Probabilistic Adjoint Sensitivity Analysis for Fast Calibration of Partial Differential Equation Models

Supplementary Material

November 21, 2020

S1 Algorithms

Algorithm S1 details the probabilistic gradient descent algorithm. Algorithm S2 describes the probabilistic line search procedure.

S2 Posterior Updating

In this section we describe the procedure used to update the posterior distribution based on new information. Suppose we have a prior $\mu = \mathcal{N}(a, C)$ and have constructed the posterior $\bar{\mu}^1$ based upon the information $\mathcal{I}^1 = f^1$, where $\mathcal{I}^1 : \mathcal{U}_{\partial P} \to \mathbb{R}^d$ and $f^1 \in \mathbb{R}^d$. For computational purposes this requires the inverse of the matrix $\mathcal{I}^1 C(\mathcal{I}^1)^\dagger$ or, equivalently, the solution of linear systems involving this matrix. Thus, we suppose that the Cholesky factorisation L_1 has been computed, so that $\mathcal{I}^1 C(\mathcal{I}^1)^\dagger = L_1 L_1^\top$, where the factor L_1 is upper-triangular.

Now suppose that new information is supplied, $\mathcal{I}^2 : \mathcal{U}_{\partial P} \to \mathbb{R}^{d'}$ with corresponding information $f^2 \in \mathbb{R}^{d'}$, and suppose that $\mathcal{I}^2$ is linearly independent of $\mathcal{I}^1$. We wish to compute the posterior distribution $\bar{\mu}^2$ by conditioning the distribution $\bar{\mu}^1$ on this new information with as low a computational cost as possible.

A natural approach would be to simply perform the conditioning procedure described in Proposition 1 with $\mu = \bar{\mu}^1$, however in practise this approach was found to suffer from a high degree of numerical instability. Instead, we advocate conditioning the original prior on this new information by updating the Cholesky factorisation computed for $\mathcal{I}^1$. The form of the new Gramian matrix whose factorisation must be computed is

$$\mathcal{I}^{1:2} C(\mathcal{I}^{1:2})^\dagger = \begin{bmatrix} \mathcal{I}^1 C(\mathcal{I}^1)^\dagger & \mathcal{I}^1 C(\mathcal{I}^2)^\dagger \\ \mathcal{I}^2 C(\mathcal{I}^1)^\dagger & \mathcal{I}^2 C(\mathcal{I}^2)^\dagger \end{bmatrix}.$$

Algorithm S1 Probabilistic version of gradient descent. The routines METRIC and INFO are problem specific and must be supplied by the user, with the former assessing the distribution of the currently computed posterior distribution to determine whether it is sufficiently narrow to accept it as a valid gradient and the latter supplying information, iteratively, based on the current distribution and location. The routine CONDITION implements Proposition 1. PLS is the probabilistic version of the Armijo line search, and is given in Algorithm S2. Of the new parameters, δ reflects how much accuracy is demanded of the posterior at each iteration, $\delta_{\min}$ specifies a maximum level of accuracy to protect against numerical instabilities resulting from large Gram matrices in CONDITION, and τ describes how rapidly δ is reduced when a valid descent direction cannot be found.

procedure PGD($p^0, g, \mu^0, \epsilon, \delta, \delta_{\min}, \tau_1$)
    Compute ν^0 from μ^0 and let X^0 be the random variable with law ν^0
    for $n = 1, 2, \ldots$ **do**
        $s^n, X^n, \gamma_n \leftarrow$ PROBJAC($X^{n-1}, g, p^{n-1}, \epsilon, \delta, \delta_{\min}$)
        if $\gamma^n < \epsilon$ **then**
            return p^{n-1}
        end if
        $p^n \leftarrow p^{n-1} + \gamma^n s^n$
    end for
end procedure
procedure PROBJAC($X, g, p, \epsilon, \delta, \delta_{\min}, \tau_1$)
    while $\delta > \delta_{\min}$ **do**
        while METRIC(X) $> \delta$ **do**
            $\mathcal{I}, f \leftarrow$ INFO(X, p)
            $X \leftarrow$ CONDITION($X, \mathcal{I}, f$)
            $s \leftarrow -\mathbb{E}(X(p))/\|\mathbb{E}(X(p))\|_2$
            $\gamma \leftarrow$ PLS(p, g, X)
            if $\gamma < \epsilon$ **then**
                $\delta \leftarrow \tau_1 \delta$
            else
                return s, X, γ
            end if
        end while
    end while
end procedure

Algorithm S2 Probabilistic line search algorithm. This is essentially a modification of the backtracking line search described in Nocedal and Wright [2006, Algorithm 3.1] to account for the fact that the gradient is a random variable rather than a constant. The parameters p, g and X are the parameter value, objective function and current posterior, respectively. The remaining parameters control the behaviour of the algorithm; we have specified sensible defaults for these and assume those defaults are used throughout the text. τ_2 controls how rapidly γ is decreased, while c controls how large a reduction in the objective function is required when a step is taken in the chosen direction and P^{crit} is the probability with which this reduction must be achieved. γ and γ^{min} control the initial and minimum values of γ respectively.

procedure PLS($p, g, X;\ \tau_2 = 0.5, c = 0.5, P^{\text{crit}}, \gamma = 1, \gamma^{\text{min}} = 10^{-10}$)
  $s \leftarrow -\mathbb{E}(X)/\|\mathbb{E}(X)\|_2$
  while $\gamma > \gamma^{\text{min}}$ **do**
    $p_\gamma \leftarrow p + \gamma s$
    if $g(p_\gamma) > g(p)$ **then**
      continue
    end if
    $Z \leftarrow -c\gamma X^\top s$
    if $\mathbb{P}(Z > g(p_\gamma) - g(p)) < P^{\text{crit}}$ **then**
      return γ
    end if
    $\gamma \leftarrow \tau_2 \gamma$
  end while
end procedure

Following [Osborne, 2010, Appendix B], we may form the Cholesky factorisation of this matrix as

$$\mathcal{I}^{1:2}C(\mathcal{I}^{1:2})^\dagger = L_{1:2}L_{1:2}^\top$$
$$L_{1:2} = \begin{bmatrix} L_1 & S_{12} \\ 0 & S_{22} \end{bmatrix}$$

where $S_{12} = (L_1^\top)^{-1}\mathcal{I}^1 C(\mathcal{I}^2)^\dagger$, and S_{22} is the Cholesky factorisation of a $\mathbb{R}^{d'\times d'}$ matrix given by

$$S_{22}S_{22}^\top = \mathcal{I}^2 C(\mathcal{I}^2)^\dagger - S_{12}S_{12}^\top.$$

Examining the cost of this procedure, we see that the only near-cubic operation required is this Cholesky factorisation, so that the update is approximately $\mathcal{O}((d')^3)$. This may then be used to compute the posterior from Proposition 1 without incurring the $\mathcal{O}((d+d')^3)$ cost that would be required to compute it *without* reusing the Cholesky factor. We note that a triangular solve of $\mathcal{O}((d+d')^2)$ is required to compute the full posterior distribution. Thus, as the size of the system grows, the cost will still increase at a quadratic rate. This was not found to be prohibitive for the experiments presented in this paper, and it is possible that with further optimisations this cost may also be reduced.

S3 Information Function for the Experiment

We now detail the implementation of the function INFO for the problem detailed in Section 5. Recall that the information functionals $\mathcal{I}_j$ correspond to finite element basis functions ϕ_j, which themselves correspond to nodal locations in the finite element mesh. Denote these locations $\{\boldsymbol{x}_j^{\text{info}}\}$. We choose new conditioning locations within this set by attempting to minimise a heuristic based on the fill distance which often appears as an upper bound in Gaussian process regression problems. To be specific, we begin by constructing an augmented point set:

$$z_{ij} = \begin{bmatrix} \boldsymbol{x}_i^{\text{info}} \\ p^j \end{bmatrix}.$$

for $i = 1, \ldots, 1089$, with $j = 1, \ldots, n$ denoting the iteration number in PROBJAC and p^j the corresponding parameter value for that iteration. The information functionals were then selected to be the $\tilde{\mathcal{I}}_j$ for which the distance between z_{in} and $z_{i'j}$, $i, i' = 1, \ldots, 1089$, $j = 1, \ldots, n-1$, is maximised. To ensure that the information f is nonzero, we enforce that when $\text{METRIC}(X^n) > \delta$, the first locations to be conditioned upon are those basis functions corresponding to x_i^{data}.

References

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer New York, 2006. doi: 10.1007/978-0-387-40065-5.

M. Osborne. *Bayesian Gaussian Processes for Sequential Prediction, Optimisation and Quadrature*. PhD thesis, PhD thesis, University of Oxford, 2010.