Published in final edited form as:
Biometrika. 2013 March ; 100(1): 75–89. doi:10.1093/biomet/ass068.

Efficient Gaussian process regression for large datasets

ANJISHNU BANERJEE, **DAVID B. DUNSON**, and **SURYA T. TOKDAR**
Department of Statistical Science, Duke University, Box 90251, Durham, North Carolina 27708-0251, U.S.A.

Summary

Gaussian processes are widely used in nonparametric regression, classification and spatiotemporal modelling, facilitated in part by a rich literature on their theoretical properties. However, one of their practical limitations is expensive computation, typically on the order of n^3 where n is the number of data points, in performing the necessary matrix inversions. For large datasets, storage and processing also lead to computational bottlenecks, and numerical stability of the estimates and predicted values degrades with increasing n. Various methods have been proposed to address these problems, including predictive processes in spatial data analysis and the subset-of-regressors technique in machine learning. The idea underlying these approaches is to use a subset of the data, but this raises questions concerning sensitivity to the choice of subset and limitations in estimating fine-scale structure in regions that are not well covered by the subset. Motivated by the literature on compressive sensing, we propose an alternative approach that involves linear projection of all the data points onto a lower-dimensional subspace. We demonstrate the superiority of this approach from a theoretical perspective and through simulated and real data examples.

Keywords

Bayesian regression; Compressive sensing; Dimensionality reduction; Gaussian process; Random projection

1. Introduction

Consider the nonparametric regression model

$$y_i=f(x_i)+\epsilon_i, \quad \epsilon_i \sim N\left(0,\sigma^2\right) \quad (i=1,\ldots,n), \quad (1)$$

where the $(x_i, y_i) \in \mathscr{X} \times \mathbb{R}$ are predictor-response pairs measured for n subjects, $f:\mathscr{X} \to \mathbb{R}$ is an unknown regression function with $\mathscr{X} \subset \mathbb{R}^p$, and the ϵ_i are measurement errors. For a Bayesian analysis of this model, a common choice is to assign the unknown function f a Gaussian process prior, $f \sim \mathrm{GP}(\mu, c)$, where μ is the mean function and c is the covariance function, so that $E\{f(x)\} = \mu(x)$ and $\mathrm{cov}\{f(x), f(x')\} = c(x, x')$ for all $x, x' \in \mathscr{X}$. Although we focus on (1) for clarity, the methods we develop can easily be generalized to more complex Bayesian hierarchical models involving latent Gaussian processes.

Gaussian process priors are widely used because of their simplicity, flexibility and substantial theoretical support (Choi & Schervish, 2007; van der Vaart & van Zanten, 2008), with the main barrier to routine implementation being computational bottlenecks that arise

ab229@stat.duke.edu dunson@stat.duke.edu tokdar@stat.duke.edu

in moderate to large samples. The realizations of $f(\cdot)$ at the sample values $x_1, \ldots, x_n$ have a multivariate Gaussian prior, with evaluations of the posterior involving $O(n^3)$ computations unless the covariance has a special structure that can be exploited. Markov chain Monte Carlo algorithms for posterior computation that allow uncertainty in the residual variance σ^2 and parameters in the mean $\mu(\cdot)$ and covariance $c(\cdot, \cdot)$ may require such computations at every one of a large number of iterations. Another concern is declining accuracy of the estimates with increasing n, as matrix inversion becomes more unstable due to the propagation of errors arising from finite machine precision. This problem is more acute if the covariance matrix is nearly rank-deficient, which is often the case when $f(\cdot)$ is considered at nearby points.

To address these problems, $f(\cdot)$ can be approximated by a process that depends on finitely many parameters. For example, one can rely on splines, kernels or other basis expansions. Cressie & Johannesson (2008) considered fixed-rank approaches for kriging in large datasets. Banerjee et al. (2008) proposed instead a predictive process that imputes $f(\cdot)$ conditionally on the values at a finite number of knots; a similar method was proposed by Tokdar (2007) for logistic Gaussian processes. The subset of regressors method (Smola & Bartlett, 2001) is a closely related approach from the machine learning literature. These methods underestimate predictive variance, with Finley et al. (2009) proposing a bias correction in the statistics literature and Snelson & Ghahramani (2006) independently developing an essentially identical approach in machine learning. Alternative methods to adjust for underestimation of predictive variance were proposed by Seeger et al. (2003) and Schwaighofer & Tresp (2002).

Quiñonero Candela & Rasmussen (2005) proposed a unifying framework that encompasses subset-of-regressor-type approximations, showing that these can be viewed as approximations to the prior on f rather than to its posterior. To address the knot selection problem, Tokdar (2007) and Guhaniyogi et al. (2011) used a reversible jump algorithm and preferential sampling. Unfortunately, such free knot methods increase the computational burden, partially eliminating the savings that come from using a low-rank method. In machine learning, various optimization methods have been proposed for knot selection, which typically assume that the knots correspond to a subset of the data points. Such methods include online learning (Csató & Opper, 2002), greedy posterior maximization (Smola & Bartlett, 2001), the maximum information criterion (Seeger et al., 2003) and matching pursuit (Keerthi & Chu, 2006), among others.

In this article, we propose a new type of approximation that uses linear projections. The method is straightforward to implement, has a theoretical justification and provides a natural generalization of knot-based methods, with pivoted factorizations (Foster et al., 2009) and the algorithm of Finley et al. (2009) arising as special cases. Motivated by Sarlos (2006) and Halko et al. (2011), we use generalized matrix factorizations to improve numerical stability. The inspiration for our method comes from the success of linear projection techniques, such as compressed sensing (Candès et al., 2006; Donoho, 2006), in machine learning. Most of this literature focuses on reconstruction of a signal from compressive measurements, with theoretical guarantees provided on the accuracy of a point estimate under sparsity assumptions. In contrast, our goal is to accurately approximate the posterior distribution for an unknown function. We explore how these approximations affect the efficiency of Markov chain Monte Carlo summaries of covariance parameters, particularly those that control the correlation range. Our experiments suggest that predictive process-type approximations may lead to a slow mixing, which can be substantially overcome by using our approximation technique.

2. Projection approximation method

2·1. Predictive processes and the subset of regressors

Letting $y = (y_1, \ldots, y_n)^{\mathrm{T}}$, $X = (x_1, \ldots, x_n)^{\mathrm{T}}$, $f_X = \{f(x_1), \ldots, f(x_n)\}^{\mathrm{T}}$, $\mu_X = \{\mu(x_1), \ldots, \mu(x_n)\}^{\mathrm{T}}$ and $C_{XX} = \{c(x_i, x_j)\}$, expression (1) implies that

$$y \sim N_n\left(f_X, \sigma^2 I_n\right), \quad f_X \sim N_n\left(\mu_X, C_{XX}\right),$$

where $\mu(\cdot)$ and $c(\cdot, \cdot)$ may involve unknown parameters. Marginalizing out f gives

$$y \sim N_n\left(\mu_X, C_{XX} + \sigma^2 I_n\right).$$

In this article, we focus primarily on approximating C_{XX} to reduce the computational burden and ill-conditioning that arise with large n.

As a first step, we place the two approximation methods, predictive process and subset of regressors, under a common umbrella. Let $X^* = \left(x_1^*, \ldots, x_m^*\right)^{\mathrm{T}}$, with $x_j^* \in \mathscr{X}$ being the jth knot. To simplify notation, we focus on the case where $\mu_X = 0$ and let $f_* = \left\{f\left(x_1^*\right), \ldots, f\left(x_n^*\right)\right\}^{\mathrm{T}}$, $C_{**} = \left\{c\left(x_i^*, x_j^*\right)\right\}$ and

$$C_{\mathrm{aug}} = \mathrm{cov}\left\{\begin{pmatrix} f_X \\ f_* \end{pmatrix}\right\} = \begin{pmatrix} C_{XX} & C_{X*} \\ C_{*X} & C_{**} \end{pmatrix}.$$

The predictive process replaces $f(\cdot)$ by $f^{\mathrm{pp}}(\cdot) = E\{f(\cdot) \mid f_*\}$, while the subset of regressors replaces C_{XX} with $C_{XX}^{\mathrm{pp}} = C_{X*} C_{**}^{-1} C_{*X}$. These approaches are identical in that both lead to $y \sim N_n\left(0, C_{XX}^{\mathrm{pp}} + \sigma^2 I_n\right)$. This marginal form is induced by using a Gaussian process with degenerate covariance function $c^{\mathrm{pp}}(x, z) = c_{x*}^{\mathrm{T}} C_{**}^{-1} c_{*z}$, where $c_{x*} = \left\{c\left(x, x_1^*\right), \ldots, c\left(x, x_m^*\right)\right\}^{\mathrm{T}}$ and $c_{*z} = \left\{c\left(x_1^*, z\right), \ldots, c\left(x_m^*, z\right)\right\}^{\mathrm{T}}$. Using the Woodbury identity (Harville, 2008), posterior computation can be carried out in $O(nm^2)$ time, involving only inversion of the $m \times m$ matrix C_{**}.

This approach underestimates variance, since at any $x \in \mathscr{X}$, $\mathrm{var}\{f(x)\} - \mathrm{var}\{f^{\mathrm{pp}}(x)\} = \mathrm{var}\{f(x) \mid f_*\}$, which is strictly positive whenever $x \notin \left\{x_1^*, \ldots, x_m^*\right\}$. To address this problem, as well as to avoid confounding with error variance (Finley et al., 2009) and to better quantify predictive uncertainty (Quiñonero Candela & Rasmussen, 2005), a nugget term is introduced: $\epsilon^{\mathrm{pp}}(x) \sim N\left\{0, c(x, x) - c_{x,*}^{\mathrm{T}} C_{**}^{-1} c_{x,*}\right\}$, with $\mathrm{cov}\{\epsilon^{\mathrm{pp}}(x_1), \epsilon^{\mathrm{pp}}(x_2)\} = 0$ for $x_1 \neq x_2$. The modified marginal form is $y \sim N\left\{0, \sigma^2 I_n + c_{XX}^{\mathrm{pp}} + \mathrm{diag}\left(C_{XX} - c_X^{\mathrm{pp}}\right)\right\}$, which is equivalent to using a Gaussian process $f^{\mathrm{pm}}(x)$ with covariance function

$$c^{\mathrm{pm}}(x, z) = c^{\mathrm{pp}}(x, z) + \delta(x, z)\left\{c(x, z) - c^{\mathrm{pp}}(x, z)\right\},$$

where $\delta(x, z) = 1$ if $x = z$ and 0 otherwise.

2·2. Generalization: projection method

The key idea in linear projection approximation is to use $f^{\mathrm{lp}}(\cdot) = E\{f(\cdot) \mid \Phi f_X\}$ instead of $f^{\mathrm{pp}}(\cdot) = E\{f(\cdot) \mid f_*\}$, where Φ is an $m \times n$ random matrix. With $C^{\mathrm{lp}}_{XX} = (\Phi C_{XX})^{\mathrm{T}}\left(\Phi C_{XX}\Phi^{\mathrm{T}}\right)^{-1}\Phi C_{XX}$, the modified marginal form is $y \sim N_n\left(0, \sigma^2 I + C^{\mathrm{lp}}_{XX}\right)$, which is induced by a Gaussian process with covariance function

$$c^{\mathrm{lp}}(x, z) = \left(\Phi c_{x,f}\right)^{\mathrm{T}}\left(\Phi C_{XX}\Phi^{\mathrm{T}}\right)^{-1}\Phi c_{f,z}$$

for $x, z \in \mathscr{X}$, where $c_{x,f} = \{c(x, x_1), \ldots, c(x, x_n)\}^{\mathrm{T}}$ and $c_{f,z} = \{c(x_1, z), \ldots, c(x_n, z)\}^{\mathrm{T}}$. As in § 2·1, to offset the underestimation of variance we introduce a nugget term and let $f^{\mathrm{lm}}(\cdot)$ denote the modified linear projection approximation having covariance function

$$c^{\mathrm{lm}}(x, z) = c^{\mathrm{lp}}(x, z) + \delta(x, z)\left\{c(x, z) - c^{\mathrm{lp}}(x, z)\right\}. \quad (2)$$

2·3. Relationships with other approaches

In this subsection, we explore some special cases of our projection representation and show its connections with other representations. Throughout the article we impose the restriction $\Phi \in \mathscr{C}$, where $\mathscr{C}$ is the class of matrices of full row-rank and with row-norm equal to unity, to avoid arbitrary scale problems.

If Φ is an $m \times n$ submatrix of an $n \times n$ permutation matrix, we obtain a predictive process whose knots are an m-dimensional subset of $\{x_1, \ldots, x_n\}$. When $m = n$ and Φ is nonsingular, it is trivial to verify that $C^{\mathrm{lp}}_{XX} = C_{XX}$, so that one obtains the full model.

Let $\mathscr{H}\{f_X, \Phi\}$ denote the Hilbert space spanned by the m-dimensional vector Φf_X, equipped with the inner product $\langle f_1, f_2\rangle = E(f_1 f_2)$ for any $f_1, f_2 \in \mathscr{H}\{f_X, \Phi\}$. The orthogonal projection of f onto $\mathscr{H}\{f_X, \Phi\}$ is $f^{\mathrm{opt}} = \arg\min_{h \in \mathscr{H}\{f_X, \Phi\}} \|f - h\| = E\{f(x) \mid \Phi f_X\}$, with similar results in kriging theory (Stein, 1999). Hence, our projection approach is optimal in this sense. The projection f^{opt} is a function of Φ and, for any $\mathscr{P} \subset \mathscr{C}$, $\min_{\Phi \in \mathscr{C}} \|f - f^{\mathrm{opt}}\| \leq \min_{\Phi \in \mathscr{P}} \|f - f^{\mathrm{opt}}\|$. As the predictive process-type approaches in § 2·1 restrict Φ to a subset of $\mathscr{C}$, the best possible approximation using such approaches is never better than that given by the best projection approximation.

The Nyström scheme (Drineas & Mahoney, 2005) considers rank-m approximations to an $n \times n$ positive semidefinite matrix A using an $m \times n$ matrix B, by giving an approximate generalized Cholesky decomposition of A as GG^{T} where $G = (BA)^{\mathrm{T}}(BAB^{\mathrm{T}})^{-1/2}$; this scheme encompasses as special cases other well-known approximate decompositions such as the partial Cholesky decomposition and the partial spectral decomposition. The accuracy of the Nyström scheme depends on how well the column space of B approximates the column space of A. The projection C^{lp}_{XX} corresponds to a Nyström approximation to C_{XX}, with $G = \{\Phi C_{XX}\}^{\mathrm{T}}\{\Phi C_{XX}\Phi^{\mathrm{T}}\}^{-1/2}$. The partial Cholesky decompositions for the covariance matrices, advocated in Foster et al. (2009) for the approaches described in § 2·1, therefore arise as special cases of our projection representation.

For any stochastic process f and any m, the truncated Karhunen–Loève expansion (Adler, 1990), $f^{\mathrm{tr}}(x) = \sum_{i=1}^{m} \eta_i(\lambda_i)^{1/2} e_i(x)$, provides the optimal m-term approximation with respect to expected squared error (Ghanem & Spanos, 2003). Here, the λ_i and e_i are eigenvalues and eigenfunctions, respectively, of the covariance function $c(\cdot\,,\,\cdot)$. The matrix $E = \{e_j(x_i)\}$ is approximately orthogonal since the eigenfunctions are orthogonal. If E were exactly orthogonal, we would have $f_X^{\mathrm{lp}}=f_X^{\mathrm{tr}}$ for Φ equal to the eigenvectors corresponding to the m largest eigenvalues of C_{XX}. In this case, $\mathrm{cov}\left(f_X^{\mathrm{tr}}\right)=\mathrm{cov}\left(f_X^{\mathrm{lp}}\right)=EDE^{\mathrm{T}}$ where $D=\mathrm{diag}(\lambda_1, \ldots, \lambda_m)$, which is a rank-$m$ spectral decomposition of C_{XX}, also corresponding to the m-term truncated Mercer expansion (Grigoriu, 2002) of $c(\cdot\,,\,\cdot)$.

3. Matrix approximations and projection construction

3·1. Reduced-rank matrix approximations

We introduce stochastic matrix approximation techniques that enable us to calculate nearly optimal projections. Let $\|\cdot\|_2$ and $\|\cdot\|_{\mathrm{F}}$ denote, respectively, the spectral and Frobenius norms for matrices, and let C be any $n \times n$ positive definite covariance matrix. Consider the spectral decomposition $C=ULU^{\mathrm{T}}$, where L is a diagonal matrix whose elements are the eigenvalues in descending order of magnitude and U is the matrix of eigenvectors. Partition C as

$$C=[U_m U_{(n-m)}]\begin{bmatrix} L_{mm} & 0 \\ 0 & L_{(n-m)(n-m)} \end{bmatrix}[U_m U_{(n-m)}]^{\mathrm{T}}.$$

From the Eckart–Young theorem (Stewart, 1993), the best rank-m approximation to C, with respect to both $\|\cdot\|_2$ and $\|\cdot\|_{\mathrm{F}}$, is $C_m=U_m L_{mm} U_m^{\mathrm{T}}$. Recall that our projection scheme replaces the covariance matrix C by $C^{\mathrm{lp}}=(\Phi C)^{\mathrm{T}}(\Phi C \Phi^{\mathrm{T}})^{-1}\Phi C$, where Φ is a projection matrix. The best projection approximation is obtained when we choose $\Phi=U_m^{\mathrm{T}}$, since in that case $C^{\mathrm{lp}}=C_m$.

The problem is that obtaining the spectral decomposition is as burdensome as computing the matrix inverse, with $O(n^3)$ computations involved. Recent articles on matrix approximation and matrix completion have proposed random approximation schemes which give near-optimal performance at lower computational cost (Sarlos, 2006; Halko et al., 2011). We can use these approaches to address the following questions: (i) given a fixed rank m, find the near-optimal projection for that rank and the corresponding error; (ii) given a fixed accuracy level $1 - \epsilon$, find the near-optimal rank for which we can achieve this accuracy, along with the corresponding projection.

First, consider question (i), the fixed-rank problem. For an $n \times r$ random matrix Ω with independent entries taken from some continuous distribution, $C\Omega$ gives r linearly independent vectors in the column space of C with probability 1. There can be at most n such independent vectors, since the dimension of the column space is n. As mentioned earlier, when we evaluate the Gaussian process on a fine grid of points, the covariance matrix C is often severely rank-deficient, and we should be able to accurately capture its column space with the span of $m \ll n$ vectors.

The first issue that arises is how to choose the random matrix Ω. The product $C\Omega$ embeds the matrix C from $\mathbb{R}^{n\times n}$ into $\mathbb{R}^{n\times r}$. Embeddings with low distortion are well studied in the compressive sensing literature, where Johnson–Lindenstrauss transforms (Johnson et al., 1986; Dasgupta & Gupta, 2003) are popular. A matrix Ω of order $n \times r$ is said to be a Johnson–Lindenstrauss transform for a subspace V of $\mathbb{R}^n$ if $|\|\Omega^{\mathrm{T}}v\| - \|v\||$ is small for all $v \in V$ with high probability. For a precise definition of the transform, see Definition 1 in Sarlos (2006). The Johnson–Lindenstrauss property is satisfied if the elements of Ω are independent draws from $N(0, r^{-1})$ or appropriate Rademacher or uniform distributions. As we have found these choices to produce essentially equivalent results, in agreement with the compressive sensing literature (Candès et al., 2006; Donoho, 2006), we shall focus on the $N(0, r^{-1})$ case for simplicity. Having formed the embedding $C\Omega$, we find Φ using Algorithm 1, which combines ideas from Sarlos (2006) with Algorithm 5.5 in Halko et al. (2011).

Algorithm—1. Given a positive definite matrix C of order $n \times n$ and a randomly generated Johnson–Lindenstrauss matrix Ω of order $n \times r$, find the projection matrix Φ of order $m \times n$ which approximates the column space and compute the approximate spectral decomposition via Nyström approximation with Φ.

Step 1. Form the matrix product $C\Omega$.

Step 2. Compute Φ^{T}, the left factor of the rank-m spectral projection of the small matrix $C\Omega$.

Step 3. Form $C_1 = \Phi C \Phi^{\mathrm{T}}$.

Step 4. Perform a Cholesky factorization of $C_1 = BB^{\mathrm{T}}$.

Step 5. Calculate the Nyström factor $C_2 = C\Phi^{\mathrm{T}}(B^{\mathrm{T}})^{-1}$.

Step 6. Compute a singular value decomposition for $C_2 = UDV^{\mathrm{T}}$.

Step 7. Calculate the approximate spectral decomposition for $C \approx C_{\mathrm{fr}} = UD^2U^{\mathrm{T}}$

We have the following result on the approximation accuracy of Algorithm 1; it is a modification of Theorem 14 in Sarlos (2006).

Theorem 1—*Take any* $0 < \epsilon \leq 1$ *and let* $r = \lfloor m/\epsilon \rfloor$. *Obtain* C_{fr} *from Algorithm* 1 for the positive definite matrix C, and let C_m be the best rank-m approximation to C in terms of $\|\cdot\|_F$. Then

$$\mathrm{pr}\left\{\|C - C_{\mathrm{fr}}\|_{\mathrm{F}} \leq (1+\epsilon)\, \|C - C_m\|_{\mathrm{F}}\right\} \geq \frac{1}{2}.$$

A parallel version of Algorithm 1 can be implemented by running Steps 1 and 2 in parallel for several copies of the matrix Ω; with $-\log\,\eta$ copies, we can sharpen the probability in Theorem 1 to $1 - \eta$. In our implementations of Algorithm 1 we used $r = m$. The algorithm decomposes the matrix $\Phi C \Phi^{\mathrm{T}}$, which involves $O(m^3)$ operations. The matrix multiplications, for instance in computing C_1, are $O(n^2m)$; this is the additional cost we pay to have the projection generalization of the algorithms in § 2·1. Matrix multiplication can be done in parallel, which is the default approach in standard linear algebra packages. Our results indicate that added computational complexity is negligible for the projection algorithm, compared with the techniques in § 2·1, and it has much better numerical stability. With the fixed-error algorithm below, we achieve lower processor times than with the predictive process-type approaches of § 2·1, because the rank required to achieve the target error level is substantially smaller with projection.

We now turn to question (ii), where the accuracy level is fixed. The $n \times n$ eigenvector matrix U from the spectral decomposition of C spans the column space of C, as is clear from the identity $C = UU^{\mathrm{T}}C$. In general, we consider the column space approximator $\Phi^{\mathrm{T}}\Phi C$ for easier error evaluation. The best rank-m column space approximator is the same as the rank-m spectral decomposition, since $U_m U_m^{\mathrm{T}} C = C_m$. It then suffices to search for good column space approximators, since Lemma 4 of Drineas & Mahoney (2005) and the discussion in § 5.4 of Halko et al. (2011) show that the error associated with the Nyström approximator is at worst as small as the error in the column space approximation and, empirically, is often substantially smaller. We need only find the projection matrix Φ for the column space approximation given the target error level, and computation of the approximate spectral decomposition using this Φ proceeds as in Steps 3–7 of Algorithm 1. We can obtain Φ to meet any target error level by modifying appropriately the steps in Algorithm 4.2 of Halko et al. (2011); this is summarized in our Algorithm 2 as follows.

Algorithm 2—Given a positive definite matrix C of order $n \times n$ and a target error $\epsilon > 0$, find the projection matrix Φ of order $m \times n$ which gives $\|C - \Phi^{\mathrm{T}}\Phi C\| < \epsilon$ with probability $1 - n/10^r$.

```
Step 1. Initialize $j = 0$ and $\Phi^{(0)} = []$, the $0 \times n$ empty matrix.
Step 2. Draw $r$ length-$n$ random vectors $\omega^{(1)}, \ldots, \omega^{(r)}$ with independent entries from $N(0, 1)$.
Step 3. Compute $\kappa^{(i)} = C\omega^{(i)}$ for $i = 1, \ldots, r$.
Step 4. Is $\max_{i=1,\ldots,r}(\|\kappa^{(j+i)}\|) < \{(\pi/2)^{1/2}\epsilon\}/10$? If yes, go to Step 11. If no, go to Step 5.
Step 5. Recompute $j = j + 1$, $\kappa^{(j)} = [I - \{\Phi^{(j-1)}\}^{\mathrm{T}}\Phi^{(j-1)}]\kappa^{(j)}$ and $\phi^{(j)} = \kappa^{(j)}/(\|\kappa^{(j)}\|)$.
Step 6. Set $\Phi^{(j)} = [\{\Phi^{(j-1)}\}^{\mathrm{T}}\ \phi^{(j)}]^{\mathrm{T}}$.
Step 7. Draw a length-$n$ random vector $\omega^{j+r}$ with independent entries from $N(0, 1)$.
Step 8. Compute $\kappa^{(j+r)} = [I - \{\Phi^{(j)}\}^{\mathrm{T}}\Phi^{(j)}]C\omega^{(j+r)}$.
Step 9. Recompute $\kappa^{(i)} = \kappa^{(i)} - \phi^{(j)}\langle \phi^{(j)}, \kappa^{(i)} \rangle$ for $i = (j + 1), \ldots, (j + r - 1)$.
Step 10. Go back to the target error check in Step 4.
Step 11. If $j = 0$, output $\Phi = \{\|\kappa^{(1)}\|^{-1}\kappa^{(1)}\}^{\mathrm{T}}$; else output $\Phi = \Phi^{(j)}$.
```

Step 9 in Algorithm 2 is not essential, but it ensures better stability when the κ vectors become very small. In our implementations of Algorithm 2 we used r such that $n \times 10^{-r} \approx 0{\cdot}1$, to maintain a probability of 0·9 of achieving the error level. The computational requirements of Algorithm 2 are similar to those of Algorithm 1; for more details we refer the reader to Halko et al. (2011, § 4.4). Posterior fit and prediction in Gaussian process regression usually involves integrating out the Gaussian process, as indicated in § 4. We end this subsection with another theorem, which states that the target error in the prior covariance matrix approximation controls the error in the marginal distribution of the data, integrating out the Gaussian process.

Theorem 2—*Let* $\pi_{\mathrm{full}} = N_n(\mu_X, C_{XX})$ be the marginal distribution of the response vector y under the original model, and let $\pi_{\mathrm{lp}} = N_n\left(\mu_X, C_{XX}^{\mathrm{lp}}\right)$ *denote its linear projection approximation. If* $\|C_{XX} - C_{XX}^{\mathrm{lp}}\|_{\mathrm{F}} \le \epsilon$, which is the error in approximation of the covariance matrix, then

$$d_{\mathrm{KL}}\left(\pi_{\mathrm{full}}, \pi_{\mathrm{lp}}\right) \le \left\{n + \left(\frac{n}{\sigma}\right)^2\right\}\epsilon$$

where $d_{\mathrm{KL}}(\cdot\,,\cdot)$ denotes the Kullback–Leibler divergence between probability densities.

3·2. Numerical stability and examples

The covariance matrix for a smooth Gaussian process tracked at a dense set of locations will be ill-conditioned and nearly rank-deficient. With propagation of round-off errors due to finite-precision arithmetic, the inverses may be highly unstable and severely degrade the quality of inference. Given two approximations with similar accuracy, as measured by the distance in Frobenius norm from the original matrix, it would be preferable to use the approximation with better conditioning. How well a covariance matrix C is conditioned can be measured by the condition number σ_l/σ_s, where σ_l and σ_s are the largest and smallest eigenvalues of C (Dixon, 1983). Larger condition numbers indicate greater numerical instability. We show empirically how conditioning can be improved greatly with the projection approximation over knot-based schemes, when considering either a fixed-rank or a fixed-target-error approach.

In this subsection and the applications presented in § 4, we use the squared exponential covariance function for convenience. Such covariance functions enjoy substantial theoretical support (Choi & Schervish, 2007; Tokdar, 2007; van der Vaart & van Zanten, 2008) and are appealing in many regression settings in favouring smooth realizations. However, expensive matrix inversion, ill-conditioning and inefficiency in accommodating unknown parameters in the covariance remain significant problems. The methods developed in this article also apply to general covariance functions, such as the Matérn covariance function, which are plagued by similar computational issues. For comparison purposes, we use two knot-based approaches: the standard predictive process approach (Banerjee et al., 2008); and the pivoted Cholesky approach for knot selection, proposed in unpublished work by the third author. The second of these gives the best performance, to our knowledge, for knot-based strategies.

Consider the covariance function $c(x, y) = \exp\{-(x - y)^2\}$, evaluate it over a uniform grid of 1000 points in [0·1, 100], and consider the resulting 1000 × 1000 covariance matrix C. The condition number of C is roughly $1{\cdot}0652 \times 10^{20}$, which indicates that the matrix is severely ill-conditioned. We now apply Algorithm 1 with $r = m$. The results are summarized in Table 1 for selected values of m. The projection approach clearly has better approximation accuracy than the other methods; this superiority becomes more marked as the dimension of the approximation increases. The condition numbers for the projection scheme can be dramatically better than those for the other two approaches, indicating superior numerical stability. We obtain similar results for other choices of m, but these are omitted from the table for brevity.

We also assess the computational efficiency in achieving a target error level. For the projection approach, we implement Algorithm 2. As a gold standard, it is useful to know the smallest rank such that the target error can be achieved using knowledge of the spectral decomposition. For this reason, we consider matrices of the form $C = ULU^{\mathrm{T}}$, where U is an orthonormal matrix and L is diagonal. The diagonal elements of L, namely the eigenvalues of C, are taken to decay at exponential rates, which is the case for smooth covariance functions (Frauenfelder et al., 2005), so that the ith element is $l_{ii} = \exp(-i\lambda)$. For the simulations tabulated, we use $\lambda = 0{\cdot}5$, $0{\cdot}08$ and $0{\cdot}04$, with U being filled with independent standard normal entries and then orthonormalized. We then pretend that this decomposition is unknown and apply Algorithm 2 for projections, as well as the two knot-based approaches, to achieve various error levels ϵ in Frobenius norm for different values of n. The results are shown in Table 2. For all the different values of ϵ and n, our projection approach achieves the desired target error level with lower rank. The times taken by the three methods were comparable, but the projection approach has lower time requirements when the rank differences become significant.

Besides the obvious advantages of computational efficiency and stability, smaller target ranks also imply lower memory requirements, and this is an important consideration when the sample size n is large. We point out that the times needed to perform matrix norm calculations for checking the target error condition in the knot-based approaches are not included in the times shown in Table 2. The projection approach benefits from the default parallel implementation of matrix multiplication in Matlab. Lower-level implementations of the algorithms would require parallel matrix multiplication to achieve similar computation times. With a graphics processor implementation for parallel matrix multiplication, projection approximation can be even more efficient.

4. Parameter estimation and illustrations

4·1. Bayesian inference for the parameters

An important component of implementing Gaussian process regression is estimation of the unknown parameters of the covariance function of the process. As mentioned earlier, we focus on the squared exponential function $c(x,y)=\theta_2^{-1}\exp\left(-\theta_1\|x-y\|^2\right)$ for simplicity, where θ_1 and θ_2 are unknown parameters corresponding to the range and inverse scale, respectively. We use Bayesian techniques for inference to fully explore the posterior over all possible values of these parameters.

For Bayesian inference, we specify prior distributions for each of the unknown parameters, namely θ_1, θ_2 and σ^2, the variance of the measurement noise in equation (1). In place of (1), with the projection approach we have

$$y_i=f^{\mathrm{lm}}(x_i)+\epsilon_i \quad (i=1,\ldots,n).$$

Using the bias-corrected form for the projection approximation, the prior for the unknown function evaluated at the data points is $f_X^{\mathrm{lm}}|\theta_1,\theta_2\sim N_n\left(0,C_{XX}^{\mathrm{lm}}\right)$ where $C_{XX}^{\mathrm{ln}}=C_{XX}^{\mathrm{lp}}+\mathrm{diag}\left(C_{XX}-C_{XX}^{\mathrm{lp}}\right)$, with $\mathrm{diag}\left(C_{XX}-C_{XX}^{\mathrm{lp}}\right)$ being the diagonal matrix as obtained from equation (2) for variance augmentation. Letting $\tau=\sigma^{-2}$ and choosing conjugate priors, we take $\tau\sim\mathrm{Ga}(a_1,b_1)$, $\theta_2\sim\mathrm{Ga}(a_2,b_2)$ and $\theta_1\sim\sum_{h=1}^{t}t^{-1}\delta_{s_h}$, a discrete uniform distribution with atoms $\{s_1,\ldots,s_t\}$. Here the gamma density $\mathrm{Ga}(a,b)$ is parameterized to have mean a/b and variance a/b^2. The priors being conditionally conjugate, we can easily derive the full conditional distributions necessary to implement a Gibbs sampling scheme for the quantities of interest:

$$\begin{aligned}
f_X^{\mathrm{lm}}|- &\sim N_n\left[\left\{\left(C_{XX}^{\mathrm{lm}}\right)^{-1}+\tau I\right\}^{-1}y,\left\{\left(C_{XX}^{\mathrm{lm}}\right)^{-1}+\tau I\right\}^{-1}\right],\\
\tau|- &\sim \mathrm{Ga}\left\{a_1+\tfrac{n}{2},b_1+\tfrac{1}{2}\left(y-f_X^{\mathrm{lm}}\right)^{\mathrm{T}}\left(y-f_X^{\mathrm{lm}}\right)\right\},\\
\theta_2|- &\sim \mathrm{Ga}\left\{a_2+\tfrac{n}{2},b_2+\tfrac{1}{2}\left(f_X^{\mathrm{lm}}\right)^{\mathrm{T}}\left(\theta_2C_{XX}^{\mathrm{lm}}\right)^{-1}f_X^{\mathrm{lm}}\right\},\\
\mathrm{pr}(\theta_1=s_i|-) &= k\left\{\det\left(C_{XX}^{\mathrm{lm}}\right)\right\}^{-1/2}\exp\left\{-\tfrac{1}{2}\left(f_X^{\mathrm{lm}}\right)^{\mathrm{T}}\left(C_{XX}^{\mathrm{lm}}\right)^{-1}f_X^{\mathrm{lm}}\right\},
\end{aligned}$$

where k is a constant such that $\sum_{i=1}^{t}\mathrm{pr}(\theta_1=s_i|-)=1$. We can integrate out the Gaussian process f_X^{lm} from the model to obtain $y\sim N_n\left(0,C_{XX}^{\mathrm{lm}}+\tau^{-1}I\right)$. This form is useful for prediction and fitting. We show in the Appendix some relevant computational details for matrix inversion using the Woodbury identity.

The covariance C_{XX}^{lm} depends on the parameters θ_1 and θ_2 as well as a random projection matrix Φ obtained via Algorithm 2. Because θ_2 is a scaling parameter, it leaves Φ from Algorithm 2 unaffected if the target error is specified on a relative scale as $\epsilon=\epsilon_1/\theta_2$ with ϵ_1

prespecified. We adopt this strategy in our implementations reported below. Because we use a finite set of candidate values for θ_1, we precompute a Φ using Algorithm 2 for each distinct value of $\theta_1 \in s_1, \dots, s_t\}$ prior to implementing our Gibbs sampler, so as to bypass the computational burden involved in implementing Algorithm 2 and conducting corresponding matrix inversions at each Gibbs iteration. Related griddy Gibbs strategies are routinely used in posterior computation for Gaussian process models.

4·2. Illustrations

We first consider a simulated data example, assuming the true functions are known on $\mathscr{X} = [0, 1]$. We consider two functions with different degrees of smoothness: a smooth function, $f_1^{\text{true}}(x) = 10\exp\left\{-0.5(x-0.5)^2\right\}$, and a wavy function, represented by $f_2^{\text{true}} = \sum_{i=1}^{5} \exp\left[-200\{x-(2i-1)/10\}^2\right]$. For each function, we take 10 000 equally spaced points in [0, 1] and add random Gaussian noise to each point to obtain the observed data y. For the smooth function, the true noise variance used is 0·1; for the wavy function, the true noise variance used is 0·01. We randomly selected 9000 points for model fitting, while the rest were used for validation.

In all our analyses, we used the squared exponential covariance function with prior specifications and implementation as described in § 3. We chose a relative target error level of 0·1 for the smooth function and a relative target error level of 0·01 for the wavy function. For the measurement noise, we used hyperparameters a_1 and b_1 such that the mean is approximately equal to the estimated noise precision from ordinary least-squares regression. In particular, for the smooth function, we took $a_1 = 1$ and $b_1 = 10$. Hyperparameter choices for covariance function parameters were guided by some trial runs; we used a grid of 2000 equispaced points in [0, 2] for θ_1 and took $a_2 = 2$ and $b_2 = 20$ for θ_2. We ran Gibbs samplers for 10 000 iterations, discarding the first 500 as burn-in. We calculated the predicted values for the held-out set, posterior means of the parameters, and the average rank required to achieve the target accuracy over the iterations. Effective sample size was calculated using the convergence diagnostics R package (R Development Core Team, 2012) CODA (Plummer et al., 2006).

Table 3 summarizes the results obtained by using our approach and the two knot-based approaches of § 3·2. Prior specifications and implementations are the same in each case, with only the approximated covariance differing. It is evident from Table 3 that our projection approximation offers substantial gains in predictive accuracy while requiring a much smaller rank than the knot-based approaches. It also improves the effective sample sizes of the covariance parameters, particularly of θ_1, which has a nonlinear effect on the covariance function and is usually difficult to explore through Markov chain sampling. With the predictive process-type approaches, we would need many more Markov chain Monte Carlo iterations to achieve similar effective sample sizes, leading to increased computational cost. These comparisons are fairly robust, in our experience. We obtained similar results with other choices of the true function, corresponding to various smoothness levels, but these are not reported here.

We also consider two large real-data examples, which have been analysed previously with several reduced-rank Gaussian process regression methods. The first is the abalone dataset (Frank & Asuncion, 2010): the interest is in modelling the age of an abalone, given other attributes which are thought to be nonlinearly related to age. The dataset consists of 4000 training and 177 test cases. The x variable is 8-dimensional and includes measurements such as length of the abalone shell, weight of the shell with and without meat, and so on. The second dataset we consider is the Sarcos robot arm data (Vijayakumar & Schaal, 2000), where one is interested in the torque of the arm given 21 other measurements, namely 7 joint positions, 7 joint velocities and 7 joint accelerations. This dataset consists of 44 484 training and 4449 test cases.

For both datasets, we use Algorithm 2 with a relative target error level of 0·01. The hyperparameters are chosen in a similar fashion to the simulated examples. This leads to $a_1 = 1$ and $b_1 = 0{\cdot}1$ for the abalone dataset, and $a_1 = 2$ and $b_1 = 0{\cdot}1$ for the Sarcos robot arm data. The grid for θ_1 in either case is made up of 2000 equispaced points in [0, 2]; for θ_2, in the abalone case we have $a_2 = 1$ and $b_2 = 1$, while for the robot arm we have $a_2 = 1$ and $b_2 = 0{\cdot}75$. The Gibbs sampler for the abalone dataset is run for 10 000 iterations, with 1000 discarded as burn-in; for the robot arm dataset, the Gibbs sampler is run for 2000 iterations, with 500 discarded as burn-in.

The results are tabulated in Table 4. In both scenarios, our projection approach gives a better prediction on the test cases than do the knot-based approaches, with an order-of-magnitude improvement for the robot arm example. For this dataset, both θ_1 and θ_2 appear to have a different posterior distribution under the projection approach than under the knot-based approaches. This could be a consequence of poor mixing of the Gibbs sampler for the knot-based approaches. Such poor mixing seems to be a general phenomenon manifested by knot-based approximations of stochastic processes (Golightly & Wilkinson, 2006). The projection approach overcomes this problem to a great extent. The inference is not very sensitive to the choice of hyperparameters; with datasets of this size we have minimal prior influence. In trial runs with a smaller number of iterations, changing the grid for θ_1 to 1000 uniformly spaced points in [0, 1] yielded very similar results, with projection again performing substantially better than the knot-based approaches.

As a final remark, we mention that although the projection approach was motivated by developing an approximation to the original Gaussian process, we end up with a covariance that is substantially modified according to some metrics and which can be compellingly argued to be an improvement upon the original model instead of just an approximation.

Acknowledgments

This work was supported by the National Institute of Environmental Health Sciences and DARPA MSEE. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute of Environmental Health Sciences, the National Institutes of Health, or DARPA MSEE.

Appendix

Appendix:

Proof of Theorem 1

By construction,

$$\begin{aligned} C_{\mathrm{fr}} &= UD^2U^{\mathrm{T}} = UDV^{\mathrm{T}}VDU^{\mathrm{T}} = C_2C_2^{\mathrm{T}} \\ &= C\Phi^{\mathrm{T}}\left(B^{\mathrm{T}}\right)^{-1}B^{-1}\Phi C = C\Phi^{\mathrm{T}}\left(BB^{\mathrm{T}}\right)^{-1}\Phi C \\ &= C\Phi^{\mathrm{T}}C_1^{-1}\Phi C = (\Phi C)^{\mathrm{T}}\left(\Phi C\Phi^{\mathrm{T}}\right)^{-1}\Phi C. \end{aligned}$$

This shows that the reduced spectral decomposition, C_{fr}, produced by Algorithm 1 is indeed equal to the projection approximation, which in turn is equal to a generalized projection matrix as explained below.

The generalized rank-m projection matrix for the projection whose column space is spanned by the columns of an $n \times m$ matrix A with $m \le n$ and whose nullity is the orthogonal complement of the column space of an $n \times m$ matrix B is given by $A(B^{\mathrm{T}} A)^{-1}B^{\mathrm{T}}$. This is a generalization of the standard projection matrix formula (Doković, 1991). Therefore $C_{\mathrm{fr}} = PC$, where $P = C\Phi^{\mathrm{T}}\{\Phi(C\Phi^{\mathrm{T}})\}^{-1}$ is the generalized projection matrix whose column space is spanned by the columns of $C\Phi^{\mathrm{T}}$ and whose nullity is the orthogonal complement of the column space of Φ^{T}. Again, by construction, the column space of Φ^{T} equals the column space of $C\Omega$, and therefore the column space of $C\Phi^{\mathrm{T}}$ is the same as the column space of $C^2\Omega$, which equals the column space of $C\Omega$. Finally, since the column space of $C\Omega$ is the same as the row space of $\Omega^{\mathrm{T}}C$, the result follows by a direct application of Theorem 14 from Sarlos (2006).

Proof of Theorem 2

The Kullback–Leibler divergence between two n-variate normal distributions $\mathcal{N}_0=N_n(\mu_0,\Sigma_0)$ and $\mathcal{N}_1=N_n(\mu_1,\Sigma_1)$ is

$$d_{\mathrm{KL}}(\mathcal{N}_0\|\mathcal{N}_1)=\frac{1}{2}\left[\mathrm{tr}\left(\Sigma_1^{-1}\Sigma_0\right)-n-\log\left\{\det\left(\Sigma_1^{-1}\Sigma_0\right)\right\}+(\mu_1-\mu_0)^{\mathrm{T}}\Sigma_1^{-1}(\mu_1-\mu_0)\right].$$

In our case, $\mathcal{N}_0=\pi_{\mathrm{full}}=N_n\left(y;0,C_{XX}+\sigma^2I\right)$ and $\mathcal{N}_1=\pi_{\mathrm{lp}}=N_n\left(y;0,C_{XX}^{\mathrm{lp}}+\sigma^2I\right)$. Therefore, $d_{\mathrm{KL}}\left(\pi_{\mathrm{full}},\pi_{\mathrm{lp}}\right)=\frac{1}{2}\left[\mathrm{tr}\left(\Sigma_1^{-1}\Sigma_0\right)-n-\log\left\{\det\left(\Sigma_1^{-1}\Sigma_0\right)\right\}\right]$, with $\Sigma_0 = C_{XX} + \sigma^2 I$ and $\Sigma_1=C_{XX}^{\mathrm{lp}}+\sigma^2I$. We have $\|\Sigma_0-\Sigma_1\|_{\mathrm{F}}=\|C_{XX}-C_{XX}^{\mathrm{lp}}\|_{\mathrm{F}}\le\epsilon$.

Break the expression for the Kullback–Leibler divergence into two parts, with the first part being

$$\mathrm{tr}\left(\Sigma_1^{-1}\Sigma_0\right)-n=\mathrm{tr}\left\{\Sigma_1^{-1}(\Sigma_0-\Sigma_1)\right\}=\sum_{i=1}^{n}\sum_{j=1}^{n}s_{ij}d_{ji},$$

where s_{ij} and dji are the (ij)th and (ji)th elements of Σ_1^{-1} and $\Sigma_0-\Sigma_1$, respectively. Then

$$\mathrm{tr}\left(\Sigma_1^{-1}\Sigma_0\right)-n\le\|\Sigma_1^{-1}\|_{\max}\sum_{i=1}^{n}\sum_{j=1}^{n}d_{ji}\le\|\Sigma_1^{-1}\|_{\max}n^2\epsilon. \quad \text{(A1)}$$

In the inequality above, we used the facts that $\|\Sigma_1^{-1}\|_{\max}=\max_{ij}s_{ij}$ and that $\|\Sigma_0-\Sigma_1\|_{\mathrm{F}}\le\epsilon$ implies $\sum_{i=1}^{n}\sum_{j=1}^{n}d_{ji}\le n^2\epsilon$. Now, $\|\Sigma_1^{-1}\|_{\max}\le\|\Sigma_1^{-1}\|2$. Since Σ_1^{-1} is symmetric positive definite, $\|\Sigma_1^{-1}\|_2$ is the largest eigenvalue of Σ_1^{-1}, which is equal to the inverse of the smallest eigenvalue of Σ_1. Recall that $\Sigma_1=C_{XX}^{\mathrm{lp}}+\sigma^2I$ and that C_{XX}^{lp} is positive semidefinite and has nonnegative eigenvalues. Therefore, all eigenvalues of Σ_1 are greater than or equal to σ^2; using this fact in conjunction with the in equality (A1), we obtain

$$\mathrm{tr}\left(\Sigma_1^{-1}\Sigma_0\right)-n\le\left(\frac{n}{\sigma}\right)^2\epsilon. \quad \text{(A2)}$$

It remains to bound the second part of the divergence expression. We have $\det\left(\Sigma_1^{-1}\Sigma_0\right)=\left(\prod_{i=1}^{n}\lambda_i^0\right)/\left(\prod_{i=1}^{n}\lambda_i^1\right)$, where λ_i^0 and λ_i^1 are eigenvalues of Σ_0 and Σ_1, respectively. Since Σ_0 and Σ_1 are symmetric, by the Hoffman–Weilandt inequality (Bhatia, 1997) there exists a permutation p such that $\sum_{i=1}^{n}\left(\lambda_{p(i)}^0-\lambda_i^1\right)^2 \le \|\Sigma_0-\Sigma_1\|_{\mathrm{F}}^2 \le \epsilon^2$. Thus, with the same permutation p, we have for each i that $\left(\lambda_{p(i)}^0/\lambda_i^1\right)\in[1-\epsilon,1+\epsilon]$. Trivial manipulation then yields $\log\left\{\det\left(\Sigma_1^{-1}\Sigma_0\right)\right\}\in[n\log(1-\epsilon),n\log(1+\epsilon)]$, so that

$$-\log\left\{\det\left(\Sigma_1^{-1}\Sigma_0\right)\right\}\le n\epsilon. \quad \text{(A3)}$$

Upon combining inequalities (A2) and (A3), we have

$$d_{\mathrm{KL}}\left(\pi_{\mathrm{full}},\pi_{\mathrm{lp}}\right)\le\left\{n+\left(\frac{n}{\sigma}\right)^2\right\}\epsilon,$$

which completes the proof.

While this is not an optimal bound, it shows that the Kullback–Leibler divergence is of the same order as the error in estimating the covariance matrix in terms of the Frobenius norm. Additional assumptions on the eigenspace of the covariance matrix would yield tighter bounds.

Example of inversion with the Woodbury matrix identity

Either Algorithm 1 or Algorithm 2 in this paper would yield $C_{XX}^{\mathrm{lp}}=UD^2U^{\mathrm{T}}$, with $U^{\mathrm{T}}U=I$. We are interested in calculating $\Sigma_1^{-1}=\left(C_{XX}^{\mathrm{lp}}+\sigma^2I\right)^{-1}$ in the marginalized form for inference or prediction. Using the Woodbury matrix identity (Harville, 2008), we have

$$\begin{aligned}\Sigma_1^{-1} &=\sigma^{-2}I-\sigma^{-2}U\left(D^{-2}+\sigma^{-2}U^{\mathrm{T}}U\right)^{-1}U^{\mathrm{T}}\sigma^{-2}\\ &=\sigma^{-2}I-\sigma^{-4}U\left(D^{-2}+\sigma^{-2}I\right)^{-1}U^{\mathrm{T}}.\end{aligned}$$

In the above, $D^{-2}+\sigma^{-2}I$ is a diagonal matrix whose inverse can be obtained by simply taking the reciprocals of the diagonal elements. Thus, direct matrix inversion can be avoided entirely with the decomposition available from the algorithms.

References

Adler, RJ. IMS Lecture Notes–Monograph Series. Vol. vol. 12. Institute of Mathematical Statistics; Hayward: 1990. An Introduction to Continuity, Extrema, and Related Topics for General Gaussian Processes; p. 75-6.

Banerjee S, Gelfand AE, Finley AO, Sang H. Gaussian predictive process models for large spatial data sets. J. R. Statist. Soc. B. 2008; 70:825–48.

Bhatia, R. Matrix Analysis. Springer; New York: 1997.

Candès EJ, Romberg J, Tao T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Info. Theory. 2006; 52:489–509.

Choi T, Schervish MJ. On posterior consistency in nonparametric regression problems. J. Mult. Anal. 2007; 98:1969–87.

Cressie N, Johannesson G. Fixed rank kriging for very large spatial data sets. J. R. Statist. Soc. B. 2008; 70:209–26.

Csató L, Opper M. Sparse on-line Gaussian processes. Neural Comp. 2002; 14:641–68.

Dasgupta S, Gupta A. An elementary proof of a theorem of Johnson and Lindenstrauss. Random Struct. Algor. 2003; 22:60–5.

Dixon JD. Estimating extremal eigenvalues and condition numbers of matrices. SIAM J. Numer. Anal. 1983; 20:812–4.

Doković D. Unitary similarity of projectors. Aequationes Math. 1991; 42:220–4.

Donoho DL. Compressed sensing. IEEE Trans. Info. Theory. 2006; 52:1289–306.

Drineas P, Mahoney MW. On the Nystrom method for approximating a Gram matrix for improved kernel-based learning. J. Mach. Learn. Res. 2005; 6:2153–75.

Finley AO, Sang H, Banerjee S, Gelfand AE. Improving the performance of predictive process modeling for large datasets. Comp. Statist. Data Anal. 2009; 53:2873–84.

Foster L, Waagen A, Aijaz N, Hurley M, Luis A, Rinsky J, Satyavolu C, Way MJ, Gazis P, Srivastava A. Stable and efficient Gaussian process calculations. J. Mach. Learn. Res. 2009; 10:857–82.

Frank, A.; Asuncion, A. UCI Machine Learning Repository. School of Information and Computer Sciences, University of California; Irvine: 2010. URL: http://archive.ics.uci.edu/ml

Frauenfelder P, Schwab C, Todor RA. Finite elements for elliptic problems with stochastic coefficients. Comp. Meth. Appl. Mech. Eng. 2005; 194:205–28.

Ghanem, R.; Spanos, PD. Stochastic Finite Elements: A Spectral Approach. Dover Publications; New York: 2003.

Golightly A, Wilkinson DJ. Bayesian sequential inference for nonlinear multivariate diffusions. Statist. Comp. 2006; 16:323–38.

Grigoriu, M. Stochastic Calculus: Applications in Science and Engineering. Birkhäuser; Boston: 2002.

Guhaniyogi R, Finley AO, Banerjee S, Gelfand AE. Adaptive Gaussian predictive process models for large spatial datasets. Environmetrics. 2011; 22:997–1007. [PubMed: 22298952]

Halko N, Martinsson PG, Tropp JA. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev. 2011; 53:217–88.

Harville, DA. Matrix Algebra from a Statistician's Perspective. Springer; New York: 2008.

Johnson WB, Lindenstrauss J, Schechtman G. Extensions of Lipschitz maps into Banach spaces. Israel J. Math. 1986; 54:129–38.

Keerthi S, Chu W. A matching pursuit approach to sparse Gaussian process regression. Adv. Neural Info. Proces. Syst. 2006; 18:643–50.

Plummer M, Best N, Cowles K, Vines K. CODA: Convergence diagnosis and output analysis for MCMC. R News. 2006; 6:7–11.

Quiñonero Candela J, Rasmussen CE. A unifying view of sparse approximate Gaussian process regression. J. Mach. Learn. Res. 2005; 6:1939–59.

R Development Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing; Vienna, Austria: 2012. ISBN 3-900051-07-0, http://www.R-project.org

Sarlos, T. Improved approximation algorithms for large matrices via random projections. Proc. 47th Ann. IEEE Symp. Found. Comp. Sci; 2006. p. 143-52.

Schwaighofer A, Tresp V. Transductive and inductive methods for approximate Gaussian process regression. Adv. Neural Info. Proces. Syst. 2002; 15:953–60.

Seeger, M.; Williams, CKI.; Lawrence, ND. Fast forward selection to speed up sparse Gaussian process regression. Proc. Ninth Int. Workshop Artif. Intel. Statist; Society for Artificial Intelligence and Statistics; 2003. p. 2003-10.

Smola AJ, Bartlett P. Sparse greedy Gaussian process regression. Adv. Neural Info. Proces. Syst. 2001; 13:619–25.

Snelson E, Ghahramani Z. Sparse Gaussian processes using pseudo-inputs. Adv. Neural Info. Proces. Syst. 2006; 18:1257–64.

Stein, M. Interpolation of Spatial Data: Some Theory for Kriging. Springer; New York: 1999.

Stewart GW. On the early history of the singular value decomposition. SIAM Rev. 1993; 35:551–66.

Tokdar ST. Towards a faster implementation of density estimation with logistic Gaussian process priors. J. Comp. Graph. Statist. 2007; 16:633–55.

van der Vaart AW, van Zanten JH. Rates of contraction of posterior distributions based on Gaussian process priors. Ann. Statist. 2008; 36:1435–63.

Vijayakumar, S.; Schaal, S. Locally weighted projection regression: An $O(n)$ algorithm for incremental real time learning in high dimensional space. Proc. 17th Int. Conf. Mach. Learn; San Francisco. Morgan Kaufmann Publishers; 2000. p. 1079-86.

Table 1

Comparative performance of three approximation methods in terms of matrix error norms. Times shown are relative to the time taken by the projection approach, scaled to 1

$m = 10$	$\lVert\cdot\rVert_F$	$\lVert\cdot\rVert_2$	Cond. no.	Relative time
LP	106·14	17·66	1·06	1
PP1	107·64	17·68	1·24	0·67
PP2	106·66	17·68	1·26	0·67
$m = 50$	$\lVert\cdot\rVert_F$	$\lVert\cdot\rVert_2$	Cond. no.	Relative time
LP	50·54	14·30	2·93	1
PP1	79·10	17·02	2803·5	0·89
PP2	69·57	15·68	876·23	0·93
$m = 100$	$\lVert\cdot\rVert_F$	$\lVert\cdot\rVert_2$	Cond. no.	Relative time
LP	6·61	2·84	20·65	1
PP1	39·96	13·20	$1{\cdot}38 \times 10^6$	0·78
PP2	10·16	6·31	1792·1	0·90

LP, linear projection approach based on Algorithm 1 for $n = 1000$; PP1, standard predictive process; PP2, predictive process with pivoted Cholesky approach for knot selection; Cond. no., condition number.

Table 2

Comparison of the ranks required to achieve specific target errors by three different algorithms

		PP1	PP2	LP
$n=100$, $\epsilon=0{\cdot}1$, optimal $m=5$	Required rank	17	9	7
	Cond. no.	298·10	54·59	20·08
	Relative time	0·43	0·57	1
$n=1000$, $\epsilon=0{\cdot}01$, optimal $m=69$	Required rank	213	97	78
	Cond. no.	$2{\cdot}30 \times 10^{7}$	2164·6	473·43
	Relative time	0·33	0·32	1
$n=10\,000$, $\epsilon=0{\cdot}01$, optimal $m=137$	Required rank	1757	793	174
	Cond. no.	$3{\cdot}19 \times 10^{19}$	$2{\cdot}30 \times 10^{9}$	1012·3
	Relative time	1·57	1·34	1

LP, linear projection approach based on Algorithm 2; PP1, standard predictive process; PP2, predictive process with pivoted Cholesky approach for knot selection; Cond. no., condition number.

Table 3

Comparison of three algorithms based on their performance for simulated datasets with a specified target error using two functions of different smoothness levels, in terms of mean squared predictive error and various posterior summaries for the unknown parameters

		PP1	**PP2**	**LP**
ϵ =0·1, smooth	MSPE	11·985	8·447	3·643
	Average required rank	1715·6	453·8	117·2
	95% interval, required rank	[1331, 2542]	[377, 525]	[97, 141]
	Posterior mean, θ_1	0·09	0·10	0·06
	95% credible interval, θ_1	[0·05, 0·14]	[0·05, 0·15]	[0·04, 0·08]
	ESS, θ_1	496	870	1949
	Posterior mean, θ_2	0·91	1·15	1·25
	95% credible interval, θ_2	[0·58, 1·58]	[0·85, 1·43]	[1·09, 1·46]
	ESS, θ_2	2941	3922	4518
	Relative time	1·23	0·91	1
ϵ =0·01, wavy	MSPE	17·41	13·82	6·93
	Average required rank	4758·5	1412·5	404·5
	95% interval, required rank	[2871, 6781]	[1247, 1672]	[312, 475]
	Posterior mean, θ_1	0·11	0·09	0·05
	95% credible interval, θ_1	[0·04, 0·17]	[0·05, 0·13]	[0·03, 0·08]
	ESS, θ_1	741	747	1049
	Posterior mean, θ_2	1·27	1·18	1·19
	95% credible interval, θ_2	[1·08, 1·43]	[1·12, 1·41]	[1·15, 1·34]
	ESS, θ_2	1521	2410	2651
	Relative time	1·90	1·22	1

LP, linear projection approach based on Algorithm 2; PP1, standard predictive process; PP2, predictive process with pivoted Cholesky approach for knot selection; MSPE, mean squared predictive error; ESS, effective sample size of the unknown parameters of the covariance function.

Table 4

Comparison of three algorithms based on their performance for the abalone and Sarcos robot arm datasets, in terms of mean squared predictive error and various posterior summaries for the unknown parameters

		PP1	**PP2**	**LP**
Abalone	MSPE	1·785	1·517	1·182
	Average required rank	417·6	328·8	57·2
	95% interval, required rank	[213, 750]	[207, 651]	[43, 71]
	Posterior mean, θ_1	0·212	0·187	0·149
	95% credible interval, θ_1	[0·112, 0·317]	[0·109, 0·296]	[0·105, 0·207]
	ESS, θ_1	516	715	1543
	Posterior mean, θ_2	0·981	1·014	1·105
	95% credible interval, θ_2	[0·351, 1·717]	[0·447, 1·863]	[0·638, 1·759]
	ESS, θ_2	1352	1427	1599
	Relative time	1·26	1·38	1
Robot arm	MSPE	0·5168	0·2357	0·0471
	Average required rank	4195	2031	376
	95% interval, required rank	[3301, 4985]	[1673, 2553]	[309, 459]
	Posterior mean, θ_1	0·496	0·352	0·105
	95% credible interval, θ_1	[0·087, 0·993]	[0·085, 0·761]	[0·042, 0·289]
	ESS, θ_1	85	119	147
	Posterior mean, θ_2	1·411	1·315	1·099
	95% credible interval, θ_2	[1·114, 1·857]	[1·065, 1·701]	[1·002, 1·203]
	ESS, θ_2	145	132	227
	Relative time	2·74	2·58	1

LP, linear projection approach; PP1, standard predictive process; PP2, predictive process with pivoted Cholesky approach for knot selection; MSPE, mean squared predictive error; ESS, effective sample size of the posterior samples of the unknown parameters of the covariance function.