Approximate Hyperparameter Marginalisation for Gaussian Processes

Anonymous Author(s)
Affiliation
Address
email

Abstract

Kernel and covariance functions are often parameterised by hyperparameters, unknown *a-priori*. We propose a novel means of approximately integrating over these hyperparameters. This approach gives rise to a novel class of covariance functions which are robust to hyperparameter uncertainty, avoiding the over-fitting that often accompanies type-II maximum-likelihood fitting of the hyperparameters. We demonstrate the efficacy of our covariances for Gaussian process regression on both synthetic and real data.

1 Introduction

Kernels and covariances are vital to much of Machine Learning.

The hyperparameters of such functions are rarely known *a-priori*. This is particularly important when we have small amounts of data, important for applications where obtaining data is expensive (computationally or otherwise).

Hyperparameters are often set by type-II maximum likelihood. Monte Carlo is too expensive. We introduce a Laplace approximation. This gives us a robust family of covariances that are tolerant to uncertainty in the hyperparameters.

2 Gaussian Processes

Gaussian processes (GPs) constitute a powerful method for performing Bayesian inference about functions using a limited set of observations [3]. A GP is defined as a distribution over the functions $f : \mathcal{X} \to \mathbb{R}$ such that the distribution over the possible function values on any finite set of $\mathcal{X}$ is multi-variate Gaussian.

A GP is completely defined by its first and second moments: a mean function $\mu : \mathcal{X} \to \mathbb{R}$, which describes the overall trend of the function, and a positive semidefinite covariance function, or kernel, $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which describes how function values are correlated as a function of their locations in the domain. Given a function $f : \mathcal{X} \to \mathbb{R}$ about which we wish to perform inference and a set of input points $\mathbf{x} \subseteq \mathcal{X}$, the Gaussian process prior distribution over the function values $\mathbf{f} = f(\mathbf{x})$ is given by:

$$\begin{aligned} p(\mathbf{f}|\mathbf{x}, \boldsymbol{\theta}, I) &:= \mathcal{N}\left(\mathbf{f}; \mu_\theta(\mathbf{x}), K_\theta(\mathbf{x}, \mathbf{x})\right) \\ &:= \frac{1}{\sqrt{\det 2\pi K_\mathbf{f}}} \exp\left(-\frac{1}{2}(\mathbf{f} - \mu_\mathbf{f})^\top K_\mathbf{f}^{-1}(\mathbf{f} - \mu_\mathbf{f})\right) \end{aligned} \tag{1}$$

where $\boldsymbol{\theta}$ is a vector containing any parameters required by μ and K: the *hyperparameters* of the model, I. Due to the ubiquity of I we henceforth drop it from explicit representation for notational

convenience. There exist a wide variety of mean and covariance functions which can be chosen in order to reflect any prior knowledge available about the function of interest.

A vector of observations $\mathbf{y} = \{y_1, ..., y_n\}$ could be viewed as a single point sampled from a n-variate Gaussian distribution. Once we have observations of the function $(\mathbf{x_s}, \mathbf{y})$ we can make predictions about the function value f_* at input x_*. As exact measurements of the function are often not available we assume a noise model, such that:

$$p(y|f, x, \sigma_n^2) := \mathcal{N}(y; f, \sigma_n^2) \tag{2}$$

which represents i.i.d Gaussian observation noise with variance σ_n^2. As should be expected, the predictive distribution over f_* is Gaussian:

$$p(f_*|x_*, \mathbf{y}, \boldsymbol{\theta}) := \mathcal{N}(f_*; m_\theta(f_*|x_*, \mathbf{y}), C_\theta(f_*|x_*, \mathbf{y})) \tag{3}$$

where the posterior mean and covariance are:

$$m_\theta(f_*|x_*, \mathbf{y}) := \mu_\theta(x_*) + K_\theta(x_*, \mathbf{x})V^{-1}(\mathbf{y} - \mu_\theta(x_*)) \tag{4}$$

$$C_\theta(f_*|x_*, \mathbf{y}) := K_\theta(x_*, x_*) - K_\theta(x_*, \mathbf{x})V^{-1}K_\theta(\mathbf{x}, x_*) \tag{5}$$

where $V := K_\theta(\mathbf{x}, \mathbf{x}) + \sigma_n^2\mathbf{I}$.

2.1 Marginalising out the Hyperparameters

The previous equations assume that the hyperparameters $\boldsymbol{\theta}$ are known; in fact we can rarely be certain about $\boldsymbol{\theta}$ *a priori*. This ignorance can be represented by a suitably uninformative prior distribution $p(\boldsymbol{\theta})$. Given such a *hyper-prior*, the hyperparameters can be marginalised to calculate the predictive distribution over f_*:

$$p(f_*|x_*, \mathbf{y}) = \frac{\int p(f_*|x_*, \mathbf{y}, \boldsymbol{\theta})p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}{\int p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}} \tag{6}$$

Unfortunately, such integrals are generally non-analytic, requiring numerical approximation. Randomized Monte Carlo techniques [1] form the most popular approaches to numerical integration, although Bayesian alternatives [4, 2] also exist. These techniques estimate the integral given the value of the integrand on a set of sample points, usually via a weighted mixture

$$p(f_*|x_*, \mathbf{y}) \simeq \sum_i \rho_i p(f_*|x_*, \mathbf{y}, \boldsymbol{\theta}_i) \tag{7}$$

for some weight vector $\boldsymbol{\rho}$. Unfortunately, the computational expense of evaluating the integrand at sufficient samples to estimate high-dimensional integrals is often prohibitive.

A less computationally demanding alternative is to select only a single sample. Type II maximum likelihood, or maximum marginal likelihood, approximates as

$$p(f_*|x_*, \mathbf{y}) \simeq p(f_*|x_*, \mathbf{y}, \boldsymbol{\theta}_{\mathrm{ML}}) \tag{8}$$

where $\boldsymbol{\theta}_{\mathrm{ML}}$ is the hyperparameter vector that maximises the marginal likelihood,

$$\boldsymbol{\theta}_{\mathrm{ML}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \; p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) \tag{9}$$

As per Figure ****, (8) is equivalent to approximating the likelihood $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ as the delta function $\delta(\theta - \theta_{\mathrm{ML}})$. This assumption is a poor representation of our true state of ignorance given only low numbers of data, for which the likelihood is typically broad and/or multi-modal. As a consequence, type II maximum likelihood can often lead to over-fitting. Nonetheless, the less onerous computational requirements of this approach have made it ubiquitous throughout machine learning.

We now present a novel means of marginalising hyperparam

3 Approximate Hyperparameter Marginalisation

In this section we introduce an approximate method for performing inference, using a Gaussian process, which accommodates the inherent uncertainty one should have about the value of hyperparameters, but which does not suffer from excessive computational expense. Rather than attempt to

approximate the integral in (6), we present a novel approach which uses a modified prior that incorporates and marginalises the uncertainty in the hyperparameters *before* it is fused with observations. The result is an adjusted covariance function which can be used interchangeably with a standard covariance function in GP inference, but whose parameters are now *hyper-hyperparameters* explaining the marginalised covariance function.

3.1 Homogenous Exponential Covariance Functions

Mike says: this section defines the general class of covariance functions our method is appropriate to, which we can call Homogenous Exponential covariance functions

As is the case in standard GP inference, a prior covariance function must be selected which is representative of any prior belief about the function of interest. In order to develop our method we first look at one of the most pervasive covariance functions used in Bayesian inference: the *Squared Exponential*, or *Gaussian*, kernel:

$$K_\theta(x, x') = h^2 \exp\left(-\frac{1}{2}\frac{\triangle^2}{L^2}\right) \tag{10}$$

where $\triangle^2 = (x - x')^2$, and h is a scaling parameter termed the output scale. The parameter L, the *length scale*, affects how closely outputs co-vary as a function of $\triangle^2$. The length parameter is perhaps the most crucial hyperparameter to train correctly, and is where we begin our exposition.

Mike says: We can also say that marginalising the output scale can be done analytically with an inverse-gamma prior on the squared output scale, but then we would no longer have a GP, which would kind of defeat the point. Alternatively, you can use our approach to approximately marginalise the output scale h by resolving $inth^2 K(x, x')p(h)dh$ where $K(x, x')$ is a normalised cov fn. Taking a Gamma prior for h^-2, rather than a Gaussian for $logh$, is a sensible thing to do here as, with reference to the previous plots, we'd rather favour infinite output scale over zero output scale as we get more and more uncertain. That is, if we have no idea of the output scale at all, it seems unreasonable for us to expect most of the data to be within epsilon of the prior mean. With a Gamma prior for h^-2 with mean mu_h and shape parameter a_h, our integral resolves to $a_h Gamma(-1 + a_h)/(mu_h Gamma(a_h))K(x, x')$ where Gamma is the Gamma function. So basically we just replace the squared output scale of our covariance function with $a_h Gamma(-1 + a_h)/(mu_h Gamma(a_h))$ and we're done. This trick has been done by previous authors- I can chase up references.

Jan says: COULD WE NOT BE MORE GENERAL SAYING: We assume our GP prior is endowed with a kernel function of the form

$$K_\theta(x, x') = h^2 \exp\left(-\frac{1}{2}\frac{d(x, x')}{L^2}\right) \tag{11}$$

where d is an arbitrary function. Typically, d will be a distance metric. For instance, choosing $d(x, x') = xMx'$ for a pos. def. matrix M yields the squared exponential kernel. Or, $d(x, x') = |x - x'|$ gives rise to the Ornstein-Uhlenbeck kernel. ... this should not affect our derivations below..

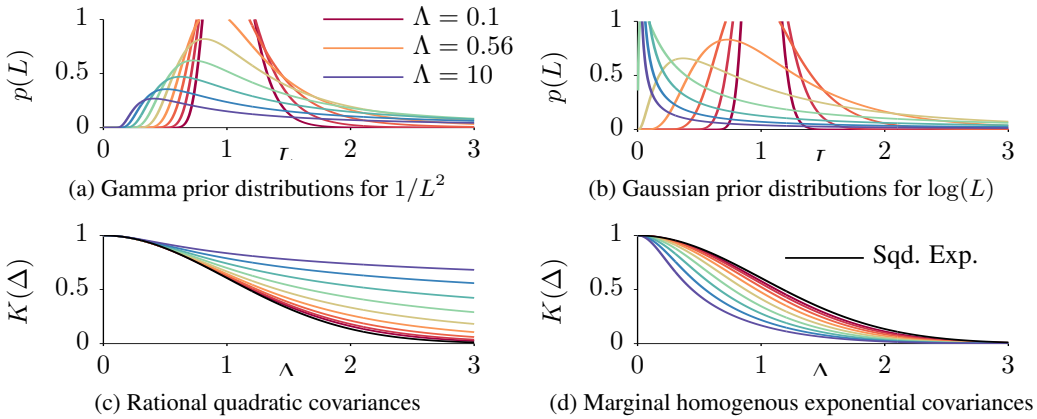
3.2 Rational Quadratic Covariance Function

The Rational Quadratic covariance funciton represents one means of approximately marginalising the input scale of a squared exponential covariance. It assumes a gamma prior, which, problematically, puts excessive weight on very lagre input scales when we are very uncertain. See Figure 1.

3.3 Marginal Homogenous Exponential Covariance Function

Mike says: now we talk about our covariance, which I've called Marginal Homogenous Exponential covariances

The standard GP prior assumes that the value of L is deterministic; any distribution an individual has over L is purely the epistemic characterisation of their uncertainty in the 'true' value of L. We



(a) Gamma prior distributions for $1/L^2$

(b) Gaussian prior distributions for $\log(L)$

(c) Rational quadratic covariances

(d) Marginal homogenous exponential covariances

Figure 1: All figures have $\nu = 1$ and share the legends of a) and d).

alter the prior such that we make L a random variable with its own generative distribution $p(L)$. The value of L should be strictly positive, hence we make the substitution $\beta := \log(L)$ and define a Gaussian distribution over β so that $p(L)$ will have strictly positive support:

$$K_\theta(x, x') = h^2 \exp\left(-\frac{1}{2} \triangle^2 \exp(-2\beta)\right) \tag{12}$$

which we will write as K_β, and

$$\begin{aligned} p(\beta|\nu, \Lambda) &= \mathcal{N}(\beta; \nu, \Lambda) \\ &= \frac{1}{\sqrt{2\pi\Lambda}} \exp\left(\frac{-(\beta - \nu)^2}{2\Lambda}\right) \end{aligned} \tag{13}$$

where ν and Λ are hyper-hyperparmeters.

As stated before, we marginalise out the uncertainty in β (or L), before fusing the prior with observations.

Jan says: Perhaps we should talk about the moment matching idea first.. As far as I can remember: Ie taylor expand the marginalized process ... it then turns out the the kernel of the matched process equals the marginalized kernel you write down in (14) below..

The marginalised kernel is calculated by evaluating the following integral:

$$K_{\nu,\Lambda} = \int_{-\infty}^{+\infty} K_\beta \; p(\beta|\nu, \Lambda) d\beta \tag{14}$$

which is non-analytic given the current form of K_β. We approximate the solution to the integral by first approximating K_β using a second-order Taylor expansion around the mean of $p(\beta)$ given by ν. In order to ensure that the approximate kernel is positive semi-definite we force K_β to be in exponential form, hence we note the following identity and substitution:

$$\begin{aligned} K_\beta &= \exp\left(-\ln \frac{1}{K_\beta}\right) \\ &= \exp\left(-A\right) \end{aligned} \tag{15}$$

and we Taylor expand around A:

$$A \approx \ln \frac{1}{K_\beta}\bigg|_\nu + (\beta - \nu)\frac{\partial A}{\partial \beta}\bigg|_\nu + \frac{1}{2}(\beta - \nu)^2 \frac{\partial^2 A}{\partial^2 \beta}\bigg|_\nu \tag{16}$$

Given (12), the derivatives of K_β can be calculated:

$$\begin{aligned} K'_\beta &= K_\beta \left(\triangle^2 \exp(-2\beta)\right) \\ K''_\beta &= K_\beta \left(\triangle^4 \exp(-4\beta) - \triangle^2 \exp(-2\beta)\right) \end{aligned}$$

which can be used for calculating the partial derivatives of (16):

$$A = \ln \frac{1}{K_\beta} \tag{17}$$

$$\frac{\partial A}{\partial \beta} = -\frac{K'_\beta}{K_\beta} = -\triangle^2 \exp(-2\beta) \tag{18}$$

$$\frac{\partial^2 A}{\partial^2 \beta} = -\frac{K''_\beta}{K_\beta} + \frac{{K'_\beta}^2}{K_\beta^2} = 2\triangle^2 \exp(-2\beta) \tag{19}$$

Using (15) and substituting (17-19) into (16) we arrive at the following expression for K_β:

$$\begin{aligned} K_\beta &\approx \exp\left(-\left(\ln \frac{1}{K_\nu} - (\beta-\nu)C + (\beta-\nu)^2 C\right)\right) \\ &= K_\nu \exp\left((\beta-\nu)C - (\beta-\nu)^2 C\right) \end{aligned} \tag{20}$$

where $C = \triangle^2 \exp(-2\nu)$

All that remains is to evaluate the integral in (14), which by using (13) and (20) is as follows:

$$\begin{aligned} K_{\nu,\Lambda} &\approx K_\nu \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi\Lambda}} \exp\left(\frac{-(\beta-\nu)^2}{2\Lambda}\right)\left((\beta-\nu)C - (\beta-\nu)^2 C\right) \\ &= K_\nu \exp\left(\frac{\Lambda C^2}{2(1+2\Lambda C)}\right)\frac{1}{\sqrt{1+2\Lambda C}} \end{aligned} \tag{21}$$

This result is a new kernel which is the product of the original kernel K_β evaluated at $\beta = \nu$, and an expression which is a function of the squared distance $\triangle^2$.

3.3.1 Proof of Positive Semi-Definiteness

In order to confirm that our new kernel $K_{\nu,\Lambda}$ is a legitimate covariance function, we must prove that it fulfils the necessary condition of kernels which is positive semi-definiteness. We first note that a sum of kernels is also a kernel, therefore:

$$K_{\nu,\Lambda} = \int K_\beta \, p(\beta) d\beta \tag{22}$$

should be a legitimate kernel if K_β is also a legitimate kernel as $p(\beta)$ just weights the contents of the integral. Using the expression for K_β in (20) and completing the square we arrive at the following expression:

$$K_\beta = K_\nu \exp\left(-\frac{1}{2}\left(\beta - \nu - \frac{1}{2}\right)^2 C\right)\exp\left(\frac{1}{4}C\right) \tag{23}$$

If we substitute in K_ν, where:

$$\begin{aligned} K_\nu &= h^2 \exp\left(-\frac{1}{2}\triangle^2 \exp(-2\nu)\right) \\ &= h^2 \exp\left(-\frac{C}{2}\right) \end{aligned} \tag{24}$$

we can write:

$$K_\beta = h^2 \exp\left(-\frac{C}{4}\right) \exp\left(-\frac{1}{2}\left(\beta - \nu - \frac{1}{2}\right)^2 C\right) \tag{25}$$

We note that the product of kernels is also a kernel. Both exponentials in (25) form legitimate kernels, which leaves us with the result that K_β is also a kernel and by definition is positive semi-definite as required.

4 Experiments

4.1 Posterior functions with differing length scales

When using standard kernels, functions drawn from the prior will have a fixed length scale; when using our kernel, the length scale of functions drawn from the prior can take a number a range of values as a result of the distribution over length scales. When our prior is fused with observations to form a posterior GP, we are able to describe functions that may have generated the observations over a range of length scales. When data is limited this is particularly crucial as a whole range of length scales may be permissible with regards to the data generated, even if one is favoured through type-II maximum likelihood methods. As data increases the range of permissible length scales reduces, eventually collapsing to the true length scale.

We demonstrate the above behaviour and its advantage over other methods in this section. We begin by drawing a function from a prior GP with known length scale. From this we can generate a training set of between X and X observations in the domain $\mathcal{X} \in [-3, 3]$. We train the hyperparameters or hyper-hyperparameters of our test methods using type-II maximum likelihood on the given training set. Now we go back to the original GP and draw a set of X functions from the posterior (so that the functions still go through the original training data) with length scales ranging from X to X. An example training set and resulting set of posterior functions can be seen in figure XXXX. A test set with 100 observations is generated from each function and a log likelihood is calculated for each method. This process is repeated with 100 different starting functions and training data.

We plot the mean and variance trials of the likelihood of the test data across all $X * 100$ against the log length scale of the posterior function.

4.2 Real world dataset

5 Conclusion

References

[1] M.H. Chen, Q.M. Shao, and J.G. Ibrahim. *Monte Carlo methods in Bayesian computation*. Springer Verlag, 2000.

[2] M.A. Osborne, R. Garnett, S.J. Roberts, C. Hart, S. Aigrain, N.P. Gibson, and S. Aigrain. Bayesian quadrature for ratios. *Fifteenth International Conference on Artificial Intelligence and Statistics*, 2012.

[3] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press Cambridge, MA, USA, 2006.

[4] C.E. Rasmussen and Z. Ghahramani. Bayesian monte carlo. *Advances in neural information processing systems*, 15:489–496, 2003.