Chapter 6, Part 1: Gaussian Processes

Advanced Topics in Statistical Machine Learning

Tom Rainforth

Hilary 2024

rainforth@stats.ox.ac.uk

Parametric Bayesian Modeling

- **Parametric models** have a fixed finite number of parameters θ, regardless of the dataset size
- This can cause a **bottleneck** as all **information** about the data must pass through θ
    - All information about data is stored in the posterior which can only encode a limited amount of information
    - For sufficiently large and rich data, the posterior cannot encapsulate all the information available
    - Posterior predictive: $p(\mathcal{D}^*|\mathcal{D}) = \mathbb{E}_{p(\theta|\mathcal{D})}[p(\mathcal{D}^*|\theta)]$
- With enough data, all posterior mass collapses on a single θ
    - $p(\mathcal{D}^*|\mathcal{D}) \to p(\mathcal{D}^*|\hat{\theta})$ for some $\hat{\theta}$
    - No matter how much data we have, our predictive model will always be limited by the complexity of $p(\mathcal{D}^*|\hat{\theta})$

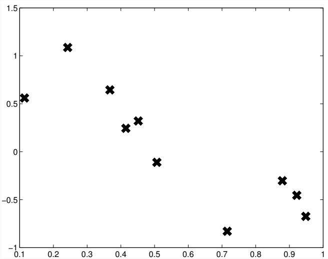
Nonparametric Bayesian Modeling

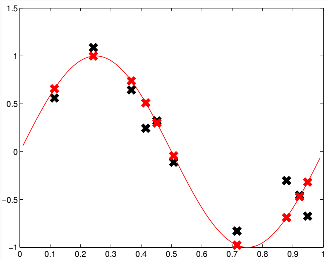
- **Nonparametric models** instead allow the number of “parameters” to grow with the dataset size.
- This allows us to construct models which can become increasingly complex as the amount of data available increases
- Removes bottleneck on predictive power of the model
- Alternative (equivalent) interpretations:
  - Use an infinite dimensional θ where we can analytically marginalize out all but some finite number of dimensions (analogous to kernel methods)
  - Use a predictive distribution that directly depends on the data, i.e. $p(\mathcal{D}^*|\theta, \mathcal{D})$ instead of $p(\mathcal{D}^*|\theta)$
- Downside: difficult to do in a way that maintains self–similarity of Bayes’ rule—limited to particular classes of models that allow analytic marginalizations

Recap: Regression

Assume we are given a supervised dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, $x_i \in \mathcal{X}$, $y_i \in \mathbb{R}$

We can think of regression in terms of reasoning about some "true" underlying function $f(x)$, with the y_i being noisy observations of $f(x_i)$, i.e. $y_i = f(x_i) + \sigma\epsilon_i$ for some constant σ and random "noise variables" ϵ_i





Recap: Kernel Ridge Regression

Kernel ridge regression (KRR) nonparametrically models f by performing regularized empirical risk minimization directly over functions in an RKHS:

$$f^* = \arg\min_{f \in \mathcal{H}_k} \left(\sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}_k}^2 \right)$$

Here the choice of the least squares loss, $(y - f(x))^2$, corresponds to the probabilistic model $\epsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, 1)$, i.e.

$$y | f(x) \sim \mathcal{N}(f(x), \sigma^2).$$

To see this, note that $f^*(x) = \mathbb{E}[Y | X = x]$ both minimizes $\mathbb{E}[(Y - f(X))^2 | X = x]$ and maximizes the likelihood of this probabilistic model.

Being Bayesian about Functions

In a Bayesian nonparametric approach we can informally consider f to be a random variable taking values in $\mathcal{H}_k$, define a corresponding prior, $p(f)$, and apply Bayes' rule to derive a posterior:

$$\begin{aligned} p(f|\mathcal{D}) &= \frac{p(\mathcal{D}|f)p(f)}{p(\mathcal{D})} \\ &\propto p(f)\prod_{i=1}^{n} p(y_i|f(x_i)) \end{aligned}$$

Note here that our likelihood function only requires f to be evaluated at the datapoints.

How Can we Put Priors over Functions?

- Conceptually, we might think of defining a prior over functions by defining a density over the elements of an RKHS
- In practice, it is much easier to just work with **evaluations** of the function: we only need to know the value of a function at points we try to evaluate and thus can directly work with the distribution of these evaluations
  - We typically only need to deal with finite sets of evaluations
  - We can be **lazy** in our evaluation of the function
- We can implicitly define a "distribution" over functions by defining the joint distribution over the evaluations $f(x_1), \dots, f(x_N)$ for all possible set of inputs $x_1, \dots, x_N$ where $N \in \mathbb{N}$, $x_i \in \mathcal{X}$
  - More formally, this defines a **stochastic process**

Gaussian Processes

Definition 1 (Gaussian Processes)

A **Gaussian process** (GP) is a stochastic process whose evaluations are jointly Gaussian. That is, $[f(x_1), \dots, f(x_N)]^T$ has a multivariate Gaussian distribution for all possible $N \in \mathbb{N}$, $x_i \in \mathcal{X}$.

A GP is fully specified by its **mean function** $m : \mathcal{X} \mapsto \mathbb{R}$ and **covariance function** $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, where $m(x) = \mathbb{E}[f(x)]$, $k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$, and

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_N) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_N) \\ \vdots & \ddots & \vdots \\ k(x_N, x_1) & \dots & k(x_N, x_N) \end{bmatrix} \right)$$

for all $N \in \mathbb{N}$, $x_i \in \mathcal{X}$.

Sampling from a Gaussian Process

Given a set of input points we wish to evaluate at $x_1, \ldots, x_N$, we can sample from a Gaussian process by simply calculating the mean vector $\mathbf{m} = [m(x_1), \ldots, m(x_N)]^T$ and covariance matrix $\mathbf{K}$ where $\mathbf{K}_{ij} = k(x_i, x_j)$ and then sampling from $\mathcal{N}(\mathbf{m}, \mathbf{K})$

Right: Samples from GP with $k(x, x') = \exp\left(-(x - x')^2/2\right)$ and $m(x) = 0$. Blue dots are output samples from a finite number of evenly spaced input points, the red and green represent a limit from an infinitesimally dense grid of inputs. Shading is $m(x) \pm 2\sqrt{k(x, x)}$.

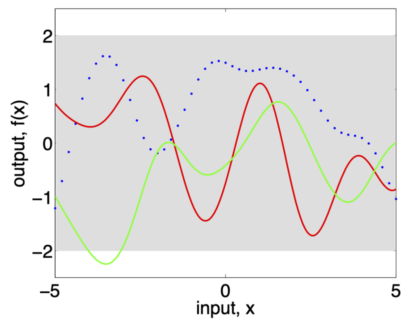


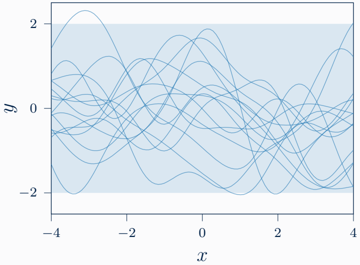
Image source: Carl Edward Rasmussen and Christopher Williams. **Gaussian Processes for Machine Learning**. The MIT Press, 2005

Intuitions

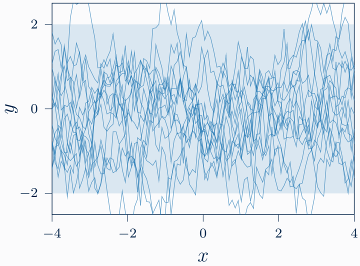
- A GP is essentially a generalization of a Gaussian to infinite dimensions with the input indexing the dimension
- We can informally describe a function as being distributed according to a GP: $f \sim \mathsf{GP}(m, k)$
- We can define an Gaussian process prior by specifying a prior mean function m_{prior} and prior covariance function k_{prior}
- The effect of the mean function is additive by linearity: if $f \sim \mathsf{GP}(m, k)$ then $f - m \sim \mathsf{GP}(0, k)$, thus one typically assumes $m_{\mathsf{prior}}(x) = 0 \, \forall x$ and adjusts the output space accordingly (e.g. regressing to $y - m(x)$)
- To ensure valid covariance matrices, the covariance function must be **positive definite**: it is thus equivalent to the notion of a **kernel** from before

Covariance Functions Specify a RKHS

Choosing a kernel, i.e. covariance function, will induce a prior over functions from a particular RKHS[1]



Squared exponential covariance



Matern 1/2 covariance

Image credit: Gabriele Abbati

[1] Slightly counter intuitively, this RKHS is not identical to that of the kernel itself: though the posterior mean will, as we show later, lie in the RKHS of the kernel, *samples* from the GP actually live in an RKHS that is closely related to, but slightly broader and less smooth than, this RKHS. This subtlety is beyond the scope of the course, but see Section 4 of `https://arxiv.org/abs/1807.02582` if you are interested.

Demo

`http://www.tmpl.fi/gp/`