Introduction to Bayesian Inference & Gaussian Processes

Trefor W. Evans

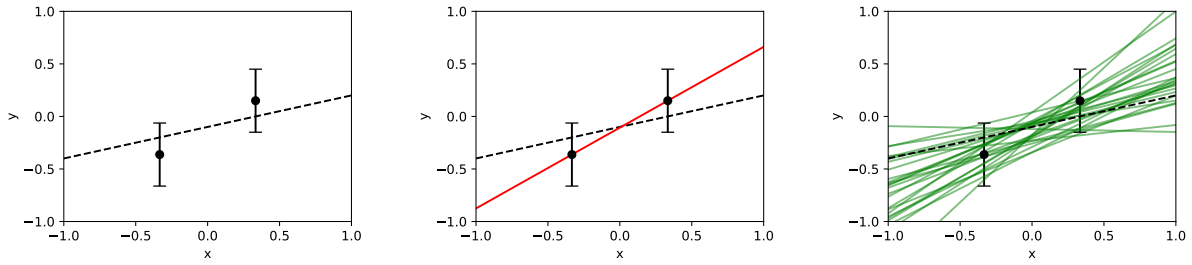
ROB313, 2021

This document provides an introduction to Bayesian inference for machine learning on the way to a comprehensive overview of Gaussian processes. There is a notable lack of clean introductory material for Gaussian processes in particular and this document aims to help fill that gap with a liberal use of visuals. As an overview, section 1 begins by introducing a Bayesian approach to machine learning followed by section 2, which then focuses on the particular modelling choice of Gaussian processes (GPs). Section 3 concludes with a discussion of techniques to perform Bayesian model selection.

1 Bayesian Learning

Machine learning ultimately aims to create algorithms that improve their performance by leveraging observed data. For example, after observing the results of two experiments, we may want a computer to predict the result of a third experiment. Machine learning algorithms rely on statistical models that will hopefully reflect reality, and these models contain parameters whose value is unknown or uncertain *a priori*. In the process of machine learning, we would like to determine the parameter values that give predictions as close to reality as possible. Given any dataset of finite size, we cannot expect to get completely certain answers about the parameter values. For example, consider fig. 1a where two noisy observations have been collected for a one-dimensional regression problem which were generated from the dashed black line with *i.i.d.* Gaussian noise applied. The goal here is to predict a value of y^* at a given value of x^*. We have chosen a linear statistical model for this problem and it is visually evident that multiple different linear curves could fit the data, each of which would give different predictions beyond the dataset. Bayesian learning differs from other approaches to machine learning in how to infer the model's parameters given the dataset. For instance, one might suggest selecting the parameter setting that best fits the data, however, this approach may perform very poorly as we move away from the training data, as seen in fig. 1b. This is known as *overfitting*, a degeneracy that must be accounted for in many approaches to machine learning. In contrast, a Bayesian approach to machine learning finds a distribution over parameter settings that agree with both the observed data and with prior knowledge. The Bayesian predictions can be seen in fig. 1c which evidently does not provide a point estimate for y^* given a value of x^* but rather a distribution over plausible values of y^*. In this example, these probabilistic

predictions compare favourably to the point predictions in fig. 1b since it reflects uncertainty due to lack of data. We will return to this example frequently throughout the document.



(a) Noisy one-dimensional dataset. (b) Maximum likelihood predictor. (c) Bayesian posterior samples.

Figure 1: Comparison of maximum likelihood and Bayesian approaches to inference. The training data is shown in black which was generated by the dashed black line and corrupted by independent Gaussian noise. The maximum likelihood predictor is given in red whereas samples from the Bayesian posterior are given by the green lines.

In Bayesian inference, probability distributions are used to reflect uncertainty. From the example in fig. 1c, samples from the probability distribution over y^* were used to visualize the fact that y^* is a random variable at any x^*. Probability theory provides a rigorous foundation that allows us to reason under uncertainty (Jaynes, 2003). We will now proceed to describe how we can use this theory in machine learning.

Parameter Inference

We begin Bayesian inference for machine learning by choosing a statistical model. Returning to the example in fig. 1, the chosen statistical model takes the form

$$\mathbf{y} = \begin{bmatrix} f(x_1;\mathbf{w}) + \epsilon_1 \\ f(x_2;\mathbf{w}) + \epsilon_2 \end{bmatrix}, \tag{1}$$

where

$$f(x;\mathbf{w}) = w_1 x + w_2 \tag{2}$$

is a linear model, $\mathbf{w} = \begin{bmatrix} w_1, w_2 \end{bmatrix}^T$ are model parameters, (x_1, y_1), (x_2, y_2) are the two data points in fig. 1a, and ϵ_1, ϵ_2 is additive noise. In a Bayesian treatment, we will consider both $\mathbf{y}$ and $\mathbf{w}$ as random variables, since before observing any data we are uncertain about these variables. We represent the uncertainty about these variables in the joint distribution under the model, $\Pr(y, \mathbf{w})$. Using the chain rule (also known as the product rule) of probability theory, we can write the joint distribution of $\mathbf{y}$ and $\mathbf{w}$ as

$$\underbrace{\Pr(\mathbf{y}, \mathbf{w})}_{\text{joint under the model}} = \underbrace{\Pr(\mathbf{y}|\mathbf{w})}_{\text{likelihood}} \underbrace{\Pr(\mathbf{w})}_{\text{prior}}. \tag{3}$$

While it is not necessary to decompose the joint under the model in this manner to perform Bayesian inference, it is often convenient for the purposes of interpretability. In our example,

eq. (1) contains all the information needed to define the *likelihood* (provided we know the statistical properties of $\epsilon_1,\ \epsilon_2$). Additionally, the *prior* $\Pr(\mathbf{w})$ can be selected based upon our belief about the value of the model parameters *a priori* (before observing any data). Specifying a good prior is important for Bayesian inference to be effective. It requires a practitioner to express their belief explicitly as a probability distribution, which can take practice.

Equation (3) contains all the information that is required to start a Bayesian modelling procedure. While decomposing the joint under the model into a prior and likelihood is attractive for the purposes of interpretability, it is often useful to consider the joint as its own entity that summarizes all information about the statistical model and the practitioner's prior beliefs.

After specifying the joint under the model, we are now ready for data. The term *inference* (or more specifically statistical inference) refers to making conclusions about uncertain variables given the observational data. This is precisely what we would like to do. To begin, consider re-writing the joint from eq. (3) using the chain rule of probability theory in a symmetric manner

$$\underbrace{\Pr(\mathbf{y}, \mathbf{w})}_{\text{joint under the model}} = \underbrace{\Pr(\mathbf{w}|\mathbf{y})}_{\text{posterior}} \underbrace{\Pr(\mathbf{y})}_{\text{model evidence}}. \tag{4}$$

Our goal is to compute the *posterior* which provides an update to our belief about $\mathbf{w}$ *after* observing the dataset. We can compute the posterior by rearranging eq. (4) to give

$$\underbrace{\Pr(\mathbf{w}|\mathbf{y})}_{\text{posterior}} = \frac{\overbrace{\Pr(\mathbf{y}, \mathbf{w})}^{\text{joint under the model}}}{\underbrace{\Pr(\mathbf{y})}_{\text{model evidence}}} = \frac{\overbrace{\Pr(\mathbf{y}|\mathbf{w})}^{\text{likelihood}} \overbrace{\Pr(\mathbf{w})}^{\text{prior}}}{\underbrace{\Pr(\mathbf{y})}_{\text{model evidence}}}. \tag{5}$$

This simple relation is referred to as *Bayes' rule* and describes how we can update our beliefs after observing data. The only element in the preceding equation that has not yet been discussed is the *model evidence* (also known as the *marginal likelihood*) which is the joint under the model with $\mathbf{w}$ marginalized that can be defined as follows

$$\underbrace{\Pr(\mathbf{y})}_{\text{model evidence}} = \int \underbrace{\Pr(\mathbf{y}, \mathbf{w})}_{\text{joint under the model}} d\mathbf{w}. \tag{6}$$

Unfortunately, this expression is usually challenging to compute since it is an integral which is often high-dimensional and cannot be computed in closed-form in many instances. Evaluating the model evidence is typically the most computationally challenging aspect of Bayesian inference, and this integral alone typically makes Bayesian inference more computationally taxing than a point-estimate approach such as the maximum likelihood procedure shown in fig. 1b for our simple example. This is perhaps not too surprising considering that Bayesian inference requires inferring distributions rather than point estimates. A wealth of approaches have been developed to ease computational burden through *approximate* Bayesian inference techniques. However, we shall see in section 3.1 that the model evidence can be computed in closed form for the majority of the models we will consider.

Predictive Inference

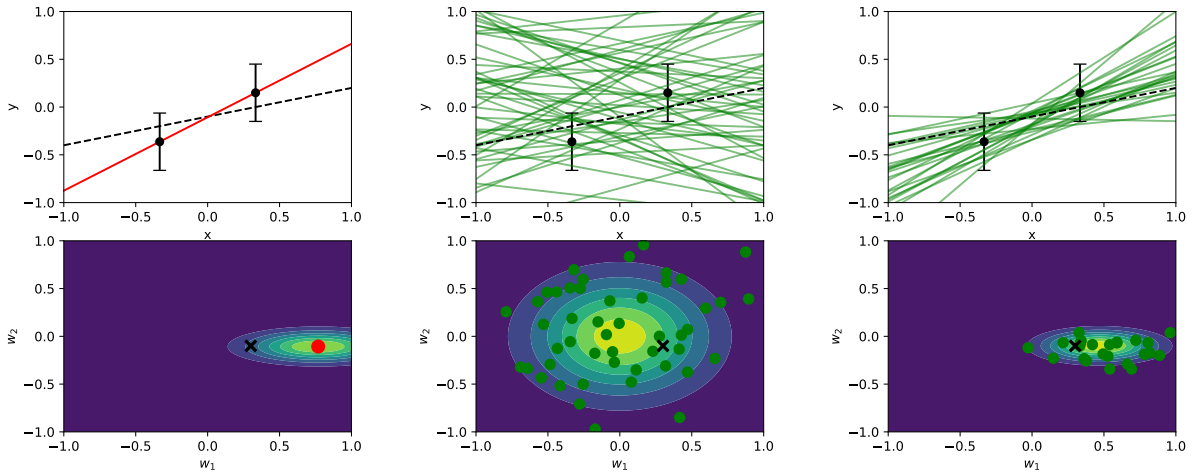
We previously showed how to update beliefs about the model parameters given data observations through the posterior, $\Pr(\mathbf{w}|\mathbf{y})$. While in some scenarios, a practitioner might care about the model parameters directly, in most machine learning scenarios, we only care about the ability to predict $f(\mathbf{x}^*)$ at a value of $\mathbf{x}^*$ that was not in the training dataset. In other words, we would like to perform *predictive inference*. This can be performed as follows

$$\underbrace{\Pr(f(\mathbf{x}^*)|\mathbf{y})}_{\text{predictive posterior}} = \int \underbrace{\Pr(f(\mathbf{x}^*)|\mathbf{w})}_{\text{predictive likelihood}} \underbrace{\Pr(\mathbf{w}|\mathbf{y})}_{\text{posterior}} d\mathbf{w}, \tag{7}$$

which is evidently a weighted average of the predictions at all values of parameters $\mathbf{w}$, weighted by the posterior. In this way, uncertainty of the parameters are taken into account to express uncertainty over predictions. Quantifying predictive uncertainty is crucial for safe or optimal decision making. In these cases, the predictive posterior would be used for downstream decision making procedures to directly assess risk and potential reward.

In fig. 1c, the predictive posterior was sampled 25 times for visualization. In that example, the *predictive likelihood* was the degenerate distribution $\Pr(f(x^*)|\mathbf{w}) = \delta(f(x^*;\mathbf{w}) - y^*)$ since there is a deterministic relationship between f and $\mathbf{w}$ given by eq. (2), where $\delta(\cdot)$ denotes the Dirac delta.

Example



(a) Maximum likelihood predictor. (b) Bayesian prior. (c) Bayesian posterior.

Figure 2: Comparison of maximum likelihood and Bayesian approaches to inference. A linear model $f(x) = w_1 x + w_2$ is employed and the top plots demonstrate inference in function space (x, y) whereas the bottom plots show inference in weight space (w_1, w_2). The green lines and dots denote samples drawn from the function and weight space, respectively. The contour plots in the bottom row from left-to-right contain the likelihood, prior and posterior.

Returning to our original example in fig. 1, we now dig into the modelling choices made, including the likelihood and prior, to better understand the inference procedures conducted.

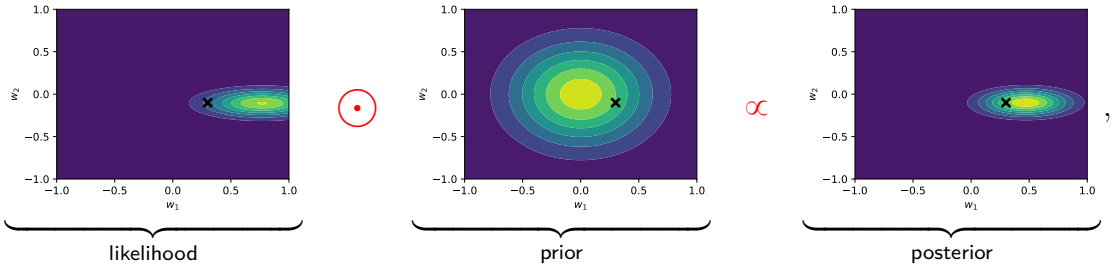
Beginning with the likelihood, the two observations in the dataset were corrupted with *i.i.d.* Gaussian noise, and therefore the random variables $\epsilon_1,\ \epsilon_2$ from eq. (1) are *i.i.d.* Gaussian with variance σ^2. The assumption of *i.i.d.* Gaussian noise is commonly employed in practice, and this assumption alone allows us to define our likelihood as follows

$$\underbrace{\mathsf{Pr}(\mathbf{y}|\mathbf{w})}_{\text{likelihood}} = \mathcal{N}\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \middle| \middle| \begin{bmatrix} f(x_1;\mathbf{w}) \\ f(x_2;\mathbf{w}) \end{bmatrix},\ \sigma^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right). \tag{8}$$

The likelihood of the dataset is plotted on the bottom of fig. 2a as a contour plot for various values of parameters $(w_1,\ w_2)$. It is evident that the red *maximum likelihood* line in the top of fig. 2a corresponds to the red dot in the bottom plot that maximizes the likelihood. If a point estimate is to be made, this choice does seem sensible, however, this simple example alone showcases the danger of making point estimates and its susceptibility to *overfitting*.

Next, we can define a prior, $\mathsf{Pr}(\mathbf{w})$. In this case we choose an *i.i.d.* Gaussian prior which is plotted on the bottom of fig. 2b as a contour plot. We can visualize the effect of this prior by drawing samples from it (green dots) and plotting each sample in the top plot (green lines). Evidently, this is not a particularly informative prior and indicates that we are not very aware of what the values of $(w_1,\ w_2)$ should be.

Now that we have specified the likelihood and prior, the posterior can be computed. There are many techniques that can be applied to compute or approximate a posterior[1]. For this simple example, we can consider a naive approach that can help to understand the posterior more clearly. From Bayes' rule in eq. (5), the posterior is evidently proportional to the product of the likelihood and prior. Visually, we can write this as



(9)

where $\odot$ denotes an "elementwise" product that multiplies the likelihood and prior at each value of $(w_1,\ w_2)$. The relation on the left-hand side of eq. (9) is proportional to the posterior up to a multiplicative factor. The multiplicative factor is the inverse model evidence as given by Bayes' rule in eq. (5), and this factor ensures that the posterior integrates to unity; a requirement for a valid probability distribution. Once again, we can visualize the effect of the posterior in the bottom plot of fig. 2c by drawing samples from it (green dots) and plotting each sample in the top plot (green lines). It can be noted that the posterior distribution is a much tighter distribution than the prior, and it is consistent with the data.

[1] In fact, this posterior can be computed in closed form as we will show in section 2.1.

2 Gaussian Processes

The previous section discussed Bayesian learning in a general setting. We now proceed towards a particular (albeit powerful) modelling choice, Gaussian processes (GPs). Beginning with a general class of basis function models, we demonstrate how a prior in weight space ($\mathbf{w}$) implies a prior in function space (f). From there, we demonstrate an equivalent view of basis function models in terms of *kernels* and show how this perspective enables i) a powerful specification of priors directly in the function space, and ii) the use of infinitely many basis functions.

2.1 Basis Function Models

Basis function models (also known as generalized linear models) are those that can be written in the form

$$f(\mathbf{x}) = \sum_{i=1}^{m} w_i \phi_i(\mathbf{x}), \tag{10}$$

where $\mathbf{x} \in \mathbb{R}^d$ is a d-dimensional input, $\phi_i : \mathbb{R}^d \to \mathbb{R}$ for $i = 1,...,m$ are d-dimensional basis functions, and $\mathbf{w} \in \mathbb{R}^m$ are parameters (or weights). This form is extremely general. In fact, almost all machine learning models can be interpreted in this way from linear models to deep learning models and Gaussian processes. The one-dimensional linear example from the previous section in eq. (2) can be written in this form where

$$\phi_1(x) = x, \quad \text{and} \quad \phi_2 = 1. \tag{11}$$

In this example, the basis functions are linear, however, they can be non-linear functions in general. We will proceed with the same likelihood as given in eq. (8) that assumes the training observations are corrupted by *i.i.d.* Gaussian noise with variance σ^2. This gives

$$\underbrace{\mathsf{Pr}\big(\mathbf{y} \big| \mathbf{w}\big)}_{\text{likelihood}} = \mathcal{N}\big(\mathbf{y} \big| \mathbf{\Phi}\mathbf{w},\, \sigma^2 \mathbf{I}_n\big), \tag{12}$$

where we have generalized our notation such that $\mathbf{y} \in \mathbb{R}^n$ contains the observations from the dataset $\{\mathbf{x}_i, y_i\}_{i=1}^n$ of size n, $\mathbf{\Phi} \in \mathbb{R}^{n \times m}$ is a matrix whose ith column contains the evaluation of ϕ_i on all n training points, and $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ is the identity matrix. As in the previous example, we will also proceed assuming a Gaussian prior on the weights $\mathsf{Pr}(\mathbf{w})$ to give

$$\underbrace{\mathsf{Pr}\big(\mathbf{w}\big)}_{\text{prior}} = \mathcal{N}\big(\mathbf{w} \big| \mathbf{0},\, \mathbf{S}^{-1}\big), \tag{13}$$

where $\mathbf{S} \in \mathbb{R}^{m \times m}$ is a symmetric positive definite precision matrix, and we have assumed a zero-prior mean for ease of exposition. Conveniently, because of the choice of the Gaussian prior, the posterior of the discussed model is also Gaussian and can be directly computed in closed form. A simple

derivation of this result can be seen by writing the natural logarithm of Bayes' rule eq. (5) to give

$$\begin{aligned}\log\Pr(\mathbf{w}|\mathbf{y}) &= \log\Pr(\mathbf{w}) + \log\Pr(\mathbf{y}|\mathbf{w}) - \log\Pr(\mathbf{y}) \\ &= -\frac{1}{2}\mathbf{w}^T\mathbf{S}\mathbf{w} - \frac{1}{2\sigma^2}(\mathbf{\Phi}\mathbf{w}-\mathbf{y})^T\mathbf{I}_n(\mathbf{\Phi}\mathbf{w}-\mathbf{y}) + \text{const.},\end{aligned}$$

where "const." contains all the terms that do not depend on $\mathbf{w}$ which includes normalizing terms from the likelihood and prior, as well as the entirety of the model evidence. Expanding, and completing the square gives

$$\log\Pr(\mathbf{w}|\mathbf{y}) = -\frac{1}{2}(\mathbf{w}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{w}-\boldsymbol{\mu}) + \text{const.},$$

where

$$\boldsymbol{\mu} = \sigma^{-2}\mathbf{\Sigma}\mathbf{\Phi}^T\mathbf{y}, \quad \text{and} \quad \mathbf{\Sigma}^{-1} = \sigma^{-2}\mathbf{\Phi}^T\mathbf{\Phi} + \mathbf{S}. \tag{14}$$

We can recognize this quadratic form as the log probability density of the multivariate Gaussian

$$\Pr(\mathbf{w}|\mathbf{y}) = \mathcal{N}\big(\mathbf{w}\big|\boldsymbol{\mu}, \mathbf{\Sigma}\big). \tag{15}$$

The proceeding equation demonstrates how inference can be analytically performed in a basis function model, requiring $\mathcal{O}(m^2n + m^3)$ time for the matrix operations involved, in general. Returning to our example, the posterior in the bottom plot of fig. 2c is evidently a Gaussian distribution. In this way, sampling from the posterior (the green dots) was performed using multivariate Gaussian sampling techniques.

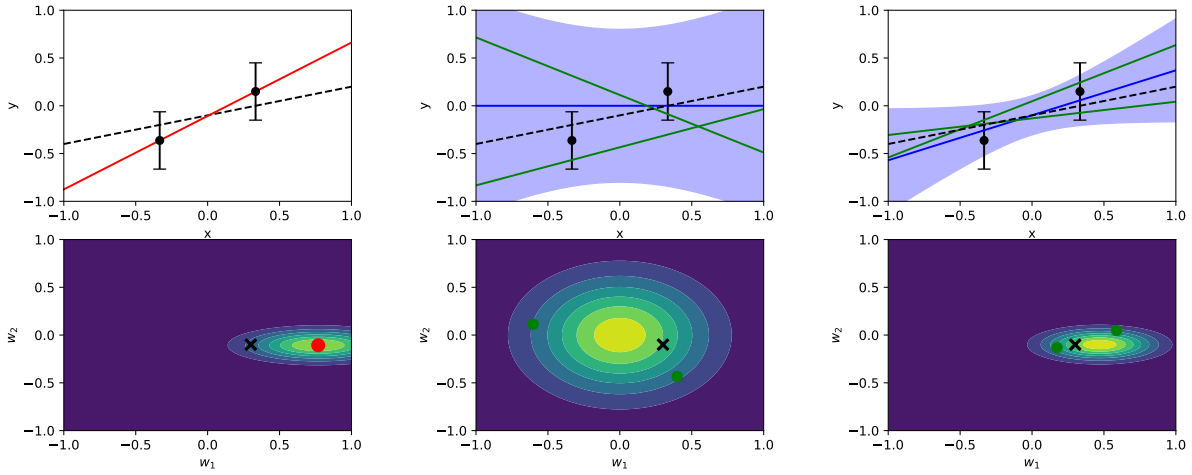
Predictive Posterior

To derive the predictive posterior, observe that the basis function model in eq. (10) is linear in $\mathbf{w}$. Observing that a linear function of Gaussian random variables is also Gaussian, we can conclude that the predictive posterior is Gaussian since the posterior $\Pr(\mathbf{w}|\mathbf{y})$ is Gaussian. Its form is given by

$$\Pr\big(f(\mathbf{x}^*)|\mathbf{y}, \mathbf{x}^*\big) = \mathcal{N}\big(f(\mathbf{x}^*)\big|\boldsymbol{\phi}(\mathbf{x}^*)^T\boldsymbol{\mu}, \boldsymbol{\phi}(\mathbf{x}^*)^T\mathbf{\Sigma}\boldsymbol{\phi}(\mathbf{x}^*)\big), \tag{16}$$

where $\boldsymbol{\phi}(\mathbf{x}^*) \in \mathbb{R}^m$ contains the evaluations of all m basis functions at $\mathbf{x}^*$. As a result of symmetry of the Gaussian distribution, it is not surprising that the predictive posterior mean is simply the evaluation of the basis function model (eq. (10)) using the posterior mean, i.e. using $\mathbf{w} = \boldsymbol{\mu}$. Additionally, note the quadratic form of the predictive posterior variance which shows that the predictive uncertainty grows with the magnitude of the basis functions. Using our example of linear basis functions in eq. (11), the uncertainty grows with the magnitude of x, as we would expect for a linear model. This can be seen in the top plot of fig. 3c where the shaded region shows two standard deviations from the mean. The top plot of fig. 3b also shows the "predictive" prior mean (blue line) and two standard deviations (shaded). This is also Gaussian (since the prior is Gaussian) and is given by

$$\Pr\big(f(\mathbf{x}^*)\big) = \mathcal{N}\big(f(\mathbf{x}^*)\big|0, \boldsymbol{\phi}(\mathbf{x}^*)^T\mathbf{S}^{-1}\boldsymbol{\phi}(\mathbf{x}^*)\big). \tag{17}$$



(a) Maximum likelihood predictor. (b) Bayesian prior. (c) Bayesian posterior.

Figure 3: Comparison of maximum likelihood and Bayesian approaches to inference. The statistical model employed is the linear model $y = w_1 x + w_2$ and the top plots demonstrate inference in function space $(x,\ y)$ whereas the bottom plots show inference in weight space $(w_1,\ w_2)$. In the top plots, the training data is shown in black which was generated by the dashed black line and corrupted by independent Gaussian noise. The parameter values of the dashed black line is shown by a black $\times$ in the bottom plots. The maximum likelihood predictor is given in red. For the Bayesian models, the blue line denotes the posterior mean, the shaded region denotes the 95% confidence interval, and the green lines and dots denote samples from function and weight space, respectively. The contour plots in the bottom row from left-to-right contain the likelihood, prior and posterior.

2.2 Function-space View

Equivalent relations to those derived in the previous section can be found by taking an alternative view. We call this different perspective a *function-space* view since inference is performed directly in function space without ever explicitly discussing basis functions or parameters/weights. We will see that in some scenarios this perspective will be computationally preferable to the previous approach (that we call a *weight-space* view), and it is certainly more interpretable.

To begin, consider the n noise-free function values $\mathbf{f} = \{f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)\}$ at the inputs $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Extending the prior from eq. (17) to multiple function values, it is easy to see that the prior over $\mathbf{f}$ is jointly Gaussian and is given by

$$\Pr(\mathbf{f}) = \mathcal{N}\big(\mathbf{f} \big| \mathbf{0},\, \mathbf{\Phi}\mathbf{S}^{-1}\mathbf{\Phi}^T\big) = \mathcal{N}\big(\mathbf{f} \big| \mathbf{0},\, \mathbf{K}(\mathbf{X},\, \mathbf{X})\big), \tag{18}$$

where we have introduced $\mathbf{K}(\mathbf{X},\, \mathbf{X}) = \mathbf{\Phi}\mathbf{S}^{-1}\mathbf{\Phi}^T \in \mathbb{R}^{n \times n}$. The matrix $\mathbf{K}(\mathbf{X},\mathbf{X})$ describes the prior covariance between the random variables $\mathbf{f}$ such that

$$[\mathbf{K}(\mathbf{X},\, \mathbf{X})]_{i,j} = \boldsymbol{\phi}(\mathbf{x}_i)^T \mathbf{S}^{-1} \boldsymbol{\phi}(\mathbf{x}_j) = \mathbb{E}\big[f(\mathbf{x}_i) f(\mathbf{x}_j)\big] = k(\mathbf{x}_i,\, \mathbf{x}_j), \tag{19}$$

where we have introduced $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ which we shall refer to the prior covariance *kernel* (also called the *covariance function)*. The kernel describes the prior covariance between the function values at two arbitrary points in d-dimensional input space. It is all that is required to define a zero mean Gaussian process:

A *Gaussian process* (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution.

A Gaussian possess is defined entirely by a covariance function k, and a mean function which we have assumed to be zero for the purpose of clarity[2]. Equation (18) demonstrates precisely that using the covariance kernel k, a finite collection of (n) observations of the target are jointly Gaussian.

Predictions with Noise-Free Observations

Writing the prior in eq. (18) using a covariance matrix constructed by the kernel k immediately suggests a simple alternative inference procedure when the observations are noise-free. Consider $\mathbf{f} \in \mathbb{R}^n$ to be the n noise-free observations of the target, and take f^* to be the response at a test input $\mathbf{x}^*$. Since a finite collection of targets is assumed to have a joint Gaussian distribution in Gaussian process modelling, we can extend the collection $\mathbf{f}$ of n random variables in eq. (18) to write the joint prior over training observations $\mathbf{f}$ and the test observation f^* as the following joint Gaussian distribution whose $(n+1)\times(n+1)$ covariance matrix is formed by the kernel k

$$\Pr(\mathbf{f}, f^*) = \mathcal{N}\left(\begin{bmatrix} \mathbf{f} \\ f^* \end{bmatrix} \middle| \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) & \mathbf{k}(\mathbf{X}, \mathbf{x}^*) \\ \mathbf{k}(\mathbf{x}^*, \mathbf{X}) & k(\mathbf{x}^*, \mathbf{x}^*) \end{bmatrix}\right). \tag{20}$$

To derive the posterior distribution, we need to restrict this joint prior distribution to contain only realizations that are consistent with the training observations $\mathbf{f}$. In probabilistic terms, this simply describes *conditioning* the joint distribution on the training observations $\mathbf{f}$. This can be performed using standard Gaussian identities as follows (e.g. (Rasmussen and Williams, 2006, appendix A.2))

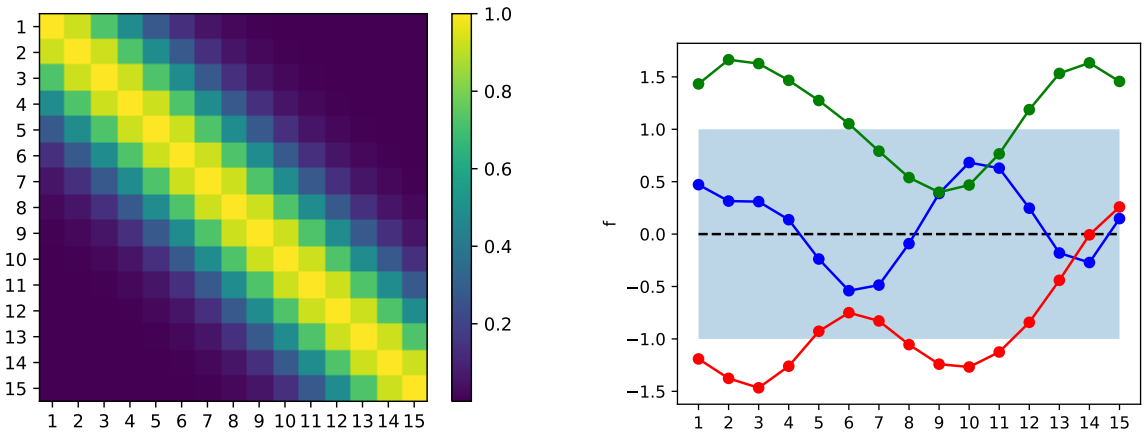
$$\Pr(f^*|\mathbf{f}) = \mathcal{N}\big(f^* \big| \mathbf{k}(\mathbf{x}^*, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{f},\, k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*, \mathbf{X})\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{k}(\mathbf{X}, \mathbf{x}^*)\big). \tag{21}$$

This is an elegant result since we were able to go from the prior directly to the posterior without explicitly considering the weight space at all. This posterior is also easy to compute, requiring only linear algebra operations, and the computations can be trivially extended to evaluate the predictive posterior on a set of test points $\mathbf{X}^*$ by simply replacing $\mathbf{x}^*$ with $\mathbf{X}^*$.

Infinite Basis Functions

Unfortunately, the posterior in eq. (21) is not defined for all basis function modelling choices. This is because the prior covariance matrix $\mathbf{K}(\mathbf{X}, \mathbf{X})$ in eq. (18) is a semi-positive definite matrix and may be singular such that $\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}$ is not defined. For instance, it will be singular if $m < n$ since the covariance is of rank at most m. One approach to help deal with this singularity is to expand the number of features m. At first glace this would appear to be an expensive solution. After all, we are considering increasing the complexity of our basis function model so it is reasonable to expect this

[2] A mean function can be easily incorporated as well. One way to account for a non-zero prior mean is to simply consider the random variables $\mathbf{f}$ to be realizations of a function with the mean function subtracted. Therefore, any realizations of $\mathbf{f}$ need to have the mean function added to it before interpretation.



(a) Prior covariance matrix, $\mathbf{K}(\mathbf{X}, \mathbf{X})$.

(b) Three samples of $\mathbf{f}$ from the prior.

Figure 4: Visualization of the Gaussian process covariance, and samples on a finite set of points $\mathbf{X}$.

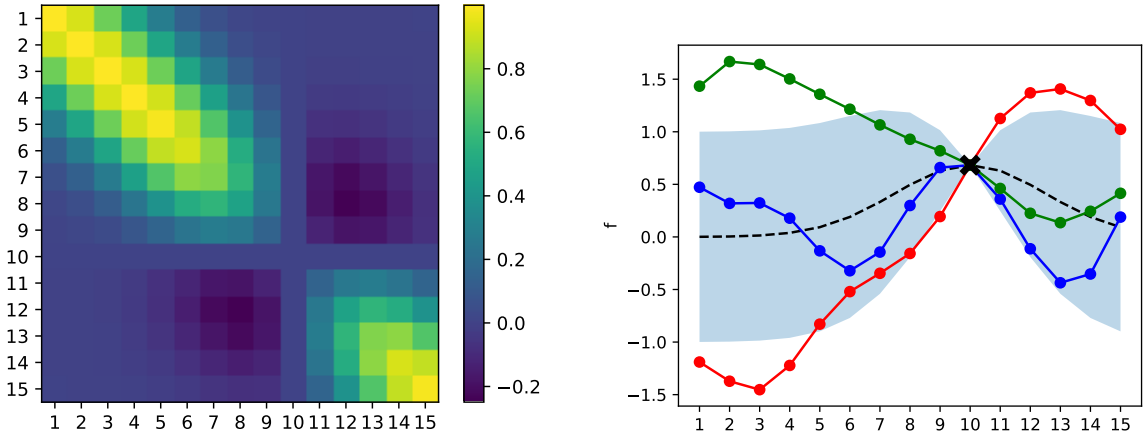
would come with an increase in cost. When we were considering inference from a weight-space perspective, computation of the posterior in eq. (15) cost $\mathcal{O}(m^2n+m^3)$ time, giving an expensive cubic scaling in the number of features. We will see, however, that by taking a function-space perspective, the cost of inference can be *independent* of the number of features if we use the kernel in a clever way.

To begin, consider that eq. (21) only requires evaluations of the kernel k, not the basis functions themselves. Therefore, if we can evaluate a kernel without directly computing the inner product between basis functions then the cost of kernel evaluation will be independent of the number of features, and so will the cost of evaluating the posterior in eq. (15). This is possible, however, we require some properties for this kernel, namely that it must admit a symmetric and positive semi-definite covariance matrix $\mathbf{K}(\mathbf{X}, \mathbf{X})$ for any collection of points in $\mathbb{R}^d$. These come directly from the requirements of the covariance of a Gaussian distribution. Fortunately, such kernels exist, for example, consider the exponentiated quadratic kernel (also known as the squared exponential kernel)

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\tfrac{1}{2}||\mathbf{x}_i - \mathbf{x}_j||_2^2\right). \tag{22}$$

It can be shown that the exponentiated quadratic kernel corresponds to the inner product of an infinite number of basis functions, i.e. $m = \infty$ when using this kernel. For example, we can obtain the exponentiated quadratic kernel as the inner product of an infinite number of Gaussian-shaped basis functions (see (Rasmussen and Williams, 2006, sec. 4.2.1)). This is a remarkable property: by using such a kernel, we can expand the capacity of our model from one using a finite number of basis functions to one using an infinite number of basis functions without incurring any additional cost.

Let us now try to get a better grasp of the Gaussian process prior defined by the covariance kernel k. Figure 4a shows the prior covariance matrix using the exponentiated quadratic kernel in eq. (22). The matrix shows the covariance between 15 points $\mathbf{X}$ in $d = 1$ dimension such that $\mathbf{x}_i = i$, i.e. the value of $\mathbf{x}$ is the same as its index. By observing fig. 4a it is immediately evident that points closer together have higher covariance. For example, $k(\mathbf{x}_1{=}1, \mathbf{x}_2{=}2)$ at index $(1, 2)$ has greater covariance then $k(\mathbf{x}_1{=}1, \mathbf{x}_3{=}3)$ at index $(1, 3)$. It seems sensible that locations far from one



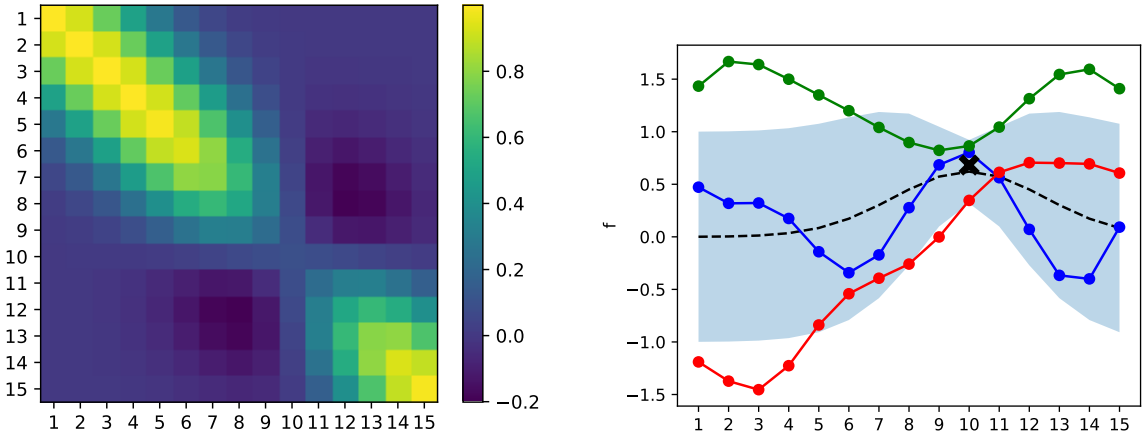
(a) Posterior covariance matrix after conditioning on index 10.

(b) Three samples from the posterior after conditioning on index 10.

Figure 5: Visualization of the Gaussian process posterior, and samples on a finite set of points. A single noise-free observation is assumed.

another should have lesser covariance, this is one of the properties of the exponentiated quadratic kernel. The three coloured curves in fig. 4b show three samples drawn from the zero-mean Gaussian distribution given in eq. (18) with the covariance matrix given in fig. 4a, and lines were drawn between the 15 points in $\mathbf{X}$ to aid visualization. Since the set $\mathbf{X}$ is arbitrary, the procedure used the create the samples in fig. 4b could be extended by expanding the set $\mathbf{X}$ to contain all infinitely many points in $\mathbb{R}$, giving a distribution over functions.

While fig. 4 visualized the GP prior, fig. 5 visualizes the GP posterior after the prior has been conditioned on a single observation at $\mathbf{x}_{10} = 10$ given by the black $\times$ in fig. 5b. Once again, the three coloured curves in fig. 5b show three samples, but this time they are drawn from the posterior Gaussian distribution given in eq. (21) with the covariance matrix given in fig. 5a and the mean given by the dashed black line in fig. 5b. Also shown in fig. 5b, is the shaded region that shows one standard deviation (square-root of the diagonal of fig. 5a) from posterior mean. Evidently the posterior variance vanishes at $\mathbf{x}_{10}$ as would be expected since we are absolutely certain about the value of the function at this point. Additionally, it can be seen that the posterior returns to the prior as we move away from $\mathbf{x}_{10}$, indicating that we know little about the function values far from this point. One of the beautiful features of Gaussian processes is that the predictive posterior statistical moments are given in closed form. This allows easy interpretation without needing to sample the posterior since the posterior mean (dashed black line) gives the expectation of the prediction as a point estimate, while the posterior variance (visualized by the shaded region) gives the posterior uncertainty which gives some measure of confidence in a prediction. Lastly, note that it is trivial to extend these computations to multidimensional inputs by simply changing the evaluation of the covariance kernel in accordance with eq. (22).



(a) Posterior covariance matrix after noisy conditioning on index 10.

(b) Three samples from the posterior after noisy conditioning on index 10.

Figure 6: Visualization of the Gaussian process posterior, and samples on a finite set of points. An observation that is corrupted with additive Gaussian noise is assumed.

Predictions with Noisy Observations

Extending the number of features m to infinity helped improve the rank of of $\mathbf{K}(\mathbf{X}, \mathbf{X})$, however, it still does not guarantee that the matrix will be full rank. As an example, consider the scenario where two input points with indices i and j are identical such that $\mathbf{x}_i = \mathbf{x}_j$. In this case the ith row (or column) of $\mathbf{K}(\mathbf{X}, \mathbf{X})$ will be identical to the jth row (or column) and the covariance will be rank deficient no matter what positive semi-definite kernel is used. A definitive way to deal with such singularities is to assume all observations are corrupted by additive independent Gaussian noise. Although other forms of noise are certainly possible, additive *i.i.d.* Gaussian noise is a very common assumption in practice. In fact, it is typical that we do not have access to the function values themselves, but noisy versions thereof (as we had assumed in eq. (1) in the original example of this document).

Additive *i.i.d.* Gaussian noise with variance σ^2 gives the following prior over the training observations

$$\Pr(\mathbf{y}) = \mathcal{N}\left(\mathbf{y} \middle| \mathbf{0}, \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}_n\right), \tag{23}$$

whose covariance differs from the prior over $\mathbf{f}$ (in eq. (18)) by the addition of a diagonal matrix. We can then modify eq. (20) to give the joint prior over noisy training observations and a test point as follows

$$\Pr(\mathbf{y}, f^*) = \mathcal{N}\left(\begin{bmatrix} \mathbf{y} \\ f^* \end{bmatrix} \middle| \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}_n & \mathbf{k}(\mathbf{X}, \mathbf{x}^*) \\ \mathbf{k}(\mathbf{x}^*, \mathbf{X}) & k(\mathbf{x}^*, \mathbf{x}^*) \end{bmatrix}\right). \tag{24}$$

We can now follow the Gaussian conditioning expression in eq. (21) to condition the joint in

eq. (24) on the noisy observations $\mathbf{y}$ as follows

$$\begin{aligned}\Pr(f^*|\mathbf{y}) &= \mathcal{N}\big(f^*\big|\mathbb{E}\big[f^*\big], \operatorname{cov}\big[f^*\big]\big), \quad \text{where} \\ \mathbb{E}\big[f^*\big] &= \mathbf{k}(\mathbf{x}^*, \mathbf{X})\Big(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2\mathbf{I}_n\Big)^{-1}\mathbf{f}, \quad \text{and} \\ \operatorname{cov}\big[f^*\big] &= k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*, \mathbf{X})\Big(\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2\mathbf{I}_n\Big)^{-1}\mathbf{k}(\mathbf{X}, \mathbf{x}^*).\end{aligned} \tag{25}$$

The preceding equation describes the key predictive equations for Gaussian process regression[3]. Also, it can be shown that the predictive posterior in the preceding equation is identical to the predictive posterior we had derived from a weight-space approach in eq. (17) provided we use the kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}(\mathbf{x}_i)^T\mathbf{S}^{-1}\boldsymbol{\phi}(\mathbf{x}_j)$ from eq. (19) (Rasmussen and Williams, 2006, chapter 2). This observation means that we arrive at the same predictive posterior if we take a weight-space or a function-space perspective, although the two approaches have differing computational complexities. Based upon the algebraic operations in their respective relations, computation of the predictive posterior generally scales as follows for the two approaches.

Perspective	Time	Storage
Weight-Space (eq. (16))	$\mathcal{O}(nm^2 + m^3)$	$\mathcal{O}(nm + m^2)$
Function-Space (eq. (25))	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$

(26)

It is therefore evident that the function-space view is generally preferred when the number of basis functions (m) is greater than the number of observations (n). When the training dataset size n is not prohibitively large, the function-space perspective is generally preferred since it allows a potentially infinite number of features $(m \to \infty)$ with no additional cost, and it allows a wealth of interpretable kernels to be used for specification of the prior (as we will discuss in the following section).

In fig. 6, we extend the visualization of fig. 5 by again conditioning the prior Gaussian distribution on index $\mathbf{x}_{10}$ but this time we assume the observation is corrupted by additive Gaussian noise with variance $\sigma^2 = 0.1$. In contrast to the example of fig. 5, we see that the variance does not vanish at $\mathbf{x}_{10}$ since we are not completely certain about the value of the function at this point because of the noise in the observation.

2.3 Covariance Kernels

Here we discuss how the choice of covariance kernel k affects Gaussian process inference. Specifically (and quite simply), choosing a kernel is synonymous with selecting a Gaussian process prior. This connection is clear when the zero-mean GP prior eq. (23) is inspected, since the kernel k is the only element we have control over in this equation. Kernels offer an elegant and easily interpretable way to specify priors, allowing practitioners to incorporate of high-level domain knowledge about a learning problem such as stationarity, differentiability, periodicity, scale, expected change over a distance, and can even enforce complicated linear operator constraints. Kernel selection is

[3] For the outline of a stable algorithmic implementation of Gaussian process predictions, please see (Rasmussen and Williams, 2006, alg. 2.1).

an extremely important topic for effective inference with Gaussian processes and this section is intended as an introduction to this field. The interested reader is referred to (Rasmussen and Williams, 2006, chapter 4) for a thorough overview.

Flexibility To begin, it is important to consider the dimension m of the feature expansion of a covariance kernel since this does affect the flexibility of a Gaussian process model. These features $\boldsymbol{\phi} : \mathbb{R}^d \rightarrow \mathbb{R}^m$ can be seen in eq. (19) and we have considered cases where m is finite, as well as infinite. In the case where the kernel can be represented exactly by a finite basis function expansion (i.e. a finite m), the resulting Gaussian process will have a finite capacity to model observations. We call a kernel with a finite basis function expansion *degenerate*. For example, consider the example in fig. 3 where linear basis functions were employed. Clearly the resultant Gaussian process does not have the flexibility to model an arbitrary non-linear function no matter how many observations are provided. Choosing a kernel with this basis function is a good choice if we know the resultant function is linear *a priori*, however, it will clearly be a poor choice if there is a possibility the function is non-linear. Conversely, some kernels used in practice have $m = \infty$ and evidently have infinite flexibility which means that the Gaussian process has the capacity to model increasingly complicated functions as more data arrives. Such a model is considered *non-parametric*. In addition, many kernels with an infinite basis function expansion admit a GP that will be *universally consistent*, meaning that any function can be approximated to arbitrary precision. The exponentiated quadratic kernel is one such example. This universal consistency implies that the Gaussian process prior has support over the space of all functions and therefore will be able to recover the true underlying function in the limit of infinite data, even if the initial prior is poorly specified. As a final note, some approaches scalable Gaussian processes make a trade-off between flexibility and computational complexity when working with very large datasets.

Differentiability The kernel can be chosen to admit a Gaussian process that has a specified level of smoothness. For example, the Matérn class of kernels can be used to specify Gaussian processes with varying levels of differentiability. Table 1 specifies several popular Matérn kernels that are zero-, one- and two-times mean-squared differentiable. The exponentiated quadratic kernel (eq. (22)) is also in the Matérn family, being infinitely differentiable. Realizations of Gaussian processes using these kernels are shown in fig. 7. It is evident that the function behaviour varies dramatically based on the level of smoothness and therefore if a given level of differentiability is known *a priori*, this is powerful information that can be used to select an appropriate kernel to restrict the class of functions under the prior appropriately.

Stationarity A covariance kernel $k(\mathbf{x}_i, \mathbf{x}_j)$ is stationary (also called translation invariant) if it can be written as a function of $\mathbf{x}_i - \mathbf{x}_j$. Stationarity means that the covariance between two points in the input space depends only on the distance between the two points and does not depend on the absolute location of the points. This has the effect of assuming the Gaussian process prior behaves similarly throughout the input space, which is often a logical presumption. For example, the exponentiated quadratic kernel in eq. (22) and the Matérn kernels in table 1 are all stationary. If desired, a stationary kernel can be made non-stationary using a non-linear input warping to give

Matérn-5/2	$k(r) = \left(1+\sqrt{5}r+\frac{5}{3}r^2\right)\exp\left(-\sqrt{5}r\right)$	Twice differentiable
Matérn-3/2	$k(r) = \left(1+\sqrt{3}r\right)\exp\left(-\sqrt{3}r\right)$	Once differentiable
Matérn-1/2	$k(r) = \exp\left(-r\right)$	Non-differentiable

Table 1: Popular Matérn kernels. The shorthand $r = ||\mathbf{x}_i - \mathbf{x}_j||_2$ was used where $\mathbf{x}_i$, $\mathbf{x}_j$ are the kernel inputs. The differentiability statements refer to mean-square differentiability of the Gaussian process that uses the respective covariance kernel. All kernels listed admit a mean-square continuous Gaussian process.

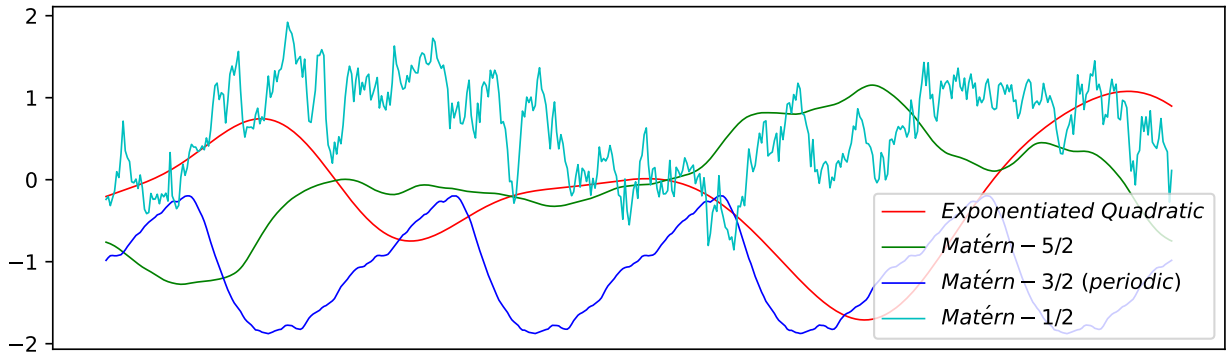


Figure 7: Realizations of zero-mean Gaussian process priors using various Matérn kernels.

the following modified kernel

$$k\big(\mathbf{g}(\mathbf{x}_i), \mathbf{g}(\mathbf{x}_j)\big), \tag{27}$$

where $\mathbf{g}: \mathbb{R}^d \to \mathbb{R}^p$ is an arbitrary non-linear function, and the number of outputs $p \geqslant 1$ can also be arbitrary. For example, Snoek et al. (2014) introduce a simple non-linear warping that can account for non-stationarity.

Periodicity Function periodicity can also be modelled through an appropriately chosen covariance kernel. A periodic prior can be employed using the warping function $\mathbf{g}(x) = \big[\cos(x), \sin(x)\big]^T$ for $d = 1$ dimensional inputs and applied as described in eq. (27). This periodic warping is applied to the Matérn-3/2 kernel in fig. 7 where the realization exhibits periodic behaviour, as expected.

Variance & Lengthscale To account for functions that have differing observation magnitudes and input scales, modifications can be made to all discussed kernels. As an example, we will re-write the exponentiated quadratic kernel originally given in eq. (22) to introduce additional hyperparameters as follows

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_0^2 \exp\Big(-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{\Lambda}^{-1} (\mathbf{x}_i - \mathbf{x}_j)\Big), \tag{28}$$

where $\sigma_0^2 > 0$ is the kernel variance, and $\mathbf{\Lambda} \in \mathbb{R}^{d \times d}$ is a symmetric positive definite matrix describing the kernel lengthscale. The kernel variance describes the magnitude of the function values of the

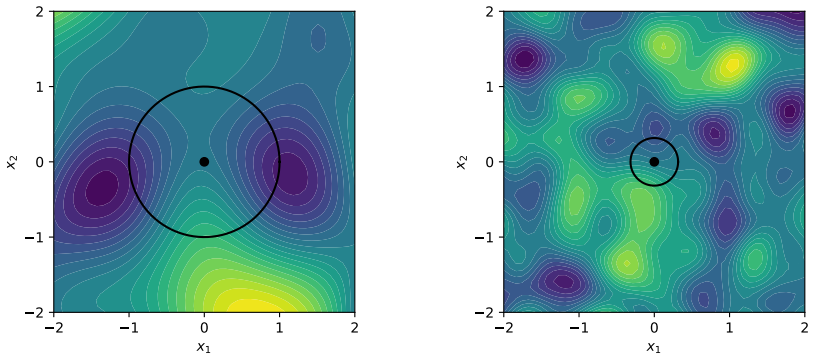
Gaussian process prior. Quite simply, if σ_0 is doubled, the vertical magnitude of the realizations in fig. 7 would double.

The kernel lengthscale $\mathbf{\Lambda}$ describes the rate at which the function is expected to change with respect to a change in input space. In the simplest case, if $\mathbf{\Lambda} = \ell^2 \mathbf{I}_d$ then the kernel is commonly called an isotropic kernel since it is simply a function of the radius $||\mathbf{x}_i - \mathbf{x}_j||_2$ from either of the input points. In this case, the parameter $\ell > 0$ describes how "sharp" the realizations will be (a smaller ℓ value gives sharper functions). As another interpretation, this prior states that you would not be able to extrapolate more than $\mathcal{O}(\ell)$ units from your data. Figures 8a and 8b plot a sample from a two-dimensional Gaussian process prior using the isotropic exponentiated quadratic kernel with two different values of ℓ. The lengthscale can also be visualized by the black curve in each plot where the radius of the black curve from the black dot indicates the lengthscale in that respective direction. Specifically, the black curves show a contour of equal prior covariance with the function value at the black dot. It can be seen that the function realization with the smaller ℓ in fig. 8b is more "wiggly" and changes value more rapidly with respect to the input coordinates.

If $\mathbf{\Lambda} = \text{diag}[\ell_1^2, \ldots, \ell_d^2]$ then the kernel has axis-aligned lengthscales and is commonly referred to as the ARD (automatic relevance determination) exponentiated quadratic kernel because it can prune irrelevant input dimensions by growing the corresponding lengthscales. Plotted in fig. 8c is a realization of a two-dimensional Gaussian process prior using an ARD exponentiated quadratic kernel where it can be seen that there are different lengthscales along the coordinate axes. Along the first input dimension x_1, the lengthscale ℓ_1 is large and the function values vary slowly. If we take $\ell_1 \to \infty$ the realization in fig. 8c would not vary at all along x_1 and the effect of this input dimension would be entirely eliminated.

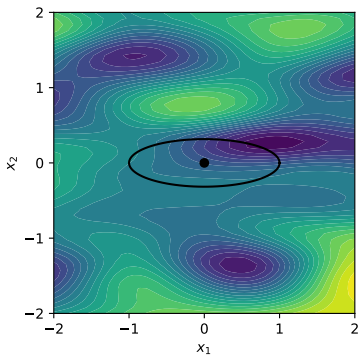
A dense $\mathbf{\Lambda}$ matrix can be seen as an application of the ARD exponentiated quadratic kernel applied after a rotation of the input space coordinate axes. Figure 8d plots a two-dimensional sample drawn from a GP prior with a dense $\mathbf{\Lambda}$ matrix. It can be seen that the black ellipse indicating the lengthscale in fig. 8d is rotated with respect to the ellipse in fig. 8c such that its principal axes are no longer aligned with the input coordinate axes. To apply lengthscales to the Matérn kernels in table 1, simply substitute $r = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{\Lambda}^{-1} (\mathbf{x}_i - \mathbf{x}_j)}$. This is simply a replacement of Euclidean distance between inputs with the Mahalanobis distance.

Neural Tangent Kernel While not necessarily a popular kernel, the neural tangent kernel is an interesting covariance function that demonstrates the power and generality of Gaussian processes. It was shown by Jacot et al. (2018) that deep, infinitely wide neural networks trained with gradient descent can be interpreted as Gaussian processes using the neural tangent kernel. This is a fascinating result which has led to many interesting and practical developments. For instance, it has allowed exact inference to be performed on deep and infinitely wide neural networks, even when complex architectures are considered such as convolutional neural networks with global average pooling (Arora et al., 2019).

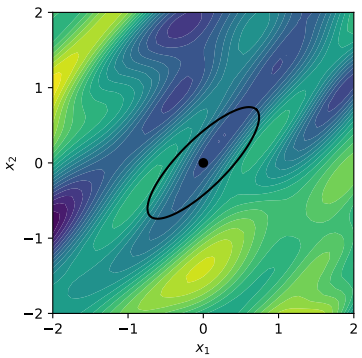


(a) $\mathbf{\Lambda} = \mathbf{I}_2$.

(b) $\mathbf{\Lambda} = 0.1\ \mathbf{I}_2$.



(c) $\mathbf{\Lambda} = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix}$.



(d) $\mathbf{\Lambda} = \begin{bmatrix} 0.55 & 0.45 \\ 0.45 & 0.55 \end{bmatrix}$.

Figure 8: Samples from a two-dimensional Gaussian process prior using the exponentiated quadratic kernel in eq. (28) with various values of $\mathbf{\Lambda}$. The black curves show a contour of equal prior covariance with the function value at the black dot. The radius of the black curve from the black dot indicates the lengthscale in that respective direction.

3 Model Selection & Model Evidence

Since the beginning of this document, we have assumed that a single model is selected *a priori* and then inference is performed. Unfortunately, in many scenarios, a practitioner may not have enough insight into a problem to specify a single good model for a learning problem. As an example, a practitioner may not know *a priori* which of two Gaussian process priors will perform better on a given learning problem. This section will discuss how to deal with uncertainty over candidate models.

We begin by posing the model selection problem more concretely. We assume that a model is defined by the vector $\boldsymbol{\theta}$ such that all candidate models can be determined by a specific value of $\boldsymbol{\theta}$. We refer to the vector $\boldsymbol{\theta}$ as a set of *hyperparameters*. As an example, the set of hyperparameters of a Gaussian process prior might include the kernel variance σ_0^2 and lengthscale $\mathbf{\Lambda}$ of the exponentiated quadratic kernel in eq. (28), as well as the training observation noise variance σ^2. The model selection problem simply involves selection of the models that perform best out of the candidates within the space of $\boldsymbol{\theta}$.

3.1 Model Evidence

The marginal likelihood or model evidence presented in eq. (6) is instrumental in Bayesian model selection. We will therefore begin by describing how the model evidence is computed for Gaussian processes, as discussed in section 2. To begin, we will update our notation such that $\Pr(\mathbf{y}|\boldsymbol{\theta})$ denotes the model evidence for the model with hyperparameters $\boldsymbol{\theta}$.

While evaluation of the model evidence is generally intractable for an arbitrary Bayesian model, in the case of a Gaussian process it can be performed analytically which is a tremendous advantage for the purposes of inference and model selection. Specifically, evaluating the marginal likelihood, of a Gaussian process simply involves evaluating the GP prior on the training dataset. By the definition of a Gaussian process, the GP prior evaluated on the training dataset is a Gaussian distribution (given in eq. (23)) and therefore evaluating the model evidence simply involves evaluating a multivariate Gaussian distribution with dimension n. To evaluate this multivariate Gaussian, there are two computational approaches that one may wish to take, a weight-space approach, and a function-space approach. While the two approaches are mathematically equivalent, the computational complexities differ in the same manner as the predictive posterior computations, whose computational complexities are summarized in eq. (26).

Weight-Space Approach From the weight-space approach of section 2.1, the log of the model evidence can be computed as

$$\log\Pr(\mathbf{y}|\boldsymbol{\theta}) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) + \frac{1}{2}\log\big(|\mathbf{S}|\big) + \frac{1}{2}\log\big(|\boldsymbol{\Sigma}|\big) - \frac{1}{2\sigma^2}\mathbf{y}^T\mathbf{y} + \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \tag{29}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are given in eq. (14). For computational reasons, the weight-space approach is generally preferred when $n > m$.

Function-Space Approach From the function-space approach of section 2.2, the log of the model evidence can be computed as

$$\log\Pr(\mathbf{y}|\boldsymbol{\theta}) = \underbrace{-\tfrac{1}{2}\log|\mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma^2\mathbf{I}_n|}_{\text{Complexity}} \underbrace{-\tfrac{1}{2}\mathbf{y}^T(\mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma^2\mathbf{I}_n)^{-1}\mathbf{y}}_{\text{Data Fit}} \underbrace{-\tfrac{n}{2}\log(2\pi)}_{\text{Normalization}}. \tag{30}$$

For computational reasons, the function-space approach is generally preferred when $n < m$. The terms in the preceding equations have been labelled for reference in later discussions.

3.2 Type-I Inference

We can now begin discussing how the model selection problem can be addressed by introducing a Bayesian approach that allows us to move from a prior over models to a posterior over models. This is effectively an inference procedure over hyperparameters, $\boldsymbol{\theta}$, rather than an inference procedure over parameters, $\mathbf{w}$ described in the earlier sections of this document. In this way, it is effectively

a meta-inference procedure. Applying Bayes' rule (eq. (5)) at the level of hyperparameters gives the posterior over $\boldsymbol{\theta}$ as follows

$$\Pr(\boldsymbol{\theta}|\mathbf{y}) = \frac{\Pr(\mathbf{y}|\boldsymbol{\theta})\ \Pr(\boldsymbol{\theta})}{\Pr(\boldsymbol{\theta})}. \tag{31}$$

$\Pr(\boldsymbol{\theta})$ is referred to as the *hyper-prior* and is a prior over models that reflects a practitioners prior belief about which model is best. $\Pr(\mathbf{y}|\boldsymbol{\theta})$ may be recognized as the marginal likelihood (eq. (6)) of a model with hyperparameters $\boldsymbol{\theta}$. Lastly, the marginal in the denominator can be computed as

$$\Pr(\boldsymbol{\theta}) = \int \Pr(\mathbf{y}|\boldsymbol{\theta}) \Pr(\boldsymbol{\theta}) d\boldsymbol{\theta}. \tag{32}$$

Figure 9 shows an example computation of the model selection posterior for two Gaussian process models where $\boldsymbol{\theta} = \boldsymbol{\Lambda}$, the lengthscale of the exponentiated quadratic kernel in eq. (28) for a $d = 1$ dimensional learning problem. For both models considered, the hyper-prior $\Pr(\boldsymbol{\Lambda})$ is equivalent (both are 0.5), and the posterior probability mass is given under each plot. Each model has a different interpretation of the data with the long lengthscale model in fig. 9a seeing a smooth curve with several outliers in the middle of the plot whereas the shorter lengthscale model in fig. 9b sees a large vertical wave in the middle of the plot. The shorter lengthscale model has greater posterior probability, however, both models have reasonable mass under the posterior indicating that there is still uncertainty about which model is preferred.

When making predictions, we would like to take into account our uncertainty over models. Applying the same rules of probability, we can write the predictive distribution over the function f^* at test input $\mathbf{x}^*$ as follows

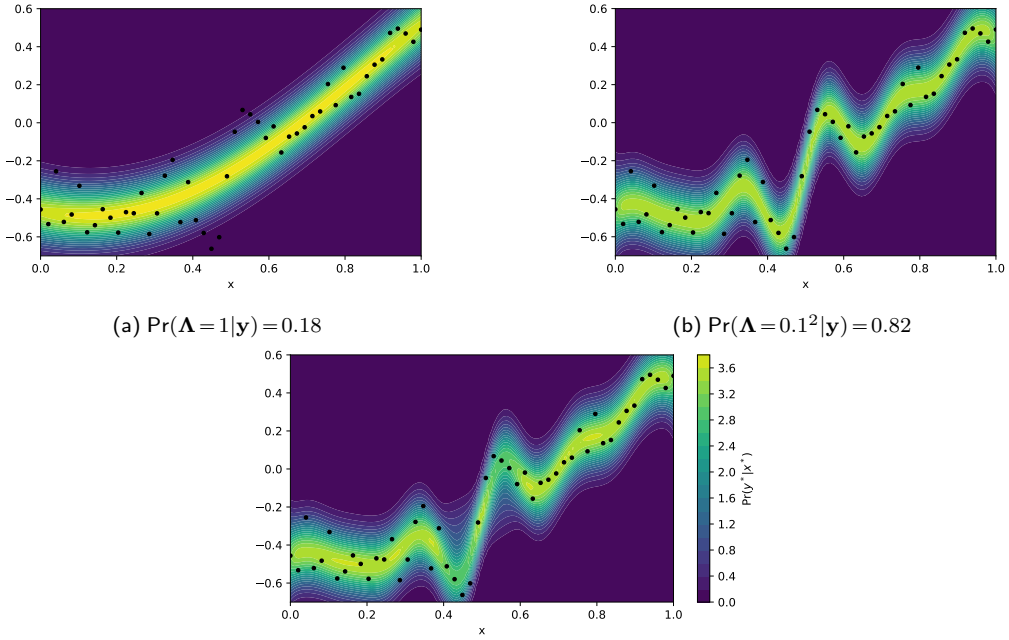
$$\Pr(f^*|\mathbf{y}) = \int \Pr(f^*|\mathbf{y}, \boldsymbol{\theta}) \Pr(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta}, \tag{33}$$

where $\Pr(f^*|\mathbf{y}, \boldsymbol{\theta})$ is the predictive posterior of a model with hyperparameters $\boldsymbol{\theta}$ that is given by eq. (7) in general (in the case of Gaussian processes, it is given by eq. (16) or eq. (25)). Ultimately, Bayesian inference is conducted at two levels simultaneously: inference over parameters and inference over models (hyperparameters). Returning to the example in fig. 9, the predictive posterior that accounts for model uncertainty is a sum of the two models weighted by the posterior over models, as follows

$$\Pr(f^*|\mathbf{y}) = 0.18\ \Pr(f^*|\mathbf{y}, \boldsymbol{\Lambda}=1) + 0.82\ \Pr(f^*|\mathbf{y}, \boldsymbol{\Lambda}=0.1^2),$$

where the predictive posteriors of each model are shown in figs. 9a and 9b, respectively, and the predictive posterior that accounts for model uncertainty is shown in fig. 9c. Note that this posterior is no longer Gaussian but is instead a mixture of Gaussians.

In general, the posterior in eq. (31) cannot be tractably computed in closed form and must be estimated numerically or approximated. A common approach for numerical estimation of the posterior can be achieved through the use of approximation or sampling techniques. The following section discusses a particular simplification of the Bayesian approach discussed here. To distinguish the two strategies, the Bayesian approach outlined here is commonly referred to as *type-I* inference. For interested readers, further details about the type-I inference procedure can be found in (Neal, 1995).



(a) $\Pr(\boldsymbol{\Lambda}=1|\mathbf{y})=0.18$

(b) $\Pr(\boldsymbol{\Lambda}=0.1^2|\mathbf{y})=0.82$

(c) Predictive posterior considering model uncertainty.

Figure 9: Demonstration of posterior computation over models (type-I inference). Both models use a Gaussian process prior given by an exponentiated quadratic covariance kernel in eq. (28) with $\sigma_0=1$ and with varying values of lengthscale $\boldsymbol{\Lambda}$. Also, training observations (black) are corrupted with *i.i.d.* Gaussian noise with variance $\sigma^2=0.1^2$ for both models. The background contours show the predictive posterior probability (the colours use the same scale for all plots). Note that the predictive posterior is plotting $\Pr(y^*|\mathbf{x}^*)$ rather than $\Pr(f^*|\mathbf{x}^*)$ which takes into account the independent Gaussian noise corrupting the input data. The hyper-prior over both models are equivalent such that $\Pr(\boldsymbol{\Lambda}=1)=\Pr(\boldsymbol{\Lambda}=0.1^2)=0.5$.

3.3 Type-II Empirical Bayes

This section discusses a simplification of the type-I Bayesian model selection problem outlined in the previous section. The simplified approach is commonly referred to as *type-II* inference, or *empirical Bayes*. Quite simply, in a type-II inference approach a single model is selected the maximizes the model evidence. This can be written as follows

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \Pr\big(\mathbf{y}\big|\boldsymbol{\theta}\big), \tag{34}$$

where $\boldsymbol{\theta}^*$ describes the selected model. Under certain conditions, the posterior eq. (31) will be tightly peaked around $\boldsymbol{\theta}^*$ and type-II inference can be seen as an approximation of type-I inference. In the presence of abundant data and relatively few hyperparameters, this approximation can be quite good[4].

The model evidence is an attractive objective for hyperparameter estimation since it naturally balances model flexibility with the models' ability to fit the dataset, admitting a Bayesian

[4]Note that in contrast, making this same maximum likelihood approximation to the posterior $\Pr(\mathbf{w}|\mathbf{y})$ over the parameters $\mathbf{w}$ will often give very poor results since the number of parameters m is typically very large and possibly infinite.

interpretation of Occam's razor. This trade-off can be seen by observing the terms in eq. (30), each of which have a clear interpretation. The term labelled "complexity" depends only on the data inputs $\mathbf{X}$ and penalizes high model flexibility[5]. The term labelled "data fit" is the only term containing the training responses $\mathbf{y}$ and reflects how well the responses are modelled by the marginal Gaussian distribution. The final term is simply a normalization constant and depends on nether the training set or the hyperparameters. Gradient-based optimization is commonly used to maximize the model evidence when the hyperparameters $\boldsymbol{\theta}$ are continuous but it should be considered that the model evidence may have multiple maxima in $\boldsymbol{\theta}$.

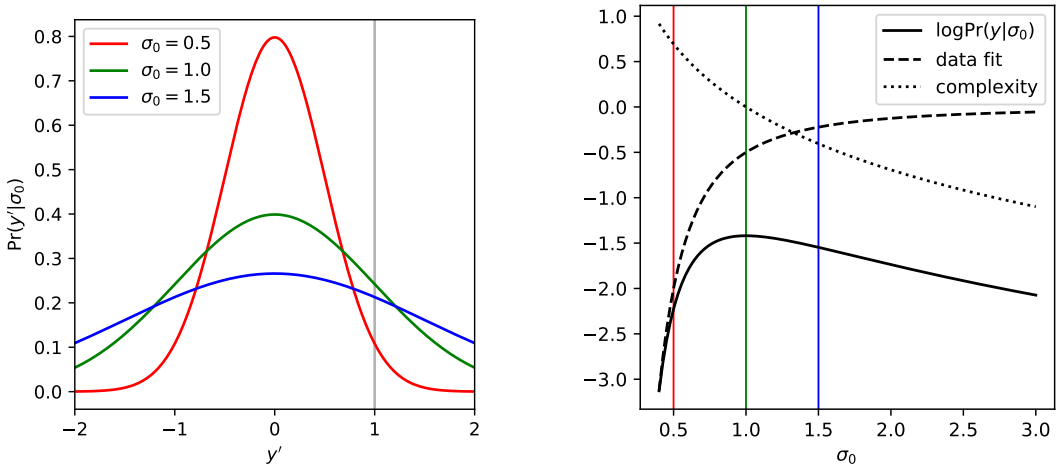
We can illustrate the Bayesian interpretation of Occam's razor through a simple example of selecting a Gaussian process prior on a dataset with a single training instance. Specifically, we will consider the exponentiated quadratic kernel in eq. (28) with various values of kernel variance σ_0, and we will assume that the single training observation $y = 1$ is noise-free (such that $f = y$). Figure 10a plots the model evidence $\text{Pr}(y'|\sigma_0)$ for three different values of σ_0 across a range of possible observation values y'. It is easily seen that of the three curves, the GP prior given by $\sigma_0 = 1$ provides the maximal model evidence at the true observation value of $y' = y = 1$ indicated by the grey vertical line. Figure 10b shows the breakdown of the log-evidence at the true training observation, $\log \text{Pr}(y|\sigma_0)$, based upon the decomposition given in eq. (30). It can be seen that the choice of $\sigma_0 = 1$ is a trade-off between data fit and complexity since of the three σ_0 choices, $\sigma_0 = 1.5$ provides the best data fit, $\sigma_0 = 0.5$ is the least complex, and $\sigma_0 = 1$ is in between on both data-fit and complexity. To understand what complexity means, observe in fig. 10a that the most complicated prior with $\sigma_0 = 1.5$ has the ability to represent a far greater range of possible observation values y' than the simplest model $\sigma_0 = 0.5$.

While empirical Bayes (evidence maximization) can be an effective means of model selection in many instances, it should be used cautiously since it can suffer from several of potential issues:

- **Type-II inference underestimates uncertainty.** This is not surprising since type-II inference is effectively ignoring uncertainty over models whereas a type-I approach takes this uncertainty into account. This effect can be seen in the example of fig. 9 where a shorter lengthscale of $\boldsymbol{\Lambda} = 0.1^2$ would have been selected out of the two options considered from type-II approach. The type-II predictive posterior is therefore shown in fig. 9b where it is clear that the predictive posterior underestimates uncertainty around $x = 0.5$ relative to the type-I predictive posterior shown in fig. 9c.

- **Type-II inference is not immune from overfitting.** Cases where many hyperparameters are being estimated by evidence maximization are liable to overfit, for instance. As an example, in fig. 10a if we were estimating the GP prior mean in addition to the variance σ_0, the maximum evidence would occur at a delta spike about $y = 1$, i.e. $\text{Pr}(y'|\boldsymbol{\theta}) = \delta(y' - 1)$. This pathological GP prior which would give infinite evidence but would clearly be a silly model to use given a single training observation.

[5] Model flexibility may be difficult to envision for a non-parametric model. Specifically, the complexity term penalizes a slowly decaying eigenspectrum of the covariance matrix which generally occurs with smaller kernel lengthscales. Therefore, a kernel with a large lengthscale (admitting smoother functions) is favoured over a kernel with a small lengthscale (i.e. admitting sharper functions) under this penalty. Also influencing the complexity is the scale of the eigenvalues which relates the kernel variance and noise variance (smaller variances are favoured).



(a) Evidence plotted over a range of possible observation values y' for three values of hyperparameter σ_0.

(b) Model evidence verses hyperparameter σ_0. The breakdown refers to the terms in eq. (30).

Figure 10: Comparison of three Gaussian process priors on a dataset with the single observation $y = 1$. The plots illustrate a Bayesian interpretation of Occam's razor for model selection by maximization of model evidence.

In general, empirical Bayes is safe from these concerns when the number of hyperparameters is far less than then number of training observations (i.e. when $|\boldsymbol{\theta}| \ll n$).