Abstract

We develop a fast deterministic variational approximation scheme for Gaussian process (GP) regression, where the spectrum of the covariance function is subjected to a sparse approximation. The approach enables uncertainty in covariance function hyperparameters to be treated without using Monte Carlo methods and is robust to overfitting. Our article makes three contributions. First, we present a variational Bayes algorithm for fitting sparse spectrum GP regression models, which makes use of nonconjugate variational message passing to derive fast and efficient updates. Second, inspired by related methods in classification, we propose a novel adaptive neighbourhood technique for obtaining predictive inference that is effective in dealing with nonstationarity. Regression is performed locally at each point to be predicted and the neighbourhood is determined using a measure defined based on lengthscales estimated from an initial fit. Weighting the dimensions according to the lengthscales effectively downweights variables of little relevance, leading to automatic variable selection and improved prediction. Third, we introduce a technique for accelerating convergence in nonconjugate variational message passing by adapting step sizes in the direction of the natural gradient of the lower bound. Our adaptive strategy can be easily implemented and empirical results indicate significant speed ups.

Keywords: Local Gaussian process, sparse approximation, variational Bayes, nonconjugate variational message passing, adaptive neighbourhood, bound optimization.
1 Introduction

Gaussian process (GP) models provide a flexible, probabilistic approach to regression and are widely used. However, the application of GP models to large data sets is challenging as the memory and computational requirements scale as $O(n^2)$ and $O(n^3)$ respectively, where $n$ is the number of training data points. Various sparse GP approximations have been proposed to overcome this limitation and a unifying framework for existing sparse methods is provided in Quiñonero-Candela and Rasmussen (2005). We consider the stationary sparse spectrum GP regression model proposed by Lázaro-Gredilla et al. (2010), where the spectrum of the covariance function is sparsified instead of the usual spatial domain. The SSGP algorithm introduced by Lázaro-Gredilla et al. (2010) for fitting this model uses conjugate gradients to optimize the marginal likelihood with respect to the hyperparameters and spectral points. Comparisons with other state-of-the-art sparse GP approximations such as the fully independent training conditional model, first introduced as sparse pseudo-input GP in Snelson and Ghahramani (2006) and the sparse multiscale GP (Walder et al. 2008) showed that the SSGP yielded significant improvements. However, optimization with respect to the spectral frequencies increases the tendency to underestimate predictive uncertainty and poses a risk of overfitting in the SSGP algorithm.

In this paper, we develop a fast deterministic variational approximation scheme for the sparse spectrum GP regression model, which enables uncertainty in covariance function hyperparameters to be treated, thus improving prediction and preventing overfitting. Our article makes three contributions. First, we present a variational Bayes (VB, Attias 1999) algorithm for fitting sparse spectrum GP regression models, which makes use of nonconjugate variational message passing (Knowles and Minka 2011) to derive fast and efficient updates. Second, we propose a novel adaptive neighbourhood technique for obtaining predictive inference inspired by adaptive nearest neighbours techniques in classification designed to mitigate the curse of dimensionality (Hastie and Tibshirani 1996). Regression is performed locally and the neighbourhood of each point to be predicted is determined
using a measure defined based on lengthscales estimated from an initial fit. Weighting
the dimensions according to the lengthscales effectively downweights variables of little
relevance, leading to automatic variable selection and improved prediction. Third, we
introduce a technique for accelerating convergence in nonconjugate variational message
passing by adapting the size of steps taken in the direction of the natural gradient of
the lower bound. Our adaptive strategy can be implemented easily and empirical results
indicate significant speed ups.

GPs have diverse applications and various methods have been developed to overcome
their computational limitations for handling large data sets. A good summary of approx-
imations used in the modelling of large spatial data sets is given in [Ren et al. (2011)]
and they also consider hybrid variational and Monte Carlo methods where importance
sampling is embedded within a VB scheme. Local GP regression significantly reduces
computational cost due to the much smaller number of training data in each partition
and has been considered by authors in machine learning (e.g. [Snelson and Ghahramani,
2007; Nguyen-Tuong et al., 2008; Park and Choi, 2010]) and in spatial statistics (e.g.
Vecchia, 1988; Haas, 1995; Stein et al., 2004; Kim et al., 2005). Urtasun and Darrell
(2008) propose fitting GP models in local neighbourhoods which are defined online for
each test point, but estimate covariance hyperparameters only for a subset of all possi-
ble local neighbourhoods. Different local experts are then combined using a mixture
model which is capable of handling multimodality. Our idea of using adaptive nearest
neighbours in GP regression is inspired by [Hastie and Tibshirani (1996)]. For each point
that we are trying to predict, we fit two models. In the first instance, the neighbour-
hood is determined using the Euclidean metric. The lengthscales estimated from the first
fitting are then used to redefine the distance measure determining the neighbourhood
for fitting the second model. Experiments suggest that this approach improves predic-
tion significantly in data with nonstationarities, as hyperparameters are allowed to vary
across neighbourhoods adapted to each query point. This approach is advantageous for
dealing with large data sets as well since a smaller number of spectral points might suffice
for each local neighbourhood. Some other interesting ideas on local GP include that of building local neighbourhoods sequentially through optimization of an empirical Bayes mean-square prediction error criterion (Gramacy and Apley, 2013) and approximation of the Gaussian likelihood by use of an ordering and conditioning on a subset of past observations (Vecchia, 1988; Stein et al., 2004). While Stein et al. (2004) suggest making neighbourhoods non-local to improve learning of covariance parameters, local neighbourhoods may work better when the motivation is handling nonstationarity. Lindgren et al. (2011) make a connection between discrete spatial Markov random fields and continuous Gaussian random fields with covariance functions in the Matern class.

VB methods approximate the intractable posterior in Bayesian inference by a factorized distribution. While this product density assumption is often unrealistic and can lead to underestimation of the posterior variance (Wang and Titterington, 2005), optimization of a factorized variational posterior can be decomposed into local computations involving only neighbouring nodes in the factor graph. This leads to fast computational algorithms which can give excellent predictive inferences and reasonably good estimates of the marginal posterior distributions (see, e.g., Blei and Jordan, 2006, Braun and McAuliffe, 2010). Variational message passing (Winn and Bishop, 2005) is an algorithmic implementation of VB for general conjugate-exponential models (Attias, 2000). Knowles and Minka (2011) propose nonconjugate variational message passing to extend variational message passing to nonconjugate models. A type of fixed point iterations algorithm, the factors in VB are assumed to be members of the exponential family in nonconjugate variational message passing and Wand (2013) derive fully simplified updates for multivariate Gaussian factors. We use nonconjugate variational message passing to derive efficient updates for the posterior distribution of the lengthscales, which is assumed to be Gaussian. Ren et al. (2011) describe the use of VB methods for spatial modelling via GP, where they also treat uncertainty in the covariance function hyperparameters. However, they propose using importance sampling within each VB iteration to handle the intractable expectations associated with the covariance function hyperparameters. Variational inference has
also been considered in machine learning for sparse GPs that selects the inducing inputs and hyperparameters by maximizing a lower bound to the exact marginal likelihood (Tit-
sias, 2009) and heteroscedastic GP regression models where the noise is input dependent (L´azaro-Gredilla and Titsias, 2011).

VB is known to suffer from slow convergence when there is strong dependence between variables in the factorized variational posterior. To speed up convergence, Qi and Jaakkola (2006) propose parameter expanded VB methods to reduce coupling in updates while Tan and Nott (2013a) considered the use of partially noncentered parametrizations. Inspired by the adaptive overrelaxed bound optimization methods proposed in Salakhutdinov and Roweis (2003), we introduce a simple adaptive strategy to accelerate convergence in non-conjugate variational message passing. Tan and Nott (2013b) showed that nonconjugate variational message passing is a natural gradient ascent algorithm with step size one and step sizes smaller than one correspond to damping. Here we propose using a step size larger than one which can accelerate convergence in fixed point iterations algorithms (see Huang et al., 2005). We do not attempt to search for the optimal step size but instead adopt an adaptive strategy which ensures that the lower bound increases at the end of each cycle of updates. Honkela et al. (2003) considered a different strategy of speeding up convergence by combining parameter-wise updates to form a diagonal direction for a line search. A general iterative algorithm for computing VB estimators (defined as the means of the variational approximation) has also been proposed by Wang and Tittering-
ton (2006) and its convergence properties investigated for normal mixture models.

This paper is organized as follows. Section 2 describes the sparse spectrum GP regression model and Section 3 develops the nonconjugate variational message passing algorithm for fitting it. Section 4 presents an adaptive strategy for accelerating convergence in nonconjugate variational message passing. Section 5 discusses how the predictive distribution can be estimated and the measures used for performance evaluation. Section 6 describes the adaptive neighbourhood approach for local regression. Section 7 considers examples including real and simulated data and Section 8 concludes.
2 Sparse spectrum Gaussian process regression

Given a data set \( \{(x_i, y_i) | i = 1, \ldots, n\} \), we assume that for each \( i = 1, \ldots, n \), the output \( y_i \in \mathbb{R} \) has been generated by an unknown latent function \( f \) evaluated at the input, \( x_i \in \mathbb{R}^d \) and independently corrupted by additive Gaussian noise such that

\[
y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \gamma^2).
\]

A GP prior is assumed over \( f(x) \) at \( x \in \mathbb{R}^d \). For any set of inputs \( \{x_i|i = 1, \ldots, n\} \), the vector \([f(x_1), \ldots, f(x_n)]^T\) has a joint Gaussian distribution, \( N(0, K) \), where \( K \) is a covariance matrix. We assume that the mean of the process is zero. It is straightforward to allow for a nonzero mean and we have experimented with the inclusion of a linear term for the mean. However, for the examples discussed in this paper, we find that it is sufficient to adopt a zero mean and this assumption also works better in the determination of relevant inputs using the adaptive neighbourhood approach described in Section 6. The entries of \( K \) are given by

\[
K_{ij} = E\{f(x_i)f(x_j)\} = k(x_i, x_j) \quad \text{for} \quad i = 1, \ldots, d, \; j = 1, \ldots, d,
\]

where \( k \) is taken to be some stationary covariance function. A stationary covariance function depends only on the difference between its inputs so that \( k(x_i, x_j) = k(x_i - x_j) = k(h) \), where \( h = (x_i - x_j) \in \mathbb{R}^d \). If the covariance function further depends only on the distance between inputs, then it is isotropic (see Adler and Taylor, 2007). Let \( C(h) = c(h^T h) \) be the correlation function of a stationary, isotropic random field on \( \mathbb{R}^d \). We consider covariance functions of the form

\[
k(h) = \sigma^2 c(h^T \Lambda h) \quad (1)
\]

where \( \sigma^2 > 0 \), \( \Lambda \) is a diagonal matrix with diagonal entries given by \( [\lambda_1^2, \ldots, \lambda_d^2] \) and \( \lambda_l \geq 0 \) for \( l = 1, \ldots, d \).

Lázaro-Gredilla et al. (2010) introduced a novel perspective on GP approximation by subjecting the spectrum of the covariance function to a sparse approximation. They
considered the approximation,

\[ f(x) \approx \sum_{r=1}^{m} \left\{ a_r \cos(2\pi s_r^T x) + b_r \sin(2\pi s_r^T x) \right\}, \tag{2} \]

where \( a_r, b_r \) are independent and identically distributed as \( N(0, \sigma^2/m) \) and \( s_r \) is a \( d \)-dimensional vector of spectral frequencies. Now, the power spectral density, \( S_k(s) \), corresponding to a stationary covariance function \( k \) is

\[ S_k(s) = \int_{\mathbb{R}^d} \exp(-2\pi is^T h) k(h) \, dh \]

and

\[ S_k(s) \propto p_k(s) \] such that

\[ S_k(s) = k(0) p_k(s). \]

When \( \{s_1, \ldots, s_m\} \) are drawn randomly from \( p_k(s) \), Lázaro-Gredilla et al. (2010) showed that (2) is a stationary sparse GP, which approximates the full GP by replacing the original spectrum with a discrete set of spectral points.

The power spectral density of \( C(h) \) is

\[ S_C(s) = \int_{\mathbb{R}^d} \exp(-2\pi is^T h) c(h^T h) \, dh \]

and

\[ S_C(s) = C(0) p_C(s) = p_C(s) \] since \( C(0) = 1. \) Thus, we have

\[
p_k(s) = \frac{1}{\sigma^2} \int_{\mathbb{R}^d} \exp(-2\pi is^T h) k(h) \, dh
= \int_{\mathbb{R}^d} \exp(-2\pi is^T h) c(h^T \Lambda h) \, dh
= |\Lambda^{-\frac{1}{2}}| \int_{\mathbb{R}^d} \exp(-2\pi i z^T \Lambda^{-\frac{1}{2}} z) c(z^T z) \, dz
= |\Lambda^{-\frac{1}{2}}| p_C(\Lambda^{-\frac{1}{2}} s).
\]

This implies that \( \{\Lambda^{\frac{1}{2}} s_1, \ldots, \Lambda^{\frac{1}{2}} s_m\} \) is a random sample from \( p_k(s) \) if \( \{s_1, \ldots, s_m\} \) is a random sample from \( p_C(s) \). For the stationary squared exponential covariance function,

\[ k(h) = \sigma^2 \exp(-\frac{1}{2} h^T \Lambda h), \tag{3} \]

\( C(h) = \exp(-\frac{1}{2} h^T h) \) from \( \textbf{(1)} \) and \( p_C \) can be shown to be the probability density of a random vector distributed as \( N(0, \frac{1}{\pi} I_d) \). Therefore, if we consider a random sample \( \{s_1, \ldots, s_m\} \) generated from \( N(0, I_d) \), then \( \left\{ \frac{1}{\sqrt{2\pi}} \Lambda^{\frac{1}{2}} s_1, \ldots, \frac{1}{\sqrt{2\pi}} \Lambda^{\frac{1}{2}} s_m \right\} \), is a random sample
from \( p_k(s) \). From (2), a sparse GP approximation to \( f(x) \) is then given by

\[
f(x) \approx \sum_{r=1}^{m} \left\{ a_r \cos(s_r \Lambda^{\frac{1}{2}} x) + b_r \sin(s_r \Lambda^{\frac{1}{2}} x) \right\}
= \sum_{r=1}^{m} \left[ a_r \cos\{(s_r \odot x)^T \lambda\} + b_r \sin\{(s_r \odot x)^T \lambda\}\right], \tag{4}
\]

where \( \Lambda^{\frac{1}{2}} = \text{diag}(\lambda) \), \( \lambda = [\lambda_1, \ldots, \lambda_d]^T \) is a vector of lengthscales and \( \odot \) denotes element by element multiplication of two vectors. In what follows we will allow the components of \( \lambda \) to be negative. This is reasonable in (4) because \( \lambda_j < 0 \) just reverses the direction of the axis for the \( j \)th spatial coordinate. Ignoring the non-negativity constraint on the \( \lambda_j \) allows us to use a normal variational posterior distribution for \( \lambda \) and then the associated expectations in the variational lower bound can be done in closed form (see the next Section and Appendix A). This is a highly novel aspect of our algorithm allowing a fast method that still handles covariance function hyperparameter uncertainty. The squared exponential covariance function also implements automatic relevance determination since the magnitude of \( \lambda_j \) is a measure of how relevant the \( j \)th variable is. When \( \lambda_j \) goes to zero, the covariance function becomes almost independent of the \( j \)th variable, essentially removing it from inference. See Rasmussen and Williams (2006) for more discussion.

Using the stationary sparse GP approximation in (4), we consider variational inference for \( y_i = \sum_{r=1}^{m} \left[ a_r \cos\{(s_r \odot x_i)^T \lambda\} + b_r \sin\{(s_r \odot x_i)^T \lambda\}\right] + \epsilon_i, \epsilon_i \sim N(0, \gamma^2) \). If we introduce \( \alpha = [a_1, \ldots, a_m, b_1, \ldots, b_m]^T, y = [y_1, \ldots, y_n]^T, Z = [Z_1, \ldots, Z_n]^T \) where \( Z_i = [\cos\{(s_1 \odot x_i)^T \lambda\}, \ldots, \cos\{(s_m \odot x_i)^T \lambda\}, \sin\{(s_1 \odot x_i)^T \lambda\}, \ldots, \sin\{(s_m \odot x_i)^T \lambda\}]^T \) and \( \epsilon = [\epsilon_1, \ldots, \epsilon_n]^T \), then this model can be written as

\[
y = Z \alpha + \epsilon, \quad \epsilon \sim N(0, \gamma^2 I_n), \tag{5}
\]

where \( \alpha \sim N(0, \sigma^2_m I_{2m}) \). For Bayesian inference, we assume the following priors: \( \lambda \sim N(\mu_\lambda^0, \Sigma_\lambda^0) \), \( \sigma \sim \text{half-Cauchy}(A_\sigma) \) and \( \gamma \sim \text{half-Cauchy}(A_\gamma) \). The hyperparameters \( \mu_\lambda^0, \Sigma_\lambda^0, A_\sigma \) and \( A_\gamma \) are assumed to be known. The density function of a random variable
x distributed as half-Cauchy($A$) is $\frac{2A}{\pi(A^2+x^2)}$, where $x > 0$ and $A > 0$. While inverse-gamma priors are more commonly used for the variance parameters in hierarchical models due to the conditional conjugacy relationship with Gaussian families, Gelman (2006) recommends the use of half-Cauchy family as priors because resulting inferences can be very sensitive to the inverse-gamma hyperparameters when the variance estimates are close to zero. We made the same observation in our experiments with inverse-gamma priors for $\sigma^2$ and $\gamma^2$. In particular, predictive inferences are very sensitive to the inverse-gamma priors in local regressions (see Section 6) where only a small neighbourhood is used for fitting at each test point.

3 Variational inference

We consider variational inference for the sparse spectrum GP regression model in (5). Let $\theta = \{\alpha, \lambda, \sigma, \gamma\}$ denote the set of unknown parameters and $p(\theta|y)$ denote the true posterior of $\theta$. In this section, we briefly review variational approximation and some important results in VB and nonconjugate variational message passing, which will be used to construct the algorithm for finding the approximate variational posterior of $\theta$.

In variational approximation, $p(\theta|y)$ is approximated by a $q(\theta)$ for which inference is more tractable. We attempt to make $q(\theta)$ a good approximation to $p(\theta|y)$ by minimizing the Kullback-Leibler divergence between $q(\theta)$ and $p(\theta|y)$, given by

$$
\int q(\theta) \log \frac{q(\theta)}{p(\theta|y)} d\theta = \int q(\theta) \log \frac{q(\theta)}{p(y, \theta)} d\theta + \log p(y),
$$

where $p(y) = \int p(y, \theta) d\theta$ is the marginal likelihood. As the Kullback-Leibler divergence is non-negative, we have

$$
\log p(y) \geq \int q(\theta) \log \frac{p(y, \theta)}{q(\theta)} d\theta = E_q\{\log p(y, \theta)\} - E_q\{\log q(\theta)\} = \mathcal{L},
$$

\text{(6)}
which gives a lower bound, $\mathcal{L}$ on the log marginal likelihood. Here $E_q$ denotes expectation with respect to $q(\theta)$. Maximization of this lower bound is equivalent to minimization of the Kullback-Leibler divergence between $p(\theta|y)$ and the variational approximation, $q(\theta)$.

In VB, $q(\theta)$ is assumed to factorize into $\prod_{i=1}^{M} q_i(\theta_i)$ for some partition $\{\theta_1, \ldots, \theta_M\}$ of $\theta$. The optimal densities $q_i(\theta_i), i = 1, \ldots, M,$ satisfying this product density restriction may be obtained using

$$q_i(\theta_i) \propto \exp\{E_{\theta_i} \log p(y, \theta)\}, \quad (7)$$

where $E_{\theta_i}$ denotes expectation with respect to the density $\prod_{j \neq i} q_j(\theta_j)$ (see, e.g. Ormerod and Wand 2010). For conjugate-exponential models, the optimal densities $q_i$ have the same form as the priors and it suffices to update the parameters of $q_i$, as is the case in variational message passing (Winn and Bishop 2005). For nonconjugate models, however, the optimal densities $q_i$ will not belong to recognizable density families. Apart from the product assumption, nonconjugate variational message passing (Knowles and Minka 2011) further assumes that $q_i(\theta_i)$ is a member of some exponential family, that is, $q_i(\theta_i) = \exp\{\eta_i^T t_i(\theta_i) - h_i(\eta_i)\},$ where $\eta_i$ is the vector of natural parameters and $t_i(\cdot)$ are the sufficient statistics. With this additional assumption, we just have to find $\eta_i$ that maximizes the lower bound $\mathcal{L}$. Nonconjugate variational message passing can be interpreted as a fixed point iterations algorithm where updates are obtained from the condition that the gradient or natural gradient (see Hoffman et al. 2013; Amari 1998) of $\mathcal{L}$ with respect to each $\eta_i$ is zero when $\mathcal{L}$ is maximized (see, e.g. Knowles and Minka 2011; Tan and Nott 2013a,b). Suppose $p(y, \theta) = \prod_a f_a(y, \theta), S_a = E_q\{\log f_a(y, \theta)\}$ and let $V_i(\eta_i) = \frac{\partial^2 h_i(\eta_i)}{\partial \eta_i \partial \eta_i^T}$ denote the variance-covariance matrix of $t_i(\theta_i)$. Provided $V_i(\eta_i)$ is invertible, Tan and Nott (2013b) showed that the natural gradient of $\mathcal{L}$ with respect to each $\eta_i$ is

$$\nabla_{\eta_i} \mathcal{L} = V_i(\eta_i)^{-1} \sum_{a \in N(\theta_i)} \frac{\partial S_a}{\partial \eta_i} - \eta_i. \quad (8)$$
The update for each $\eta_i$ in nonconjugate variational message passing is given by

$$\eta_i \leftarrow \mathcal{V}_i(\eta_i)^{-1} \sum_{a \in N(\theta_i)} \frac{\partial S_a}{\partial \eta_i}, \quad (9)$$

where the summation is over all factors in $N(\theta_i)$, the neighbourhood of $\theta_i$ in the factor graph of $p(y, \theta)$. The updates in nonconjugate variational message passing reduce to variational message passing for conjugate factors (see Knowles and Minka, 2011; Tan and Nott, 2013a). Unlike variational message passing however, the lower bound $\mathcal{L}$ is not guaranteed to increase at each step and convergence problems may be encountered sometimes. Knowles and Minka (2011) suggest using damping to fix convergence problems.

When $q_i$ is Gaussian, say, $q_i(\theta_i) = N(\mu_{\theta_i}^q, \Sigma_{\theta_i}^q)$, Wand (2013) showed that the update in (9) can be expressed in terms of the mean and covariance of $q_i$, and simplified as

$$\Sigma_{\theta_i}^q \leftarrow -\frac{1}{2} \left[ \text{vec}^{-1} \left( \sum_{a \in N(\theta_i)} \frac{\partial S_a}{\partial \text{vec}(\Sigma_{\theta_i}^q)} \right) \right]^{-1}, \quad \mu_{\theta_i}^q \leftarrow \mu_{\theta_i}^q + \Sigma_{\theta_i}^q \sum_{a \in N(\theta_i)} \frac{\partial S_a}{\partial \mu_{\theta_i}^q}. \quad (10)$$

We use $\text{vec}(A)$ to denote the $d^2 \times 1$ vector obtained by stacking the columns of a $d \times d$ matrix $A$ under each other, from left to right in order.

### 3.1 Nonconjugate variational message passing algorithm

We consider a variational approximation of the form

$$q(\theta) = q(\alpha)q(\lambda)q(\sigma, \gamma) \quad (11)$$

for the model in (5). The optimal densities $q(\alpha)$ and $q(\sigma, \gamma)$ satisfying (11) may be obtained from (7) and can be shown to be $q(\alpha) = N(\mu_{\alpha}^q, \Sigma_{\alpha}^q)$ and $q(\sigma, \gamma) = q(\sigma)q(\gamma)$ where

$$q(\sigma) = \frac{\exp(-C_{\sigma}'/\sigma^2)}{\mathcal{H}(2m - 2, C_{\sigma}', A_{\sigma}^2)\sigma^{2m}(A_{\sigma}^2 + \sigma^2)}; \quad q(\gamma) = \frac{\exp(-C_{\gamma}'/\gamma^2)}{\mathcal{H}(n - 2, C_{\gamma}', A_{\gamma}^2)\gamma^{n}(A_{\gamma}^2 + \gamma^2)},$$

where

- $C_{\sigma}' = 2\mu_{\sigma}^q A_{\sigma}^2 - 2\nu_{\sigma}^q A_{\sigma}^2$;
- $C_{\gamma}' = 2\mu_{\gamma}^q A_{\gamma}^2 - 2\nu_{\gamma}^q A_{\gamma}^2$. 


and \(\mathcal{H}(p, q, r) = \int_{0}^{\infty} x^{p} \exp\{-qx^{2} - \log(r + x^{-2})\} \, dx, p \geq 0, r > 0\), following the notation in [Wand et al. (2011)]. The variational parameter updates of \(\mu_{\alpha}, \Sigma_{\alpha}, C_{\alpha}^{\gamma}\) and \(C_{\gamma}^{\gamma}\) can be obtained from [11] as well. As \(\mathcal{H}(p, q, r)\) can be arbitrarily large or small, [Wand et al. (2011)] suggest evaluating \(\log \mathcal{H}(p, q, r)\) efficiently using quadrature. A discussion can be found in [Wand et al. (2011)], Appendix B and we follow their methods. For \(q(\lambda)\), \(p(y|\alpha, \lambda, \gamma)\) is not a conjugate factor and we make use of nonconjugate variational message passing. Assuming \(q(\lambda) = N(\mu_{\lambda}^{q}, \Sigma_{\lambda}^{q})\), updates for \(\mu_{\lambda}^{q}\) and \(\Sigma_{\lambda}^{q}\) can be derived from (10) using matrix differential calculus (see Magnus and Neudecker, 1988). The expectations in (10) are given in Appendices A and B. Let \(\vartheta = \{\mu_{\alpha}^{q}, \Sigma_{\alpha}^{q}, \mu_{\lambda}^{q}, \Sigma_{\lambda}^{q}, C_{\alpha}^{\gamma}, C_{\gamma}^{\gamma}\}\) denote the set of variational parameters. An iterative scheme for finding \(\vartheta\) is presented in Algorithm 1.

---

**Initialize** \(\vartheta\).

**Cycle:**

1. \(\Sigma_{\lambda}^{q} \leftarrow \left\{\Sigma_{\lambda}^{q} + \frac{\mathcal{H}(n, C_{\alpha}^{T}, A_{\alpha})}{\mathcal{H}(n-2, C_{\alpha}^{T}, A_{\alpha})}(F_{1} + F_{2})\right\}^{-1}\)
2. \(\mu_{\lambda}^{q} \leftarrow \mu_{\lambda}^{q} + \Sigma_{\lambda}^{q}^{-1}(\mu_{\lambda}^{0} - \mu_{\lambda}^{q}) - \frac{\mathcal{H}(n, C_{\alpha}^{T}, A_{\alpha})}{2\mathcal{H}(n-2, C_{\alpha}^{T}, A_{\alpha})}(F_{3} + F_{4})\)
3. \(\Sigma_{\alpha}^{q} \leftarrow \left\{\frac{\mathcal{H}(n, C_{\alpha}^{T}, A_{\alpha})}{\mathcal{H}(n-2, C_{\alpha}^{T}, A_{\alpha})} E_{q}(Z^{T}Z) + m \frac{\mathcal{H}(2m, C_{\alpha}^{T}, A_{\alpha})}{\mathcal{H}(2m-2, C_{\alpha}^{T}, A_{\alpha})} I_{2m}\right\}^{-1}\)
4. \(\mu_{\alpha}^{q} \leftarrow \frac{\mathcal{H}(n, C_{\alpha}^{T}, A_{\alpha})}{\mathcal{H}(n-2, C_{\alpha}^{T}, A_{\alpha})} \Sigma_{\alpha}^{q} E_{q}(Z^{T}y)\)
5. \(C_{\alpha}^{\gamma} \leftarrow \frac{m}{2} \left\{\mu_{\alpha}^{T} \mu_{\alpha}^{q} + \text{tr}(\Sigma_{\alpha}^{q})\right\}\)
6. \(C_{\gamma}^{\gamma} \leftarrow \frac{1}{2} \left[y^{T}y - 2y^{T}E_{q}(Z)\mu_{\alpha}^{q} + \text{tr}\{E_{q}(Z^{T}Z)\}E_{q}(Z^{T}Z)\}\)

until the increase in the lower bound \(\mathcal{L}\) is negligible.

**Algorithm 1:** Nonconjugate variational message passing algorithm for sparse spectrum GP regression model in (5).

A unique aspect of our variational scheme is the way that covariance function uncertainty is handled, with the required expectations involving \(\lambda\) in the lower bound computable in closed form. In particular, \(E_{q}(Z)\) and \(E_{q}(Z^{T}Z)\) can be evaluated in closed form and details are given in Appendix A. We define \(F_{1}, F_{2}, F_{3}\) and \(F_{4}\) in Algorithm 1 as follows. Let \(\mu_{\alpha}^{q}\mu_{\alpha}^{T} + \Sigma_{\alpha}^{q}\) be partitioned as \(\begin{bmatrix} A & B^{T} \\ B & D \end{bmatrix}\) where \(A, B\) and \(D\) are all \(m \times m\) matrices. Let \(t_{ir} = s_{r} \odot x_{i}, t_{ir}' = t_{ir} - t_{il}, t_{ir}^{+} = t_{ir} + t_{il}, v_{ir}^{+} = \exp(-\frac{1}{2}t_{ir}^{T}\Sigma_{\lambda}^{q}t_{ir}^{+})\) and
\( \nu_{irl}^+ = \exp\left(-\frac{1}{2} t_{irl}^T \Sigma_{\lambda_{irl}}^q \right) \) for \( i = 1, \ldots, n, r = 1, \ldots, m, l = 1, \ldots, m \). We have

\[
F_1 = \sum_{i=1}^{n} \sum_{r=1}^{m} y_i \exp\left(-\frac{1}{2} t_{irl}^T \Sigma_{\lambda}^q r \right) \left\{ \mu_{\alpha r}^q \cos(t_{irl}^T \mu_{\lambda}^q) + \mu_{\alpha m+r}^q \sin(t_{irl}^T \mu_{\lambda}^q) \right\} t_{irl}^T
\]

\[
F_2 = -\frac{1}{4} \sum_{i=1}^{n} \sum_{r=1}^{m} \sum_{l=1}^{m} \nu_{irl}^- \left\{ (A_{rl} + D_{rl}) \cos(t_{irl}^T \mu_{\lambda}^q) + 2 B_{rl} \sin(t_{irl}^T \mu_{\lambda}^q) \right\} t_{irl}^- t_{irl}^- T
\]

\[
+ \nu_{irl}^+ \left\{ (A_{rl} - D_{rl}) \cos(t_{irl}^T \mu_{\lambda}^q) + 2 B_{rl} \sin(t_{irl}^T \mu_{\lambda}^q) \right\} t_{irl}^+ t_{irl}^+ T
\]

\[
F_3 = -2 \sum_{i=1}^{n} \sum_{r=1}^{m} y_i \exp\left(-\frac{1}{2} t_{irl}^T \Sigma_{\lambda}^q r \right) \left\{ \mu_{\alpha r+m}^q \cos(t_{irl}^T \mu_{\lambda}^q) - \mu_{\alpha r}^q \sin(t_{irl}^T \mu_{\lambda}^q) \right\} t_{irl}
\]

\[
F_4 = \frac{1}{2} \sum_{i=1}^{n} \sum_{r=1}^{m} \sum_{l=1}^{m} \nu_{irl}^- \left\{ 2 B_{rl} \cos(t_{irl}^T \mu_{\lambda}^q) - (A_{rl} + D_{rl}) \sin(t_{irl}^T \mu_{\lambda}^q) \right\} t_{irl}^- \]

\[
+ \nu_{irl}^+ \left\{ 2 B_{rl} \cos(t_{irl}^T \mu_{\lambda}^q) + (D_{rl} - A_{rl}) \sin(t_{irl}^T \mu_{\lambda}^q) \right\} t_{irl}^+ \].

The lower bound \( \mathcal{L} \), as defined in (6), is commonly used for monitoring convergence. It can be evaluated in closed form (see Appendix B for details) and is given by

\[
\mathcal{L} = m \log m + \log(\frac{4A}{\pi})^q + \frac{1}{2} \log |\Sigma_{\lambda}^{-1} \Sigma_{\lambda}^q| - \frac{1}{2} (\mu_{\lambda}^q - \mu_{\lambda}^q) (\Sigma_{\lambda}^{-1} \Sigma_{\lambda}) (\mu_{\lambda}^q - \mu_{\lambda}^q) - \frac{1}{2} \text{tr}(\Sigma_{\lambda}^{-1} \Sigma_{\lambda})
\]

\[
+ \frac{1}{2} \log |\Sigma_{\alpha}^q| + \log \mathcal{H}(n - 2, C_{\gamma}^q, A_{\alpha}^2) + \log \mathcal{H}(2m - 2, A_{\gamma}^2, C_{\alpha}^q) + m + \frac{d}{2} - \frac{n}{2} \log(2\pi).
\]

(12)

Note that the above expression applies only after the parameter updates in steps 5 and 6 of Algorithm 1 have been made.

4 Adaptive nonconjugate variational message passing

For the sparse spectrum GP regression model in (5), we have \( y = Z \alpha + \epsilon \) where \( Z \) and \( \alpha \) are intimately linked. Each time the lengthscales (\( \lambda \)) are changed by a small amount, the amplitudes (\( \alpha \)) will also have to respond to this change in order to match the observed \( y \). In our variational approximation in (11), we have assumed that the posteriors of \( \lambda \) and \( \alpha \) are independent so that the expectations with respect to \( q \) are tractable and closed form updates can be derived for a fast algorithm. However, the strong dependence between \( \lambda \) and \( \alpha \).
and $\alpha$ implies that only small steps can be taken in each cycle of updates and a large number of iterations will likely be required for Algorithm 1 to converge.

To accelerate convergence, we propose making some simple modification to the updates in steps 1 and 2, derived using nonconjugate variational message passing. Let $\eta_\lambda$ denote the natural parameter of $q(\lambda) = N(\mu_\lambda^q, \Sigma_\lambda^q)$ and $\hat{\eta}_\lambda$ denote the update of $\eta_\lambda$ in nonconjugate variational message passing given by (9). Tan and Nott (2013b) showed that nonconjugate variational message passing is a type of natural gradient method with a step size of one. That is, if we update $\eta_\lambda$ at iteration $t$ using $\eta_\lambda^{(t)} = \eta_\lambda^{(t-1)} + a_t \nabla_{\eta_\lambda} \mathcal{L}|_{\eta_\lambda = \eta_\lambda^{(t-1)}}$, we get

$$\eta_\lambda^{(t)} = \eta_\lambda^{(t-1)} + a_t (\hat{\eta}_\lambda^{(t)} - \eta_\lambda^{(t-1)})$$

from (8) where $\hat{\eta}_\lambda^{(t)}$ is $\mathcal{V}_\lambda(\eta_\lambda)^{-1} \sum_{a \in \mathcal{N}(\lambda)} \frac{\partial S_a}{\partial \eta_\lambda}$ evaluated at $\eta_\lambda = \eta_\lambda^{(t-1)}$. This update reduces to the update in nonconjugate variational message passing when $a_t = 1$. Earlier, we note that updates in nonconjugate variational message passing do not necessarily lead to an increase in $\mathcal{L}$. In such situations, we have found it helpful to reduce the step size by taking $a_t < 1$. From our observations, instability in nonconjugate variational message passing usually happens at the first few iterations and beyond that the algorithm is usually quite stable. It is thus important to choose the variational parameters for initialization carefully. However, once the algorithm has reached stability, taking larger steps with $a_t > 1$ can result in significant speed ups.

Recall that nonconjugate variational message passing is a type of fixed point iterations algorithm. We illustrate in Figure 1 how taking steps larger than one can accelerate convergence in a very simple one-dimensional single variable case, where we are solving $x = f(x)$. Instead of taking $x^{(t)} = f(x^{(t-1)})$, we consider $x^{(t)} = x^{(t-1)} + a_t (\hat{x}^{(t)} - x^{(t-1)})$, where $\hat{x}^{(t)} = f(x^{(t-1)})$ and $a_t > 1$. The solid line starting from $x^{(0)}$ indicates the conventional path in fixed point iterations while the dot dash line indicates the path to convergence with a step size greater than 1. While the dot dash line moves to a region close to the point of convergence faster than the solid line, it may also overshoot if $a_t$ is too
Figure 1: Solid line starting from $x^{(0)}$ indicates conventional path in fixed point iterations while the dot dash line indicates path to convergence with a step size greater than 1.

large. In Algorithm 2 presented below, we borrow ideas from Salakhutdinov and Roweis (2003) to construct an adaptive algorithm where the step size $a_t$ is allowed to increase by a factor $\rho > 1$ after each cycle of updates whilst $\mathcal{L}$ is on an increasing trend and we revert to $a_t = 1$ when $\mathcal{L}$ decreases. This strategy ensures convergence as $\mathcal{L}$ is guaranteed to increase at the end of each cycle provided the original nonconjugate variational message passing updates results in an increase in $\mathcal{L}$.

It is shown in Appendix C that (13) simplifies to the updates:

$$
\Sigma_\lambda^q \leftarrow \left[ (1 - a_t) \Sigma_\lambda^{q-1} - 2a_t \text{vec}^{-1} \left( \sum_{a \in \mathcal{N}(\lambda)} \frac{\partial S_a}{\partial \text{vec}(\Sigma_\lambda^q)} \right) \right]^{-1},
$$

$$
\mu_\lambda^q \leftarrow \mu_\lambda^q + a_t \Sigma_\lambda^q \sum_{a \in \mathcal{N}(\lambda)} \frac{\partial S_a}{\partial \mu_\lambda^q},
$$

(14)

The adaptive nonconjugate variational message passing algorithm is presented in Algorithm 2. We assume in Algorithm 2 that by reverting $a_t$ to 1, the nonconjugate variational message passing updates will lead to an increase in the lower bound so that the user will not be stuck in an endless loop. Step 2(b) has been added as a safeguard as we have found that due to rounding off errors or when the step size is very large, the updated $\Sigma_\lambda^q$ may not satisfy the symmetric positive definite condition. In that case, we propose reducing the step size by a factor of $\rho$ until the eigenvalues of the updated $\Sigma_\lambda^q$ are all positive. It is useful to insert step 2(b) in Algorithm 1 after $\Sigma_\lambda^q$ has been updated as well as it can serve
Initialize $\vartheta^{(0)}$. Set $t = 0$ and $a_0 = 1$.

While ($\delta > \text{tolerance}$ and $t < \text{maximum number of iterations}$),

1. $t \leftarrow t + 1$.
   
   Compute $F_5 = \sum_{\lambda}^{q-1} + \frac{\mathcal{H}(n,C^q_{\gamma},A^2_{\gamma})}{\mathcal{H}(n-2,C^q_{\gamma},A^2_{\gamma})}(F_1 + F_2)$ and $F_6 = \sum_{\lambda}^{q-1} (\mu^0_{\lambda} - \mu^q_{\lambda}) - \frac{\mathcal{H}(n,C^q_{\gamma},A^2_{\gamma})}{2\mathcal{H}(n-2,C^q_{\gamma},A^2_{\gamma})}(F_3 + F_4)$.

2. (a) Compute $\Sigma^q_{\lambda} \leftarrow \left[ (1 - a_t)\Sigma^q_{\lambda}^{-1} + a_tF_5 \right]^{-1}$.
   
   (b) If $\Sigma^q_{\lambda}$ is symmetric positive definite, proceed to step 3.
   
   Else, $a_t \leftarrow a_t/\rho$ and return to step 2(a).

3. $\mu^q_{\lambda} \leftarrow \mu^q_{\lambda} + a_t\Sigma^q_{\lambda}F_6$.

4. Compute updates in steps 3–6 of Algorithm 1.

5. (a) Compute $\delta = \mathcal{L}|_{\vartheta(t)} - \mathcal{L}|_{\vartheta(t-1)}$.
   
   (b) If $\delta > 0$, $a_t = \rho a_{t-1}$ and return to step 1.
   
   Else, $a_t \leftarrow 1$, $t \leftarrow t + 1$ and return to step 2.

Algorithm 2: Adaptive nonconjugate variational message passing algorithm for sparse spectrum GP regression model in (5).

as damping. For both Algorithms 1 and 2, we initialize $\mu^q_{\lambda}$ as $[0.5, \ldots, 0.5]^T$ (which is one half of the amplitudes of the inputs after any rescaling), $\Sigma^q_{\lambda}$ as $\text{diag}[0.5, \ldots, 0.5]^T$, $C^q_{\gamma}$ as $(\frac{n}{2} - 1) \cdot \text{var}(y)/4$, $C^q_{\sigma}$ as $(m - 1) \cdot \text{var}(y)$, and $\mu^q_{\alpha}$ and $\Sigma^q_{\alpha}$ are initialized using the updates in steps 3–4 of Algorithm 1. We set the maximum number of iterations as 500 and the algorithms are deemed to have converged if the relative increase in $\mathcal{L}$ is less than $10^{-6}$.

Salakhutdinov and Roweis (2003) recommends taking the factor $\rho$ to be close to but more than 1 in their adaptive overrelaxed bound optimization methods. Using this as a guide, we have experimented with $\rho$ taking values of 1.1, 1.5 and 2. While all these values lead to improvement in efficiency, we find that $\rho = 1.5$ is more favourable, as the step sizes increase rather slowly when $\rho = 1.1$ while the step sizes increase too fast when $\rho = 2$, leading to many failed attempts to improve $\mathcal{L}$. While Algorithm 2 does not necessarily converge to the same local mode as Algorithm 1, we find that most of the time, results from the two algorithms are very close. Sometimes, Algorithm 2 also demonstrates the ability to avoid local modes with the larger steps that it takes. We compare and quantify
the performance of the two algorithms in Section 7.1. Note that in Algorithm 2, each failed attempt to improve $L$ is also being counted as an additional iteration in step 5(b) even though step 1 does not have to be reevaluated.

5 Predictive distribution and performance evaluation

Let $D = \{(x_i, y_i)|i = 1, \ldots, n\}$ and $T = \{(x_j^*, y_j^*)|j = 1, \ldots, n^*\}$ be the training and testing data sets respectively. Let $S = \{s_1, \ldots, s_m\}$ be the set of spectral frequencies randomly generated from $N(0, I_d)$. Bayesian predictive inference is based on the predictive distribution, $p(y_j^*|x_j^*, S, D) = \int p(y_j^*|x_j^*, S, \alpha, \lambda, \gamma) p(\alpha, \lambda, \gamma|D, S) d\alpha d\lambda d\gamma$, assuming $y_j^*$ is conditionally independent of $D$ given $\alpha$, $\lambda$ and $\gamma$. We replace $p(\alpha, \lambda, \gamma|D)$ with our variational approximation $q(\alpha, \lambda, \gamma)$ so that

$$p(y_j^*|x_j^*, S, D) \approx \int p(y_j^*|x_j^*, S, \alpha, \lambda, \gamma)q(\alpha)q(\lambda)q(\gamma) d\alpha d\lambda d\gamma. \quad (15)$$

From (15), the posterior predictive mean of $y_j^*$ is

$$\mu_j^* = \int y_j^* p(y_j^*|x_j^*, S, D) dy_j^*$$

$$\approx E_q \left\{ \int y_j^* p(y_j^*|x_j^*, S, \alpha, \lambda, \gamma) dy_j^* \right\}$$

$$= E_q(Z_j^*)^T \mu_q,$$

where $Z_j^* = [\cos((s_1 \odot x_j^*)^T \lambda), \ldots, \cos((s_m \odot x_j^*)^T \lambda), \sin((s_1 \odot x_j^*)^T \lambda), \ldots, \sin((s_m \odot x_j^*)^T \lambda)]^T$ and $E_q(Z_j^*)$ can be computed by the results in Appendix A. The posterior predictive variance is

$$\sigma_j^*2 \approx \int y_j^*2 p(y_j^*|x_j^*, S, D) dy_j^* - \{E_q(Z_j^*)^T \mu_q\}^2$$

$$\approx E_q(\gamma^2 + (Z_j^* \alpha)^2) - \mu_q^T E_q(Z_j^*) E_q(Z_j^*)^T \mu_q$$

$$= \frac{\mathcal{H}(n-4,C_q^{-1},A_q^{-2})}{\mathcal{H}(n-2,C_q^{-1},A_q^{-2})} + \text{tr}\{\mu_q^T \mu_q + \Sigma_q\} E_q(Z_j^*Z_j^*) - \mu_q^T E_q(Z_j^*) E_q(Z_j^*)^T \mu_q.$$
In the examples, following Lázaro-Gredilla et al. (2010), we evaluate performance using two quantitative measures: the normalized mean square error (NMSE) test and the mean negative log probability (MNLP) test which are defined as

\[
\text{NMSE} = \frac{1}{n^*} \sum_{j=1}^{n^*} \frac{(y_j^* - \mu_j^*)^2}{\sum_{j=1}^{n^*} (y_j^* - \bar{y})^2} \quad \text{and} \quad \text{MNLP} = \frac{1}{2n^*} \sum_{j=1}^{n^*} \left\{ \frac{(y_j^* - \mu_j^*)^2}{\sigma_j^*} + \log \sigma_j^* + \log(2\pi) \right\}
\]

The MNLP is implicitly based on a normal predictive distribution for \( y_j^* \) with mean \( \mu_j^* \) and variance \( \sigma_j^* \), \( j = 1, \ldots, n^* \).

6 Adaptive neighbourhoods approach for predictive inference

We propose a new technique of obtaining predictive inference by fitting models locally using adaptive neighbourhoods. Suppose \( D = \{(x_i, y_i) | i = 1, \ldots, n\} \) is the training data set, and \( T = \{(x_j^*, y_j^*) | j = 1, \ldots, n^*\} \) is the testing data set. Our proposed approach consists of two stages: For each test point \( x_j^* \), \( j = 1, \ldots, n^* \),

1. we first find the \( k \) nearest neighbours of \( x_j^* \) in \( D \) (that are closest to \( x_j^* \) in terms of Euclidean distance) and denote the index set of these \( k \) neighbours by \( N_1 \). We use Algorithm 2 to fit a sparse spectrum GP regression model, \( M_1 \), to \( \{(x_i, y_i) | i \in N_1\} \).

2. Next, we use the variational posterior mean of the lengthscales, \( \mu_{q\lambda}^l \), from \( M_1 \) to define a new distance measure: \( d(x_j^*, x_i) = \sqrt{(x_j^* - x_i)^T \text{diag}(\mu_{q\lambda}^2)(x_j^* - x_i)} \), where the dimensions are weighted according to \( \mu_{q\lambda}^2 \). This will effectively downweight or remove variables of little or no relevance. Using this new distance measure, we find the \( k \) nearest neighbours of \( x_j^* \) in \( D \) and denote the index set of these \( k \) neighbours by \( N_2 \). We use Algorithm 2 to fit a sparse spectrum GP regression model, \( M_2 \), to \( \{(x_i, y_i) | i \in N_2\} \) and use the variational posterior from \( M_2 \) for predictive inference.

In summary, the first fitting (\( M_1 \)) is used to find out which variables are more relevant in determining the output. From (3), a large value of \( \lambda_l \) indicates that the covariance drops rapidly along the dimension of \( l \) and hence the neighbourhood should be shrunk.
along the \( l \)th dimension. Using \( \mu_\lambda^q \) from the first fit as an estimate of the lengthscales, the neighbourhood is then adapted before performing a second fitting (\( M_2 \)) to improve prediction. In the examples, when the SSGP algorithm is implemented using this adaptive neighbourhood approach, we replace \( \mu_\lambda^q \) by the estimated lengthscales \( \hat{\lambda} \).

This adaptive neighbourhood approach is well-placed to handle data with nonstationarities as stationarity is only assumed locally and local fitting can adapt the noise and the degree of smoothing to the nonstationarities. Adapting the neighbourhood can also be very helpful in improving prediction when there are many irrelevant variables due to automatic relevance determination implemented via the lengthscales. A major advantage of the variational approach is that it allows uncertainty in the covariance hyperparameters to be modelled within a fast computational scheme. This is especially important when fitting using local neighbourhoods as plug in approaches to estimating hyperparameters will tend to underestimate predictive uncertainty. This approach is advantageous for dealing with large data sets as well since we only consider fitting models to a small subset \( k \) of the \( n \) data points at each test point for which a smaller number of basis functions (\( m \)) might suffice. While the computational requirements grow linearly with the number of prediction locations, this approach is trivially parallelizable to get a linear speed up with the number of processors.

7 Examples

We compare the performance of Algorithm 2 with the SSGP algorithm (Lázaro-Gredilla et al., 2010) using two real data sets: the pendulum data set and the rainfall-runoff data set. In addition, we generate ten artificial covariates on top of the existing two covariates in the rainfall-runoff example to test the capability of Algorithm 2 in automatic relevance determination (since only two of the twelve covariates are relevant). The implementation of SSGP in Matlab is obtained from http://www.tsc.uc3m.es/~miguel/simpletutorialssgp.php. We consider two versions of the SSGP algorithm:
Figure 2: Pendulum data set. NMSE (left) and MNLP (right) values produced by Algorithm 2 (global VA), global SSGP (fixed) and global SSGP (optimized) and averaged over ten repetitions plotted against the number of basis functions ($m$).

SSGP (fixed) uses fixed spectral points while SSGP (optimized) optimizes the marginal likelihood with respect to the spectral points. We observe some sensitivity in predictive performance to the basis functions and adopt the following strategy for better results: for each implementation of Algorithm 1 (or 2), we randomly generate ten sets of spectral points from $N(0, I_d)$, perform 2 iterations of the algorithm, and select the set with the highest attained lower bound to continue to full convergence. A similar strategy was used by Lázaro-Gredilla et al. (2010) to initialize the SSGP algorithm. Due to the zero mean assumption, we center all target vectors, $\mathbf{y}$, by subtracting the mean $\bar{\mathbf{y}}$ from $\mathbf{y}$. In the examples, “VA” refers to the variational approximation approach implemented via Algorithm 2, “global” refers to using the entire training set for fitting while “local” refers to the adaptive neighbourhood approach described in Section 6. All code was written in R and run on a dual processor Windows PC 3.30 GHz workstation.

7.1 Pendulum Data Set

The pendulum data set has $d = 9$ covariates and contains 315 training points and 315 test points. It is available at http://www.tsc.uc3m.es/~miguel/simpletutorialssgp.php. The target variable is the change in angular velocity of a simulated mechanical pendulum.
Figure 3: Pendulum data set. Left: Plot of the lower bound attained at convergence against index of runs. Right: Plot of number of iterations required for convergence against index of runs. Solid line corresponds to Algorithm 2 with $\rho = 1.5$ while dashed line corresponds to Algorithm 1.

over 50 ms and the covariates consist of different parameters of the system. Lázaro-Gredilla et al. (2010) used this example to show that SSGP (optimized) can sometimes fail due to overfitting. We rescale the input variables in the training set to lie in $[-1, 1]$ and consider the number of basis functions, $m \in \{10, 25, 50, 100, 200\}$. We compare the performance of Algorithm 2 with SSGP (optimized) and SSGP (fixed) using NMSE and MNLP values averaged over ten repetitions. We set $\rho = 1.5$, $A_{\sigma} = A_{\gamma} = 25$ for the half-Cauchy priors, following Gelman (2006) and Wand et al. (2011) and $\mu_{\lambda} = 0$, $\Sigma_{\lambda} = 10I_d$ for the lengthscales in Algorithm 2. For this data set which is quite small, we note that the adaptive neighbourhood approach did not yield significant improvements as all inputs are relevant and there is no strong nonstationarity. Hence we report only results for global fits, which are shown in Figure 2. The NMSE and MNLP values produced by Algorithm 2 are comparable with that of SSGP (fixed) for small $m$ and are better for large $m$. SSGP (optimized) performs well in NMSE but does quite badly in MNLP due to overfitting. On the whole, Algorithm 2 produces reasonably good NMSE performance and is less prone to overfitting than the SSGP algorithm. The ability of the variational approach to treat uncertainty in the covariance function hyperparameters reduces the underestimation of predictive uncertainty, resulting in better MNLP performance.
Next, we compare the performance of Algorithm 1 with Algorithm 2 both in terms of efficiency and the lower bound attained at convergence. We use Algorithm 1 to re-perform the runs for $m \in \{10, 25, 50, 100\}$, using the same sets of spectral points that were used in Algorithm 2. These runs are indexed from 1 to 40 (there are ten repetitions for each $m$). Figure 3 shows a plot of the lower bound attained at convergence on the left and a plot of the number of iterations required for convergence on the right for each of the 40 runs. The solid line corresponds to Algorithm 2 with $\rho = 1.5$ while the dashed line corresponds to Algorithm 1. Figure 3 indicates that, except for runs 3 and 40, the lower bound attained by Algorithms 1 and 2 are almost indistinguishable. However, Algorithm 2 required a much smaller number of iterations to converge than Algorithm 1. Excluding runs 3 and 40 where the lower bound attained by Algorithms 1 and 2 differs significantly, using Algorithm 2 instead of Algorithm 1 leads on average to a reduction of 49% in the number of iterations required for convergence. The highest reduction observed is 84% at run 9. At run 3, Algorithm 2 was able to escape a local mode and attained a higher lower bound at convergence. However, at run 40, it was caught in a local mode. We re-perform run 40 using $\rho = 1.1$ and it turns out that Algorithm 2 was then able to attain the same lower bound as Algorithm 1 but in around half the number of iterations.

The typical behaviour of Algorithm 2 is illustrated in Figure 4. On the left is a plot
of the lower bound against iteration number (solid line corresponds to Algorithm 2 with \( \rho = 1.5 \) while dashed line corresponds to Algorithm 1) and on the right is a plot of the adaptive step size \( (a_t) \) used in Algorithm 2 against iteration number \( (t) \) for run 33. The step size typically increases by a factor of 1.5 at each iteration but falls back to 1 when the lower bound fails to increase. The step size may also be reduced by factors of 1.5 due to the requirement that the covariance matrix be symmetric positive definite in step 2(b) of Algorithm 2. The reduction in the number of iterations that Algorithm 2 takes to converge as compared to Algorithm 1 is 74% for run 33.

### 7.2 Rainfall-runoff data

In this example, we consider data from a deterministic rainfall-runoff model, which is a simplification of the Australian Water Balance Model (AWBM, Boughton, 2004). The AWBM estimates catchment streamflow using time series of rainfall and evapotranspiration data and is widely used in Australia for estimating catchment water yield or design flood estimation. The model has three parameters - the maximum storage capacity \( S \), the base flow index \( BFI \), and the baseflow recession factor \( K \). We have model simulations for around eleven years of average monthly potential evapotranspiration and daily rainfall data for the Barrington River catchment, located in New South Wales, Australia. The model was run for 500 different values of the parameters \( (S, K, BFI) \) generated using a maximin Latin hypercube design. We consider the AWBM streamflow response at a time of peak rainfall input as the target \( y \), and \( S \) and \( K \) as covariates, omitting \( BFI \). A small amount of independent normal random noise with standard deviation 0.01 was added to \( y \) to avoid degeneracies in regions of the space where the response tends to be identically zero. This data set has been studied in Nott et al. (2012) and is available as part of their supplementary materials. We randomly selected 100 data points as the test set and use the remaining 400 data points as the training set. This data is highly nonstationary with large flat regions, a few rapidly varying regions and the noise level changes a lot over the space.
Figure 5: Rainfall-runoff data. NMSE and MNLP values averaged over ten repetitions plotted against the number of basis functions (first column) and against the number of neighbours (second, third and fourth columns). Number of basis functions used in the local methods was 20.

We fit this data globally using the SSGP algorithm and compare these results with the adaptive neighbourhood approach, which is implemented using both the SSGP algorithm and Algorithm 2 with factor $\rho = 1.5$. For the priors, we set $A_\sigma = A_\gamma = 25$ and $\mu_\lambda^0 = 0$, $\Sigma_\lambda^0 = 100I_d$. Figure 5 shows the NMSE and MNLP values averaged over ten repetitions. For the global SSGP method, we consider the number of basis functions $m \in \{10, 25, 50, 100, 200\}$. While there is slight improvement in the NMSE values as $m$ increases, the MNLP values remain largely constant at around 3.75 even for large $m$. Due to the nonstationary nature of this data, a global stationary fit does very poorly in MNLP. For the adaptive neighbourhood approach, we consider neighbourhoods of size $k = 20, 40, 60, 80, 100$, fixing the number of basis functions, $m = 20$. For the local methods, the dotted lines correspond to results from the initial fitting where the $k$ near-
Figure 6: Rainfall-runoff data. Plot of neighbourhood of test point determined using Euclidean distance (left) and new weighted distance measure (right). Circles denote neighbours and solid circle denotes test point.

The nearest neighbours are determined based on Euclidean distance. The solid lines correspond to results from the final fit where the $k$ nearest neighbours are determined using the new distance measure with dimensions weighted according to the length scales. The performance of SSGP (fixed) and SSGP (optimized) are quite similar for large neighbourhoods. However, for small neighbourhoods, SSGP (optimized) performs poorly, especially in MNLP. The improvement brought about by adapting the neighbourhood is more apparent in the variational approach than in SSGP. It is clear that the adaptive neighbourhood approach is critical for this data set where noise levels vary considerably over the space. The variational approach performs very well when using just a small neighbourhood about each test point both in terms of NMSE and MNLP.

Figure 6 illustrates how the neighbourhood of a test point changes from the initial to the final fit for the case $k = 60$, when Algorithm 2 was being used. The plot on the left shows the neighbours (denoted by circles) of a test point (denoted by a solid circle) determined using Euclidean distance. The plot on the right shows the neighbours of the same test point determined using the new distance measure. In this case, the component of $\mu^q_\lambda$ corresponding to the covariate $S$ is much larger than that corresponding to the covariate $K$, resulting in the neighbourhood being shrunk along the $S$ axis. The adapted
neighbourhood leads to an improvement in the estimation of the predictive mean and especially the predictive variance of the test point.

7.3 Simulated data

We consider again the rainfall-runoff data in Section 7.2 and generate ten additional covariates artificially. As both covariates $S$ and $K$ lie in the interval $[0,1]$, we simulate each of the ten additional covariates randomly from the uniform distribution on the interval $[0,1]$. We compare the performance of the SSGP algorithm using a global fit with the adaptive neighbourhood approach implemented using SSGP and Algorithm 2. We set $\rho = 1.5$ in Algorithm 2 and consider the same priors as in Section 7.2. For the global SSGP method, we consider the number of basis functions, $m \in \{10, 25, 50, 100, 200\}$ and for the adaptive local neighbourhood approach, we consider neighbourhoods of size $k = 20, 40, 60, 80, 100$, fixing the number of basis functions, $m = 20$. The results are shown in Figure 7.

For the global approach, the results of SSGP (fixed) is quite similar to those in the 2 covariates case. However, there is some fluctuation in SSGP (optimized) and the results only approach that of SSGP (fixed) for large $m$. It seems that SSGP (optimized) has some difficulty discerning the relevant covariates particularly when $m$ is small. For the local approach, a small neighbourhood with $k = 20$ does not work well for both the SSGP algorithm and the variational method, indicating that a larger neighbourhood is likely required for high dimensional problems. There is some instability in both SSGP (optimized) and SSGP (fixed) for small $m$, while Algorithm 2 performs quite steadily across all $m$. There is a clear improvement in the MNLP values from adapting the neighbourhood according to the lengthscales and Algorithm 2 achieved the lowest MNLP values among the methods that were studied, using a smaller neighbourhood. The MNLP values achieved by Algorithm 2 are close to those attained in Section 7.2, indicating that the adaptive neighbourhood approach is effective in eliminating covariates of little relevance. We note that the poor performance of SSGP (optimized) in terms of MNLP is due mainly
to the underestimation of predictive variance. The variational approach is able to provide significant improvement in this aspect and is much more robust to overfitting. However, the NMSE values obtained in the adaptive neighbourhood approach are higher than those obtained in the global approach. Finally, we note that a good neighbourhood size is dependent on the number of covariance function parameters to be estimated and on the degree of nonstationarity, which is very much problem specific. Some experimentation with different neighbourhood sizes is probably necessary.

8 Conclusion

In this paper, we have presented a nonconjugate variational message passing algorithm for fitting sparse spectrum GP regression models where closed form updates are possible for
all variational parameters, except for the evaluation of $\mathcal{H}(p, q, r)$. We note that $\mathcal{H}(p, q, r)$ can be evaluated very efficiently using quadrature and there is almost no computational overhead when compared to updates based on conditionally conjugate inverse-gamma priors for the variance parameters. However, half-Cauchy priors lead to much better predictive inference especially in the adaptive neighbourhood approach where the amount of training data is small. A Bayesian approach has been adopted for parameter estimation which allows covariance function hyperparameter uncertainty to be treated and empirical results suggest that this improves prediction (especially in the MNLP values) and prevents overfitting. We also propose a novel adaptive neighbourhood technique for obtaining predictive inference which is adept at handling data with nonstationarities and this approach can be extended to large data sets as well. The simulated data set showed that weighting the dimensions according to the lengthscales estimated from an initial fit is very effective at downweighting variables of little relevance, leading to automatic variable selection and improved prediction. In addition, we introduce a technique for accelerating convergence in nonconjugate variational message passing by taking step sizes larger than one in the direction of the natural gradient of the lower bound. We do not attempt to search for the optimal step size but adopt an adaptive strategy that can be easily implemented, and empirical results indicate significant speed ups. Algorithm 2 is thus an attractive alternative for fitting sparse spectrum GP regression models, which is stable, robust to overfitting and capable of dealing with highly nonstationary data as well when used in combination with the adaptive neighbourhood approach.

9 Acknowledgements

We thank Lucy Marshall for supplying the rainfall-runoff data set. Linda S. L. Tan was partially supported as part of the Singapore Delft Water Alliance’s tropical reservoir research programme.
References

Adler, R. J. and Taylor, J. E. (2007). Random fields and geometry. Springer Monographs in Mathematics. Springer, New York.

Amari, S. (1998). Natural gradient works efficiently in learning. Neural Computation, 10, 251–276.

Attias, H. (1999). Inferring parameters and structure of latent variable models by variational Bayes. In Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (eds. K. Laskey, H. Prade), 21–30. Morgan Kaufmann, San Francisco, CA.

—— (2000). A variational Bayesian framework for graphical models. In Advances in Neural Information Processing Systems 12 (eds. S. A. Solla, T. K. Leen and K.-R. Müller), 209–215. MIT Press, Cambridge, MA.

Blei, D. M. and Jordan, M. I. (2006). Variational inference for Dirichlet process mixtures. Bayesian Analysis, 1, 121–144.

Boughton, W. (2004). The Australian water balance model. Environmental Modelling and Software, 19, 943–956.

Braun, M. and McAuliffe, J. (2010). Variational inference for large-scale models of discrete choice. Journal of the American Statistical Association, 105, 324–335.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models. Bayesian Analysis, 1, 515–533.

Gramacy, R. B. and Apley, D. W. (2013). Local Gaussian process approximation for large computer experiments. Available at arXiv:1303.0383.

Haas, T. C. (1995). Local prediction of a spatio-temporal process with an application to wet sulfate deposition. Journal of the American Statistical Association, 90, 1189–1199.

Hastie, T. and Tibshrani, R. (1996). Discriminant adaptive nearest neighbor classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 18, 607–616.
Hoffman, M. D., Blei, D. M., Wang, C. and Paisley, J. (2013). *Journal of Machine Learning Research*, in press. Available at arXiv:1206.7051.

Honkela, A., Valpola, H. and Karhunen, J. (2003). Accelerating cyclic update algorithms for parameter estimation by pattern searches. *Neural Processing Letters*, 17, 191–203.

Huang, H., Yang, B. and Hsu, C. (2005). Triple jump acceleration for the EM algorithm. In *Proceedings of the 5th IEEE International Conference on Data Mining*, 649–652. IEEE Computer Society, Washington, DC, USA.

Kim, H.-M., Mallick, B. K. and Holmes, C. C. (2005). Analyzing nonstationary spatial data using piecewise Gaussian processes. *Journal of the American Statistical Association*, 100, 653–668.

Knowles, D. A., Minka, T. P. (2011). Non-conjugate variational message passing for multinomial and binary regression. In *Advances in Neural Information Processing Systems 24* (eds. J. Shawe-Taylor, R. S. Zemel, P. Bartlett, F. Pereira and K. Q. Weinberger), 1701–1709. Neural Information Processing Systems, La Jolla, CA.

Lázaro-Gredilla, M., Quiñonero-Candela, J., Rasmussen, C. E. and Figueiras-Vidal, A. R. (2010). Sparse spectrum Gaussian process regression. *Journal of Machine Learning Research*, 11, 1865–1881.

Lázaro-Gredilla, M. and Titsias, M. K. (2011). Variational heteroscedastic Gaussian process regression. In *Proceedings of the 28th International Conference on Machine Learning* (eds. L. Getoor and T. Scheffer), 841–848. Omnipress, Madison, MI, USA.

Lindgren, F., Rue, H. and Lindström, J. (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B*, 73, 423–498.

Magnus, J. R. and Neudecker, H. (1988). Matrix differential calculus with applications in statistics and econometrics. Wiley, Chichester, UK.

Nguyen-Tuong, D., Seeger, M. and Peters, J. (2009). Model learning with local Gaussian process regression. *Advanced Robotics*, 23, 2015–2034.
Nott, D. J., Tan, S. L., Villani, M. and Kohn, R. (2012). Regression density estimation with variational methods and stochastic approximation. Journal of Computational and Graphical Statistics, 21, 797–820.

Ormerod, J. T. and Wand, M. P. (2010). Explaining variational approximations. The American Statistician, 64, 140–153.

Park, S. and Choi, S. (2010). Hierarchical Gaussian process regression. Proceedings of 2nd Asian Conference on Machine Learning (eds. M. Sugiyama and Q. Yang), 95–110.

Qi, Y. and Jaakkola, T. S. (2006). Parameter expanded variational Bayesian methods. In Advances in Neural Information Processing Systems 19 (eds. B. Schölkopf, J. Platt and T. Hofmann), 1097–1104. MIT Press, Cambridge.

Quiñonero-Candela, T. and Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. Journal of Machine Learning Research, 6, 1939–1959.

Rasmussen, C. E. and Williams, C. K. I. (2006). Gaussian Processes for Machine Learning. MIT Press, Cambridge, MA.

Ren, Q., Banerjee, S., Finley, A. O. and Hodges, J. S. (2011). Variational Bayesian methods for spatial data analysis. Computational Statistics and Data Analysis, 55, 3197–3217.

Salakhutdinov, R. and Roweis, S. (2003). Adaptive overrelaxed bound optimization methods. Proceedings of the 20th International Conference on Machine Learning (eds. T. Fawcett and N. Mishra), 664–671. AAAI Press, Menlo Park, California.

Snelson, E. and Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In Advances in Neural Information Processing Systems 18 (eds. Y. Weiss, B. Schölkopf and J. Platt), 1257–1264. MIT Press, Cambridge, MA.

—— (2007). Local and global sparse Gaussian process approximations. In Journal of Machine Learning Research - Proceedings Track, Vol. 2: 11th International Conference on Artificial Intelligence and Statistics, 524–531.
Stein, M. L., Chi, Z. and Welty, L. J. (2004). Approximating likelihoods for large spatial data
sets. *Journal of the Royal Statistical Society: Series B*, 66, 275–296.

Tan, L. S. L. and Nott, D. J. (2013a). Variational inference for generalized linear mixed models
using partially non-centered parametrizations. *Statistical Science*, 28, 168–188.

—— (2013b). A stochastic variational framework for fitting and diagnosing generalized linear
mixed models. Available at arXiv:1208.4949.

Titsias, M. K. (2009). Variational learning of inducing variables in sparse Gaussian processes.
In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*
(eds. D. van Dyk and M. Welling), 567-574.

Urtasun, R. and Darrell, T. (2008). Sparse probabilistic regression for activity-independent
human pose inference. In *IEEE Conference on Computer Vision and Pattern Recognition 2008*,
pg. 1-8.

Vecchia, A. V. (1988). Estimation and model identification for continuous spatial processes.
*Journal of the Royal Statistical Society: Series B*, 50, 297–312.

Walder, C., Kim, K. I. and Schölkopf, B. (2008). Sparse multiscale Gaussian process regression.
In *Proceedings of the 25th International Conference on Machine Learning* (eds. A. McCallum
and S. am Roweis), 1112–1119. ACM Press, New York.

Wand, M. P., Ormerod, J. T., Padoan, S. A. and Frühwirth, R. (2011). Mean field variational
Bayes for elaborate distributions. *Bayesian Analysis*, 6, 847–900.

Wand, M. P. (2013). Fully simplified multivariate normal updates in non-conjugate variational
message passing. Available at [http://works.bepress.com/matt_wand/8](http://works.bepress.com/matt_wand/8)

Wang, B. and Titterington, D. M. (2005). Inadequacy of interval estimates corresponding to
variational Bayesian approximations. In *Proceedings of the 10th International Workshop on
Artificial Intelligence and Statistics* (eds. R. G. Cowell and Z. Ghahramani), 373–380. Society
for Artificial Intelligence and Statistics.
— (2006). Convergence properties of a general algorithm for calculating variational Bayesian estimates for a normal mixture model. *Bayesian Analysis*, 3, 625–650.

Winn, J. and Bishop, C.M. (2005). Variational message passing. *Journal of Machine Learning Research*, 6, 661–694.

## A Derivation of $E_q(Z)$ and $E_q(Z^T Z)$

**Lemma 1.** Suppose $\lambda \sim N(\mu, \Sigma)$ and $t_1$, $t_2$ are fixed vectors the same length as $\lambda$. Let $t_{12} = t_1 - t_2$ and $t_{12}^+ = t_1 + t_2$, then

$$E\{\cos(t_1^T \lambda) \cos(t_2^T \lambda)\} = \frac{1}{2} \left[ \exp(-\frac{1}{2} t_{12}^{-T} \Sigma t_{12}^{-}) \cos(t_{12}^{-T} \mu) + \exp(-\frac{1}{2} t_{12}^{+T} \Sigma t_{12}^{+}) \cos(t_{12}^{+T} \mu) \right]$$

$$E\{\sin(t_1^T \lambda) \sin(t_2^T \lambda)\} = \frac{1}{2} \left[ \exp(-\frac{1}{2} t_{12}^{-T} \Sigma t_{12}^{-}) \cos(t_{12}^{-T} \mu) - \exp(-\frac{1}{2} t_{12}^{+T} \Sigma t_{12}^{+}) \cos(t_{12}^{+T} \mu) \right]$$

$$E\{\sin(t_1^T \lambda) \cos(t_2^T \lambda)\} = \frac{1}{2} \left[ \exp(-\frac{1}{2} t_{12}^{-T} \Sigma t_{12}^{-}) \sin(t_{12}^{-T} \mu) + \exp(-\frac{1}{2} t_{12}^{+T} \Sigma t_{12}^{+}) \sin(t_{12}^{+T} \mu) \right]$$

By setting $t_2 = 0$ in the first and third expressions, we get

$$E\{\cos(t_1^T \lambda)\} = \exp(-\frac{1}{2} t_1^T \Sigma t_1) \cos(t_1^T \mu) \quad \text{and} \quad E\{\sin(t_1^T \lambda)\} = \exp(-\frac{1}{2} t_1^T \Sigma t_1) \sin(t_1^T \mu).$$

**Proof.** $E[\exp\{i\lambda^T(t_1 - t_2)\}] = \exp\{i \mu^T(t_1 - t_2) - \frac{1}{2}(t_1 - t_2)^T \Sigma (t_1 - t_2)\}$ implies

$$E[\cos(\lambda^T(t_1 - t_2))] = E\{\cos(t_1^T \lambda) \cos(t_2^T \lambda) + \sin(t_1^T \lambda) \sin(t_2^T \lambda)\}$$

$$= \exp\left\{-\frac{1}{2}(t_1 - t_2)^T \Sigma (t_1 - t_2) \right\} \cos\{\mu^T(t_1 - t_2)\} \quad (16)$$

$$E[\sin(\lambda^T(t_1 - t_2))] = E\{\sin(t_1^T \lambda) \cos(t_2^T \lambda) - \cos(t_1^T \lambda) \sin(t_2^T \lambda)\}$$

$$= \exp\left\{-\frac{1}{2}(t_1 - t_2)^T \Sigma (t_1 - t_2) \right\} \sin\{\mu^T(t_1 - t_2)\}. \quad (17)$$

---

33
Replacing $t_2$ by $-t_2$, we get

$$E[\cos\{\lambda^T(t_1 + t_2)\}] = E\{\cos(t_1^T \lambda) \cos(t_2^T \lambda) - \sin(t_1^T \lambda) \sin(t_2^T \lambda)\}$$

$$= \exp\{-\frac{1}{2}(t_1 + t_2)^T \Sigma(t_1 + t_2)\} \cos\{\mu^T(t_1 + t_2)\}$$

(18)

$$E[\sin\{\lambda^T(t_1 + t_2)\}] = E\{\sin(t_1^T \lambda) \cos(t_2^T \lambda) + \cos(t_1^T \lambda) \sin(t_2^T \lambda)\}$$

$$= \exp\{-\frac{1}{2}(t_1 + t_2)^T \Sigma(t_1 + t_2)\} \sin\{\mu^T(t_1 + t_2)\}. \quad (19)$$

(16) + (18) gives the first equation of the lemma, (16)–(18) gives the second and (17) + (19) gives the third.

Using Lemma 1, we have $E_q(Z) = [E_q(Z_1), \ldots, E_q(Z_n)]^T$ where

$$E_q(Z_i^T) = \left[ \exp\{-\frac{1}{2}t_{i_1}^T \Sigma^q_{\lambda_{i_1}} t_{i_1}\} \cos(t_{i_1}^T \mu_{\lambda}^q), \ldots, \exp\{-\frac{1}{2}t_{im}^T \Sigma^q_{\lambda_{im}} t_{im}\} \cos(t_{im}^T \mu_{\lambda}^q), \exp\{-\frac{1}{2}t_{i_1}^T \Sigma^q_{\lambda_{i_1}} \sin(t_{i_1}^T \mu_{\lambda}^q), \ldots, \exp\{-\frac{1}{2}t_{im}^T \Sigma^q_{\lambda_{im}} \sin(t_{im}^T \mu_{\lambda}^q) \right]$$

and $t_{ir} = s_r \odot x_i$ for $i = 1, \ldots, n$, $r = 1, \ldots, m$. We also have $E_q(Z^T Z) = \sum_{i=1}^n E_q(Z_i Z_i^T)$ where $E_q(Z_i Z_i^T) = \left[ P_{ir}, Q_{ir}, R_{ir} \right]$, where $P_i, Q_i, R_i$ are all $m \times m$ matrices and

$$P_{ir} = \frac{1}{2} \left\{ \exp\{-\frac{1}{2}t_{ir}^T \Sigma_{\lambda_{ir}} t_{ir}\} \cos(t_{ir}^T \mu_{\lambda}^q) + \exp\{-\frac{1}{2}t_{ir}^T \Sigma_{\lambda_{ir}} t_{ir}\} \cos(t_{ir}^T \mu_{\lambda}^q) \right\},$$

$$Q_{ir} = \frac{1}{2} \left\{ \exp\{-\frac{1}{2}t_{ir}^T \Sigma_{\lambda_{ir}} t_{ir}\} \sin(t_{ir}^T \mu_{\lambda}^q) + \exp\{-\frac{1}{2}t_{ir}^T \Sigma_{\lambda_{ir}} t_{ir}\} \sin(t_{ir}^T \mu_{\lambda}^q) \right\},$$

$$R_{ir} = \frac{1}{2} \left\{ \exp\{-\frac{1}{2}t_{ir}^T \Sigma_{\lambda_{ir}} t_{ir}\} \cos(t_{ir}^T \mu_{\lambda}^q) - \exp\{-\frac{1}{2}t_{ir}^T \Sigma_{\lambda_{ir}} t_{ir}\} \cos(t_{ir}^T \mu_{\lambda}^q) \right\},$$

$t_{ir}^+ = t_{ir} - t_{il}, t_{ir}^- = t_{ir} + t_{il}$ for $r = 1, \ldots, m$, $l = 1, \ldots, m$. 

34
B Derivation of lower bound

From (6), the lower bound is given by \( \mathcal{L} = E_q\{\log p(y, \theta)\} - E_q\{\log q(\theta)\} \) where

\[
E_q\{\log p(y, \theta)\} = E_q\{\log p(y|\alpha, \lambda, \gamma)\} + E_q\{\log p(\alpha|\sigma)\} + E_q\{\log p(\lambda)\} 
+ E_q\{\log p(\sigma)\} + E_q\{\log p(\gamma)\},
\]

\[
E_q\{\log q(\theta)\} = E_q\{\log q(\alpha)\} + E_q\{\log q(\lambda)\} + E_q\{\log q(\sigma)\} + E_q\{\log q(\gamma)\}.
\]

The terms in the lower bound can be evaluated as follows:

- \( E_q\{\log p(y|\alpha, \beta, \lambda, \gamma)\} = -\frac{n}{2} \log(2\pi) - \frac{n}{2} E_q(\log \gamma^2) - \frac{1}{2} \mathcal{H}(n, C^q_{\gamma}, A^2_{\gamma}) \left[ y^T y - 2y^T E_q(Z)\mu^q_{\alpha} + \text{tr}\{\left(\mu^q_{\alpha}\mu^q_{\alpha}^T + \Sigma^q_{\alpha}\right) E_q(Z^T Z)\}\right] \)

- \( E_q\{\log p(\alpha|\sigma)\} = -m \log(2\pi) - m E_q\{\log \sigma^2\} + m \log m 
- \frac{m}{2} \mathcal{H}(2m, C^q_{\sigma}, A^2_{\sigma}) \left\{\mu^q_\alpha \Sigma^{-1} \Sigma^q_\alpha + \text{tr}(\Sigma^q_\alpha)\right\} \)

- \( E_q\{\log p(\lambda)\} = -\frac{d}{2} \log(2\pi) - \frac{1}{2} \log |\Sigma^q_\lambda| - \frac{1}{2} (\mu^q_\lambda - \mu^0_\lambda)^T \Sigma^{-1}_\lambda (\mu^q_\lambda - \mu^0_\lambda) - \frac{1}{2} \text{tr}(\Sigma^{-1}_\lambda \Sigma^q_\lambda) \)

- \( E_q\{\log p(\sigma)\} = \log(2A_\sigma) - \log \pi - E_q\{\log(A^2_\sigma + \sigma^2)\} \)

- \( E_q\{\log p(\gamma)\} = \log(2A_\gamma) - \log \pi - E_q\{\log(A^2_\gamma + \gamma^2)\} \)

- \( E_q\{\log q(\alpha)\} = -m \log(2\pi) - \frac{1}{2} \log |\Sigma^q_\alpha| - m \)

- \( E_q\{\log q(\lambda)\} = -\frac{d}{2} \log(2\pi) - \frac{1}{2} \log |\Sigma^q_\lambda| - \frac{d}{2} \)

- \( E_q\{\log p(\sigma)\} = -C^q_{\sigma} \frac{\mathcal{H}(2m, C^q_{\sigma}, A^2_{\sigma})}{\mathcal{H}(2m-2, C^q_{\sigma}, A^2_{\sigma})} - \log(2m - 2, C^q_{\sigma}, A^2_{\sigma}) - 2m E_q\{\log \sigma\} 
- E_q\{\log(A^2_\sigma + \sigma^2)\} \)

- \( E_q\{\log p(\gamma)\} = -C^q_{\gamma} \frac{\mathcal{H}(n, C^q_{\gamma}, A^2_{\gamma})}{\mathcal{H}(n-2, C^q_{\gamma}, A^2_{\gamma})} - \log(n - 2, C^q_{\gamma}, A^2_{\gamma}) - n E_q\{\log \gamma\} 
- E_q\{\log(A^2_\gamma + \gamma^2)\} \)

Putting these terms together and making use of the updates in steps 5 and 6 of Algorithm 3 gives the lower bound in (12).
C Derivation of simplified updates in Algorithm 2

It can be shown (see [Wand 2013; Tan and Nott 2013a]) that the natural parameter of $q(\lambda) = N(\mu^q_\lambda, \Sigma^q_\lambda)$ is $\eta_\lambda = \begin{bmatrix} -\frac{1}{2} D_d^T \text{vec}(\Sigma^{-1}_\lambda) \\ \Sigma^{-1}_\lambda \mu^q_\lambda \end{bmatrix}$, where $D_d$ is a unique $d^2 \times \frac{d^2}{2}(d + 1)$ matrix that transforms $\text{vech}(A)$ into $\text{vec}(A)$ for any $d \times d$ symmetric square matrix $A$, that is, $D_d\text{vech}(A) = \text{vec}(A)$. We use $\text{vech}(A)$ to denote the $\frac{1}{2}d(d + 1)$ vector obtained from $\text{vec}(A)$ by eliminating all supradiagonal elements of $A$. [Magnus and Neudecker 1988] is a good reference for the matrix differential calculus involved in the derivation below. From (13) and (Tan and Nott 2013a, pg. 7), we have

$$\begin{bmatrix} -\frac{1}{2} D_d^T \text{vec}(\Sigma^{-1}_\lambda(t)) \\ \Sigma^{-1}_\lambda(t) \mu^q_\lambda(t) \end{bmatrix} = (1 - a_t) \begin{bmatrix} -\frac{1}{2} D_d^T \text{vec}(\Sigma^{-1}_\lambda(t-1)) \\ \Sigma^{-1}_\lambda(t-1) \mu^q_\lambda(t-1) \end{bmatrix} + a_t \begin{bmatrix} D_d^T \\ -2(\mu^{q(t-1)}_\lambda \otimes I)D_d^T D_d^T \end{bmatrix} \sum_{a \in N(\lambda)} \begin{bmatrix} \frac{\partial S_a}{\partial \text{vec}(\Sigma^q_\lambda)} \\ \frac{\partial S_a}{\partial \mu^q_\lambda} \end{bmatrix}, \quad (20)$$

where $\frac{\partial S_a}{\partial \text{vec}(\Sigma^q_\lambda)}$ and $\frac{\partial S_a}{\partial \mu^q_\lambda}$ are evaluated at $\Sigma^q_\lambda = \Sigma^q_\lambda(t-1)$ and $\mu^q_\lambda = \mu^q_\lambda(t-1)$. Let $\sum_{a \in N(\lambda)} \frac{\partial S_a}{\partial \text{vec}(\Sigma^q_\lambda)} = \frac{1}{2} \text{vec}(G)$. The first line of (20) simplifies to

$$\Sigma^{-1}_\lambda(t) = (1 - a_t) \Sigma^{-1}_\lambda(t-1) + a_t G \quad \Rightarrow \Sigma^q(t) = \{(1 - a_t) \Sigma^{-1}_\lambda(t-1) + a_t G\}^{-1}.$$ 

The second line of (20) gives

$$\Sigma^{-1}_\lambda(t) \mu^q_\lambda(t) = (1 - a_t) \Sigma^{-1}_\lambda(t-1) \mu^q_\lambda(t-1) + a_t \Sigma^q(t) \mu^q_\lambda(t-1) + a_t \sum_{a \in N(\lambda)} \frac{\partial S_a}{\partial \mu^q_\lambda} = \Sigma^{-1}_\lambda(t) \mu^q_\lambda(t-1) + a_t \sum_{a \in N(\lambda)} \frac{\partial S_a}{\partial \mu^q_\lambda} \Rightarrow \mu^q_\lambda(t) = \mu^q_\lambda(t-1) + a_t \sum_{a \in N(\lambda)} \frac{\partial S_a}{\partial \mu^q_\lambda}.$$ 

36