Gaussian Processes
SC4/SM4 Data Mining and Machine Learning, Hilary Term 2017
Dino Sejdinovic

8.1 Different views of regression

Regression with least squares loss $L(y, f(x)) = (y - f(x))^2$ implies that we are fitting the *conditional mean* function $f^*(x) = \mathbb{E}[Y|X = x]$. This loss also corresponds to the probabilistic model where y_i is a noisy version of the underlying function f evaluated at input x_i:

$$y_i|f(x_i) \sim \mathcal{N}(f(x_i), \sigma^2), \quad \text{independently for} \quad i = 1, \ldots, n. \tag{8.1}$$

There are different ways to model the class of functions f.

- *Frequentist Parametric* approach: model f as f_θ for some parameter vector θ. Fit θ by ML / ERM with squared loss (**linear regression**).
- *Frequentist Nonparametric* approach: model f as the unknown parameter taking values in an infinite-dimensional space of functions (RKHS). Fit f by *regularized* ML / ERM with squared loss (**kernel ridge regression**)
- *Bayesian Parametric* approach: model f as f_θ for some parameter vector θ. Put a prior on θ and compute a posterior $p(\theta|\mathcal{D})$ (**Bayesian linear regression**).
- *Bayesian Nonparametric* approach: treat f as the random variable taking values in an infinite-dimensional space of functions. Put a prior over functions $f \in \mathcal{F}$, and compute a posterior $p(f|\mathcal{D})$ (**Gaussian Process regression**).

8.2 Gaussian Process Regression

Gaussian processes (GPs) are a widely used class of models that allow us to place a prior distribution directly on the space of functions rather than on parameters in a particular family of functions. This prior can then be converted into a posterior distribution once we have seen some data. One can think of a Gaussian process as an infinite-dimensional generalisation of a multivariate normal distribution. Namely, given an *index set* $\mathcal{X}$, a collection of random variables $\{A_x\}_{x \in \mathcal{X}}$ is said to be a Gaussian process if and only if for every finite set of indices $x_1, \ldots, x_n$, vector $[A_{x_1}, \ldots, A_{x_n}]^\top$ has a multivariate normal distribution on $\mathbb{R}^n$. Thus, to any Gaussian process, we can associate a random function $f \colon \mathcal{X} \to \mathbb{R}$ by setting $f(x) = A_x$, for all $x \in \mathcal{X}$. Gaussian process is fully specified by its mean and covariance functions, i.e.

$$\begin{aligned} m(x) &= \mathbb{E}[f(x)], \\ k(x, x') &= \mathbb{E}\left[(f(x) - m(x))\left(f(x') - m(x')\right)\right], \end{aligned}$$

where expectations are taken over f (x and x' are fixed elements in the index set $\mathcal{X}$). This means that for any finite set $x_1, \ldots, x_n$, $\mathbf{f} = [f(x_1), \ldots, f(x_n)]^\top \in \mathbb{R}^n$ has a distribution $\mathcal{N}(\mathbf{m}, \mathbf{K})$, where

$\mathbf{m}_i = m(x_i)$ and $\mathbf{K}$ is the covariance matrix given by $\mathbf{K}_{ij} = k(x_i, x_j)$. We will typically assume that the mean function $m(x)$ is zero under the Gaussian process prior. If we know before seeing any data that the distribution of the function evaluations should be centered around some other mean, we could easily include that into the model. Equivalently, we could also subtract that known mean from the data and just use the zero mean model. If we are looking at the data to estimate the mean function, then often the zero mean GP suffices – in fact, structural information about mean functions (constant, linear) can be included into the choice of the covariance function (*exercises*). Covariance functions $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ obviously has to be positive definite so they are essentially equivalent to the kernel functions we have seen before. In fact, there is a rich connection between RKHS methods and Gaussian processes, an example of which we will discuss below.

8.2.1 Gaussian Conditioning and Regression Model

The convenience of manipulating multivariate normal distributions carries over to Gaussian processes. Let us review the rules for Gaussian conditioning, which are key to Gaussian process regression.

Gaussian Conditioning. Let $\mathbf{z} \sim \mathcal{N}(\mu, \Sigma)$ be a multivariate normal random vector and let us split its dimensions into two parts, i.e.

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}, \quad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}. \tag{8.2}$$

Note that $\Sigma_{21} = \Sigma_{12}^\top$ due to symmetry of covariance matrices. Then the conditional density of $\mathbf{z}_2$ given $\mathbf{z}_1$ is also normal and given by

$$p(\mathbf{z}_2|\mathbf{z}_1) = \mathcal{N}(\mathbf{z}_2;\, \mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(\mathbf{z}_1 - \mu_1),\, \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}). \tag{8.3}$$

For a given set of inputs $\mathbf{x} = \{x_i\}_{i=1}^n$, we denote the vector of evaluations of f by $\mathbf{f} = [f(x_1), \ldots, f(x_n)]^\top \in \mathbb{R}^n$ and the vector of observed outputs by $\mathbf{y} = [y_1, \ldots, y_n]^\top \in \mathbb{R}^n$. Note that since we treat f as a random function, $\mathbf{f}$ is a random n-dimensional vector. The Gaussian process regression model, assuming likelihood function in (8.1), is then given by

$$\begin{aligned} \mathbf{f} &\sim \mathcal{N}(0, \mathbf{K}) \\ \mathbf{y}|\mathbf{f} &\sim \mathcal{N}(\mathbf{f}, \sigma^2 I), \end{aligned}$$

where $\mathbf{K}$ is the covariance (kernel) matrix given by $\mathbf{K}_{ij} = k(x_i, x_j)$. But because both the prior and the likelihood are normal this simply means that $\mathbf{f}$ and $\mathbf{y}$ are *jointly normal* with

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K} \\ \mathbf{K} & \mathbf{K} + \sigma^2 I \end{bmatrix} \right). \tag{8.4}$$

For example, to find the cross-covariance between $\mathbf{f}$ and $\mathbf{y}$ note that

$$\mathbb{E}\left[\mathbf{f}\mathbf{y}^\top\right] = \mathbb{E}\left[\mathbf{f}(\mathbf{f} + \sigma\epsilon)^\top\right] = \mathbb{E}\left[\mathbf{f}\mathbf{f}^\top\right] + \sigma\mathbb{E}\left[\mathbf{f}\epsilon^\top\right] = \mathbf{K}, \tag{8.5}$$

where $\epsilon \sim \mathcal{N}(0, I)$ is independent of $\mathbf{f}$. Now, we can simply apply the Gaussian conditioning to find the posterior distribution

$$\mathbf{f}|\mathbf{y} \sim \mathcal{N}(\mathbf{K}(\mathbf{K} + \sigma^2 I)^{-1}\mathbf{y}, \mathbf{K} - \mathbf{K}(\mathbf{K} + \sigma^2 I)^{-1}\mathbf{K}).$$

This gives as the posterior distribution of the evaluations of the unknown function at the set of inputs where we have observed noisy evaluations $\mathbf{y}$.

8.2.2 Posterior Predictive Distribution

But we can continue with this formalism further and construct the *posterior predictive distribution*. Suppose $\mathbf{x}' = \{x'_j\}_{j=1}^m$ is a test set. We can extend our model to include the function values $\mathbf{f}' = [f(x'_1), \ldots, f(x'_m)]^\top \in \mathbb{R}^m$ at the test set. The prior can now be extended to include $\mathbf{f}'$ (recall that our prior was on the whole function – not on its values at specific locations!), so that the model reads:

$$\begin{bmatrix}\mathbf{f} \\ \mathbf{f}'\end{bmatrix} |\mathbf{x}, \mathbf{x}' \sim \mathcal{N}\left(\begin{bmatrix}0 \\ 0\end{bmatrix}, \begin{bmatrix}\mathbf{K}_{\mathbf{xx}} & \mathbf{K}_{\mathbf{xx}'} \\ \mathbf{K}_{\mathbf{x}'\mathbf{x}} & \mathbf{K}_{\mathbf{x}'\mathbf{x}'}\end{bmatrix}\right)$$
$$\mathbf{y}|\mathbf{f} \sim \mathcal{N}(\mathbf{f}, \sigma^2 I)$$

where $(\mathbf{K}_{\mathbf{xx}})_{ij} = k(x_i, x_j)$, $(\mathbf{K}_{\mathbf{x}'\mathbf{x}'})_{ij} = k(x'_i, x'_j)$, $\mathbf{K}_{\mathbf{xx}'}$ is an $n \times m$ matrix with (i, j)-th entry $k(x_i, x'_j)$ and $\mathbf{K}_{\mathbf{x}'\mathbf{x}} = \mathbf{K}_{\mathbf{xx}'}^\top$ We are now making use of the joint normality of $\mathbf{f}'$ and $\mathbf{y}$:

$$\begin{bmatrix}\mathbf{f}' \\ \mathbf{y}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}0 \\ 0\end{bmatrix}, \begin{bmatrix}\mathbf{K}_{\mathbf{x}'\mathbf{x}'} & \mathbf{K}_{\mathbf{x}'\mathbf{x}} \\ \mathbf{K}_{\mathbf{xx}'} & \mathbf{K}_{\mathbf{xx}} + \sigma^2 I\end{bmatrix}\right) \tag{8.6}$$

and from Gaussian conditioning rules again, we can read off the posterior predictive distribution as

$$\mathbf{f}'|\mathbf{y} \sim \mathcal{N}\left(\mathbf{K}_{\mathbf{x}'\mathbf{x}}(\mathbf{K}_{\mathbf{xx}} + \sigma^2 I)^{-1}\mathbf{y}, \mathbf{K}_{\mathbf{x}'\mathbf{x}'} - \mathbf{K}_{\mathbf{x}'\mathbf{x}}(\mathbf{K}_{\mathbf{xx}} + \sigma^2 I)^{-1}\mathbf{K}_{\mathbf{xx}'}\right). \tag{8.7}$$

Thus, we also have a closed form expression for the posterior distribution of the evaluations of the unknown function at any collection of inputs in $\mathcal{X}$. While this follows directly from the joint normality and Gaussian conditioning rules, it is instructive to notice that we could have arrived at the posterior predictive by integrating $p(\mathbf{f}'|\mathbf{f})$ through the posterior $p(\mathbf{f}|\mathbf{y})$, i.e.

$$p(\mathbf{f}'|\mathbf{y}) = \int p(\mathbf{f}'|\mathbf{f})p(\mathbf{f}|\mathbf{y})d\mathbf{f}. \tag{8.8}$$

This follows from $\int \mathcal{N}(a|Bc, D)\mathcal{N}(c|e, F)dc = \mathcal{N}(a|Be, D + BFB^\top)$ (*exercises*). Namely, even if we no longer have the Gaussian observation model (8.1) and $\mathbf{y}$ and $\mathbf{f}$ are no longer jointly normal, we can still use (8.8) to reason about the posterior predictive distribution.

8.2.3 Kernel Ridge Regression vs Gaussian Process Regression

If kernel ridge regression (KRR) uses the same kernel as the covariance function in Gaussian process regression (GPR) and moreover, if the regularisation parameter λ in KRR is the same as the noise

variance σ^2 in GPR, KRR estimate of the function coincides with the GPR posterior mean. Indeed, recall that in KRR we are solving empirical risk minimisation

$$\min_{f\in\mathcal{H}_k} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \sigma^2 \|f\|^2_{\mathcal{H}_k},$$

and are fitting a function of the form $f(x) = \sum_{i=1}^{n} \alpha_i k(\cdot, x_i)$. Closed form solution is given by $\alpha = \left(\mathbf{K_{xx}} + \sigma^2 I\right)^{-1} \mathbf{y}$. But then if we wish to predict function values at a new set $\mathbf{x}' = \{x'_j\}_{j=1}^m$ of input vectors, we have

$$f(x'_j) = \sum_{i=1}^{n} \alpha_i k\left(x'_j, x_i\right) = \left[k(x'_j, x_1), \ldots, k(x'_j, x_n)\right] (\mathbf{K_{xx}} + \sigma^2 I)^{-1}\mathbf{y},$$

and $\left[k(x'_j, x_1), \ldots, k(x'_j, x_n)\right]$ is the j-th row of $\mathbf{K_{x'x}}$, so this is the same as the mean in (8.7). Note that GPR also gives predictive variance, a measure of uncertainty, which can be important when making predictions far away from the input data. There are other important differences between the two approaches: KRR is frequentist, while GPR is Bayesian, and thus the hyperparameters are fitted in different ways. KRR typically uses cross-validation and grid search, while GPR, as we discuss next, uses maximum marginal likelihood or a fully Bayesian treatment with hyperparameters integrated out.

8.3 Hyperparameter Selection

Probabilistic model given by Gaussian processes allows principled selection of hyperparameters in the model (parameters of the kernel function and the noise variance in the likelihood (8.1)) using *maximum marginal likelihood*.

Marginal likelihood of the hyperparameter vector $\theta = (\nu, \sigma^2)$ which would generally include kernel parameters ν as well as the standard deviation σ^2 of the noise in the observation model, is given by

$$p(\mathbf{y}|\theta) = \int p(\mathbf{y}|\mathbf{f}, \theta) p(\mathbf{f}|\theta) d\mathbf{f} = \mathcal{N}\left(\mathbf{y}; 0, \mathbf{K}_\nu + \sigma^2 I\right).$$

We will introduce the shorthand $\mathbf{K}_{\theta+} = \mathbf{K}_\nu + \sigma^2 I$. Thus, we can write the marginal log-likelihood as

$$\log p(\mathbf{y}|\theta) = -\frac{1}{2}\log|\mathbf{K}_{\theta+}| - \frac{1}{2}\mathbf{y}^\top \mathbf{K}^{-1}_{\theta+}\mathbf{y} - \frac{n}{2}\log(2\pi). \tag{8.9}$$

In general, marginal log-likelihood is a nonconvex function of the parameter vector θ and it can have multiple maxima - thus we typically resort to numerical optimisation methods, such as gradient ascent. The derivative with respect to θ_i (*exercise*) has the form

$$\frac{\partial}{\partial \theta_i} \log p(\mathbf{y}|\theta) = -\frac{1}{2}\mathrm{Tr}\left(\mathbf{K}_{\theta+}^{-1}\frac{\partial \mathbf{K}_{\theta+}}{\partial \theta_i}\right) + \frac{1}{2}\mathbf{y}^\top \mathbf{K}_{\theta+}^{-1}\frac{\partial \mathbf{K}_{\theta+}}{\partial \theta_i}\mathbf{K}_{\theta+}^{-1}\mathbf{y}. \tag{8.10}$$

Some common kernel choices in this context involve *Automatic Relevance Determination (ARD)* kernel

$$k(x, x') = \tau^2 \exp\left(-\sum_{j=1}^{p} \frac{(x^{(j)} - x'^{(j)})^2}{\eta_j^2}\right), \tag{8.11}$$

which has a global scale parameter τ as well as one *bandwidth* parameter η_j per covariate dimension j. If in the hyperparameter selection, very large values of η_j are selected, this essentially means that the dimension j is switched off (does not contribute to the kernel function). This is very useful in applications where it is likely that not all dimensions will be relevant.

In addition to maximum marginal likelihood, we can also perform full Bayesian inference for hyperparameters. Namely, we could start with a prior $p(\theta)$ on θ and draw samples from the posterior

$$p(\theta|\mathbf{y}) \quad \propto \quad p(\theta)p(\mathbf{y}|\theta) = p(\theta)\int p(\mathbf{y}|\mathbf{f}, \theta)p(\mathbf{f}|\theta)d\mathbf{f}.$$

This means that we can integrate uncertainty over hyperparameters into predictions as well, and approximate (integral is typically not available in closed form)

$$p(\mathbf{f}'|\mathbf{y}) = \int p(\mathbf{f}'|\mathbf{y}, \theta)p(\theta|\mathbf{y})d\theta.$$

8.4 Gaussian Processes for Classification

In Bayesian classification problems, we are interested in modelling the posterior probabilities of the categorical response variable given a set of training examples and a new input vector. These probabilities must lie in the interval $(0, 1)$ while a Gaussian process models functions that have output on the entire real axis. Thus, it is necessary to adapt Gaussian processes by transforming their outputs using an appropriate nonlinear activation function. Consider the binary classification model with classes -1 and $+1$, using the logistic sigmoid:

$$p(y_i = +1|f(x_i)) = \sigma(f(x_i)) = \frac{1}{1 + e^{-f(x_i)}}. \tag{8.12}$$

This non-Gaussian form of the likelihood function, however, renders exact posterior inference intractable and approximate methods are needed. There are a number of approximate schemes that can be used but we will focus here on Laplace approximation. We know that

$$\begin{aligned} \log p(\mathbf{f}|\mathbf{y}) &= \text{const} + \log p(\mathbf{f}) + \log p(\mathbf{y}|\mathbf{f}) \\ &= \text{const} - \frac{1}{2}\mathbf{f}^\top \mathbf{K}^{-1}\mathbf{f} + \sum_{i=1}^{n} \log \sigma(y_i f(x_i)). \end{aligned}$$

Thus, we can compute the gradient

$$\frac{\partial \log p(\mathbf{f}|\mathbf{y})}{\partial \mathbf{f}} = -\mathbf{K}^{-1}\mathbf{f} + \mathbf{q}_\mathbf{f},$$

where $\mathbf{q}_\mathbf{f} = [q_1, \ldots, q_n]^\top$ with $q_i = \sigma(-y_i f(x_i))y_i$. The Hessian is given by

$$\frac{\partial^2 \log p(\mathbf{f}|\mathbf{y})}{\partial \mathbf{f} \partial \mathbf{f}^\top} = -\mathbf{K}^{-1} - \mathbf{D}_\mathbf{f},$$

where $\mathbf{D}_\mathbf{f}$ is an $n \times n$ diagonal matrix with $(\mathbf{D}_\mathbf{f})_{ii} = \sigma(f(x_i))\sigma(-f(x_i))$. This Hessian is negative definite, since $\mathbf{K}^{-1}$ is positive definite and $(\mathbf{D}_\mathbf{f})_{ii} \geq 0$. Thus, there is a unique posterior mode. Note that $\mathbf{D}_\mathbf{f}$ depends on $\mathbf{f} = [f(x_1), \ldots, f(x_n)]^\top$ but not on the labels $\mathbf{y}$. We can now employ numerical optimisation (gradient ascent or Newton-Raphson method) to find the posterior mode $\hat{\mathbf{f}}^{\text{MAP}}$ and approximate the posterior $p(\mathbf{f}|\mathbf{y})$ with a normal distribution:

$$\tilde{p}(\mathbf{f}|\mathbf{y}) \;=\; \mathcal{N}\left(\mathbf{f} \,\middle|\, \hat{\mathbf{f}}^{\text{MAP}}, \left(\mathbf{K}^{-1} + \mathbf{D}_{\hat{\mathbf{f}}^{\text{MAP}}}\right)^{-1}\right).$$

Note that this can be rewritten as

$$\tilde{p}(\mathbf{f}|\mathbf{y}) \;=\; \mathcal{N}\left(\mathbf{f} \,\middle|\, \hat{\mathbf{f}}^{\text{MAP}}, \mathbf{K} - \mathbf{K}\left(\mathbf{K} + \mathbf{D}^{-1}_{\hat{\mathbf{f}}^{\text{MAP}}}\right)^{-1}\mathbf{K}\right),$$

using the Woodbury identity[1] $\left(\mathbf{K}^{-1} + \mathbf{D}\right)^{-1} = \mathbf{K} - \mathbf{K}\left(\mathbf{K} + \mathbf{D}^{-1}\right)^{-1}\mathbf{K}$ for invertible matrices $\mathbf{K}$ and $\mathbf{D}$.

We can use this further to construct an approximation of the predictive posterior as well, writing

$$\tilde{p}(\mathbf{f}'|\mathbf{y}) = \int p(\mathbf{f}'|\mathbf{f})\tilde{p}(\mathbf{f}|\mathbf{y})d\mathbf{f}, \tag{8.13}$$

which can now be solved in the closed form since $p(\mathbf{f}'|\mathbf{f})$ is also normal,

$$p\left(\mathbf{f}'|\mathbf{f}\right) = \mathcal{N}\left(\mathbf{f}' \,|\, \mathbf{K}_{\mathbf{x}'\mathbf{x}}\mathbf{K}_{\mathbf{x}\mathbf{x}}^{-1}\mathbf{f}, \mathbf{K}_{\mathbf{x}'\mathbf{x}'} - \mathbf{K}_{\mathbf{x}'\mathbf{x}}\mathbf{K}_{\mathbf{x}\mathbf{x}}^{-1}\mathbf{K}_{\mathbf{x}\mathbf{x}'}\right),$$

giving

$$\tilde{p}\left(\mathbf{f}'|\mathbf{y}\right) = \mathcal{N}\left(\mathbf{f}' \,|\, \mathbf{K}_{\mathbf{x}'\mathbf{x}}\mathbf{K}_{\mathbf{x}\mathbf{x}}^{-1}\hat{\mathbf{f}}^{\text{MAP}}, \mathbf{K}_{\mathbf{x}'\mathbf{x}'} - \mathbf{K}_{\mathbf{x}'\mathbf{x}}\left(\mathbf{K}_{\mathbf{x}\mathbf{x}} + \mathbf{D}^{-1}_{\hat{\mathbf{f}}^{\text{MAP}}}\right)^{-1}\mathbf{K}_{\mathbf{x}\mathbf{x}'}\right). \tag{8.14}$$

8.5 Large-Scale Approximations[2]

Gaussian processes and kernel methods require computational cost that scales at least as $O(n^2)$ and often as $O(n^3)$ in the number of observations n (due to the need to compute, store and invert the $n \times n$ kernel matrix $\mathbf{K}$). This is the price we pay for having a nonparametric model, i.e.

[1] Woodbury matrix identity or matrix inversion lemma in its general form is $(A+UCV)^{-1} = A^{-1} - A^{-1}U\left(C^{-1} + VA^{-1}U\right)^{-1}VA^{-1}$ for matrices A,U,C,V of conformable sizes.

[2] not examinable

for performing the computation in terms of the dual coefficients. For large datasets, e.g. where $n \sim 10^5$, this becomes a prohibitive computational cost and memory requirement, however. Many methods have been proposed to deal with this issue, here we will overview the basic approaches based on the reduced-rank approximation of $\mathbf{K_{xx}}$ - see Chapter 8 of [2] for an in-depth overview.

8.5.1 Nyström method

GP regression and kernel ridge regression both require inversion of the matrix $\mathbf{K_{xx}} + \sigma^2 I$. Let us assume for the moment that $\mathbf{K_{xx}}$ can be approximated by a rank m matrix, with $m \ll n$, i.e. $\mathbf{K_{xx}} \approx QQ^\top$, where Q is an $n \times m$ matrix. Then we can apply the matrix inversion lemma and write

$$\left(QQ^\top + \sigma^2 I\right)^{-1} = \sigma^{-2} I - \sigma^{-2} Q \left(\sigma^2 I + Q^\top Q\right)^{-1} Q^\top, \tag{8.15}$$

such that the inversion of an $n \times n$ matrix has been transformed into an inversion of an $m \times m$ matrix. However, in order to derive the optimal reduced-rank approximation to $\mathbf{K_{xx}}$, we need to perform the eigendecomposition of $\mathbf{K_{xx}}$ which is itself a costly operation, requiring $O(n^3)$ computation. Instead, an often used approach is the *Nyström approximation:*

$$\tilde{\mathbf{K}}_{\mathbf{xx}} = \mathbf{K_{xz}} \mathbf{K_{zz}^{-1}} \mathbf{K_{zx}},$$

where $\{z_j\}_{j=1}^m$ is a collection of a small number of inputs in $\mathcal{X}$ (which could be a subset of the training set, but could also be some auxiliary pseudo-inputs) called *inducing variables* or *landmark points*, and we denoted as usual $(\mathbf{K_{zz}})_{ij} = k(z_i, z_j)$, $(\mathbf{K_{xz}})_{ij} = k(x_i, z_j)$ and $\mathbf{K_{zx}} = \mathbf{K_{xz}^\top}$. Now, we can set $Q = \mathbf{K_{xz}} \mathbf{K_{zz}^{-1/2}}$ and apply the formula (8.15). Note that this is equivalent to using a finite-dimensional feature map $\phi : x \mapsto \mathbf{K_{zz}^{-1}} \left[k(z_1, x), \ldots, k(z_m, x)\right]^\top$ and an *approximate kernel*:

$$\tilde{k}(x, x') = \phi(x)^\top \phi\left(x'\right).$$

8.5.2 Random Fourier Features

Another popular method within the frequentist kernel methods are random Fourier features (RFF) of [1]. The idea behind RFF is to use Bochner's representation of *translation-invariant* kernels on $\mathbb{R}^p$, i.e. if a real-valued kernel $k(x, x')$ depends only on the difference $x - x'$, then it can be written as

$$\begin{aligned} k(x, x') &= \int_{\mathbb{R}^p} \exp\left(i\omega^\top (x - x')\right) d\Lambda(\omega) \\ &= \int_{\mathbb{R}^p} \left\{\cos\left(\omega^\top x\right) \cos\left(\omega^\top x'\right) + \sin\left(\omega^\top x\right) \sin\left(\omega^\top x'\right)\right\} d\Lambda(\omega) \end{aligned} \tag{8.16}$$

for some positive measure (w.l.o.g. a probability distribution) Λ called *spectral measure* of k. For many widely used kernels, spectral measure takes a simple form, e.g. if $k(x, x') = \exp\left(-\frac{1}{2\gamma^2} \|x - x'\|_2^2\right)$, then Λ is a multivariate normal $\mathcal{N}(0, \gamma^{-2} I)$. Now, for a given Λ, we sample m frequencies $\{\omega_j\} \sim \Lambda$

and use a Monte Carlo estimator of the kernel function given by the integral in (8.16):

$$\begin{aligned}\tilde{k}(x,y) &= \frac{1}{m}\sum_{j=1}^{m}\left\{\cos\left(\omega_j^\top x\right)\cos\left(\omega_j^\top y\right)+\sin\left(\omega_j^\top x\right)\sin\left(\omega_j^\top y\right)\right\}\\ &= \langle\phi_\omega(x),\phi_\omega(y)\rangle_{\mathbb{R}^{2m}},\end{aligned}$$

which is an approximate kernel corresponding to an explicit set of features $\phi_\omega(x)\in\mathbb{R}^{2m}$ given by

$$x\mapsto\frac{1}{\sqrt{m}}\left[\cos\left(\omega_1^\top x\right),\sin\left(\omega_1^\top x\right),\ldots,\cos\left(\omega_m^\top x\right),\sin\left(\omega_m^\top x\right)\right]$$

With this set of features, we can now run algorithms in the primal representation which is less costly than paying the computational cost in n.

References

[1] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 2007.

[2] C.E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.