GP3: Gaussian Processes with Probabilistic Programming (In progress)

Anuj Sharma

January 19, 2018

Abstract

This document provides an overview of the models and inference algorithms implemented in GP3.

1 Background

1.1 Gaussian Processes

Assume we have a dataset of N observations $X = \{x_i\}_{i=1}^N$ in some P dimensional space. We say that a function $f(x)$ is distributed as a Gaussian process $f \sim \mathcal{GP}(\mu(\cdot), k(\cdot,\cdot))$ if any set of function values $\{f(x_i)\}_{i=1}^n$ has a multivariate normal distribution defined by mean function $\mu(\cdot)$ and covariance kernel $k(\cdot,\cdot)$:

$$\{f(x_i)\}_{i=1}^n \sim \mathcal{N}(\mu(X), K(X,X)) \tag{1}$$

where $K(X,X)_{i,j} = k(x_i, x_j)$. A Gaussian Process can be thought of as a distribution over functions, where the covariance kernel encodes properties such as smoothness and periodicity. Typically, applications of Gaussian Processes work with a Gaussian observation likelihood (i.e. $y_i \sim \mathcal{N}(f(x_i), \sigma^2)$ for some observation variance σ^2). With Gaussian likelihoods, we have closed-form solutions for predictive means and covariances. I will forgo reviewing this since GP3 is intended primarily for complex custom likelihoods.

See Rasmussen and Williams (2006) for a full introduction to GPs.

1.2 Structure Exploiting Inference

Assume we have a dataset of N observations $\{x_i\}_{i=1}^N$ that lie on a rectilinear grid of dimension P, $x \in \chi_1 \times ...\chi_P$. Also assume that our kernel function k decomposes as a product over dimensions

$$k(x, x') = \prod_{i=1}^{P} k(x_p, x'_p) \tag{2}$$

With these two constraints, our covariance matrix $K_{X,X}$ decomposes as a Kronecker product over dimensions:

$$K = K_1 \otimes ... \otimes K_P \tag{3}$$

Typically, GP inference (even with non-Gaussian likelihoods) requires solving a linear system of the form

$$(K_{X,X} + \sigma^2 I)^{-1} y \tag{4}$$

The standard practice is to solve the above using Cholesky decompositions. This scales $O(n^3)$. With the Kronecker structure, we can take advantage of efficient matrix-vector products using the following property:

$$(A \otimes B)\text{vec}(X) = \text{vec}(BXA^T) \tag{5}$$

This is useful for solving systems with conjugate gradients, where we can leverage efficient matrix-vector products with K. For intuition, consider the case where we have X that lies on a $10 \times 10 \times 10$ grid. With traditional GP inference, we would have matrix-vector products with a 1000×1000 covariance matrix (10^6 elements). With the above constraints, we only have to work with 3 10×10 matrices (300 elements).

See Wilson et al (2014) and Wilson, Dann, and Nickish (2015) for an overview of structure exploiting inference.

2 Inference

2.1 Laplace Approximation

2.1.1 Newton's Method

I follow Flaxman et al (2015)'s algorithm for Laplace inference in GPLVMs with non-Gaussian likelihoods. We are interested in optimizing

$$\Psi(s) := \log p(s|\mathcal{D}) =^{const} \log p(y|s) + \log p(s|X) \tag{6}$$

We use Newton's method where

$$\nabla\Psi(s) = \nabla \log p(y|s) + K^{-1}(s - \mu) \tag{7}$$

$$\nabla\nabla\Psi(s) = \nabla\nabla \log p(y|s) - K^{-1} \tag{8}$$

To find the MAP estimate, we use the following Newton step:

$$s \leftarrow s - (\nabla\nabla\Psi)^{-1}\nabla\Psi \tag{9}$$

This gives us a Laplace approximation around the MAP estimate

$$p(s|y) \approx \mathcal{N}(\hat{s}, (K^{-1} + W)^{-1}) \tag{10}$$

where $\hat{s}$ denotes the MAP and W denotes the Hessian of the likelihood at the MAP. As derived in Flaxman et al (2015), this inference procedure requires $O(Dn^{\frac{D+1}{D}})$ operations and $O(Dn^{\frac{2}{D}})$ space.

2.1.2 Variance Approximation

For the variance approximations, I extend the methods described in Flaxman et al (2015), Wilson, Dann and Nickish (2015), and Papandreou and Yuille (2011).

Using the transformations for numerical stability used in Rasmussen and Williams (2006) and Flaxman et al (2015) we can write the posterior covariance as $(K^{-1} + W)^{-1} = K - KQK$, where $B = I + W^{\frac{1}{2}}KW^{\frac{1}{2}}$ and $Q = W^{\frac{1}{2}}B^{-1}W^{\frac{1}{2}}$. Using this, we can note that the posterior variance is equal to

$$\text{diag}(K) - \text{var}(KW^{\frac{1}{2}}r) \tag{11}$$

where $r \sim \mathcal{N}(0, B^{-1})$. This gives us the following procedure for estimating the posterior variance:

First, draw $g_n \sim \mathcal{N}(0, I)$ and $g_m \sim \mathcal{N}(0, I)$. Then, solve the following for r:

$$Br = (W^{\frac{1}{2}}KW^{\frac{1}{2}})^{\frac{1}{2}}g_m + \sigma g_n \tag{12}$$

where σ is our noise variance. Using n_s iterations of the above procedure, we can estimate our posterior variance as

$$\max(0, K - \frac{1}{n_s}\sum_{i=1}^{n_s}(KW^{\frac{1}{2}}r_i)^2) \tag{13}$$

where the max is taken element-wise. We use $n_s = 30$ for our approximations (the papers above recommend $n_s = 20$).

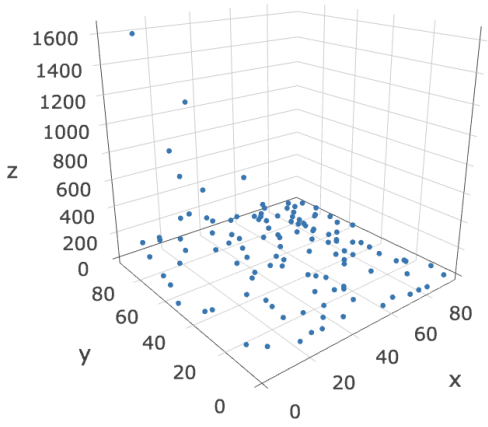
2.2 Stochastic Variational Inference

I also adapt the stochastic variational inference algorithm described in Wilson et al (2016). In this form of inference, we approximate our posterior $p(s|y)$ using a variational distribution $q(s; \lambda)$. We optimize the following variational lower bound over the variational parameters λ:

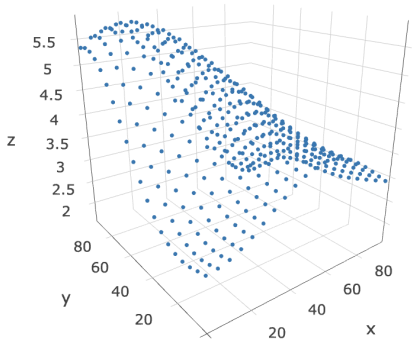
$$\log p(y) \geq \mathbb{E}_{q(s)p(y|s)}[\log p(y|s)] - \text{KL}[q(s)||p(s)] \tag{14}$$

I use a multivariate normal $\mathcal{N}(\mu_q, S_q)$ as the variational distribution. I assume that S_q has a Cholesky decomposition $R_q^T R_q$ (where R_q is upper triangular) and R_q decomposes as a Kronecker product over dimensions (i.e. $R_q = R_{q1} \otimes ... \otimes R_{qP}$). This allows us to exploit Kronecker structure in both the GP prior and the variational distribution.

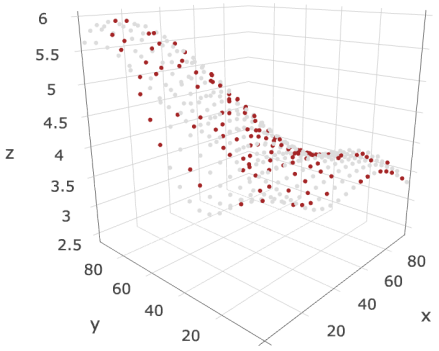
I use the "reparametrization trick" and Monte Carlo estimates of expectations of gradients to optimize the first term. See Wilson et al (2016) for a complete derivation. See Figure 3 for an example run of the SVI inference.



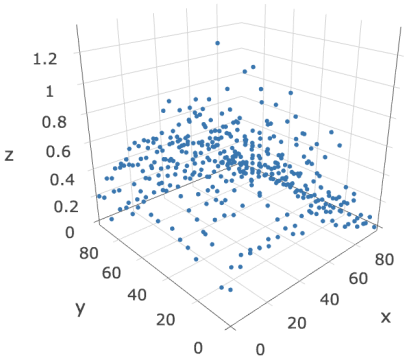
(a) Observed data



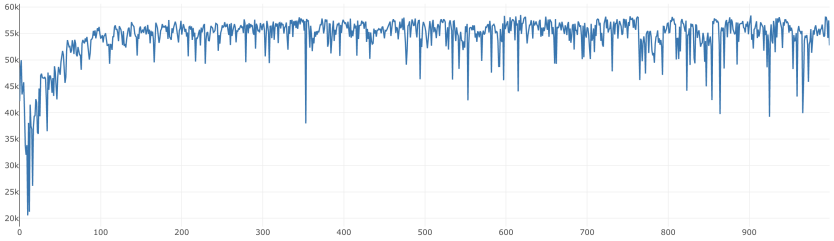
(b) True function



(c) Inferred Function



(d) Posterior Variances



(e) Trajectory of variational lower bound

Figure 1: Example run of SVI

2.3 Mean Field Stochastic Variational Inference

References

Flaxman, Seth, Wilson, Andrew Gordon, Neil, Daniel B., Nickish, Hannes, Smola, Alexander J. (2015). *Fast Kronecker Inference in Gaussian Processes with Non-Gaussian Likelihoods.* Proceedings of the 32nd International Conference on Machine Learning.

Papandreou, G. and Yuille, A. L. (2011). Ecient variational inference in large-scale Bayesian compressed sensing. In Proc. IEEE Workshop on Information Theory in Com- puter Vision and Pattern Recognition (in conjunction with ICCV-11), pages 1332-1339.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for Machine Learning.*The MIT Press.

Wilson, Andrew Gordon, Gilboa, Elad, Nehorai, Arye, and Cunningham, John P. (2014). *Fast Kernel Learning for Multidimensional Pattern Extrapolation.* 27th Conference on Neural Information Processing Systems (NIPS 2014).

Wilson, Andrew Gordon, Dann, Christoph, Nickish Hannes (2015). *Thoughts on Massively Scalable Gaussian Processes*

Wilson, Andrew Gordon, Hu, Zhiting, Salakhutdinov, Ruslan, Xing, Eric P. (2016). *Stochastic Variational Deep Kernel Learning.* 29th Conference on Neural Information Processing Systems (NIPS 2016).