Gaussian Processes with Probabilistic Programming

Anonymous Author(s)
Affiliation
Address
email

Abstract

Abstract

1 Introduction

MCMC lends itself to Bayesian interpretations of Gaussian Processes since they can provide a vehicle to express otherwise intractable integrals necessary for a fully Bayesian representation.

2 Gaussian Processes

In the following, we will introduce GP related theory and notations. We will exclusively work on two variable regression problems. Let the data be real-valued scalars $\{x_i, y_i\}_{i=1}^n$ (complete data will be denoted by column vectors $\mathbf{x}$, $\mathbf{y}$). GPs present a non-parametric way to express prior knowledge on the space of possible functions f that we assume to have generated the data. f is assumed latent and the GP prior is given by a multivariate Gaussian with mean and covariance $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, where $m(\mathbf{x})$ is a function of the mean of all functions that map to y_i at x_i and $k(\mathbf{x}, \mathbf{x}')$ is a kernel or covariance function that summarizes the covariance of all functions that map to y_i at x_i. We can absorb the mean function into the covariance function so without loss of generality we can set the mean to zero. The marginal likelihood can be expressed as:

$$p(\mathbf{y}|\mathbf{x}) = \int p(\mathbf{y}|\mathbf{f}, \mathbf{x})\, p(\mathbf{f}|\mathbf{x})\, d\mathbf{f} \tag{1}$$

where the prior is Gaussian $\mathbf{f}|\mathbf{x} \sim \mathcal{N}\big(0, k(\mathbf{x}, \mathbf{x}')\big)$. For a zero mean Gaussian Process this results in a Gaussian posterior $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean:

$$\boldsymbol{\mu} = \mathbf{K}(\mathbf{x}, \mathbf{x}^*)\, \mathbf{K}(\mathbf{x}^*, \mathbf{x}^*)^{-1}\, \mathbf{y} \tag{2}$$

and covariance

$$\boldsymbol{\Sigma} = \mathbf{K}(\mathbf{x}, \mathbf{x}) + \mathbf{K}(\mathbf{x}, \mathbf{x}^*)\mathbf{K}(\mathbf{x}^*, \mathbf{x}^*)^{-1}\mathbf{K}(\mathbf{x}^*, \mathbf{x}). \tag{3}$$

where $\mathbf{K}$ is a covariance function. The covariance function covers general high-level properties of the observed data such as linearity, periodicity and smoothness. The most widely used type of covariance function is the squared exponential covariance function:

$$k(x, x') = \sigma^2 \exp(-\frac{(x - x')^2}{2\ell^2}) \tag{4}$$

where σ and ℓ are hyper-parameters. σ is a scaling factor and ℓ is the typical length-scale. Smaller variations can be achieved by exchanging these hyper-parameters. Below, we see how we can express simple GP smoothing with a few

lines of Venture code while allowing users to custom design covariance functions.

Listing 1: GP Smoothing

```
[ASSUME l  1] ∈ {hyper-parameters}
[ASSUME sf 2] ∈ {hyper-parameters}

k(x, x') := σ^2 exp(-(x-x')^2 / 2ℓ^2)

[ASSUME f VentureFunction(k, σ, ℓ) ]
[ASSUME SE make-se (apply-function f l sf) ]
[ASSUME (make-gp 0 SE) ]

[SAMPLE GP (array 1 2 3)] % Prior

[OBSERVE GP D]

[SAMPLE GP (array 1 2 3)]

[INFER  (MH {hyper-parameters} one 100) ]

[SAMPLE GP (array 1 2 3)] % Posterior
```

The first two lines depict the hyper-parameters. We tag both of them to belong to the set {hyper-parameters}. Every member of this set belongs to the same inference scope. This scope controls the application of the inference procedure used. In this paper, we use MH throughout. Each scope is further subdivided into blocks that allow to do block-proposals. In the following we omit the block notation for readability, since we always choose the block of a certain scope at random.

The ASSUME directives describe the assumptions we make for the GP model, we assume the hyper-parameters l and sf (corresponding to ℓ, σ) to be 1 and 2. The squared exponential covariance function can be defined outside the Venture code with foreign conventional programming languages, e.g. Python. In that way, the user can define custom covariance functions without being restricted to the most common ones. We then integrate the foreign function into Venture as VentureFunction. In the next line this function is associated with the hyper-parameters. Finally, we assume a Gaussian Process SP with a zero mean and the previously assumed squared exponential covariance function.

In the case where hyper-parameters are unknown they can be found deterministically by optimizing the marginal likelihood using a gradient based optimizer. Non-deterministic, Bayesian representations of this case are also known (Neal, 1997). Extending the program described in listing 1 for a Bayesian treatment of hyper-parameters is simple using the build in stochastic procedure that simulates drawing samples from a gamma distribution:

Listing 2: Bayesian GP Smoothing

```
[ASSUME l (gamma 1 3)]
[ASSUME sf (gamma 1 2)]

k(x, x') := σ^2 exp(-(x-x')^2 / 2ℓ^2)

[ASSUME f VentureFunction(k, σ, ℓ) ]
[ASSUME SE make-se (apply-function f l sf) ]
[ASSUME (make-gp 0 SE ) ]
```

Larger variations are achieved by changing the type of the covariance function structure. A different type could be a linear covariance function:

$$k(x, x') = \sigma^2 (x - \ell)(x' - \ell). \tag{5}$$

Note that covariance function structures are compositional. We can add covariance functions if we want to model globally valid structures

$$k_3(x, x') = k_1(x, x') + k_2(x, x') \tag{6}$$

and we can multiply covariance functions if the data is best explained by local structure

$$k_4(x, x') = k_1(x, x') \times k_2(x, x'); \tag{7}$$

both, k_3 and k_4 are valid covariance function structures. This leads to an infinite space of possible structures that could potentially explain the observed data best (e.g. Fig. **??**). In the following, we will refer to covariance functions that are not composite as base covariance functions. Note that this form of composition can be easily expressed in Venture, for example if one wishes to add a linear and a periodic kernel:

Listing 3: LIN $\times$ PER

```
[ASSUME l (gamma 1 3)]
[ASSUME sf (gamma 1 2)]
[ASSUME a (gamma 2 2)]

k_LIN(x, x') = σ_1^2 (x − ℓ)(x' − ℓ)

k_PER(x, x') := σ_2^2 exp(− 2 sin^2(π(x−x')/p) / ℓ^2 )

[ASSUME f_LIN VentureFunction(k_LIN, σ_1) ]
[ASSUME f_PER VentureFunction(k_PER, σ_2, ℓ, p) ]
[ASSUME LIN (make−LIN (apply−function f_LIN a)) ]
[ASSUME PER (make−PER (apply−function f_PER l sf)) ]
[ASSUME (make−gp 0 (function−times LIN PER)) ]
```

Knowledge about the composite nature of covariance functions is not new, however, until recently, the choice and the composition of covariance functions were done ad-hoc. The Automated Statistician Project came up with an approximate search over the possible space of kernel structures (Duvenaud et al., 2013; Lloyd et al., 2014).

2.1 A Bayesian interpretation

In the following, we will explore a Bayesian representation of GP. The probability of the hyper-parameters of a GP with assumptions as above and given covariance function structure $\mathbf{K}$ can be described as:

$$P(\boldsymbol{\theta} \mid \mathbf{D}, \mathbf{K}) = \frac{P(\mathbf{D} \mid \boldsymbol{\theta}, \mathbf{K}) P(\boldsymbol{\theta} \mid \mathbf{K})}{P(\mathbf{D} \mid \mathbf{K})}. \tag{8}$$

We are interested in the case where covariance structure is not given. Our probabilistic programming based MCMC framework approximates the following intractable integrals of the expectation for the prediction:

$$\mathbb{E}[y^* \mid x^*, D, \mathbf{K}_\Omega^s] = \iint f(x^*, \boldsymbol{\theta}, \mathbf{K})\, P(\boldsymbol{\theta} \mid \mathbf{D}, \mathbf{K})\, P(\mathbf{K}|\boldsymbol{\Omega}, s, n)\, \mathrm{d}\boldsymbol{\theta}\mathrm{d}\mathbf{K}. \tag{9}$$

This is done by sampling from the posterior probability distribution of the hyper-parameters and the possible kernel:

$$y^* \approx \frac{1}{T} \sum_{t=1}^{T} f(x^* | \boldsymbol{\theta}^{(t)}, \mathbf{K}^{(t)}). \tag{10}$$

3 Stochastic Processes

In order to provide the sampling of the kernel, we introduce a stochastic process to the SP that simulates the grammar for algebraic expressions of kernel algebra. Here, we start with a set of

possible kernels and draw a random subset. For this subset of size n, we sample a set of possible operators that operate on the base kernels.

The marginal probability of a kernel structure which allows us to sample is characterized by the probability of a uniformly chosen subset of the set of n possible covariance functions times the probability of sampling a global or a local structure which is given by a binomial distribution:

$$P(\mathbf{K} \mid \mathbf{\Omega}, s, n) = P(\mathbf{\Omega} \mid s, n) \times P(s \mid n) \times P(n), \tag{11}$$

with

$$P(\mathbf{\Omega} \mid s, n) = \binom{n}{r} p_{+\times}^{k} (1 - p_{+\times})^{n-k} \tag{12}$$

and

$$P(s \mid n) = \frac{n!}{\mid s \mid!} \tag{13}$$

where $P(n)$ is a prior on the number of base kernels used. It is possible to also assign a prior for the probability to sample global or local priors, however, we have assigned complete uncertainty to this with the binomial $p = 0.5$.

4 Experiments

Neal suggested the treatment of outliers as a use-case for a Bayesian treatment of Gaussian processes (1997). He evaluates his MCMC setting using the following synthetic data problem. Let f be the underlying function that generates the data:

$$f(x) = 0.3 + 0.4x + 0.5\sin(2.7x) + \frac{1.1}{(1 + x^2)} + \eta \quad with \;\; \eta \sim \mathcal{N}(0, \sigma) \tag{14}$$

We synthetically generate outliers by setting $\sigma = 0.1$ in 95% of the case and to $\sigma = 1$ in the remaining cases. Venture GPs can capture the true underlying function within only 100 MH steps (see Fig. 2). Note that Neal devices an additional noise model and performs large numbe of Hybrid-Monte Carlo and Gibbs steps.

4.1 Structure Learning

Many equivalent covariance structures can be sampled due to covariance function algebra and equivalent representations with different parameterization (Lloyd et al., 2014). Certain covariance functions can differ in terms of the hyper-parameterization but can be absorbed into a single covariance function with a different parameterization. To inspect the posterior of these equivalent structures we convert each kernel expression into a sum of products and subsequently simplify expressions using the following grammar:

Listing 4: Grammar to simplify expressions

SE $\times$ SE	$\rightarrow$ SE
{SE, PER, C, WN} $\times$ WN	$\rightarrow$ WN
LIN $+$ LIN	$\rightarrow$ LIN
{SE, PER, C, WN, LIN} $\times$ C	$\rightarrow$ {SE, PER, C, WN, LIN}

For reproducing results from the Automated Statistician Project in a Bayesian fashion we first define a prior on the hypothesis space. Note that, as in the implementation of the Automated Statistician, we upper-bound the complexity of the space of covariance functions we want to explore. We also put vague priors on hyper-parameters.

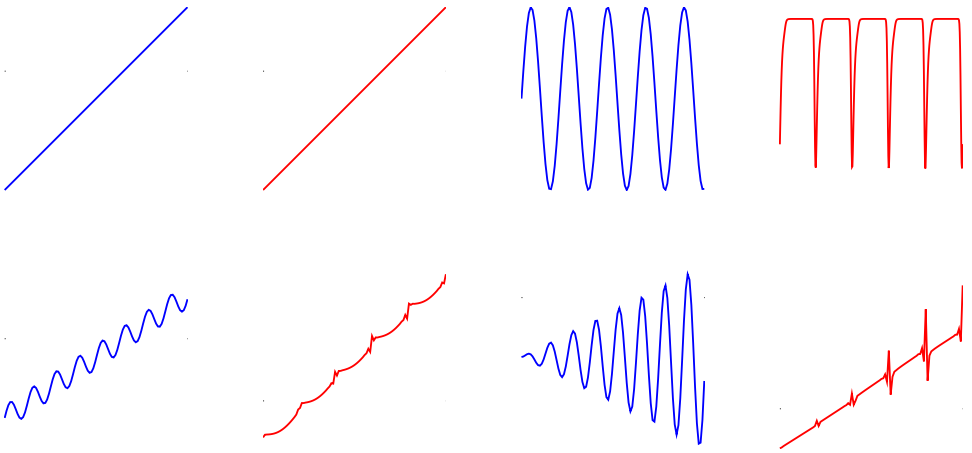
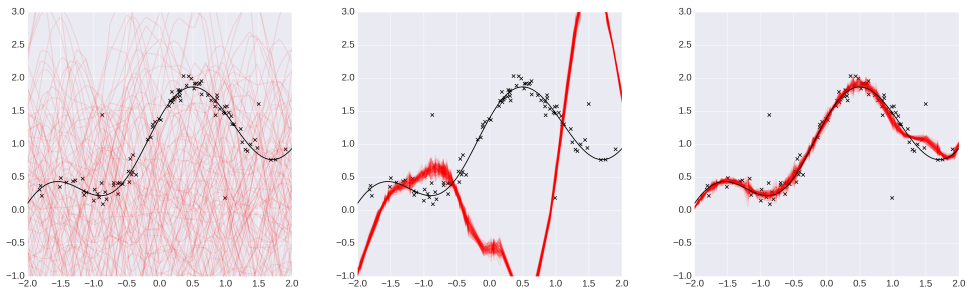


Figure 1: Composition of covariance functions (blue, left) and samples from the distribution of curves they can produce (red, right).



(a) Prior (b) Observed (c) Inferred

Figure 2: Running a Venture GP on Neal's example for MCMC showing the prior, after having observed the data and after performing inference on the hyper-parameters. Note how the GP is choosing outliers to smooth instead of essential data.

Listing 5: Venture Code for Bayesian GP Structure Learning

```
[ASSUME S (array K_1,K_2,···,K_n)] // (defined as above)
[ASSUME p_n (uniform_structure n)]
[ASSUME S (array K_1,K_2,···,K_n)]
[ASSUME K* (grammar S p_n)]
[ASSUME GP (make-gp 0 K* ]

[OBSERVE GP D]

[INFER (REPEAT 3000 (DO
                        (MH 10 p_n one 1)
                        (MH 10 K* one 1)
                        (MH 10 {hyper-parameters} one 10)) ]
```

We defined a set of covariance structures so that we could reproduce results for covariance function structure learning as in the Automated Statistician. Our results are very similar to what has been reported by previous work (Duvenaud et al., 2013; see Fig. 3).

4.2 Log-Likelihood

4.3 Residuals

References

Duvenaud, D., Lloyd, J. R., Grosse, R., Tenenbaum, J., and Ghahramani, Z. (2013). Structure discovery in nonparametric regression through compositional kernel search. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1166–1174.

Lloyd, J. R., Duvenaud, D., Grosse, R., Tenenbaum, J., and Ghahramani, Z. (2014). Automatic construction and natural-language description of nonparametric regression models. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Neal, R. M. (1997). Monte carlo implementation of gaussian process models for bayesian regression and classification. *arXiv preprint physics/9701026*.

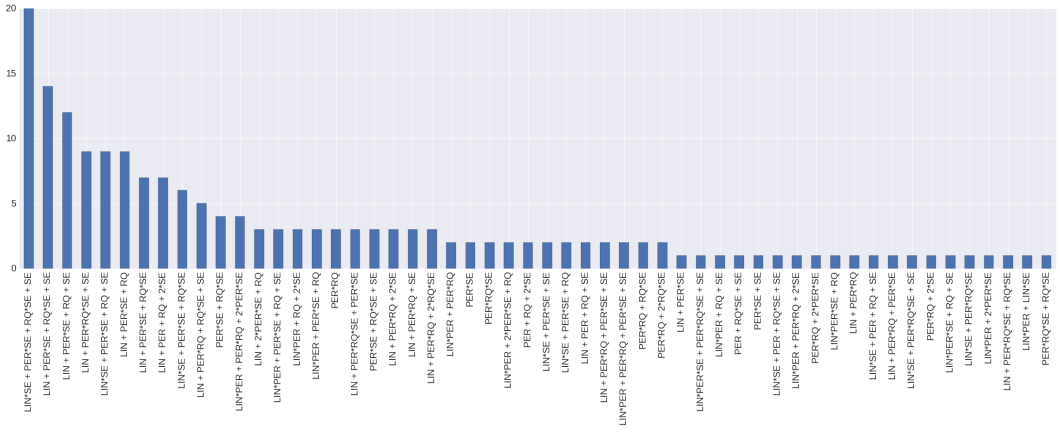
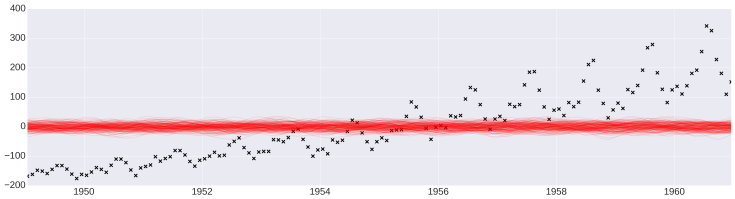
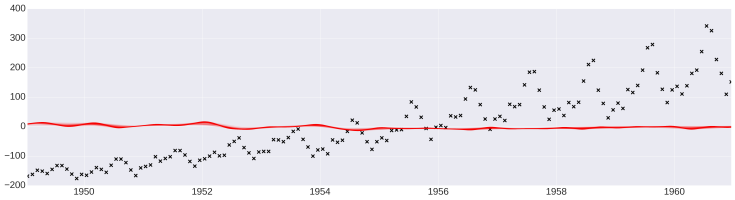


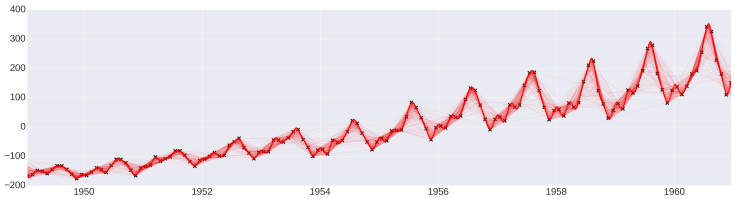
Figure 3: Preliminary results from the cross-validation on the CO2 data. Note that Duvenaud et al. (2013) report LIN $\times$ SE + PER $\times$ SE + RQ $\times$ SE. We have run a leave one out cross-validation on this data set. Above we see the preliminary results on 181 validations (of a total of 545 $\times$ 2 runs).



(a) Prior, before having seen any data.



(b) After having seen any data but before inference



(c) After 1000 MH steps.

Figure 4: Running a Venuter GP with covariance structure PER x SE on the airline data