Supplementary materials

A Reverse mode sensitivities derivations

We are interested in reverse mode differentiation. That is, we propagate the gradient backward in the chain of operations from the scalar objective to the upstream parameters.

For a chain of operations $X \to Y \to \cdots \to c$ (with X, Y matrices), we have

$$dc = \sum_{ij} \frac{\partial c}{\partial Y_{ij}} dY_{ij}$$

Noting $\bar{Y} = \left[\frac{\partial c}{\partial Y_{ij}}\right]$, we can rewrite this as $dc = Tr(\bar{Y}^T dY)$. When doing reverse mode differentiation, we propagate the errors from the objective. From the mapping from $X \to Y$, we figure out matrices L_X and R_X such that $dY = L_X(dX)R_X$. It follows that $dc = Tr(\bar{Y}^T L_X(dX)R_X)$ and, after a rotation in the trace, we can then identify $\bar{X}^T = R_X \bar{Y}^T L_X$.

A.1 Outer product $m, v \to mv^T$

$o = mv^T$, so $\mathrm{d}o = \mathrm{d}m\, v^T + m\, \mathrm{d}v^T$
For fixed m: $\mathrm{d}o = m\, \mathrm{d}v^T$ so

$$\bar{v} = \bar{o}^T m.$$

For fixed v: $\mathrm{d}o = \mathrm{d}m v^T$ so

$$\bar{m}^T = v^T \bar{o}^T,$$
$$\bar{m} = \bar{o} v.$$

A.2 Product $B_1^X, B_2^Y \to B_1^X B_2^Y$

X and Y are both either T (for transpose) or 1 (for identity). This allows to write all cases of argument transposition at once.
$P = B_1^X B_2^Y$, so $\mathrm{d}P = \mathrm{d}(B_1^X)B_2^Y + B_1^X \mathrm{d}(B_2^Y)$.
For fixed B_1^X: $\mathrm{d}P = B_1^X \mathrm{d}(B_2^Y)$ so

$$\bar{B}_2^{YT} = \bar{P}^T B_1^X = \mathbb{P}(\bar{P}^T, B_1^X).$$

For fixed B_2^Y: $\mathrm{d}P = \mathrm{d}(B_1^X)B_2^Y$ so

$$\bar{B}_1^{XT} = B_2^Y \bar{P}^T = \mathbb{P}(B_2^Y, \bar{P}^T).$$

A.3 Solve $L^X, B^Y \to L^{-X} B^Y$

X and Y are as defined in the product section.

$S = L^{-X} B^Y$, so

$$\begin{aligned}\mathrm{d}S &= \mathrm{d}(L^{-X})B^Y + L^{-X}\mathrm{d}(B^Y) \\ &= -L^{-X}\mathrm{d}(L^X)L^{-X}B^Y + L^{-X}\mathrm{d}(B^Y).\end{aligned}$$

For fixed B^Y: $\mathrm{d}S = -L^{-X}\mathrm{d}(L^X)L^{-X}B^Y$ so $\bar{L}^{XT} = -L^{-X}B^Y\bar{S}^T L^{-X}$

$$\begin{aligned}\bar{L}^{XT} &= -L^{-X}B^Y\bar{S}^T L^{-X} \\ &= -(L^{-X}B^Y)(L^{-XT}\bar{S})^T \\ &= -\mathbb{P}(\$(L^X, B^Y), \$(L^{XT}, \bar{S})^T).\end{aligned}$$

For fixed L^X: $\mathrm{d}S = L^{-X}\mathrm{d}(B^Y)$ so

$$\begin{aligned}\bar{B^Y}^T &= \bar{S}^T L^{-X} \\ \bar{B^Y} &= L^{-XT}\bar{S} \\ &= \mathbb{S}(L^{XT}, \bar{S}).\end{aligned}$$

A.4 Product $B, v \to Bv$

$p = Bv$, so $\mathrm{d}p = \mathrm{d}Bv + B\mathrm{d}v$.

For fixed v, $\mathrm{d}p = \mathrm{d}Bv$, so

$$\begin{aligned}\bar{B}^T &= v\bar{p}^T \\ \bar{B} &= \mathbb{O}(\bar{p}, v^T).\end{aligned}$$

For fixed B, $\mathrm{d}p = B\mathrm{d}v$, so

$$\begin{aligned}\bar{v}^T &= \bar{p}^T B \\ \bar{v} &= \mathbb{P}(B^T, \bar{p}).\end{aligned}$$

A.5 Solve $L, v \to L^{-1}v$

$u = L^{-1}v$, so

$$\begin{aligned}\mathrm{d}u &= \mathrm{d}(L^{-1}v) \\ &= \mathrm{d}(L^{-1})v + L^{-1}\mathrm{d}v \\ &= -L^{-1}(\mathrm{d}L)L^{-1}v + L^{-1}\mathrm{d}v.\end{aligned}$$

For fixed L, $\mathrm{d}u = L^{-1}\mathrm{d}v$, so

$$\begin{aligned}\bar{v}^T &= \bar{u}^T L^{-1} \\ &= \mathbb{S}(L^T, u)^T.\end{aligned}$$

For fixed v, $\mathrm{d}u = -L^{-1}(\mathrm{d}L)L^{-1}v$, so

$$\begin{aligned}\bar{L}^T &= -L^{-1}v\bar{u}^T L^{-1} \\ &= \mathbb{P}(\mathbb{S}(L, v), \mathbb{S}(L^T, u)^T).\end{aligned}$$

B Algorithms

We report algorithms operating on a *dense* representation of the banded matrices (the usual i, j indexing). In practice these only read and write into the band of the matrices involved as input or output. A derivation using the (*diag, column*) indexing would reflect this fact more clearly but blurs the inner working of the algorithms. In a similar spirit, our algorithms sometimes include avoidable instantiations of temporary variables that are kept for clarity. For reverse mode differentiation we refer the reader to (Giles, 2008) from which we follow the notation.

Algorithm 1 Cholesky algorithm for banded matrices.

Input : Q: an $n \times n$ symmetric positive definite matrix with l lower sub-diagonals.
Output : L: an $n \times n$ lower triangular matrix with l lower sub-diagonals.

```
Initialize L with zeros
for i in 0 to n − 1 do
    for j in max(i − l, 0) to i + 1 do
        m = max(i − l, j − l, 0)
        s = sum(L[i, m : j] ∗ L[j, m : j])
        if i==j then
            L[i, j] = sqrt(Q[i, j] − s)
        else
            L[i, j] = (Q[i, j] − s)/L[j, j]
        end
    end
end
```

Algorithm 2 Reverse mode differentiation for the Cholesky operator for banded matrices.

Input : L: the $n \times n$ Cholesky factor of Q, with l lower sub-diagonals.
L_b: the $n \times n$ reverse mode derivatives of L with respect to the objective function.
Output : Q_b: the $n \times n$ reverse mode derivatives of Q with respect to the objective function.

```
Initialize Q_b with zeros
for i in n-1 downto 0 do
    j_stop = max(i − l, 0)
    for j in i to j_stop do
        if j == i then
            Q_b[i, i] = 1/2 ∗ L_b[i, i]/L[i, i]
        else
            Q_b[i, j] = L_b[i, j]/L[j, j]
            L_b[j, j] − = L_b[i, j] ∗ L[i, j]/L[j, j]
        end
        for l in j-1, in j_stop do
            L_b[i, l] − = Q_b[i, j] ∗ L[j, l]
            L_b[j, l] − = Q_b[i, j] ∗ L[i, l]
        end
    end
end
```

Algorithm 3 Algorithm for the inverse of a matrix from its Cholesky decomposition.

Input : L: an $n \times n$ lower triangular matrix with l lower sub-diagonals.
Output : S: an $n \times n$ lower triangular matrix with l lower sub-diagonals of $[LL^T]^{-1}$.

Initialize Sym as a (symmetric) $n \times n$ banded matrix of *both* lower and upper bandwidth l
$vec = \text{diag}(L)$
$U = \text{transpose}(L/vec)$
for j **in** $n-1,$ **downto** 0 **do**
  for i **in** j **downto** $max(j-l+1, 0)$ **do**
    $Sym[i,j] = -\text{sum}(U[i, i+1:i+l]Sym[i+1:i+l, j])$
    $Sym[j,i] = Sym[i,j]$
    if i==j **then**
      $Sym[i,i] += 1/(vec[i])^2$
    end
  end
end
$S = \text{lower\_band}(Sym)$

Algorithm 4 Reverse mode differentiation of the inverse from Cholesky.

Input : L: L: the $n \times n$ Cholesky factor of Q, with l lower sub-diagonals.
S: The $n \times n$ output of the subset inverse from Cholesky with l lower sub-diagonals.
bS: the $n \times n$ reverse mode derivatives of B with respect to the objective function.
Note that both S and bS should be treated as symmetric *banded matrices, with bS copied locally (modified in place).*

Output : bL: $n \times n$ the reverse mode derivatives of L with respect to the objective function.

Initialize bU as a banded matrix similar to U, filled with zeros.
Initialize *bvec_inv_2* as a vector of size n filled with zeros.

$vec = \text{diag}(L)$
$U = \text{transpose}(L/vec)$

for j **in** 0 **to** $n-1$ **do**
  for i **in** $\max(0, j-k+1)$ **to** j **do**
    if $i == j$ **then**
      $bvec\_inv\_2[i] += bS[i,i]$
    end
    % *Grad of:* $S[j,i] = S[i,j]$
    $tmp = bS[j,i]$
    $bS[j,i] = 0$
    $bS[i,j] += tmp$

    % *Grad of:* $S[i,j] = -sum(U[i,i+1:i+k]S[i+1:i+k,j])$
    $bU[i,i+1:i+k] -= S[i+1:i+k,j] * bS[i,j]$
    $bS[i+1:i+k,j] -= U[i,i+1:i+k] * bS[i,j]$
    $bS[i,j] = 0$
  end
end

% *Grad of:* $U = transpose(L/vec)$
$bL = \text{transpose}(bU)/vec$
% *Grad of* $1/vec^2$
$bvec = -2 * bvec\_inv\_2/vec^3$
% *Grad of:* $1/vec$
$bvec -= \text{sum}(\text{transpose}(bU)L, 0)/(vec^2)$
% *Grad of:* $vec = diag(L)$
$bL += \text{diag}(bvec)$

Algorithm 5 Solve of a lower triangular banded matrix by an arbitrary banded matrix.

Input : L: An $n \times n$, lower-triangular banded matrix with l sub-diagonal
R: An $n \times n$ banded matrix with u upper-diagonals

Output : O: An $n \times n$ with l lower and u upper sub-diagonals of $L^{-1}R$.

for k **in** $-u$ **to** l **do**
  for i **in** $\min(n+k-1, n-1)$ **downto** $\max(0, k)$ **do**
    r = ***if*** $(i, i-k) \in band(R)$ ***then*** $R[i, i-k]$ ***else*** 0
    $dot\_product = \text{dot}(L[i,:], O[:, i-k])$
    $O(i, i-k) = (r - dot\_product)/L(i,i)$
  end
end

C Pseudocode of algorithms

We here provide details on how we used our operators for banded matrices in the algorithms we proposed. Implementation details are also given for the pre-existing algorithms we used as comparisons.

C.1 log likelihood evaluation (for HMC)

In section 3.2, we explained how to perform gradient based MCMC in GMRF models. Algorithm 6 gives some details on how we evaluate the likelihood given sampled parameters θ and whitened latent v.

Algorithm 6 Log Likelihood evaluation

Input : white noise sample v: vector of length n
observations Y: vector of length n
precision Q: $n \times n$ symmetric matrix with l sub-diagonals
parameter θ and density $p(\theta)$
Output : Log likelihood $\log p(v, \theta, Y)$

% Cholesky factor of Q
$L_Q \leftarrow C(Q)$ % $n \times n$, l sub-diagonals
% Construct observation sample
$F \leftarrow \mathbb{S}(L_Q^T, v)$
% Compute log-likelihood
$\log p(v, \theta, Y) \leftarrow \log\, p(v) + \log\, p(\theta) + \sum_{i=1}^{n} \log\, p(y_i|\theta, F_i)$.
return $\log p(v, \theta, Y)$

C.2 Variational Inference Objective

In section 3.3, we introduced a novel algorithm to perform variational inference in GMRF models using a precision parameterised Gaussian variational distribution. Algorithm 7 provides some details of how we use our operators for banded matrices to evaluate the variational lower bound to the marginal likelihood.

Algorithm 7 Variational Inference objective evaluation.

Input : prior parameters:
m_p: vector of length n
L_p: $n \times n$ lower triangular matrix with l_p sub-diagonals
variational parameters:
m_q: vector of length n
L_q: $n \times n$ lower triangular matrix with l_q sub-diagonals
observations Y: vector of length n
Output : variational objective $\mathcal{L}$

% Compute marginal variance
$\sigma_F^2 \leftarrow diag[\mathbb{I}(L_F)]$.
% Evaluate KL divergence
$KL \leftarrow \frac{1}{2}\left(\text{tr}(\mathbb{I}(L_q)\mathbb{P}(L_p, L_p^T)) + 2\sum_i(\log[L_q]_{ii} - \log[L_p]_{ii}) + ||\mathbb{P}(L_p^T, m_p - m_q)||^2 - n\right)$
% Evaluate variational expectations
$VE \leftarrow \sum_{i=1}^{n} \mathbb{E}_{q(F_i)=\mathcal{N}(\mu_{F,i}, \sigma_{F,i}^2)} \log p(Y_i|F_i)$
% Return variational lower bound
$\mathcal{L} \leftarrow VE - KL$
return $\mathcal{L}$

C.3 Marginal Likelihood

In section 3.1, we explained how to efficiently compute marginal likelihood compuation in conjugate GMRFs with partial observations. In Algorithm 8, we give further details about how we use our operators for banded matrices to achieve these efficient compuations.

The generative model is the following: latent vector f has a zero mean multivariate Gaussian prior with precision Q, observation model is additive white noise with variance τ^2.

Algorithm 8 Marginal Likelihood

Input : parameter θ
prior precision Q: $n \times n$ symmetric matrix with l sub-diagonals
observations Y: vector of length n
indicator matrix E: matrix of size $n \times N$
Output : $\log p(Y|\theta)$

% Cholesky factorization of the posterior precision
$L \leftarrow \mathbb{C}(Q + \tau^{-2} E^T E)$ % note that $E^T E$ is diagonal
% Cholesky factorization of the prior precision
$L' \leftarrow \mathbb{C}(Q)$
% Evaluation of the Marginal likelihood
$R \leftarrow \mathbb{S}(L, E^T Y)$
$\log p(Y|\theta) \leftarrow -\frac{n}{2}\log(2\pi) - \log|L| + \log|L'| - \frac{1}{2\tau^2} Y^T Y + \frac{1}{2\tau^4} R^T R$
return $\log p(Y|\theta)$

C.4 Classical Kalman filter recursions

In section 5.2, we compared our implementation of Gaussian process regression (for kernels have a state-space representation) using our operators to traditional implementations using Kalman filtering. In this section and summarized in Algorithm 9 are derivations of the classic Kalman filter recursions.

We implement the filtering recursions for a linear state-space model with state dimension d and observation dimension e defined as

$$F_t \sim \mathcal{N}(F_t; A_t F_{t-1} + b_t; Q_t), \quad Y_t \sim \mathcal{N}(Y_t; H F_t + c, R),$$

with $F_0 \sim \mathcal{N}(F_0; \mu_0, \Sigma_0)$. For a sequence of T observations stacked into $Y^T = [Y_1^T, \dots, Y_T^T] \in \mathbb{R}^{Te}$, the aim is to evaluate $p(Y)$ which can be expressed in terms of conditionals densities $p(Y) = \prod_t p(Y_t|Y_{1\dots t-1})$. These conditional densities can be obtained after one pass of the Kalman filtering recursions as follows:

Define $p(F_t|Y_{1\dots t-1}) = \mathcal{N}(F_t; \mu_t, \Sigma_t)$. We update the belief on F_t after observing Y_t:

$$\begin{aligned} p(F_t|Y_{1\dots t}) &= p(F_t|Y_{1\dots t-1}, Y_t) \\ &\propto p(Y_t|F_t, Y_{1\dots t-1}) p(F_t|Y_{1\dots t-1}) \\ &\propto p(Y_t|F_t) p(F_t|Y_{1\dots t-1}) \\ &= \mathcal{N}(F_t; \bar{\mu}_t, \bar{\Sigma}_t) \end{aligned}$$

with

$$\begin{aligned} \bar{\Sigma}_t &= (\Sigma_t^{-1} + H^T R^{-1} H)^{-1} \\ &= \Sigma_t - \Sigma_t H^T (R + H \Sigma_t H^T)^{-1} H \Sigma_t \\ &= (I - K_t H)\Sigma_t, \quad K_t = \Sigma_t H^T (R + H \Sigma_t H^T)^{-1} \\ \bar{\mu}_t &= \bar{\Sigma}_t (\Sigma_t^{-1} \mu_t + H^T R^{-1} (Y_t - c)) \\ &= (I - K_t H)\mu_t + K_t (Y_t - c). \end{aligned}$$

K_t corresponds to the Kalman gain. Then we predict one step ahead:

$$\begin{aligned} p(F_{t+1}|Y_{1\dots t}) &= \int dF_t \, p(F_{t+1}|F_t) p(F_t|Y_{1\dots t}) \\ &= \mathcal{N}(F_t; \mu_{t+1}, \Sigma_{t+1}), \end{aligned}$$

so

$$\begin{aligned}\Sigma_{t+1} &= A_{t+1}\bar{\Sigma}_t A_{t+1}^T + Q_{t+1}\\ \mu_{t+1} &= A_{t+1}\bar{\mu}_t + b_{t+1}.\end{aligned}$$

In the end, for the marginal likelihood computation, we need:

$$\begin{aligned}p(Y_t|Y_{1\dots t-1}) &= \int dF_t p(Y_t|F_t)p(F_t|Y_{1\dots t-1})\\ &= \mathcal{N}(Y_t; H\mu_t, H\Sigma_t H^T + R).\end{aligned}$$

Algorithm 9 Kalman Filtering

Input : parameters $\theta = [A_t, b_t, Q_t, H, c, R]$, Observations Y
Output : $\log p(Y|\theta)$

```
L ← 0
for t = 1...T do
  % incorporation of observation Y_t
  K_t ← Σ_t H^T (R + H Σ_t H^T)^{-1}
  Σ̄_t ← (I − K_t H) Σ_t
  μ̄_t ← (I − K_t H) μ_t + K_t (Y_t − c)
  % prediction one step ahead
  Σ_{t+1} ← A_{t+1} Σ̄_t A_{t+1}^T + Q_{t+1}
  μ_{t+1} ← A_{t+1} μ̄_t + b_{t+1}
  % evaluation of the log conditional log p(Y_t|Y_{1...t-1})
  L ← L + log N(Y_t; H μ_t, H Σ_t H^T + R)
end for
return L
```

C.5 Gaussian process regressiong with banded precision

Our implementation of Gaussian process regression, for the experiment of section 5.2, treats the chain of latent states $F^T = [F_0^T, ..., F_T^T] \in \mathbb{R}^{(T+1)d}$ and observations $Y^T = [Y_1^T, ..., Y_T^T] \in \mathbb{R}^{Te}$ as a big multivariate normal distribution $p(Y, F)$ that factorises into a prior and a likelihood as $p(Y, F) = p(Y|F)p(F)$. Both terms can be seen as multivariate normal densities on F, albeit a possibly degenerate one for $p(Y|F)$. They can be expressed in terms of their natural parameters:

$$\begin{aligned}p(F) &= \mathcal{N}(F; \eta_0, \Lambda_0)\\ p(Y|F) &= \mathcal{N}(F; \eta_1, \Lambda_1),\end{aligned}$$

where only η_1 depends on the data Y.

The density of a multivariate normal distribution with natural parameterisation η, Λ is:

$$\begin{aligned}\mathcal{N}(F; \eta, \Lambda) &= \exp\left(a(\eta, \Lambda) + \eta^T F + \frac{1}{2}F^T \Lambda F\right)\\ a(\eta, \Lambda) &= -\frac{1}{2}\left(-\log|\Lambda| + Td\log 2\pi + \eta^T \Lambda^{-1}\eta\right)\end{aligned}$$

Hence the joint over latent F and observations Y is

$$\begin{aligned}p(F, Y) &= \mathcal{N}(F, \eta_0, \Lambda_0)\mathcal{N}(F, \eta_1, \Lambda_1)\\ &= \exp\left(a(\Lambda_0, \eta_0) + a(\Lambda_1, \eta_1) - a(\Lambda_2, \eta_2)\right)\mathcal{N}(F; \eta_2, \Lambda_2)\end{aligned}$$

with $(\eta_2, \Lambda_2) = (\eta_0, \Lambda_0) + (\eta_1, \Lambda_1)$ the parameters of the Gaussian posterior on F.

Marginalizing over F provides the log marginal

$$\log p(Y) = a(\Lambda_0, \eta_0) + a(\Lambda_1, \eta_1) - a(\Lambda_2, \eta_2).$$

Due to the structure of the model, the prior and likelihood terms can be further simplified:
Since $p(F) = p(F_0)\prod_t p(F_t|F_{t-1})$, we have $|\Lambda_0| = |\Sigma_0|^{-1}\Pi_t|Q_t|^{-1}$.
Since $p(Y|F) = \prod_t p(Y_t|F_t)$ and noting $p(Y_t|F_t) = \mathcal{N}(F_t; u_t, R)$, we have $|\Lambda_1| = |R|^{-T}$ and $\eta_1^T\Lambda_1^{-1}\eta_1 = \sum_t u_t^T R u_t$.

All the other log determinants and quadratic terms are expressed from the Cholesky factor C_j of Λ_j as $\log|\Lambda_j| = 2\log|C_j| = 2\sum_i \log[C_j]_{ii}$ and $\eta_j^T\Lambda_j^{-1}\eta_j = ||C_j^{-1}\eta_j||^2$.

So we can rewrite the likelihood as

$$\begin{aligned}\log p(Y) = &\frac{1}{2}\left(Td\log 2\pi - \log|\Sigma_0 R^T\Pi_t Q_t| + 2\log|C_2|\right)\\ &+\frac{1}{2}(||C_0^{-1}\eta_0||^2 + \sum_t u_t^T R u_t - ||C_2^{-1}\eta_2||^2)\end{aligned}$$

Algorithm 10 Gaussian process regression (using banded matrices)

Input : parameters $\theta = [A_{1\dots T}, b, Q_{1\dots T}, H, c, R]$,
observation Y: vector of length n
Output : $\log\, p(Y|\theta)$

% building the prior and likelihood natural parameters [see Grigorievskiy et al 2017]
$\eta_0, \Lambda_0 \leftarrow f(A_{1\dots T}, b, Q_{1\dots T})$
$\eta_1, \Lambda_1 \leftarrow g(H, c, R, Y)$
% Cholesky factorization of the prior and posterior precisions
$C_0 \leftarrow \mathbb{C}(\Lambda_0)$
$C_2 \leftarrow \mathbb{C}(\Lambda_0 + \Lambda_1)$
% evaluation of the log-likelihood
$\log p(Y) \leftarrow \frac{1}{2}\left(Td\log 2\pi - \log|\Sigma_0 R^T\Pi_t Q_t| - 2\log|C_2| + ||C_0^{-1}\eta_0||^2 + \sum_t u_t^T R u_t - ||C_2^{-1}\eta_2||^2\right)$
return L

The bottleneck in computational terms is the Cholesky factorization which scales as $O(Tl^3)$, because Λ_{post} is a banded precision of length Tl and bandwidth $2l$.

C.6 Gaussian process regression

In section 5.2, we compare different implementations of Gaussian process regression to the naive one that builds a dense covariance matrix K for the latent. For this latter case, the marginal likelihood is evaluated as:

$$\log\, p(Y|\theta) = -\frac{1}{2}Y^T(K+\sigma^2 I)^{-1}Y - \frac{1}{2}\log|K+\sigma^2 I| - \frac{N}{2}\log 2\pi$$

Algorithm 11 provides details on how we implemented it in practice.

Algorithm 11 Gaussian process regression Likelihood

Input : parameter θ
observation Y: vector of length n
Output : $\log\, p(Y|\theta)$

% Cholesky factorization of the posterior covariance
$L \leftarrow \text{Cholesky}(K + \sigma^2 I)$
% marginal likelihood evaluation
$\log\, p(Y|\theta) \leftarrow -\frac{1}{2}||L^{-1}Y||^2 - \sum_i \log|L_{ii}| - \frac{N}{2}\log 2\pi$
return $\log\, p(Y|\theta)$