

Gaussian Process Learning via Fisher Scoring of Vecchia’s Approximation

Joseph Guinness

Cornell University, Department of Statistics and Data Science

Abstract

We derive a single pass algorithm for computing the gradient and Fisher information of Vecchia’s Gaussian process loglikelihood approximation, which provides a computationally efficient means for applying the Fisher scoring algorithm for maximizing the loglikelihood. The advantages of the optimization techniques are demonstrated in numerical examples and in an application to Argo ocean temperature data. The new methods are more accurate and much faster than an optimization method that uses only function evaluations, especially when the covariance function has many parameters. This allows practitioners to fit nonstationary models to large spatial and spatial-temporal datasets.

1 Introduction

The Gaussian process model is an indispensable tool for the analysis of spatial and spatial-temporal datasets and has become increasingly popular as a general-purpose model for functions. Because of its high computational burden, researchers have devoted substantial effort to developing numerical approximations for Gaussian process computations. Much of the work focuses on efficient approximation of the likelihood function. Fast likelihood evaluations are crucial for optimization procedures that require many evaluations of the likelihood, such as the default Nelder-Mead algorithm (Nelder and Mead, 1965) in the R `optim` function. The likelihood must be repeatedly evaluated in MCMC algorithms as well.

Compared to the amount of literature on efficient likelihood approximations, there has been considerably less development of techniques for numerically maximizing the likelihood (see Geoga et al. (2018) for one recent example). This article aims to address the disparity by providing:

1. Formulas for evaluating the gradient and Fisher information for Vecchia’s likelihood approximation in a single pass through the data, so that the Fisher scoring algorithm can be applied. Fisher scoring is a modification of the Newton-Raphson optimization method, replacing the Hessian matrix with the Fisher information matrix.
2. Numerical examples with simulated and real data demonstrating the practical advantages that the new techniques provide over an optimizer that uses function evaluations alone.

Among the sea of Gaussian process approximations proposed over the past several decades, Vecchia’s approximation (Vecchia, 1988) has emerged as a leader. It can be computed in linear time and with linear memory burden, and it can be parallelized. Maximizing the approximation corresponds to solving a set of unbiased estimating equations, leading to desirable statistical properties (Stein et al., 2004). It is general in that it does not require gridded data nor a stationary model assumption. The approximation forms a valid multivariate normal model, and so it can be used for simulation and conditional simulation. As an approximation to the target model, it is highly accurate relative to competitors (Guinness, 2018). Vecchia’s approximation also forms a

conceptual hub in the space of Gaussian process approximations, since a generalization includes many well-known approximations as special cases (Katzfuss and Guinness, 2017). Lastly, there are publicly available R packages implementing it (Finley et al., 2017; Guinness and Katzfuss, 2018).

The numerical examples in this paper show that, in realistic data and model scenarios, the new techniques offer significant computational advantages over default optimization techniques. Although it is more expensive to evaluate the gradient and Fisher information in addition to the likelihood, the Fisher scoring algorithm converges in a small number of iterations, leading to a large advantage in total computing time over an optimization method that uses only the likelihood. For isotropic Matérn models, the speedup is roughly 2 to 4 times, and on more complicated models with more parameters, the new techniques can be more than 40 times faster. This is a significant practical improvement that will be attractive to practitioners choosing among various methods.

2 Background

Let s_1, \dots, s_n be locations in a domain D . At each s_i , we observe a scalar response y_i , collected into column vector $y = (y_1, \dots, y_n)^T$. Along with the response, we observe covariates $x_i = (x_{i1}, \dots, x_{ip})$ collected into an $n \times p$ design matrix X . In the Gaussian process, we model y as a multivariate normal vector Y with expected value $E(Y) = X\beta$ ($\beta \in \mathbb{R}^p$), and covariance matrix $E((Y - X\beta)(Y - X\beta)^T) = \Sigma_\theta$, where the (i, j) entry of Σ_θ is $K_\theta(s_i, s_j)$. The function K_θ is positive definite on $D \times D$ and depends on covariance parameters θ . The loglikelihood for β and θ is

$$\log f_{\beta, \theta}(y) = -\frac{n}{2} \log(2\pi) - \frac{1}{2} \log \det \Sigma_\theta - \frac{1}{2} (y - X\beta)^T \Sigma_\theta^{-1} (y - X\beta). \quad (1)$$

Unless Σ_θ has some exploitable structure, evaluation of the loglikelihood involves storing the n^2 entries of Σ_θ and performing $O(n^3)$ floating point operations to obtain the Cholesky factor of Σ_θ , both of which are computationally prohibitive when n is large.

Vecchia’s loglikelihood approximation is a modification of the conditional representation of a joint density function. Let $g(1) = \emptyset$, $g(i) \subset (1, \dots, i - 1)$ and $y_{g(i)}$ be the corresponding subvector of y . Vecchia’s loglikelihood approximation is

$$\ell(\beta, \theta) = \sum_{i=1}^n \log f_{\beta, \theta}(y_i | y_{g(i)}), \quad (2)$$

leading to computational savings when $|g(i)|$ is small. As mentioned in the introduction, Vecchia’s likelihood approximation corresponds to a valid multivariate normal distribution with mean $X\beta$ and a covariance matrix $\tilde{\Sigma}_\theta$. To motivate why obtaining the gradient and Fisher information poses an analytical challenge, consider the partial derivative of Vecchia’s loglikelihood with respect to θ_j :

$$\frac{\partial \ell(\beta, \theta)}{\partial \theta_j} = \frac{1}{2} (y - X\beta)^T \tilde{\Sigma}_\theta^{-1} \frac{\partial \tilde{\Sigma}_\theta}{\partial \theta_j} \tilde{\Sigma}_\theta^{-1} (y - X\beta) - \frac{1}{2} \text{Tr} \left(\tilde{\Sigma}_\theta^{-1} \frac{\partial \tilde{\Sigma}_\theta}{\partial \theta_j} \right), \quad (3)$$

where $(\partial \tilde{\Sigma}_\theta / \partial \theta_j)$ is an $n \times n$ matrix of partial derivatives of $\tilde{\Sigma}_\theta$ with respect to θ_j . Not only is $\partial \tilde{\Sigma}_\theta / \partial \theta_j$ too large to store in memory, the covariances $\tilde{\Sigma}_\theta$ are not easily computable, nor are their partial derivatives. In the next section, we outline a simple reframing of Vecchia’s likelihood that leads to a computationally tractable method of evaluating the gradient and Fisher information.

3 Derivations for Single Pass Algorithm

To derive formulas for the gradient and Fisher information, it is helpful to rewrite the conditional likelihoods in terms of marginals. To this end, define $u_i = y_{g(i)}$ and $v_i = (y_{g(i)}, y_i)$. Define the design matrices for u_i and v_i , respectively, as Q_i and R_i , and define the covariance matrices for u_i and v_i , respectively as A_i and B_i (suppressing dependence on θ). The notation is chosen to follow the mnemonic device that the first of the two letters alphabetically is a subvector or submatrix of the second letter. Vecchia's loglikelihood can then be rewritten as

$$\ell(\beta, \theta) = \sum_{i=1}^m \log f_{\beta, \theta}(v_i) - \log f_{\beta, \theta}(u_i) \quad (4)$$

$$= -\frac{1}{2} \sum_{i=1}^n [\log \det B_i - \log \det A_i] \quad (5)$$

$$- \frac{1}{2} \sum_{i=1}^n [(v_i - R_i \beta)^T B_i^{-1} (v_i - R_i \beta) - (u_i - Q_i \beta)^T A_i^{-1} (u_i - Q_i \beta)] - \frac{n}{2} \log(2\pi). \quad (6)$$

Our proposed algorithm for obtaining the likelihood, gradient, and Fisher information involves computing the following quantities in a single pass through the data.

$$(\text{logdet}) = \sum_{i=1}^n (\log \det B_i - \log \det A_i) \quad (7)$$

$$(\text{dlogdetj}) = \sum_{i=1}^n (\text{Tr}(B_i^{-1} B_{i,j}) - \text{Tr}(A_i^{-1} A_{i,j})) \quad (8)$$

$$(\text{ySy}) = \sum_{i=1}^n (v_i^T B_i^{-1} v_i - u_i^T A_i^{-1} u_i) \quad (9)$$

$$(\text{XSy}) = \sum_{i=1}^n (R_i^T B_i^{-1} v_i - Q_i^T A_i^{-1} u_i) \quad (10)$$

$$(\text{XSX}) = \sum_{i=1}^n (R_i^T B_i^{-1} R_i - Q_i^T A_i^{-1} Q_i) \quad (11)$$

$$(\text{dySyj}) = - \sum_{i=1}^n (v_i^T B_i^{-1} B_{i,j} B_i^{-1} v_i - u_i^T A_i^{-1} A_{i,j} A_i^{-1} u_i) \quad (12)$$

$$(\text{dXSyj}) = - \sum_{i=1}^n (R_i^T B_i^{-1} B_{i,j} B_i^{-1} v_i - Q_i^T A_i^{-1} A_{i,j} A_i^{-1} u_i) \quad (13)$$

$$(\text{dXSXj}) = - \sum_{i=1}^n (R_i^T B_i^{-1} B_{i,j} B_i^{-1} R_i - Q_i^T A_i^{-1} A_{i,j} A_i^{-1} Q_i) \quad (14)$$

$$(\text{Trjk}) = \sum_{i=1}^n [\text{Tr}(B_i^{-1} B_{i,j} B_i^{-1} B_{i,k}) - \text{Tr}(A_i^{-1} A_{i,j} A_i^{-1} A_{i,k})], \quad (15)$$

where $A_{i,j}$ and $B_{i,j}$ are the matrices of partial derivatives of A_i and B_i , respectively, with respect to θ_j . The quantities having the form $(\text{d} * \text{j})$ are simply the partial derivatives of the corresponding quantity $(*)$ with respect to θ_j . Each of these quantities can be updated at each $i = 1, \dots, n$, and so all can be evaluated in a single pass through the data. We refer to them collectively as our single-pass quantities.

3.1 Profile Likelihood, Gradient, and Fisher Information

Given covariance parameter θ , denote the maximum Vecchia likelihood estimate of β as $\widehat{\beta}(\theta)$. Since $\widehat{\beta}(\theta)$ has a closed form expression (Section 3.2), we can maximize the profile likelihood $\ell(\widehat{\beta}(\theta), \theta)$ over θ alone. The profile likelihood can be written in terms of our single-pass quantities as

$$\ell(\widehat{\beta}(\theta), \theta) = -\frac{n}{2} \log(2\pi) - \frac{1}{2} (\text{dlogdet}) - \frac{1}{2} \left[(\mathbf{yS}\mathbf{y}) - 2(\mathbf{X}\mathbf{S}\mathbf{y})\widehat{\beta}(\theta) + \widehat{\beta}(\theta)^T (\mathbf{X}\mathbf{S}\mathbf{X})\widehat{\beta}(\theta) \right]. \quad (16)$$

Therefore the partial derivatives can also be written in terms of the single pass quantities as

$$\begin{aligned} \frac{\partial \ell(\widehat{\beta}(\theta), \theta)}{\partial \theta_j} &= -\frac{1}{2} (\text{dlogdet } \mathbf{j}) - \frac{1}{2} \left[(\text{d}\mathbf{yS}\mathbf{y} \mathbf{j}) - 2(\text{d}\mathbf{X}\mathbf{S}\mathbf{y} \mathbf{j})\widehat{\beta}(\theta) + \widehat{\beta}(\theta)^T (\text{d}\mathbf{X}\mathbf{S}\mathbf{X} \mathbf{j})\widehat{\beta}(\theta) \right] \\ &\quad - \frac{1}{2} \left[-2(\mathbf{X}\mathbf{S}\mathbf{y}) \frac{\partial \widehat{\beta}(\theta)}{\partial \theta_j} + 2\widehat{\beta}(\theta)^T (\mathbf{X}\mathbf{S}\mathbf{X}) \frac{\partial \widehat{\beta}(\theta)}{\partial \theta_j} \right], \end{aligned} \quad (17)$$

where $(\partial \widehat{\beta}(\theta) / \partial \theta_j)$ is the column vector of partial derivatives of the p entries of $\widehat{\beta}(\theta)$ with respect to covariance parameter θ_j . The Fisher information is

$$\mathcal{I}(\theta)_{jk} = \frac{1}{2} \sum_{i=1}^n [\text{Tr}(B_i^{-1} B_{i,j} B_i^{-1} B_{i,k}) - \text{Tr}(A_i^{-1} A_{i,j} A_i^{-1} A_{i,k})] = \frac{1}{2} (\text{Tr } \mathbf{j}\mathbf{k}).$$

It remains to be shown that $\widehat{\beta}(\theta)$ and $\partial \widehat{\beta}(\theta) / \partial \theta_j$ can be computed using our single-pass quantities.

3.2 Mean Parameters

The profile likelihood estimate $\widehat{\beta}(\theta)$ satisfies $\partial \ell(\beta, \theta) / \partial \beta_j = 0$ for every $j = 1, \dots, p$. These partial derivatives are

$$\begin{bmatrix} \partial \ell(\beta, \theta) / \partial \beta_1 \\ \vdots \\ \partial \ell(\beta, \theta) / \partial \beta_p \end{bmatrix} = \sum_{i=1}^n R_i^T B_i^{-1} (v_i - R_i \beta) - Q_i^T A_i^{-1} (u_i - Q_i \beta), \quad (18)$$

giving the equation

$$\left[\sum_{i=1}^n (R_i^T B_i^{-1} R_i - Q_i^T A_i^{-1} Q_i) \right] \widehat{\beta}(\theta) = \left[\sum_{i=1}^n (R_i^T B_i^{-1} v_i - Q_i^T A_i^{-1} u_i) \right]. \quad (19)$$

Therefore, the profile likelihood estimate of β is

$$\widehat{\beta}(\theta) = (\mathbf{X}\mathbf{S}\mathbf{X})^{-1} (\mathbf{X}\mathbf{S}\mathbf{y}), \quad (20)$$

a function of our single pass quantities. Taking partial derivatives with respect to θ_j yields

$$\frac{\partial \widehat{\beta}(\theta)}{\partial \theta_j} = (\mathbf{X}\mathbf{S}\mathbf{X})^{-1} (\text{d}\mathbf{X}\mathbf{S}\mathbf{y} \mathbf{j}) - (\mathbf{X}\mathbf{S}\mathbf{X})^{-1} (\text{d}\mathbf{X}\mathbf{S}\mathbf{X} \mathbf{j}) (\mathbf{X}\mathbf{S}\mathbf{X})^{-1} (\mathbf{X}\mathbf{S}\mathbf{y}), \quad (21)$$

also a function of our single pass quantities.

4 Numerical Studies

This section contains timing results, comparing the R `optim` implementation of the Nelder-Mead algorithm to the Fisher scoring algorithm. In Fisher scoring, we reject steps that do not increase the loglikelihood, dividing the step size by 2 iteratively. If we cannot increase the loglikelihood along the Fisher step direction, we attempt to step along the gradient. For both Nelder-Mead and Fisher scoring, we first generate estimates using Vecchia’s approximation with $|g(i)| = 10$ nearest neighbors, then refine the estimates using $|g(i)| = 30$. In Nelder-Mead, we evaluate only the likelihood, not the gradient and Fisher information. The Fisher scoring algorithm stops when the dot product between the step and the gradient is less than $1e-4$. Default stopping criteria were used for Nelder-Mead algorithm. We simulate all datasets from the same model:

$$Y(s) = \mu + Z(s) + \varepsilon(s), \tag{22}$$

where $\mu = 0$, Z is a Gaussian process with exponential covariance function $K(s_1, s_2) = \sigma^2 \exp(-\|s_1 - s_2\|/\alpha)$, and $\varepsilon(s)$ are i.i.d. $N(0, \tau^2)$ with $\tau^2 = 0.2$. We take $(\sigma^2, \alpha) = (2, 0.3)$. Data are simulated on an evenly spaced grid of 4900 locations on $[0, 1]^2$. In addition to the exponential covariance with unknown variance and range, we estimate parameters in three covariance models that generalize the exponential:

$$K(s_1, s_2) = \frac{\sigma^2}{\Gamma(\nu)2^{\nu-1}} \left(\frac{\|s_1 - s_2\|}{\alpha} \right)^\nu \mathcal{K}_\nu \left(\frac{\|s_1 - s_2\|}{\alpha} \right) \tag{23}$$

$$K(s_1, s_2) = \frac{\sigma^2}{\Gamma(\nu)2^{\nu-1}} (\|Ls_1 - Ls_2\|)^\nu \mathcal{K}_\nu (\|Ls_1 - Ls_2\|) \tag{24}$$

$$K(s_1, s_2) = \exp \left(\sum_{j=1}^J b_j (\phi_j(s_1) + \phi_j(s_2)) \right) \frac{\sigma^2}{\Gamma(\nu)2^\nu} \left(\frac{\|s_1 - s_2\|}{\alpha} \right)^\nu \mathcal{K}_\nu \left(\frac{\|s_1 - s_2\|}{\alpha} \right). \tag{25}$$

The first is an isotropic Matérn covariance function. The second is a geometrically anisotropic Matérn covariance, with anisotropy parameterized by the 2×2 lower triangular matrix L . The third is a Matérn covariance with a nonstationary variance function. The nonstationary variances are defined in terms of pre-specified known basis functions ϕ_j and unknown parameters b_j . For identifiability purposes, the $J = 8$ basis functions are an orthogonal basis that is also orthogonal to a constant function. The orthogonal basis is formed by applying Gram-Schmidt orthogonalization to a set of Gaussian basis functions.

Excluding μ , which is estimated by profile maximum likelihood, but including the nugget variance τ^2 , the four models have 3, 4, 6, and 12 unknown parameters. Each model has a multiplicative variance parameter σ^2 . In the Nelder-Mead algorithm, we profile out σ^2 , whereas In Fisher scoring, we do not. We found that profiling σ^2 does not substantially influence convergence speed in Fisher scoring. All positive parameters are mapped to the real line by a log transform. Each model was fit to each dataset on an independent R process running on a single thread of an 8-core (16 thread) Intel Xeon W-2145 (3.7GHz, 4.5GHz Turbo) processor with 16GB RAM. Fifteen datasets were sent to each of 14 threads, yielding 210 simulated datasets.

Hisotograms of the timing results are given in Figure 1. Considering the median times, Fisher scoring is 2-3 times faster for the isotropic models (3 and 4 parameters), more than 10 times faster for the stationary Matérn model (6 parameters), and 47 times faster for the nonstationary model (12 parameters). There is also no noticeable loss in accuracy, which we can evaluate by comparing the maximum loglikelihoods returned by Fisher scoring to the loglikelihoods returned by Nelder-Mead. For the isotropic models, the two loglikelihoods never differed by more than 0.001

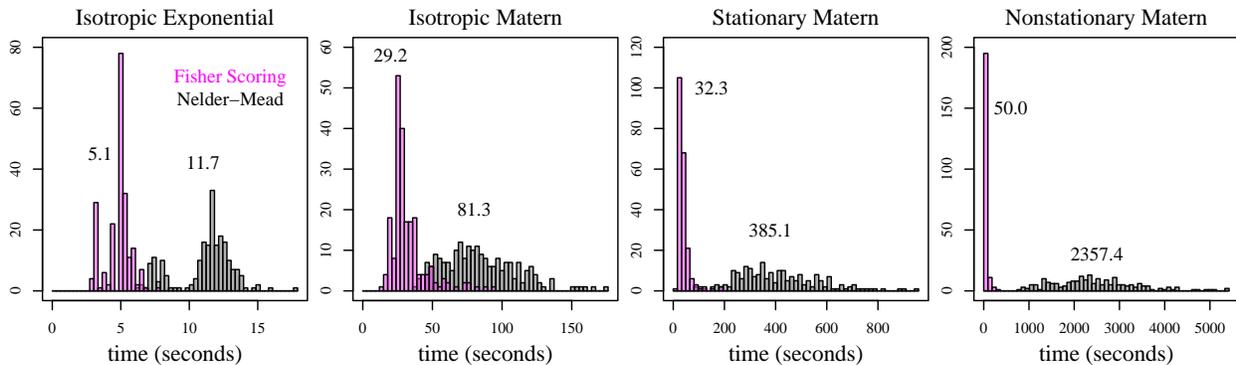


Figure 1: Results of optimization timing study. Each plot shows histograms of time (in seconds) until convergence for one of the four covariance functions over 210 replicates. The plotted numbers indicate the median time until convergence.

units. In the stationary model, the Fisher scoring loglikelihood was more than 0.01 larger than the Nelder-Mead loglikelihood in 11 of the 210 datasets, whereas the Nelder-Mead loglikelihood was never more than 0.01 larger than the Fisher scoring loglikelihood. For the nonstationary model, the Fisher scoring loglikelihoods were more than 0.01 larger than the Nelder-Mead loglikelihoods in 199 of the 210 datasets, whereas the Nelder-Mead loglikelihoods were more than 0.01 larger in just 3 of the 210 datasets. Not only does Nelder-Mead take nearly 50 times longer than Fisher scoring in the nonstationary model, it usually does not find the actual maximum likelihood estimates.

4.1 Identifiability

Surprisingly, the maximum likelihood estimate of the nugget variance can be a negative number. This is not an error of the optimization algorithm (a triumph rather); in these cases, the algorithm finds a negative nugget with a higher likelihood than any nonnegative nugget can produce. Negative nugget estimates occur most frequently when the data are evenly spaced and when the maximum likelihood estimate of the smoothness parameter is small ($\nu < 0.25$). In this scenario, the covariance function has a narrow peak at distances smaller than the minimum spacing between locations. This happened frequently enough in our testing of the Fisher scoring algorithm that we found it necessary to impose a penalty on very small values of the nugget and smoothness parameters, since it is not sensible to return negative nugget estimates. The penalties are

$$\text{pen}(\tau^2) = -0.01 \log(1 + 0.01/\tau^2), \quad \text{pen}(\nu) = -0.01 \log(1 + 0.2/\nu).$$

The likelihood function also has difficulty jointly identifying variance and range parameters when the range parameter estimate is much larger than the maximum distance between points in the dataset. This is a theoretically well-studied problem (Zhang, 2004) that no optimization routine can overcome. We have found that penalizing large variance parameters helps improve convergence of Fisher scoring without sacrificing accuracy. We used the penalty

$$\text{pen}(\sigma^2) = \log(1 + e^{\sigma^2/\tilde{\sigma}^2-6}),$$

where $\tilde{\sigma}^2$ is the estimate of the residual variance parameter in a least squares fit of the response to the constant covariate. This imposes essentially no penalty on the parameter unless it is several times larger than the least squares estimate, after which the penalty increases roughly linearly in σ^2 . These two identifiability problems can be handled more elegantly in a Bayesian framework, but we do not pursue that here because identifiability is not the focus of this paper.

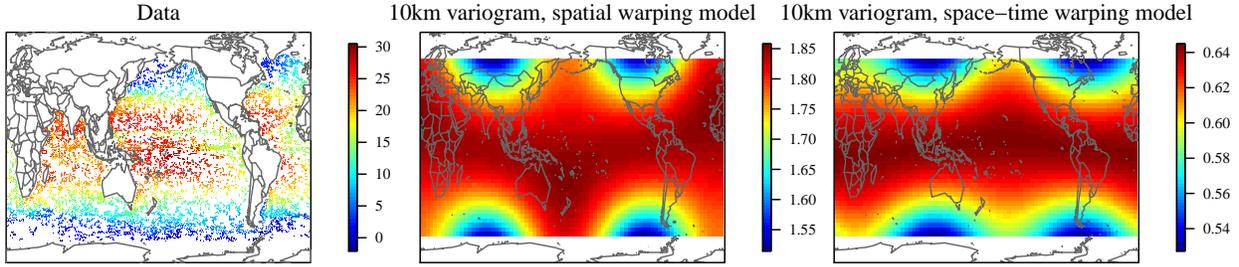


Figure 2: Plot of Argo data and variograms evaluated at 10km for the spatial-only model, and the spatial-temporal model.

5 Case Study: Argo Ocean Temperature Data

Argo is a global program that deploys floating ocean temperature sensors (International Argo Program, 2019). Each Argo float operates on a 10 day cycle, during which it descends to a 2000m depth and returns to the surface, collecting temperature and salinity measurements along the depth profile. The floats drift freely in the horizontal direction with ocean currents. As of May 2019, 3,799 floats covered the globe. We analyze a subset of the observations collected at 100 dbar (approximately 100m depth) between January 1 and March 31, 2016. Preprocessed data were provided by Mikael Kuusela and are described in more detail in Kuusela and Stein (2018). In total, there are 32,492 measurements over the three month period. The data are plotted in Figure 2.

We model the data from day t and location s on the sphere $\mathbb{S} \subset \mathbb{R}^3$ as

$$Y(s, t) = \beta_0 + \beta_1 L(s) + \beta_2 L^2(s) + Z(s + \Phi(s), t) + \varepsilon(s, t),$$

where $L(s)$ is the latitude of location s , $Z(s, t)$ is a Gaussian process with covariance function K_θ , $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is a spatial warping function, and $\varepsilon(s, t)$ are i.i.d. mean zero normals with variance τ^2 . We consider both spatial and spatial-temporal models for K_θ :

$$K_\theta((s_1, t_1), (s_2, t_2)) = \frac{\sigma^2}{2^{\nu-1}\Gamma(\nu)} (d_\alpha(s_1, s_2))^\nu \mathcal{K}_\nu(d_\alpha(s_1, s_2)),$$

$$K_\theta((s_1, t_1), (s_2, t_2)) = \frac{\sigma^2}{2^{\nu-1}\Gamma(\nu)} (d_\alpha((s_1, t_1), (s_2, t_2)))^\nu \mathcal{K}_\nu(d_\alpha((s_1, t_1), (s_2, t_2))).$$

The function d_α is Euclidean distance scaled by either a spatial range parameter or spatial and temporal range parameters

$$d_\alpha(s_1, s_2) = \frac{\|s_1 - s_2\|}{\alpha}, \quad d_\alpha((s_1, t_1), (s_2, t_2)) = \left(\frac{\|s_1 - s_2\|^2}{\alpha_1^2} + \frac{|t_1 - t_2|^2}{\alpha_2^2} \right)^{1/2}.$$

The warping function Φ is assumed to be a linear combination of the gradients of the five spherical harmonic functions of degree 2, where the gradient is with respect to the three Euclidean coordinates. We use degree 2 because the degree 0 function is constant, and the degree 1 spherical harmonics have constant partial derivatives (as a function of s), and so degree 1 warpings simply translate all points by the same vector and do not affect the covariances. We also consider the special case of $\Phi(s) = 0$ for all s , which corresponds to isotropic models in space and time. The spatial warping model has 9 parameters, while the space-time warping model has 10. The isotropic models have 4 and 5 parameters.

Model	loglikelihood		time (minutes)	
	Fisher Scoring	Nelder-Mead	Fisher Scoring	Nelder-Mead
Isotropic Spatial	-6167.675	-6167.676	2.76	7.06
Warping Spatial	-5812.902	-5812.912	6.56	79.96
Isotropic Space-Time	-237.038	-237.039	3.62	10.61
Warping Space-Time	0.000	-2.038	12.13	190.31

Table 1: Optimization results for four models fit to Argo float data. Reported loglikelihoods are differences from largest loglikelihood

We fit each model using both Fisher scoring and Nelder-Mead, with the results given in Table 1. Fisher scoring is able to fit the space-time warping model in 12.13 minutes, whereas Nelder-Mead ran for 190 minutes and returned a loglikelihood value 2.038 units lower. In the spatial-only warping model, Fisher Scoring finished in 6.56 minutes, whereas Nelder-Mead returned a loglikelihood value 0.01 lower after 80 minutes. The two methods produced nearly the same loglikelihoods on the isotropic models, with Fisher scoring running more than twice as fast. The results closely mirror the numerical study, where Fisher scoring had its largest improvements in both speed and accuracy when fitting models with the many parameters. Finally, in Figure 2, we plot $\text{Var}(Y(s, t) - Y(s+h, t))$ as a function of s , with $\|h\| = 10\text{km}$. The images show that the warping model produces an anisotropic variogram, with larger increment variances near the equator.

6 Discussion

We believe that practitioners will benefit from the availability of high quality algorithms for fitting nonstationary Gaussian process models to large spatial and spatial-temporal datasets. The methods are applicable to any covariance function that is differentiable with respect to its parameters. This is important because it separates the tasks of constructing models and developing methods for fitting the models, freeing us to select the most appropriate covariance function for the data rather than the most appropriate model for which a specialized method exists. The Fisher scoring algorithm, as well as anisotropic, nonstationary variance, and warping covariance functions, will be implemented in version 0.2.0 of the GpGp R package (Guinness and Katzfuss, 2018).

Acknowledgements

This work was supported by the National Science Foundation under grant No. 1613219 and the National Institutes of Health under grant No. R01ES027892.

References

- Finley, A., Datta, A., Banerjee, S., and Mckim, A. (2017). *spNNGP: Spatial Regression Models for Large Datasets using Nearest Neighbor Gaussian Processes*. R package version 0.1.1.
- Geoga, C. J., Anitescu, M., and Stein, M. L. (2018). Scalable Gaussian process computations using hierarchical matrices. *arXiv preprint arXiv:1808.03215*.
- Guinness, J. (2018). Permutation and grouping methods for sharpening Gaussian process approximations. *Technometrics*, 60(4):415–429.

- Guinness, J. and Katzfuss, M. (2018). *GpGp: Fast Gaussian Process Computation Using Vecchia's Approximation*. R package version 0.1.0.
- International Argo Program (2019). <http://www.argo.ucsd.edu/>. Online: accessed 2019-05-19.
- Katzfuss, M. and Guinness, J. (2017). A general framework for Vecchia approximations of Gaussian processes. *arXiv preprint arXiv:1708.06302*.
- Kuusela, M. and Stein, M. L. (2018). Locally stationary spatio-temporal interpolation of Argo profiling float data. *Proceedings of the Royal Society A*, 474(2220):20180400.
- Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *The computer journal*, 7(4):308–313.
- Stein, M. L., Chi, Z., and Welty, L. J. (2004). Approximating likelihoods for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(2):275–296.
- Vecchia, A. V. (1988). Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 50(2):297–312.
- Zhang, H. (2004). Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. *Journal of the American Statistical Association*, 99(465):250–261.