \to \infty$, a BART is equivalent to a GP of a particular type. Figure 3 shows the application of the BART method to the noisy sine wave data we discussed above with a solid white line representing the mean and the orange bands the uncertainty around each prediction.

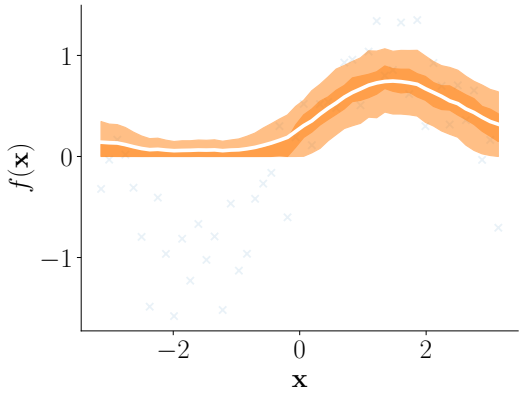

Figure 3: BART prediction and uncertainty of the sine data as implemented in pyMC