Fast Kronecker Inference in Gaussian Processes with non-Gaussian Likelihoods

Seth Flaxman[1] SFLAXMAN@CS.CMU.EDU
Andrew Gordon Wilson[1] ANDREWGW@CS.CMU.EDU
Daniel B. Neill[1] NEILL@CS.CMU.EDU
Hannes Nickisch[2] HANNES@NICKISCH.ORG
Alexander J. Smola[1,3] ALEX@SMOLA.ORG

[1]Carnegie Mellon University, [2]Philips Research Hamburg, [3]Marianas Labs

Abstract

Gaussian processes (GPs) are a flexible class of methods with state of the art performance on spatial statistics applications. However, GPs require $\mathcal{O}(n^3)$ computations and $\mathcal{O}(n^2)$ storage, and popular GP kernels are typically limited to smoothing and interpolation. To address these difficulties, Kronecker methods have been used to exploit structure in the GP covariance matrix for scalability, while allowing for expressive kernel learning (Wilson et al., 2014). However, fast Kronecker methods have been confined to Gaussian likelihoods. We propose new scalable Kronecker methods for Gaussian processes with non-Gaussian likelihoods, using a Laplace approximation which involves linear conjugate gradients for inference, and a lower bound on the GP marginal likelihood for kernel learning. Our approach has near linear scaling, requiring $\mathcal{O}(Dn^{\frac{D+1}{D}})$ operations and $\mathcal{O}(Dn^{\frac{2}{D}})$ storage, for n training data-points on a dense $D > 1$ dimensional grid. Moreover, we introduce a log Gaussian Cox process, with highly expressive kernels, for modelling spatiotemporal count processes, and apply it to a point pattern ($n =$ 233,088) of a decade of crime events in Chicago. Using our model, we discover spatially varying multiscale seasonal trends and produce highly accurate long-range local area forecasts.

1. Introduction

Gaussian processes were pioneered in geostatistics (Matheron, 1963) where they are commonly known as *kriging* models (Ripley, 1981). O'Hagan (1978) instigated their general use, pursuing applications to optimal design, curve fitting, and time series. GPs remain a mainstay of spatial and spatiotemporal statistics (Diggle & Ribeiro, 2007; Cressie & Wikle, 2011) and have gained widespread popularity in machine learning (Rasmussen & Williams, 2006).

Unfortunately, the $\mathcal{O}(n^3)$ computations and $\mathcal{O}(n^2)$ storage requirements for GPs has greatly limited their applicability. Kronecker methods have recently been introduced (Saatçi, 2011) to scale up Gaussian processes, with no losses in predictive accuracy. While these methods require that the input space (predictors) are on a multidimensional lattice, this structure is present in many spatiotemporal statistics applications, where predictors are often indexed by a grid of spatial coordinates and time.

A variety of approximate approaches have been proposed for scalable GP inference, including inducing point methods (Quiñonero-Candela & Rasmussen, 2005), and finite basis representations through random projections (Lázaro-Gredilla et al., 2010; Yang et al., 2015). Groot et al. (2014) use Kronecker based inference and low-rank approximations for GP classification, scaling to moderately sized datasets ($n < 7000$).

Our contributions include:

- We extend Kronecker methods for non-Gaussian likelihoods, enabling applications outside of standard regression settings. We use a Laplace approximation on the likelihood, proposing linear conjugate gradients for inference, and a lower bound on the GP marginal likelihood, for kernel learning. Moreover, our methodology extends to incomplete grids – caused by, for example, water or political boundaries. The Laplace approximation naturally harmonizes with Kronecker methods, providing a scalable general purpose approach for GPs with non-Gaussian likelihoods. Alternatives such as EP or VB would require another layer of approximation as the required marginal variance approximations are not tractable for large n: EP has sequential local updates, one per dat-

Proceedings of the 32^{nd} *International Conference on Machine Learning*, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

apoint, while VB can be understood as a sequence of marginal variance reweighted Laplace approximations with a smoothed effective likelihood, i.e., computing the marginal variances is required in any case.

- We perform a detailed comparison with Groot et al. (2014), and demonstrate that our new approach has significant advantages in scalability and accuracy.
- We have implemented code as part of the GPML toolbox (Rasmussen & Nickisch, 2010). See `http://www.cs.cmu.edu/~andrewgw/pattern` for updates and demos.
- We develop a spatiotemporal log Gaussian Cox process (LGCP), with highly expressive spectral mixture covariance kernels (Wilson & Adams, 2013). Our model is capable of learning intricate structure on large datasets, allowing us to derive new scientific insights from the data, and to perform long range extrapolations. This is the first use of structure learning with expressive kernels, enabling long-range forecasts, for GPs with non-Gaussian likelihoods.
- We apply our model to a challenging public policy problem, that of small area crime rate forecasting. Using a decade of publicly available date-stamped and geocoded crime reports we fit the $n = 233{,}088$ point pattern of crimes coded as "assault" using the first 8 years of data to train our model, and forecast 2 years into the future. We produce very fine-grained spatiotemporal forecasts, which we evaluate in a fully probabilistic framework. Our forecasts far outperform predictions made using popular alternatives. We interpret the learned structure to gain insights into the fundamental properties of these data.

We begin with a review of Gaussian processes in section 2, and then introduce the log-Gaussian Cox process (LGCP) in Section 3 as a motivating example for non-Gaussian likelihoods. In Section 4 we describe the standard Laplace approximation approach to GP inference and hyperparameter learning. In sections 5 and 6 we present our new Kronecker methods for scalable inference, hyperparameter learning, and missing observations. In Section 7, we detail the LGCP model specification we pursue for most experiments, including our use of spectral mixture kernels (Wilson & Adams, 2013). We detail our experiments on synthetic and real data in Section 8.

2. Gaussian processes

We assume a basic familiarity with Gaussian processes (GPs) (Rasmussen & Williams, 2006). We are given a dataset $\mathcal{D} = (\boldsymbol{y}, X)$ of targets (responses), $\boldsymbol{y} = \{y_1, \ldots, y_n\}$, indexed by predictors (inputs) $X = \{x_1, \ldots, x_n\}$. The targets could be real-valued, categorical, counts, etc., and the predictors, for example, could be spatial locations, times, and other covariates. We assume the relationship between the predictors and targets is determined by a latent Gaussian process $f(x) \sim \mathcal{GP}(m, k_\theta)$, and an observation model $p(y(x)|f(x))$. The GP is defined by its mean m and covariance function k_θ (parametrized by $\boldsymbol{\theta}$), such that any collection of function values $\boldsymbol{f} = f(X) \sim \mathcal{N}(\boldsymbol{\mu}, K)$ has a Gaussian distribution with mean $\boldsymbol{\mu}_i = m(x_i)$ and covariance matrix $K_{ij} = k(x_i, x_j|\boldsymbol{\theta})$.

Our goal is to infer the predictive distribution $p(f_*|\boldsymbol{y}, x_*)$, for any test input x_*, which allows us to sample from $p(\boldsymbol{y}_*|\boldsymbol{y}, x_*)$ via the observation model $p(y(x)|f(x))$:

$$p(f_*|\mathcal{D}, x_*, \boldsymbol{\theta}) = \int p(f_*|\boldsymbol{X}, x_*, \boldsymbol{f}, \boldsymbol{\theta}) p(\boldsymbol{f}|\mathcal{D}, \boldsymbol{\theta}) d\boldsymbol{f} \quad (1)$$

We also wish to infer the *marginal likelihood* of the data, conditioned only on kernel hyperparameters $\boldsymbol{\theta}$,

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \int p(\boldsymbol{y}|\boldsymbol{f}) p(\boldsymbol{f}|\boldsymbol{\theta}) d\boldsymbol{f}\,, \quad (2)$$

so that we can optimize this likelihood, or use it to infer $p(\boldsymbol{\theta}|\boldsymbol{y})$, for kernel learning. Having an expression for the marginal likelihood is particularly useful for kernel learning, because it allows one to bypass the extremely strong dependencies between $\boldsymbol{f}$ and $\boldsymbol{\theta}$ in trying to learn $\boldsymbol{\theta}$. Unfortunately, for all but the Gaussian likelihood (used for standard GP regression), where $p(\boldsymbol{y}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}, \Sigma)$, equations (1) and (2) are analytically intractable.

3. A motivating example: Cox Processes

In this section, we describe the log-Gaussian Cox Process (LGCP), a particularly important spatial statistics model for point process data (Møller et al., 1998; Diggle et al., 2013). While the LGCP is a general model, its use has been limited to small datasets. We focus on this model because of its importance in spatial statistics and its suitability for the Kronecker methods we propose. Note, however, that our methods are generally applicable to Gaussian process models with non-Gaussian likelihoods, such as Gaussian process classification.

An LGCP is a Cox process (inhomogeneous Poisson process with stochastic intensity) driven by a latent log intensity function $\log \lambda := f$ with a GP prior:

$$f(s) \sim \mathcal{GP}(\mu(s), k_\theta(\cdot, \cdot))\,. \quad (3)$$

Conditional on a realization of the intensity function, the number of points in a given space-time region S is:

$$y_S|\lambda(s) \sim \text{Poisson}\left(\int_{s \in S} \lambda(s)\, ds\right). \quad (4)$$

Following a common approach in spatial statistics, we introduce a "computational grid" (Diggle et al., 2013) on the observation window and represent each grid cell with its centroid, $s_1, \ldots, s_n$. Let the count of points inside grid cell i be y_i. Thus our model is a Gaussian process with a Poisson observation model and exponential link function:

$$y_i|f(s_i) \sim \text{Poisson}\left(\exp[f(s_i)]\right) . \quad (5)$$

4. Laplace Approximation

The Laplace approximation models the posterior distribution of the Gaussian process, $p(\boldsymbol{f}|\boldsymbol{y}, X)$, as a Gaussian distribution, to provide analytic expressions for the predictive distribution and marginal likelihood in Eqs. (1) and (2). We follow the exposition in Rasmussen & Williams (2006).

Laplace's method uses a second order Taylor expansion to approximate the unnormalized log posterior,

$$\Psi(\boldsymbol{f}) := \log p(\boldsymbol{f}|\mathcal{D}) \overset{\text{const}}{=} \log p(\boldsymbol{y}|\boldsymbol{f}) + \log p(\boldsymbol{f}|X) , \quad (6)$$

centered at the $\hat{\boldsymbol{f}}$ which maximizes $\Psi(\boldsymbol{f})$. We have:

$$\nabla\Psi(\boldsymbol{f}) = \nabla \log p(\boldsymbol{y}|\boldsymbol{f}) - K^{-1}(\boldsymbol{f} - \boldsymbol{\mu}) \quad (7)$$

$$\nabla\nabla\Psi(\boldsymbol{f}) = \nabla\nabla \log p(\boldsymbol{y}|\boldsymbol{f}) - K^{-1} \quad (8)$$

$W := -\nabla\nabla \log p(\boldsymbol{y}|\boldsymbol{f})$ is an $n \times n$ diagonal matrix since the likelihood $p(\boldsymbol{y}|\boldsymbol{f})$ factorizes as $\prod_i p(y_i|f_i)$.

We use Newton's method to find $\hat{\boldsymbol{f}}$. The Newton update is

$$\boldsymbol{f}^{\text{new}} \leftarrow \boldsymbol{f}^{\text{old}} - (\nabla\nabla\Psi)^{-1}\nabla\Psi . \quad (9)$$

Given $\hat{\boldsymbol{f}}$, the Laplace approximation for $p(\boldsymbol{f}|\boldsymbol{y})$ is given by a Gaussian:

$$p(\boldsymbol{f}|\boldsymbol{y}) \approx \mathcal{N}(\boldsymbol{f}|\hat{\boldsymbol{f}}, (K^{-1} + W)^{-1}) . \quad (10)$$

Substituting the approximate posterior of Eq. (10) into Eq. (1), and defining $A = W^{-1} + K$, we find the approximate predictive distribution is

$$p(f_*|\mathcal{D}, x_*, \boldsymbol{\theta}) \approx \mathcal{N}(\boldsymbol{k}_*^\top \nabla \log p(\boldsymbol{y}|\hat{\boldsymbol{f}}), k_{**} - \boldsymbol{k}_*^\top A^{-1}\boldsymbol{k}_*) \quad (11)$$

where $\boldsymbol{k}_* = [k(x_*, x_1), .., k(x_*, x_n)]^\top$ and $k_{**} = k(x_*, x_*)$.

This completes what we refer to as inference with a Gaussian process. We have so far assumed a fixed set of hyperparameters $\boldsymbol{\theta}$. For *learning*, we train these hyperparameters through marginal likelihood optimization. The Laplace approximate marginal likelihood is:

$$\log p(\boldsymbol{y}|X, \boldsymbol{\theta}) = \log \int \exp[\Psi(\boldsymbol{f})] d\boldsymbol{f} \quad (12)$$

$$\approx \log p(\boldsymbol{y}|\hat{\boldsymbol{f}}) - \frac{1}{2}\boldsymbol{\alpha}^\top K^{-1}\boldsymbol{\alpha} - \frac{1}{2}\log|I + KW| , \quad (13)$$

where $\boldsymbol{\alpha} := K^{-1}(\hat{\boldsymbol{f}} - \boldsymbol{\mu})$. Standard practice is to find the $\hat{\boldsymbol{\theta}}$ which maximizes the approximate marginal likelihood of Eq. (13), and then condition on $\hat{\boldsymbol{\theta}}$ in Eq. (11) to perform inference and make predictions.

Learning and inference require solving linear systems and determinants with $n \times n$ matrices. This takes $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ storage, using standard approaches, e.g., Cholesky decomposition (Rasmussen & Williams, 2006).

5. Kronecker Methods

Kronecker approaches have recently been exploited in various GP settings (e.g., Bonilla et al., 2007; Finley et al., 2009; Stegle et al., 2011). We briefly review Kronecker methods for efficient GPs, following Saatçi (2011), Gilboa et al. (2013), and Wilson et al. (2014), extending these methods to non-Gaussian likelihoods in the next section.

The key assumptions enabling the use of Kronecker methods is that the GP kernel is formed by a product of kernels across input dimensions and the inputs are on a Cartesian product grid (multidimensional lattice), $x \in \mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_D$. (This grid need not be regular and the $\mathcal{X}_i$ can have different cardinalities.) Given these two assumptions, the covariance matrix K decomposes as a Kronecker product of covariance matrices $K = K_1 \otimes \cdots \otimes K_D$.

Saatçi (2011) shows that the computationally expensive steps in GP regression can be accelerated by exploiting Kronecker structure. Inference and learning require solving linear systems $K^{-1}v$ and computing log-determinants $\log|K|$. Typical approaches require $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ space. Using Kronecker methods, these operations only require $\mathcal{O}(Dn^{\frac{D+1}{D}})$ operations and $\mathcal{O}(Dn^{\frac{2}{D}})$ storage, for n datapoints and D input dimensions. In Section A.1, we review the key Kronecker algebra results, including efficient matrix-vector multiplication and eigendecomposition.

Wilson et al. (2014) extend these efficient methods to partial grids, by augmenting the data with imaginary observations to form a complete grid, and then ignoring the effects of the imaginary observations using a special noise model in combination with linear conjugate gradients. Partial grids are common, and can be caused by, e.g., government boundaries, which interfere with grid structure.

6. Kronecker Methods for Non-Gaussian Likelihoods

We introduce our efficient Kronecker approach for Gaussian processes inference (Section 6.2) and learning (Section 6.3) with non-Gaussian likelihoods, after introducing some notation and transformations for numerical conditioning.

6.1. Numerical Conditioning

For numerical stability, we use the following transformations: $B = I + W^{1/2}KW^{1/2}$, $Q = W^{1/2}B^{-1}W^{1/2}$, $\boldsymbol{b} = W(\boldsymbol{f} - \boldsymbol{\mu}) + \nabla \log p(\boldsymbol{y}|f)$, and $\boldsymbol{a} = \boldsymbol{b} - QK\boldsymbol{b}$. Now $(K^{-1} + W)^{-1} = K - KQK$, from the matrix inversion lemma, and the Newton update in Eq. (9) becomes:

$$\boldsymbol{f}^{\text{new}} \leftarrow K\boldsymbol{a} \quad (14)$$

The predictive distribution in Eq. (11) becomes:

$$p(f_*|\mathcal{D}, x_*, \boldsymbol{\theta}) \approx \mathcal{N}(\boldsymbol{k}_*^\top \nabla \log p(\boldsymbol{y}|\hat{\boldsymbol{f}}), k_{**} - \boldsymbol{k}_*^\top Q\boldsymbol{k}_*) \quad (15)$$

6.2. Inference

Existing Kronecker methods do not apply to non-Gaussian likelihoods because we are no longer working solely with the covariance matrix K. We use linear conjugate gradients (LCG), an iterative method for solving linear systems which only involves matrix-vector products, to efficiently calculate the key steps of the inference algorithm in Section 4. Our full algorithm is shown in Algorithm 1. The Newton update step in Eq. (14) requires costly matrix-vector multiplications and inversions of $B = (I + W^{1/2}KW^{1/2})$. We replace Eq. (14) with the following two steps:

$$Bz = W^{-1/2}\boldsymbol{b} \tag{16}$$

$$\boldsymbol{\alpha}^{\text{new}} = W^{1/2}z \tag{17}$$

For numerical stability, we follow (Rasmussen & Williams, 2006, p. 46) and apply our Newton updates to $\boldsymbol{\alpha}$ rather than $\boldsymbol{f}$. The variable $\boldsymbol{b} = W(\boldsymbol{f} - \boldsymbol{\mu}) + \nabla \log p(\boldsymbol{y}|\boldsymbol{f})$ can still be computed efficiently because W is diagonal, and Eq. (16) can be solved efficiently for $\boldsymbol{z}$ using LCG because matrix-vector products with B are efficient due to the diagonal and Kronecker structure.

The number of iterations required for convergence of LCG to within machine precision is in practice independent of n (the number of columns in B), and depends on the conditioning of B. Solving Eq. (16) requires $\mathcal{O}(Dn^{\frac{D+1}{D}})$ operations and $\mathcal{O}(Dn^{\frac{2}{D}})$ storage, which is the cost of matrix vector products with the Kronecker matrix K. No modifications are necessary to calculate the predictive distribution in Eq. (15). We can thus efficiently evaluate the approximate predictive distribution in $\mathcal{O}(mDn^{\frac{D+1}{D}})$ where $m \ll n$ is the number of Newton steps. For partial grids, we apply the extensions in Wilson et al. (2014) without modification.

6.3. Hyperparameter learning

To evaluate the marginal likelihood in Eq. (13), we must compute $\log|I + KW|$. Fiedler (1971) showed that for Hermitian positive semidefinite matrices U and V:

$$\prod_i (u_i + v_i) \le |U + V| \le \prod_i (u_i + v_{n-i+1}) \tag{18}$$

where $u_1 \le u_2 \le \ldots \le u_n$ and $v_1 \le \ldots \le v_n$ are the eigenvalues of U and V. To apply this bound let $e_1 \le e_2 \le \ldots \le e_n$ be the eigenvalues of K and $w_1 \le w_2 \le \ldots \le w_n$ be the eigenvalues of W. Then we use that the eigenvalues of W^{-1} are $w_n^{-1} \le w_{n-1}^{-1} \le \ldots \le w_1^{-1}$:

$$\begin{aligned}\log|I + KW| &= \log(|K + W^{-1}||W|)\\ &\le \log \prod_i (e_i + w_i^{-1}) \prod_i w_i \qquad (19)\\ &= \sum_i \log(1 + e_i w_i)\end{aligned}$$

Putting this together with Equation (13) we have our bound on the Laplace approximation's log-marginal likelihood:

$$\log p(\boldsymbol{y}|X, \boldsymbol{\theta}) \ge \log p(\boldsymbol{y}|\hat{\boldsymbol{f}}) - \frac{1}{2}\hat{\boldsymbol{\alpha}}^\top K^{-1}\hat{\boldsymbol{\alpha}} - \frac{1}{2}\sum_i \log(1+e_i w_i) \tag{20}$$

We chose the lower bound as we use non-linear conjugate gradients for our learning approach to find the best $\hat{\boldsymbol{\theta}}$ to maximize the approximate marginal likelihood. We approximate the necessary gradients using finite differences.

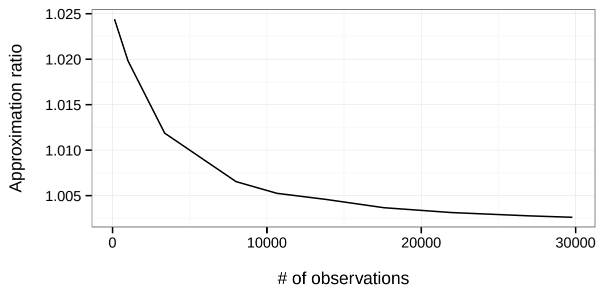
6.4. Evaluation of our Learning Approach

The bound we used on the Laplace approximation's log-marginal likelihood has been shown to be the closest possible bound on $|U + V|$ in terms of the eigenvalues of Hermitian positive semidefinite U and V (Fiedler, 1971), and has been used for heteroscedastic regression (Gilboa et al., 2014). However, its most appealing quality is computational efficiency. We efficiently find the eigendecomposition of K using standard Kronecker methods, where we calculate the eigenvalues of $K_1, \ldots, K_D$, each in time $\mathcal{O}(n^{\frac{3}{D}})$. We immediately know the eigenvalues of W because it is diagonal. Putting this together, the time complexity of computing this bound is $\mathcal{O}(Dn^{\frac{3}{D}})$. The log-determinant is recalculated many times during hyperparameter learning, so its time complexity is quite important to scalable methods.[1]

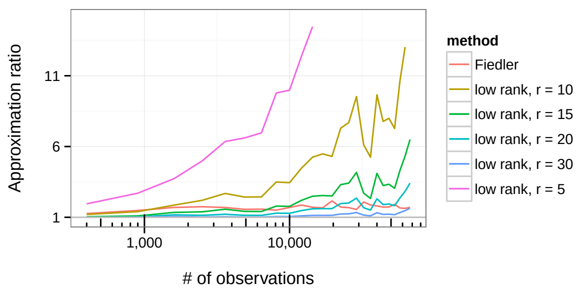
As shown in Figure 1(a), as the sample size increases the lower bound on the negative log marginal likelihood approaches the negative log marginal likelihood calculated with the true log determinant. This result makes perfect sense for our Bayesian model, because the log-determinant is a complexity penalty term defined by our prior, which becomes less influential with increasing datasizes compared to the data dependent model fit term, leading to an approximation ratio converging to 1.

Next, we compare the accuracy and run-time of our bound to a recently proposed (Groot et al., 2014) log-det approximation relying on a low-rank decomposition of K. In Figure 1(b) we generated synthetic data on an $\sqrt{n} \times \sqrt{n}$ grid and calculated the approximation ratio by dividing the approximate value $\log|I + KW|$ by the true value $\log|I + KW|$ calculated with the full matrix. Our bound always has an approximation ratio between 1 and 2, and it gets slightly worse as the number of observations increases.

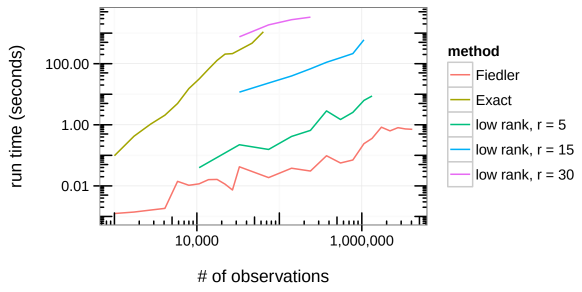
[1]An alternative would be to try to exactly compute the eigenvalues of $I + KW$ using LCG. But this would require performing at least n matrix-vector products, which could be computationally expensive. Note that this was not an issue in computing the Laplace predictive distribution, because LCG solves linear systems to within machine precision for $J \ll n$ iterations. Our approach, with the Fiedler bound, provides an approximation to the Laplace marginal likelihood, and a lower bound which we can optimize, at the cost of a single eigendecomposition of K, which is in fact more efficient than a single matrix vector product $B\boldsymbol{v}$.



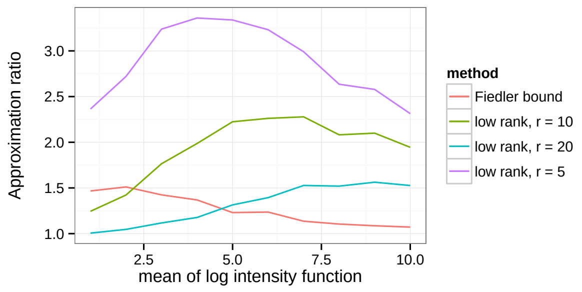
(a) Negative log-marginal likelihood approximation ratio



(b) Log-determinant approximation ratio



(c) Log-determinant runtime



(d) Log-determinant approximation ratio

Figure 1. We evaluate our bounds on the log determinant in Eq. (19) and the Laplace marginal likelihood in Eq. (20), compared to exact values and low rank approximations. In a), the approximation ratio is calculated as our bound (Fiedler) on the negative marginal likelihood divided by the Laplace negative marginal likelihood. In b) and d), the approximation ratios are calculated as a given approximation for the log-determinant divided by the exact log-determinant. In c) we compare the runtime of the various methods.

This contrasts with the low-rank approximation. When the rank r is close to $\sqrt{n}$ the approximation ratio is reasonable, but quickly deteriorates as the sample size increases.

In Figure 1(c) we compare the running times of these methods, switching to a 3-dimensional grid. The exact method quickly becomes impractical. For a million observations, a rank-5 approximation takes 6 seconds, a rank-15 approximation takes 600 seconds, while our bound takes only 0.24 seconds. While we cannot compare to the true log-determinant, our bound is provably an upper bound, so the ratio between the low rank approximation and ours is a lower-bound on the true approximation ratio. Here the low-rank approximation ratio is at least 2.8 for the rank-15 approximation and at least 30 for the rank-5 approximation.

Finally, we know theoretically that Fiedler's bound is exact when the diagonal matrix W is equal to spherical noise $\sigma^2 I$, which is the case for a Gaussian observation model.[2] Since the Gaussian distribution is a good approximation to the Poisson distribution in the case of a large mean parameter, we evaluated our log-determinant bound while varying the prior mean $\boldsymbol{\mu}$ of $\boldsymbol{f}$ from 0 to 10. As shown in Figure 1(d), for larger values of $\boldsymbol{\mu}$, our bound becomes more accurate. There is no reason to expect the same behavior from a low-rank approximation, and in fact the rank-20 approximation becomes worse as the mean of λ increases.

[2]The entries of W are equal to the second derivative of the likelihood of the observation model, so in the case of the Poisson observation model with exponential link function, $W_{ii} = -\nabla\nabla \log p(\boldsymbol{y}|\boldsymbol{f}) = \exp[\hat{\boldsymbol{f}}_i]$.

6.5. Algorithm Details and Analysis

For inference, our approach makes no further approximations in computing the Laplace predictive distribution, since LCG converges to within machine precision. Thus, unlike inducing points methods like FITC or approximate methods like Nyström, our approach to inference gives the same answer as if we used standard Cholesky methods.

Pseudocode for our algorithm is shown in Algorithm 1. Given $K_1, \ldots, K_D$ where each matrix is $n^{1/D} \times n^{1/D}$, line 2 takes $\mathcal{O}(Dn^{2/D})$. Line 5 repeatedly applies Equation (A22), and matrix-vector multiplication $(\bigotimes K_d)\, v$ reduces to D matrix-matrix multiplications VK_j where V is a matrix with n entries total, reshaped to be $n^{\frac{D-1}{D}} \times n^{\frac{1}{D}}$. This matrix-matrix multiplication is $\mathcal{O}(n^{\frac{D-1}{D}} n^{\frac{1}{D}} n^{\frac{1}{D}}) = \mathcal{O}(n^{\frac{D+1}{D}})$ so the total run-time is $\mathcal{O}(Dn^{\frac{D+1}{D}})$. Line 7 is elementwise vector multiplication which is $\mathcal{O}(n)$. Line 8 is calculated with LCG as discussed in Section 6 and takes $\mathcal{O}(Dn^{\frac{D+1}{D}})$. Lines 4 through 12 comprise the Newton update. Newton's method typically takes a very small number of iterations $m \ll n$ to converge, so the overall run-time is $\mathcal{O}(mDn^{\frac{D+1}{D}})$. Line 13 requires D eigendecompositions of matrices $K_1, \ldots, K_D$ which takes time $\mathcal{O}(Dn^{\frac{3}{D}})$ as discussed in Section 6.4. Line 14 is elementwise vector multiplication and addition so it is $\mathcal{O}(n)$. Overall, the runtime is $\mathcal{O}(Dn^{\frac{D+1}{D}})$. There is no speedup for $D = 1$, and for $D > 1$ this is nearly linear time. This is much faster than the standard Cholesky approach which requires $\mathcal{O}(n^3)$ time. The memory requirements are given by the total number of entries in $K_1, \dots K_p$: $\mathcal{O}(Dn^{\frac{2}{D}})$. This is smaller than the storage required for the n observations, so it is not a major factor. But it is worth noting because it is much less memory than required by the standard Cholesky approach of $\mathcal{O}(n^2)$ space.

Algorithm 1 Kronecker GP Inference and Learning

```
 1: **Input:** $\boldsymbol{\theta}, \boldsymbol{\mu}, K, p(\boldsymbol{y}|\boldsymbol{f}), \boldsymbol{y}$
 2: Construct $K_1, \dots, K_D$
 3: $\boldsymbol{\alpha} \leftarrow 0$
 4: **repeat**
 5:     $\boldsymbol{f} \leftarrow \boldsymbol{K\alpha} + \boldsymbol{\mu}$                # Eq. (A22)
 6:     $W \leftarrow -\nabla\nabla \log p(\boldsymbol{y}|\boldsymbol{f})$       # Diagonal
 7:     $\boldsymbol{b} \leftarrow W(\boldsymbol{f} - \boldsymbol{\mu}) + \nabla p(\boldsymbol{y}|\boldsymbol{f})$
 8:     Solve $B\boldsymbol{z} = W^{-\frac{1}{2}}\boldsymbol{b}$ with CG     # Eq. (16)
 9:     $\Delta\boldsymbol{\alpha} \leftarrow W^{\frac{1}{2}}\boldsymbol{z} - \boldsymbol{\alpha}$               # Eq. (17)
10:     $\hat{\xi} \leftarrow \arg\min_{\xi} \Psi(\boldsymbol{\alpha}+\xi\Delta\boldsymbol{\alpha})$ # Line Search
11:     $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} + \hat{\xi}\Delta\boldsymbol{\alpha}$               # Update
12: **until** convergence of $\Psi$
13: $\boldsymbol{e} = \text{eig}(K)$                      # exploit Kronecker structure
14: $Z \leftarrow \boldsymbol{\alpha}^\top(\boldsymbol{f} - \boldsymbol{\mu})/2 + \sum_i \log(1 + \boldsymbol{e}_i W_i)/2 - \log p(\boldsymbol{y}|\boldsymbol{f})$
15: **Output:** $\boldsymbol{f}, \boldsymbol{\alpha}, Z$
```

7. Model Specification

We propose to combine our fast Kronecker methods for non-Gaussian likelihoods, discussed in Section 6, with Cox processes, which we introduced in Section 3. We will use this model for crime rate forecasting in Section 8.

With large sample sizes but little prior information to guide the choice of appropriate covariance functions, we turn to a class of recently proposed expressive covariance functions called Spectral Mixture (SM) kernels (Wilson & Adams, 2013). These kernels model the *spectral density* given by the Fourier transform of a stationary kernel ($k = k(\tau) = k(x - x')$) as a scale-location mixture of Gaussians. Since mixtures of Gaussians are dense in the set of all distribution functions and Bochner's theorem shows a deterministic relationship between spectral densities and stationary covariances, SM kernels can approximate any stationary covariance function to arbitrary precision. For 1D inputs z, and $\tau = z - z'$, an SM kernel with Q components has the form

$$k(\tau) = \sum_{q=1}^{Q} w_q \exp(-2\pi^2\tau^2 v_q) \cos(2\pi\tau\mu_q) \,. \tag{21}$$

w_q is the weight, $1/\mu_q$ is the period, and $1/\sqrt{v_q}$ is the length-scale associated with component q. In the spectral domain, μ_q and v_q are the mean and variance of the Gaussian for component q. Wilson et al. (2014) showed that a combination of Kronecker methods and spectral mixture kernels distinctly enables structure discovery on large multidimensional datasets – structure discovery that is not possible using other popular scalable approaches, due to the limiting approximations in these alternatives.

For our space-time data, in which locations s are labeled with coordinates (x, y, t), we specify the following separable form for our covariance function k_θ:

$$k_\theta((x, y, t), (x', y', t')) = k_x(x, x')k_y(y, y')k_t(t, t')$$

where k_x and k_y are Matérn-5/2 kernels for space and k_t is a spectral mixture kernel with $Q = 20$ components for time. We used Matérn-5/2 kernels because the spatial dimensions in this application vary smoothly, and the Matérn kernel is a popular choice for spatial data (Stein, 1999).

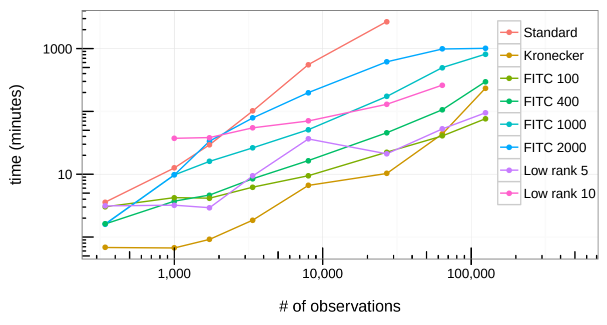
We also consider the negative binomial likelihood as an alternative to the Poisson likelihood. This is a common alternative choice for count data (Hilbe, 2011), especially in cases of overdispersion and we find that it has computational benefits. The GLM formulation of the negative binomial distribution has mean m and variance $m + \frac{m^2}{r}$. It approaches the Poisson distribution as $r \to \infty$.

8. Experiments

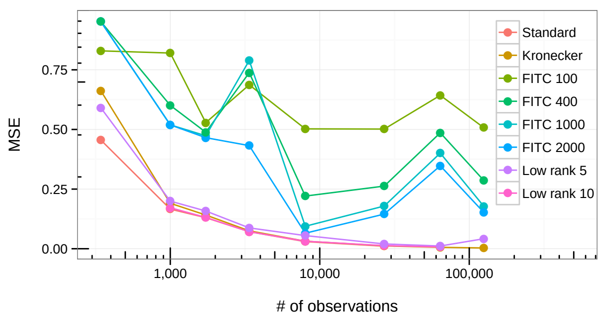
We evaluate our methods on synthetic and real data, focusing on runtime and accuracy for inference and hyperparameter learning. Our methods are implemented in GPML (Rasmussen & Nickisch, 2010). We apply our methods to spatiotemporal crime rate forecasting, comparing with FITC, SSGPR (Lázaro-Gredilla et al., 2010), low rank Kronecker methods (Groot et al., 2014), and Kronecker methods with a Gaussian observation model.

8.1. Synthetic Data

To demonstrate the vast improvements in scalability offered by our method we simulated a realization from a GP on a grid of size $n \times n \times n$ with covariance function given by the product of three SM kernels. For each realization $f(s_i)$, we then drew $y_i \sim \text{NegativeBinomial}(\exp(f(s_i) + 1)$. Using this as training data, we ran non-linear conjugate gradients to learn the hyperparameters that maximized the lower bound on the approximate marginal likelihood in equation (20), using the same product of SM kernels. We initialized our hyperparameters by taking the true hyperparameter values and adding random noise. We compared our new Kronecker methods to standard methods and FITC with varying numbers of inducing points. In each case, we used the Laplace approximation. We used 5-fold crossvalidation, relearning the hyperparameters for each fold and making predictions for the latent function values $\boldsymbol{f}_i$ on the 20% of data that was held out. The average MSE and running times for each method on each dataset are shown in Figure 2. We also calculated the log-likelihood of our posterior predictions for varying numbers of observations n for FITC-100, as shown in Table A3 in the Appendix. Our method achieved significantly higher predictive log-likelihood than FITC-100 for $n \geq 1000$.



(a) Run-time



(b) Accuracy

Figure 2. Run-time and accuracy (mean squared error) of optimizing the hyperparameters of a GP with the Laplace approximation, comparing our new Kronecker inference methods to standard GP inference, FITC, and Kronecker with low rank. The standard method has cubic running time. Each experiment was run with 5-fold crossvalidation but error bars are not shown for legibility. There is no significant difference between the standard and Kronecker methods in terms of accuracy. For grids of size $10 \times 10 \times 10$ observations and greater, FITC has significantly lower accuracy than Kronecker and standard methods.

In our final synthetic test, we simulated *100 million observations* from a GP on an 8 dimensional grid, possibly the largest dataset that has ever been modeled with a Gaussian process. This is particularly exceptional given the non-Gaussian likelihood. In this case, we had a simple covariance structure given by a squared exponential (RBF) kernel with different length-scales per dimension. We successfully evaluated the marginal likelihood in 27 minutes.

8.2. Crime Rate Forecasting in Chicago

The City of Chicago makes geocoded, date-stamped crime report data publicly available through its data portal[3]. For our application, we chose crimes coded as "assault" which includes all "unlawful attacks" with a weapon or otherwise. Assault has a marked seasonal pattern, peaking in the summer. We used a decade of data from January 1, 2004 to December 31, 2013, consisting of 233,088 reported incidents of assault. We trained our model on data from the first 8 years of the dataset (2004-2011), and made forecasts for each week of 2012 and 2013. Forecasting this far into the future goes well beyond what is currently believed to be possible by practitioners.

LGCPs have been most widely applied in the 2-dimensional case, and we fit spatial LGCPs to the training data, discretizing our data into a 288×446 grid for a total of 128,448 observations. Posterior inference and learned hyperparameter are shown in Section A.3 of the Appendix.

For our spatiotemporal forecasting, we used Spectral Mixture (SM) kernels for the time dimension, as discussed in Section 7. Specifically, we consider $Q = 20$ mixture components. For hyperparameter learning, our spatial grid was 17×26, corresponding to 1 mile by 1 mile grid cells, and our temporal grid was one cell per week, for a total of 416 weeks. Thus, our dataset of 233,088 assaults was discretized to a grid of size 183,872. Both of these sample sizes far exceed the state-of-the-art in fitting LGCPs, and indeed in fitting most GP regression problems without extreme simplifying assumptions or approximations.

[3]http://data.cityofchicago.org

To find a good starting set of hyperparameters, we used the hyperparameter initialization procedure in Wilson et al. (2014) with a Gaussian observation model. We also rescaled counts by the maximum count at that location, log-transformed, and then centered so that they would have mean 0. We ran non-linear conjugate gradient descent for 200 iterations. Using the hyperparameters learned from this stage, we switched to the count data and a negative binomial likelihood. We then ran non-linear conjugate gradient descent for another 200 iterations to relearn the hyperparameters and also the variance of the negative binomial.

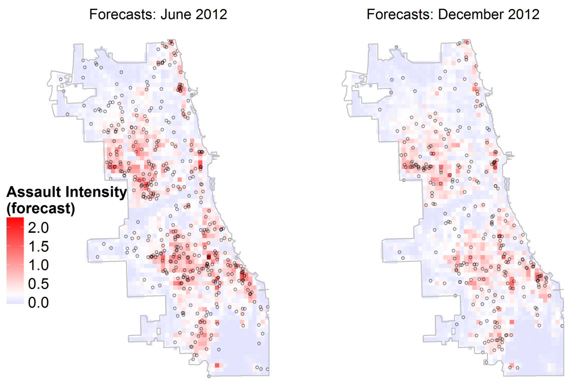


Figure 3. Local area posterior forecasts of assault one year into the future with the actual locations of assaults shown as black dots. The model was fit to data from January 2004 to December 2011, and the forecasts were made for the first week of June 2012 (left) and December 2012 (right).

The spatial hyperparameters that we learned are $\sigma^2 = 0.2231$, $\lambda_1 = 0.11$ and $\lambda_2 = 0.02$. This means that at this high resolution, with so much temporal data, there was little smoothing in space, with nearby locations allowed to be very different. Yet due to the multiplicative structure of our covariance function, our posterior inference is able to "borrow strength" such that locations with few observations follow a globally-learned time trend. We learned 60 temporal hyperparameters, and the spectral mixture components with the highest weights are shown in Figure 4, visualized in the covariance and frequency domains. We also show what posterior time series predictions would be if only a particular spectral component had been used, roughly giving an idea of the "explanatory" power of separate spectral components. We interpret the components, by decreasing weight, as follows: component 1 has a period and length-scale larger than the observation window thus picking up a decreasing trend over time. Components 2 (with period 1 month) and 4 pick up very-short-scale time variation, enabling the model to fit the observed data well. Component 3 picks up the yearly periodic trend (the spike in the spectral domain is at $0.02 = \frac{1}{52.1}$). Component 5 picks up a periodic trend with length longer than a year – 97 weeks, a feature for which we do not have any explanation. The exact hyperparameters are in Table A2 in the Appendix.

After learning the hyperparameters, we made predictions for the entire 8 years of training data and 2 years of forecasts. In Figure A6 in the Appendix we show the time series of assaults for 9 neighborhoods with our predictions, forecasts, and uncertainty intervals. Next, we rediscretized our original point pattern to a grid of size 51×78 ($n = 1.6$ million observations) and made spatial predictions 6 months and 1 year into the future, as shown in Figure 3, which also includes the observed point pattern of crimes. Visually, our forecasts are quite accurate. The accuracy and runtime of our method and competitors is shown in Table 1. The near 0 RMSE for predictions at the training data locations (i.e. the training error) for Kronecker Gaussian SM-20 indicates overfitting, while our model, Kronecker NegBinom SM-20, has a more reasonable RMSE of 0.79, out-performing the other models. The forecasting RMSE of our model was not significantly different than SSGPR or Kronecker Gaussian, while it outperformed FITC. But RMSE does not take forecasting intervals (posterior uncertainty) into account. Kronecker Gaussian and SSGPR had overly precise posterior estimates. Forecast log-likelihood is the probability of the out-of-sample data (marginalizing out the model parameters), so we can use it to directly compare the models, where higher likelihoods are better. The Kronecker Gaussian approach has the lowest forecast log-likelihood. FITC was not overconfident, but its posterior forecasts were essentially constant. Our model has the highest forecast log-likelihood, showing a balance between a good fit and correct forecasting intervals. Kronecker Gaussian methods showed the fastest run-times due to the availability of a closed form posterior. FITC was very slow, even though we only used 100 inducing points.

	KronNB SM-20	KronNB SM-20 Low Rank	KronGauss SM-20	FITC-100 NB SM-20	SSGPR-200
Training RMSE	0.79	1.13	10^{-11}	2.14	1.45
Forecast RMSE	1.26	1.24	1.28	1.77	1.26
Forecast log-likelihood	-33,916	-172,879	-352,320	-42,897	-82,781
Run-time	2.8 hours	9 hours	22 min.	4.5 hours	2.8 hours

Table 1. Kron NB SM-20 (our method) uses Kronecker inference with a negative binomial observation model and an SM kernel with 20 components. KronNB SM-20 Low Rank uses a rank 5 approximation. KronGauss SM-20 uses a Gaussian observation model. FITC 100 uses the same observation model and kernel as KronNB SM-20 with 100 inducing points and FITC inference. SSGPR-200 uses a Gaussian observation model and 200 spectral points. Carrying forward the empirical mean and variance has a forecast RMSE of 1.84 and log-likelihood of -306,430.

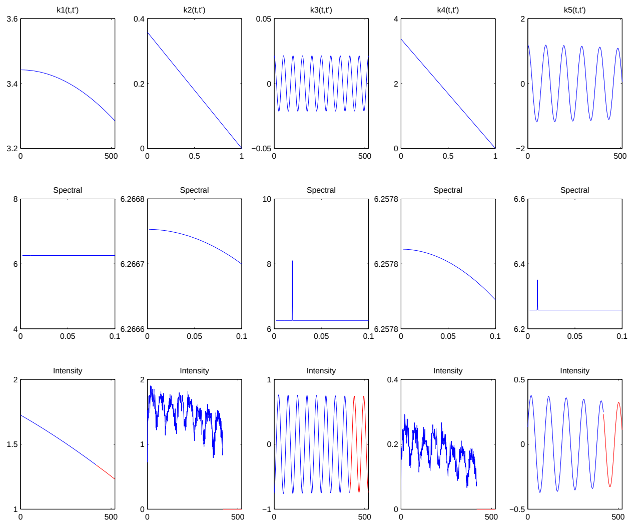


Figure 4. The five spectral mixture components with highest weights learned by our model are shown as a covariance (top) and spectral density (middle). In the bottom row, time series predictions were made on the dataset (ignoring space) using only that component. Red indicates out-of-sample forecasts.

9. Conclusion

We proposed a new scalable Kronecker method for Gaussian processes with non-Gaussian likelihoods, achieving near linear run-times for inference and hyperparameter learning. We evaluated our method on synthetic data, where it outperformed the alternatives, and demonstrated its real-world applicability to the challenging problem of small area crime rate forecasting. Our kernel learning automatically discovered multiscale seasonal trends and our inference generated highly accurate long-range forecasts, with accurate uncertainty intervals.

ACKNOWLEDGEMENTS

This work was partially supported by the National Science Foundation, grant IIS-0953330. AGW thanks ONR grant N000141410684 and NIH grant R01GM093156.

References

Bonilla, Edwin V, Chai, Kian M, and Williams, Christopher. Multi-task gaussian process prediction. In *Advances in Neural Information Processing Systems*, pp. 153–160, 2007.

Cressie, N. and Wikle, C.K. *Statistics for spatio-temporal data*, volume 465. Wiley, 2011.

Diggle, Peter and Ribeiro, Paulo Justiniano. *Model-based geostatistics*. Springer, 2007.

Diggle, Peter J, Moraga, Paula, Rowlingson, Barry, Taylor, Benjamin M, et al. Spatial and spatio-temporal log-gaussian cox processes: extending the geostatistical paradigm. *Statistical Science*, 28(4):542–563, 2013.

Fiedler, Miroslav. Bounds for the determinant of the sum of hermitian matrices. *Proceedings of the American Mathematical Society*, pp. 27–31, 1971.

Finley, Andrew O, Banerjee, Sudipto, Waldmann, Patrik, and Ericsson, Tore. Hierarchical spatial modeling of additive and dominance genetic variance for large spatial trial datasets. *Biometrics*, 65(2):441–451, 2009.

Gilboa, E., Saatci, Y., and Cunningham, J. Scaling multidimensional inference for structured gaussian processes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PP(99):1–1, 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2013.192.

Gilboa, Elad, Cunningham, John P, Nehorai, Arye, and Gruev, Viktor. Image interpolation and denoising for division of focal plane sensors using gaussian processes. *Optics express*, 22(12):15277–15291, 2014.

Groot, Perry, Peters, Markus, Heskes, Tom, and Ketter, Wolfgang. Fast laplace approximation for gaussian processes with a tensor product kernel. In *Proceedings of 22th Benelux Conference on Artificial Intelligence (BNAIC 2014)*, 2014.

Hilbe, Joseph M. *Negative binomial regression*. Cambridge University Press, 2011.

Lázaro-Gredilla, Miguel, Quiñonero-Candela, Joaquin, Rasmussen, Carl Edward, and Figueiras-Vidal, Aníbal R. Sparse spectrum gaussian process regression. *The Journal of Machine Learning Research*, 11:1865–1881, 2010.

Matheron, Georges. Principles of geostatistics. *Economic geology*, 58(8):1246–1266, 1963.

Møller, J., Syversveen, A.R., and Waagepetersen, R.P. Log gaussian cox processes. *Scandinavian Journal of Statistics*, 25(3):451–482, 1998.

Neal, Radford M. *Bayesian Learning for Neural Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996. ISBN 0387947248.

O'Hagan, Anthony. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society*, B (40):1–42, 1978.

Quiñonero-Candela, Joaquin and Rasmussen, Carl Edward. A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

Rasmussen, Carl Edward and Nickisch, Hannes. Gaussian processes for machine learning (gpml) toolbox. *The Journal of Machine Learning Research*, 11:3011–3015, 2010.

Rasmussen, Carl Edward and Williams, Christopher KI. Gaussian processes for machine learning, 2006.

Ripley, Brian D. *Spatial Statistics*. Wiley, New York, 1981.

Saatçi, Yunus. *Scalable inference for structured Gaussian process models*. PhD thesis, University of Cambridge, 2011.

Steeb, W-H and Hardy, Yorick. *Matrix calculus and Kronecker product: a practical approach to linear and multilinear algebra*. World Scientific, 2011.

Stegle, Oliver, Lippert, Christoph, Mooij, Joris M, Lawrence, Neil D, and Borgwardt, Karsten M. Efficient inference in matrix-variate gaussian models with iid observation noise. In *Advances in neural information processing systems*, pp. 630–638, 2011.

Stein, Michael L. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 1999.

Wilson, Andrew G and Adams, Ryan P. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 1067–1075, 2013.

Wilson, Andrew Gordon, Gilboa, Elad, Nehorai, Arye, and Cunningham, John P. Fast kernel learning for multidimensional pattern extrapolation. In *Advances in Neural Information Processing Systems*. MIT Press, 2014. URL `http://www.cs.cmu.edu/~andrewgw/manet.pdf`.

Yang, Z., Smola, A.J., Song, L., and Wilson, A.G. A la carte - learning fast kernels. *Artificial Intelligence and Statistics*, 2015.