FastGP: an R package for Gaussian processes

Giri Gopalan
Harvard University

Luke Bornn
Harvard University

Abstract

Despite their promise and ubiquity, Gaussian processes (GPs) can be difficult to use in practice due to the computational impediments of fitting and sampling from them. Here we discuss a short R package for efficient multivariate normal functions which uses the **Rcpp** and **RcppEigen** packages at its core. GPs have properties that allow standard functions to be sped up; as an example we include functionality for Toeplitz matrices whose inverse can be computed in $O(n^2)$ time with methods due to Trench and Durbin (Golub & Van Loan 1996), which is particularly apt when time points (or spatial locations) of a Gaussian process are evenly spaced, since the associated covariance matrix is Toeplitz in this case. Additionally, we include functionality to sample from a latent variable Gaussian process model with elliptical slice sampling (Murray, Adams, & MacKay 2010).

Keywords: Gaussian processes, multivariate normal distributions, **Rcpp**, **RcppEigen**.

1. Introduction

Many methodologies involving a Gaussian process rely heavily on computationally expensive functions such as matrix inversion and the Cholesky decomposition. Rather than create a package to solve a particular high-level Gaussian process (GP) task (e.g., expectation propagation, variational inference, regression and classification (Neal 1998)), the aim of **FastGP** is to improve the performance of these fundamental functions in order to help all researchers working with GPs. While there exist R packages for sampling from a multivariate normal distribution (MVN) or evaluating the density of an MVN, notably **MASS** and **mvtnorm** on CRAN (Genz & et al. 2014; Venables et al. 2002), we have found such packages can be slow in the context of GPs, partially due to unnecessary checks for symmetry and positive definiteness (which hold for GPs with commonly used kernels such as squared exponential or Matern (Rasmussen & Williams 2005)) or not accounting for the structure (e.g. Toeplitz) of the underlying covariance matrix. Hence, we write functions optimized with **Rcpp** and **RcppEigen** (Bates & Eddelbuettel 2013; Eddelbuettel 2013) to make these tasks more computationally efficient, and demonstrate their efficiency by benchmarking them against built-in R functions and methods from the **MASS** and **mvtnorm** libraries. Additionally, we include functionality to sample from the posterior of a Bayesian model for which the prior distribution is multi-

variate normal using elliptical slice sampling, a task which is often used alongside GPs and due to its iterative nature, benefits from a C++ version (Murray, Adams, & MacKay 2010).

To elaborate, a Gaussian process (GP) is a collection of random variables (i.e., a stochastic process) (X_t) such that any finite subset of these random variables has a joint multivariate normal distribution (Grimmet & Stirzaker 2001). Such processes have been used extensively in recent decades, particularly in machine learning, spatial and temporal statistics, and computer experiments (Rasmussen & Williams 2005). However, GPs can be difficult to work with in practice because they are computationally onerous; to be precise, the density of a multivariate normal (MVN) vector in $\mathbb{R}^n$ is:

$$f(x) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp^{-(x-\mu)^T\Sigma^{-1}(x-\mu)/2} \tag{1}$$

This involves the computation of a determinant and inverse of a matrix in $\mathbb{R}^{n\times n}$, generally taking $O(n^3)$ operations to complete, and the computation of the Cholesky decomposition is typically a prerequisite for sampling from a multivariate normal, which also takes $O(n^3)$ time to complete (Golub & Van Loan 1996). Many spatial models, including those which tackle nonstationarity as in Bornn et al. (2012), parametrize the covariance matrix Σ, and hence for Monte Carlo-based inference require repeated recalculations of Σ^{-1} and $|\Sigma|$.

2. Key functions and benchmark results

2.1. Key functions and package organization

The core functions of **FastGP** can be categorized into three sets. The first set of functions are matrix operations that are necessary for sampling from and evaluating the density of a multivariate normal random variable. These are: a function for inversion using **RcppEigen**, a function for inverting a symmetric positive definite Toeplitz matrix in $O(n^2)$ time (which, as aforementioned, can be useful for inverting a covariance matrix in which the underlying points are evenly spaced (Storkey, A.J. 1999)) which uses methods due to Trench and Durbin (Golub & Van Loan 1996) written in **Rcpp**, and a function for evaluating the Cholesky decomposition of a matrix using **RcppEigen**. To be as explicit as possible, the inversion and Cholesky decomposition come directly from **RcppEigen** (Bates & Eddelbuettel 2013). The second set of functions are those which directly simulate from and evaluate the log density of a multivariate normal, both in the general case and when the underlying covariance matrix is Toeplitz. The final major function included in the package is the elliptical slice sampling algorithm for simulating from the posterior of a Bayesian model in which the prior is jointly multivariate normal, which tends to outperform classical methods such as Metropolis-Hastings computationally, as is evidenced empirically by Murray, Adams, & MacKay (2010).

2.2. Benchmark results

Here we use the **rbenchmark** (Eugster & Leisch 2008) package to demonstrate the efficacy of these methods. In particular we test these functions with a mock covariance matrix from a square exponential Gaussian process on 200 evenly spaced time points with σ and ϕ arbitrarily set to 1 (and hence the covariance matrix is Toeplitz in this case).

Table 1: Benchmarking the runtime for functions included in **FastGP**. The numbers indicate how many times faster the functions performed using our package versus standard R, **MASS**, and **mvtnorm** functions, using the `benchmark` function from the **rbenchmark** package.

FastGP Function	Standard Function	Relative Speed Improvement
`rcppeigen_invert_matrix`	`solve`	x3.287
`tinv`	`solve`	x14.374
`rcppeigen_get_det`	`det`	x1.851
`rcpp_log_dmvnorm, istoep= TRUE`	`dmvnorm`	x1.966
`rcpp_log_dmvnorm, istoep= FALSE`	`dmvnorm`	x.6592
`rcpp_rmvnorm`	`rmvnorm`	x23.462
`rcpp_rmvnorm`	`mvrnorm`	x22.812

2.3. Demonstration of elliptical slice sampling

We consider a model where we observe a "warped" signal $s = A\sin((t + w)/T) + \epsilon$ where w is drawn according to a 0 mean GP with squared exponential kernel, ϵ is drawn according to a normal distribution with 0 mean and $\sigma = .001$, and A and T are known constants. Our objective is to perform inference on the latent warping w, and we can do this with the elliptical slice sampling function included with **FastGP**, as in `FastGPdemo.r`. The function implements the algorithm as described by Murray, Adams, & MacKay (2010) and benefits from the use of the optimized functions contained within **FastGP** since each iteration requires several log-likelihood evaluations (which may require the evaluation of the log-density of a multivariate normal distribution) and drawing from a multivariate normal distribution. This results in the following posterior draws for w illustrated in **Figure 1**. Additionally we benchmark an elliptical slice sampler with **FastGP** functions versus an elliptical slice sampler with standard functions below.

Table 2: Benchmarking the runtime for elliptical slice sampling using functions from **FastGP** versus elliptical slice sampling using standard functions from R, **mvtnorm**, and **MASS**.

FastGP Function	Standard Function	Relative Speed Improvement
`rcpp_ess`	`standard_ess`	x1.508

3. Conclusion

To summarize, we have written an R package **FastGP** using **Rcpp** and **RcppEigen** for handling multivariate normal distributions in the context of Gaussian processes efficiently. Additionally we have included functionality to perform Bayesian inference for latent variable Gaussian process models with elliptical slice sampling (Murray, Adams, & MacKay 2010).

References

Bates, D., & Eddelbuettel, D. 2013. Journal of Statistical Software, 52(5), 1-24.

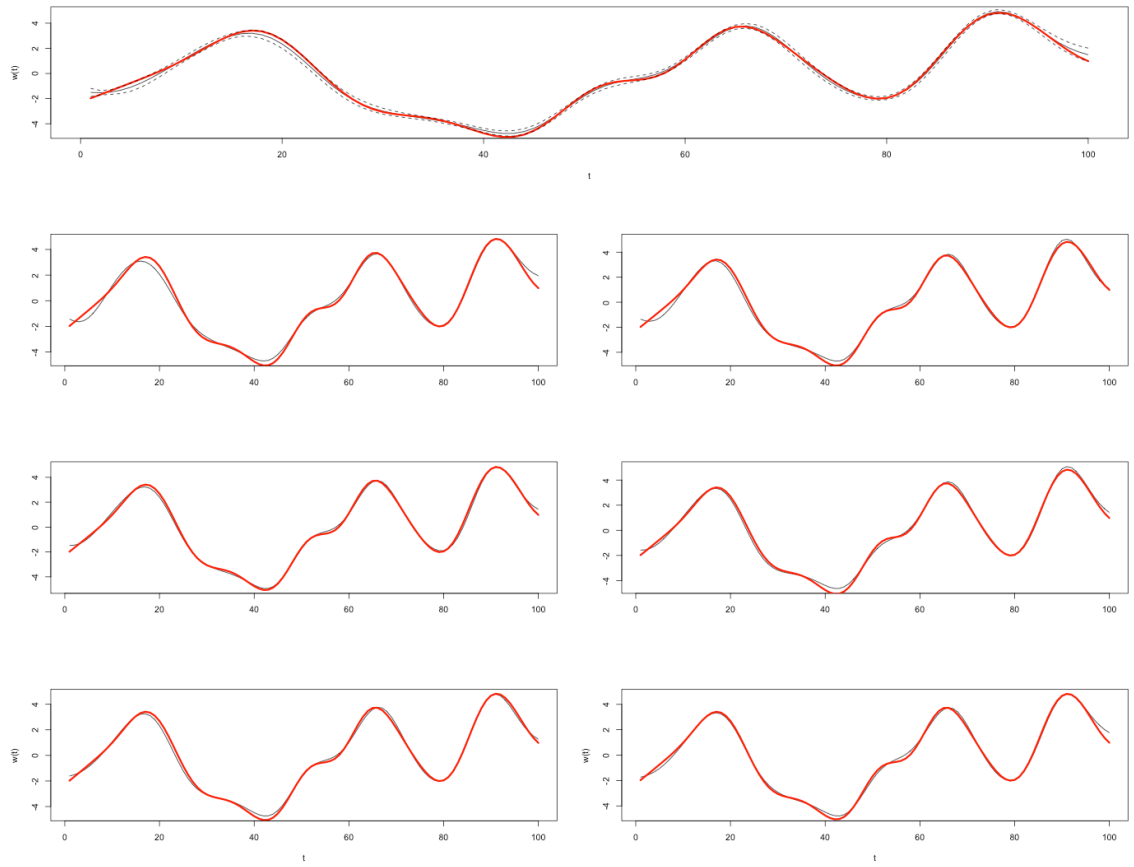

Figure 1: Top: The red line indicates ground truth, the black line indicates the mean of the posterior latent samples, and the dotted lines indicate the 2 standard deviations above and below the mean, respectively, of the latent samples. Remaining rows show the 100th, 300th, 500th, 700th, 900th, and 1000th MCMC samples for the warping w respectively, in black, compared to ground truth in red.

Bornn, L., Shaddick, G., & Zidek, J. V. (2012). Modeling nonstationary processes through dimension expansion. Journal of the American Statistical Association, 107(497), 281-289.

Eddelbuettel, D. 2013. Seamless R and C++ Integration with Rcpp. Springer, New York.

Eddelbuettel, D., & Sanderson, C. 2014, Computational Statistics and Data Analysis, 71, pp 1054-1063.

Eugster, Manuel., & Leisch, Friedrich. 2008, Bench Plot and Mixed Effects Models: First Steps toward a Comprehensive Benchmark Analysis Toolbox. Compstat 2008—Proceedings in Computational Statistics, 299-306

Genz et al. mvtnorm: Multivariate Normal and t Distributions. R package version 1.0-2. http://CRAN.R-project.org/package=mvtnorm

Golub, Gene., & Van Loan, C.F. 1996, Matrix Computations. Johns Hopkins University Press.

Grimmet, G., & Stirzaker, D. 2001. Probability and Random Processes. Oxford University Press.

Murray, I., Adams, R., & MacKay, D. 2010, JMLR

Neal, RM. 1998 Regression and classification using Gaussian process priors. Bayesian Statistics 6.

Rasmussen, C., & Williams, C. 2005, Gaussian Processes for Machine Learning (The MIT Press)

Storkey A.J. 1999, Truncated covariance matrices and Toeplitz methods in Gaussian processes, in ICANN99: Artificial Neural Networks. pages 55-60.

Venables, W. N. & Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth Edition. Springer, New York. ISBN 0-387-95457-0