Bayesian Spatio-temporal Model Solution With Metho

Draft

Dan B MacKinlay[*]
CSIRO's Data61
Dan.MacKinlay@data61.csiro.au

Dan Pagendam
CSIRO's Data61

Russell Tsuchida
CSIRO's Data61

Petra Kuhnert
CSIRO's Data61

Sreekanth Janardhanan
CSIRO

November 25, 2022

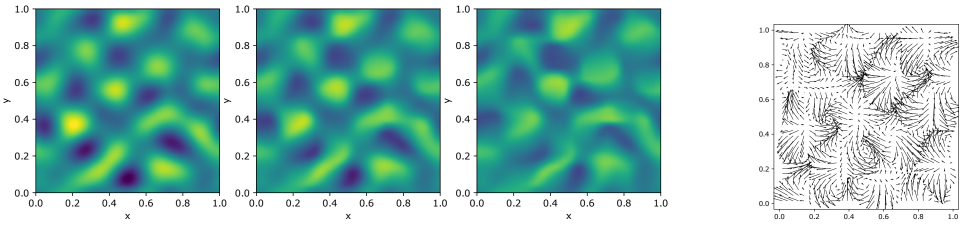
Abstract

We study the problem of stochastic model inversion for physical dynamical systems. This paper introduces MaTHerOn Ensemble inversion (Metho), an efficient means of using a forward prediction operator to sample from a posterior distribution over unobserved function-valued system parameters by means of an approximate Monte Carlo sampling method.

Our main contribution is the use of several recent innovations to expand the scale and power of Ensemble Kalman inversion techniques. Specifically, we combine ensemble representation of the prior and posterior, using Lanczos approximations of the posterior precision, and an incremental version of the Matheron update rules for jointly Gaussian random field to perform inference.

The resulting algorithm extends the many advantages of Ensemble Kalman methods to function valued estimators. Specifically, the cost of the resulting algorithm scales sub-cubically with the dimension of the estimands, does not require the calculation or storage of large covariance matrices, or require artificial evolution dynamics be applied to static latent parameters. We do not require exact likelihood evaluations of the prior or posterior samples, merely the ability to simulate from the prior and sufficient regularity of the forward operator. Our method achieves orders-of-magnitude speed-up over a classical physical model inversion approach, while maintaining acceptable accuracy and increases the resolution at which inference is possible.

Additionally, we derive an incremental version of the Matheron update for non-linearly observed Gaussian processes which is of independent interest.



(a) State z_0, z_1, z_2, (Density field)

(b) Unobserved forcing u

Figure 1: Example of the types of systems that we conduct inference with here; In this case it is a compressible Navier-Stokes fluid flow system over 3 time steps.

[*] Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

1 Introduction

This paper introduces a new method for inversion of models of dynamical systems whose state and parameters are function-valued. Problems of this kind arise in geophysics, hydrology, and medical imaging, among other areas. In fact any study where a hidden cause must be inferred through its influence on some dynamical behaviour amounts to solving this kind of inverse problem. Current solutions include the classic Kalman inversion or a local low-rank approximation such as a Proper Orthogonal Decomposition or possibly a domain-specific model.

We propose the MaTHeron Ensemble inversion approach (METHO), which efficiently utilises a forward operator to sample from a posterior distribution over unobserved parameters. METHO advances our capability to handle such models, by treating the observations as approximate Gaussian processes. It is general enough to be applicable to any sufficiently well-behaved dynamic model, scales to very large systems, and requires few assumptions. The prior covariances may be supplied by any model from which we can sample, and are not required to, for example, be stationary or homogeneous.

Is it possible to be any more specific about what constitutes "well-behavedness"? Surely, but I don't know how. Popular bounds are in terms of spectral radius of operators; I don't even know what that means for nonlinear operators

To illustrate the concept, consider a recursive function-valued time series of observations, z_t, $t = 0, 1, 2, \ldots, T$ that is defined in terms of a known stochastic forward predictor, $\mathscr{P}$ and an unknown forcing, u, such that

$$\mathsf{z}_t = \mathscr{P}(\mathsf{z}_{t\text{-}1}, \mathsf{u}), t = 1, 2, \cdots, T.$$

The inverse problem infers u, from the time series of observations $\mathsf{z}_0 = z_0, \mathsf{z}_1 = z_1, \cdots, \mathsf{z}_T = z_T$. Figure 1 illustrates such a system, a realisation of the compressible Navier-Stokes fluid flow equations on a square domain. Our method allows us to sample efficiently from the approximate posterior

$$\mathsf{u} \mid \mathsf{z}_0 = z_0, \mathsf{z}_1 = z_1, \cdots, \mathsf{z}_T = z_T.$$

2 Preliminaries

2.1 Notation

For clarity in what follows, we adopt the following notation conventions: a variable displayed like z_t represents the realisation of a function (e.g. a spatial field) indexed at time t; the variable displayed sans-serif z_t represents a random variable used to model (statistically) z_t; and $|X|$ is used to denote the cardinality of a set/dataset X.

2.2 Related work

PEST, GLUE, Foreman-Mackey

Generic Bayes inversion of models Stuart [2010], Dashti and Stuart [2015], Tarantola [2005].

Stochastic inversion via adversarial learning Xu and Darve [2019], Bao et al. [2020], Chu et al. [2021]. Gradient descent inversion Xu and Darve [2020], MacKinlay et al. [2021].

Operator inversion with generative score based on diffusion models, which in hindsight is probably faster and easier than this Song et al. [2022], Jalal et al. [2021].

Neural operator inversion with graph networks Li et al. [2020]. Bayes operators with Laplace ops Magnani et al. [2022]. Functional Bayes NN Watson et al. [2020].

3 Problem setting

We consider discrete-time function-valued processes of the following form: At each time step t the state of the system is characterised by a random state z_t taking values in a function space $\mathcal{Z}$ and each successive step is defined in terms of some *forward operator* $\mathscr{P} : \mathcal{Z} \times \mathcal{U} \to \mathcal{Z}$ which predicts successive states in terms of the previous state and some constant[2] forcing taking values in $\mathcal{U}$. Recursive application of the forward operator

$$\mathsf{z}_t = \mathscr{P}(\mathsf{z}_{t\text{-}1}, \mathsf{u}), \quad t = 1, 2, \cdots, T \tag{1}$$

given some initial state z_0, defines a time series of functions, $\mathsf{z}_{0:T} := (\mathsf{z}_0, \mathsf{z}_1, \ldots, \mathsf{z}_T)$. Viewed from the perspective of a field-valued observation this defines a relatively simple dependence structure (Figure 2). However, *within* a

[2] In this section we consider the case of a static forcing and static operator, but extension to time-varying u and/or time-varying, known, operator is straightforward

(function-valued) node there are potentially arbitrary dependencies between sites; and sites within these nodes may have a fully connected undirected dependence structure. We use the fact that we may filter over such a time series, repeatedly using the posterior estimate for $\mathsf{u} \mid \tilde{\mathsf{z}}_{t\text{-}1}{=}\tilde{z}_{t\text{-}1}, \tilde{\mathsf{z}}_t{=}\tilde{z}_t$ as the prior when calculating an update for $\mathsf{u} \mid \mathsf{z}_t{=}\tilde{z}_t, \tilde{\mathsf{z}}_{t+1}{=}\tilde{z}_{t+1}$.

In the spatio-temporal context, we assume are interested in $\mathcal{Z}, \mathcal{U}$ as spatial fields, i.e. their elements map some spatial index to some observation value. $\mathcal{Z} \subset \{z : \mathcal{S} \to \mathbb{R}^{d_z}\}, \mathcal{U} \subset \{u : \mathcal{S} \to \mathbb{R}^{d_u}\}$. The (compact) domain $\mathcal{S} \subset \mathbb{R}^{d_s}$ has interpretation as the spatial domain of the system, and (typically $d_s \in \{1, 2, 3\}$). In this paper it is always the unit square. d_u is typically small; here we focus on problems where $d_u = 2$, but our methods can be used for $d_u \in \mathbb{N}$. Such systems arise naturally in, for example, the forward solutions to partial differential equations observed at discrete times. More general Hilbert spaces without a spatial interpretation may also be amenable to our methods.

We initially take the operator to be a deterministic function of its arguments. So that we may solve such systems computationally, we operate not directly in the continuously-indexed function spaces, but on discrete representations. Each of $\mathcal{Z}, \mathcal{U}$ is a Hilbert space with a finite, orthogonal basis and associated inner product $\langle f, g\rangle = \int_{s \in \mathcal{S}} f(s)g(s)\mathrm{d}\lambda(\boldsymbol{s})$ for λ the Lebesgue measure on $\mathcal{S}$. Fixing the basis for $\mathcal{Z}$ as $\mathcal{X} = \begin{bmatrix}\phi_1 & \cdots & \phi_{|\mathcal{X}|}\end{bmatrix}^\top$ wherein each $\phi_i \in \mathcal{Z}$, a given function z_t may be represented by a projection $\mathrm{P}_\mathcal{X} : \mathcal{Z} \to \mathbb{R}^{|\mathcal{X}|}$, as $\mathcal{X} = \begin{bmatrix}\phi_1 & \cdots & \phi_{|\mathcal{X}|}\end{bmatrix}^\top$ Explicitly,

countable but truncated?

$$\mathrm{P}_\mathcal{X}\{f\} = \begin{bmatrix}\langle f, \phi_1\rangle & \cdots & \langle f, \phi_{|\mathcal{X}|}\rangle\end{bmatrix}^\top.$$

We fix a (possibly different) orthogonal basis for $\mathcal{U}$. Projected vector we denote using tildes, $\tilde{f} := \mathrm{P}_\mathcal{X}\{f\}$, and we make the projection explicit only when it cannot be inferred from context. The inverse of the projection $\mathrm{P}_\mathcal{X}^{-1} : \mathbb{R}^{|\mathcal{X}|} \to \mathcal{Z}$ is

$$\mathrm{P}_\mathcal{X}^{-1}\left\{\begin{bmatrix}\tilde{f}_1 & \cdots & \tilde{f}_{|\mathcal{X}|}\end{bmatrix}^\top\right\} = \sum_{i=1}^{|\mathcal{X}|} \tilde{f}_i \phi_i.$$

Similarly, we associate with $\mathcal{U}$ a (possibly different) basis $\mathcal{Y}$ and associated projection operator $\mathrm{P}_\mathcal{Y}$ and inverse. By some abuse of notation we use the same representation for the forward operator $\mathscr{P}$ and its discretized version, writing $\mathscr{P}(\tilde{z}, \tilde{u})$ to mean $\mathrm{P}_\mathcal{X}\mathscr{P}(\mathrm{P}_\mathcal{X}^{-1}\tilde{z}, \mathrm{P}_\mathcal{Y}^{-1}\tilde{u})$ where now $\mathscr{P} : \mathbb{R}^{|\mathcal{X}|} \times \mathbb{R}^{|\mathcal{Y}|} \to \mathbb{R}^{|\mathcal{X}|}$.

We note in passing that due to this representation, there is nothing special about the spatial quality of these fields. Therefore, it is a trivial extension of our approach to infer scalar-valued parameters or a simultaneously infer spatial, function-valued and scalar-valued parameters. A scalar parameter is just a trivial field with a trivial basis.

Throughout this work, our examples use a basis of piecewise constant, disjoint top-hat functions partitioning the space, centred on some regular lattice of points $\mathcal{X}$. This is the same basis used in Finite Element methods. Other bases, such as piecewise linear, Fourier, or polynomial bases, are substantively similar. Selecting a basis to guarantee convergence to some continuous true solution can be complicated in general Lassas et al. [2009] but we assume that solutions in the discrete projected space are close to those in a notionally continuous space if the discretization is "fine enough" that the basis spans enough of the solution space. This requires that our methods must be feasible for large basis sets, on the order of thousands or millions of basis functions.

We take as our primary example the statistical nverse problem of inferring static forcing, u of a PDE given observation data $\mathcal{D} := [\tilde{z}_0, \tilde{z}_1, \cdots, \tilde{z}_T]$ such that $\tilde{\mathsf{z}}_0 = \tilde{z}_0, \tilde{\mathsf{z}}_1 = \tilde{z}_1, \cdots, \tilde{\mathsf{z}}_T = \tilde{z}_T$. Extension to a forward operator taking an additional unobserved random argument is considered in the sequel.

3.1 Inversion by prediction error minimisation

One straightforward approach is to choose by optimisation a "best" solution in terms of giving a prediction with the smallest discrepancy from observations, treating all terms as deterministic.

$$\hat{u} := \operatorname{argmin}_u L\left(\tilde{z}_t, \mathscr{P}(\tilde{z}_{t\text{-}1}, \tilde{u})\right). \tag{2}$$

In the case that gradients of the model output, $\mathscr{P}(\tilde{z}_{t\text{-}1}, \tilde{u})$, with respect to inputs are available, (2) may be efficiently solved even for high-dimensional parameters by gradient descent.

The optimisation problem described in (2) is often under-determined, in the sense there may be many solutions that have the same loss given the observations. In ill-conditioned and unspecified settings this is potentially problematic; For one it may not convey the uncertainty in the solutions identified, and for another, it may not convey the uncertainty in solution.

mention that we need a regularising prior even with true model

This approach can be generalised to a stochastic forward operator, by taking expectations

$$\hat{u} := \operatorname{argmin}_u \mathbb{E}[L\left(\tilde{z}_t, \mathscr{P}(\tilde{z}_{t\text{-}1}, \tilde{u})\right)]. \tag{3}$$

diagram of excessively wobbly posterior

Solution in that case may be estimated by a Monte Carlo method.

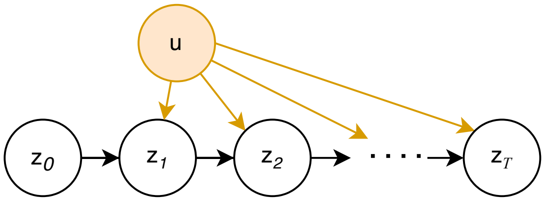


Figure 2: DAG for this model

3.2 Bayesian inversion

A Bayesian probabilistic model represents the quantities of interest with random variables by assigning some probability measure over the state space. An appropriate choice of prior measure regularises our inference towards some plausible region of solution space. Probabilistic methods provide distributions over the set of feasible solutions whilst simultaneously regularizing these solutions via a prior measure on the solution function space u.

For spatial random fields, this means dealing in random spatial functions, i.e. representing states and forcings with measures over sets $\mathcal{Z}$ and $\mathcal{U}$ respectively. Rewriting (1) as a structural equation model,

this is getting convoluted. just mention the probability space/sigma algebra?

$$\mathsf{z}_t = \mathscr{P}(\mathsf{z}_{t\text{-}1}, \mathsf{u}), \quad t = 1, \cdots, T. \tag{4}$$

we note that the corresponding directed graphical model structure is given by Figure 2. The pushforward of $\mathscr{P}$ can be arbitrarily complicated, and thus the joint distribution induced by (4) will be typically intractable. A pragmatic choice for probabilistic modelling of the random functions z and u is to approximate the distribution of the latent forcing and the initial value with Gaussian processes, and to approximate all subsequent steps also by a Gaussian process

$$\begin{aligned}
\mathsf{z}_0 &\sim \mathcal{GP}(m_{\mathsf{z}_0}, k_{\mathsf{z}_0}) && (5)\\
\mathsf{u} &\sim \mathcal{GP}(m_{\mathsf{u}}, k_{\mathsf{u}}) && (6)\\
\mathsf{z}_t &\sim \mathcal{GP}(\mathscr{M}_{\mathsf{z}}\{\mathsf{z}_{t\text{-}1}, \mathsf{u}\}, \mathscr{K}_{\mathsf{z}}\{\mathsf{z}_{t\text{-}1}, \mathsf{u}\}), \quad t = 1, 2, \cdots, T && (7)\\
&\overset{d}{\approx} \mathscr{P}(\mathsf{z}_{t\text{-}1}, \mathsf{u}).
\end{aligned}$$

This presumes that there exists some mean function $\mathscr{M}_{\mathsf{z}}$ and covariance kernel $\mathscr{K}_{\mathsf{z}}$ such that the forward predictive density is well approximated by the pushfoward. (4).

Various choices for calculating $\mathscr{M}_{\mathsf{z}}\{\mathsf{z}_{t\text{-}1}, \mathsf{u}\}, \mathscr{K}_{\mathsf{z}}\{\mathsf{z}_{t\text{-}1}, \mathsf{u}\}$ may be motivated by different senses in which we *approximate* the target distributions, leading to different algorithms for inference, with different tradeoffs of accuracy and computational cost. It is the strategic selection of these choices which is at the core of this paper.

We repeatedly use the following construction. (Conditionally-)jointly Gaussian variates,

$$\left[\begin{array}{c} \tilde{\mathsf{z}}_t \\ \tilde{\mathsf{u}} \end{array}\right] \Bigg| (\tilde{\mathsf{z}}_{t\text{-}1} {=} \tilde{z}_{t\text{-}1}) \sim \mathcal{N}\left(\left[\begin{array}{c} m_{\tilde{\mathsf{z}}_t} \\ m_{\tilde{\mathsf{u}}} \end{array}\right], \left[\begin{array}{cc} \mathrm{K}_{\tilde{\mathsf{z}}_t\tilde{\mathsf{z}}_t} & \mathrm{K}_{\tilde{\mathsf{z}}_t\tilde{\mathsf{u}}}^{\top} \\ \mathrm{K}_{\tilde{\mathsf{z}}_t\tilde{\mathsf{u}}} & \mathrm{K}_{\tilde{\mathsf{u}}\tilde{\mathsf{u}}} \end{array}\right] \right). \tag{8}$$

have a well-known, data-conditional update, namely,

$$\begin{aligned}
\tilde{\mathsf{u}} | (\tilde{\mathsf{z}}_{t\text{-}1} {=} \tilde{z}_{t\text{-}1}, \tilde{\mathsf{z}}_t {=} \tilde{z}_t) &\sim \mathcal{N}\left(\hat{m}_{\tilde{\mathsf{u}}}, \hat{\mathrm{K}}_{\tilde{\mathsf{u}}\tilde{\mathsf{u}}}\right), \text{ where} && (9)\\
\hat{m}_{\tilde{\mathsf{u}}} &= m_{\tilde{\mathsf{u}}} + \mathrm{K}_{\tilde{\mathsf{z}}_t\tilde{\mathsf{u}}} \mathrm{K}_{\tilde{\mathsf{z}}_t\tilde{\mathsf{z}}}^{-1} (m_{\tilde{\mathsf{z}}_t} - \tilde{\mathsf{z}}_t) && (10)\\
\hat{\mathrm{K}}_{\tilde{\mathsf{u}}\tilde{\mathsf{u}}} &= \mathrm{K}_{\tilde{\mathsf{u}}\tilde{\mathsf{u}}} - \mathrm{K}_{\tilde{\mathsf{z}}_t\tilde{\mathsf{u}}} \mathrm{K}_{\tilde{\mathsf{z}}_t\tilde{\mathsf{z}}}^{-1} \mathrm{K}_{\tilde{\mathsf{z}}_t\tilde{\mathsf{u}}}^{\top}. && (11)
\end{aligned}$$

Bayesian inversion with linearised Gaussian processes

A classic choice is to define the $\mathrm{K}_{\tilde{\mathsf{u}}\tilde{\mathsf{u}}}$ in terms of a linearized approximation to the forward operator of the system, $\mathscr{P}$. This is the basic technique of the propgation of errors, the Extended Kalman Filter, and more generally, Gaussian Belief propagation, e.g. Ortiz et al. [2021]. Specifically, for some expansion point $\tilde{z}_0, \tilde{u}_0$, we choose

$$\begin{aligned}
\mathscr{P}_{\tilde{z}_0,\tilde{u}_0}(\tilde{z}, \tilde{u}) &\approx \widehat{\mathscr{P}}_{\tilde{z}_0,\tilde{u}_0}(\tilde{z}, \tilde{u})\\
&:= \mathscr{P}(\tilde{z}_0, \tilde{u}_0) + \mathrm{J}_{\tilde{u}_0}(\tilde{u} - \tilde{u}_0) \text{ where} && (12)\\
\mathrm{J}_{\tilde{u}_0} &:= \nabla_{\tilde{\mathsf{u}}} \mathscr{P}(\tilde{z}, \tilde{u})\big|_{\tilde{z}=\tilde{z}_0, \tilde{u}=\tilde{u}_0}.
\end{aligned}$$

We use the linearised form (12) to approximate (7), expanding about the prior mean $m_{\tilde u} = \mathbb{E}\tilde{\mathsf u}$,

$$\begin{bmatrix}\tilde{\mathsf z}_t\\ \tilde{\mathsf u}\end{bmatrix}\Bigg|\,(\tilde{\mathsf z}_{t\text{-}1}{=}\tilde z_{t\text{-}1}) \overset{\mathrm d}{\approx} \begin{bmatrix}\widehat{\mathscr P}_{\tilde z, m_{\tilde u}}(\tilde z_{t\text{-}1},\tilde{\mathsf u})+\tilde\varepsilon\\ \tilde{\mathsf u}\end{bmatrix} \tag{13}$$

where $\tilde\varepsilon \sim \mathcal N(0,\sigma^2\mathrm I)$ is a noise term that admits a degree of model misfit. Under this approximation, the moments in the form (9) are

$$m_{\tilde{\mathsf z}_t} = \mathscr P(\tilde m_{z_{t-1}}, \tilde m_u) \tag{14}$$

$$\mathrm K_{\tilde{\mathsf z}_t\tilde{\mathsf z}_t} = J_{m_u}\mathrm K_{\tilde{\mathsf u}\tilde{\mathsf u}}J_{m_u}^\top + \sigma^2\mathrm I \tag{15}$$

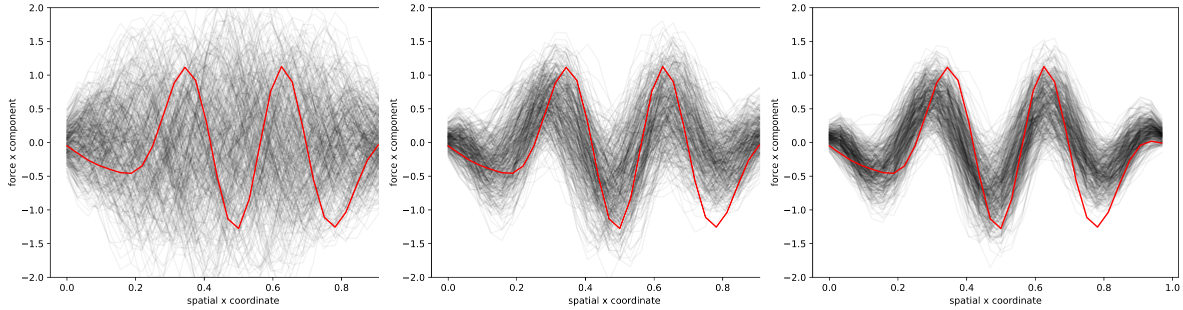
$$\mathrm K_{\tilde{\mathsf z}_t\tilde{\mathsf u}} = \mathrm K_{\tilde{\mathsf u}\tilde{\mathsf u}}J_{m_u}^\top \tag{16}$$

since $\widehat{\mathscr P}$ is linear.

This provides us a recipe for iteratively updating beliefs about $\mathsf u$ given new observation pairs $\tilde{\mathsf z}_{t\text{-}1}{=}\tilde z_{t\text{-}1}, \tilde{\mathsf z}_t{=}\tilde z_t$ by iteratively applying (9) to the linearized approximation (13).

This approach is unsatisfactory for various reasons. For one, by linearising the forward operator about the prior mean we have restricted the influence of non-linear dynamics on the system. For another, the storage requirements of the posterior are large, comprising a mean function and also a covariance matrix between all sites, requiring storage of an $|\mathcal X| \times |\mathcal X|$ covariance matrix. Further, computing this update can be prohibitively expensive, as the updates in (11) and (10) require solving a linear system involving $\mathrm K_{\tilde{\mathsf z}_t\tilde{\mathsf z}}$, which incurs a $\mathcal O(|\mathcal X|^3)$ computational time cost in the absence of any exploitable structure. Further, the capacity of this model to handle non-deterministic predictions is limited, since only linearly additive noise may be easily applied to the prediction model. In the next section we attempt to address these problems simultaneously using a sampling method.

3.3 METHO: Ensemble inversion



(a) Prior samples of $\mathsf u$ (b) METHO samples $\mathsf u \mid \tilde{\mathsf z}_0, \tilde{\mathsf z}_1$. (c) METHO samples $\mathsf u \mid \tilde{\mathsf z}_0, \cdots \tilde{\mathsf z}_5$.

Figure 3: A slice along the x spatial axis of some sampled solution from the METHO update. $N = 400$, $k = 100$. The solid red line shows ground-truth, and each transparent grey line is one sample from the solution space.

Two difficulties of the linearised Gaussian model are the widespread use of linear approximations and large, uninterpretable posterior distributions. . These issues can be simultaneously eased by moving to Monte Carlo inference, wherein distributions are summarised by samples drawn from those distributions. Specifically, our method, METHO, is in the family of Ensemble Kalman methods of Data Assimilation Evensen [2009] and which have been generalised to solve inverse problems Iglesias et al. [2013]. In these ensembles, prior beliefs about parameters are represented by samples from the prior distribution, and the ensemble of samples is successively modified to update our posterior beliefs based on observations. Ultimately, this approach leverages the summary statistics of the ensemble to model uncertainties surrounding the estimates. Such methods can frequently be more computationally efficient and flexible than the error-propagation method or section 3.2. Further, ensemble methods frequently attain superior performance in non-linear systems **?**.

what do you mean by large and uninterpretable?

In this setting, we construct an empirical approximation to the prior distribution for $\mathsf u$ by simulating an i.i.d. ensemble of N realisations from it,

name the prior

$$\mathrm U = \begin{bmatrix}\tilde u^{(1)} & \cdots & u^{(N)}\end{bmatrix}.$$

We associate with it a predictive ensemble

it looks like you've also included the projection below, but it's not

$$\begin{aligned}\mathrm{Z} &= \begin{bmatrix} \mathscr{P}(z_{t\text{-}1}, u^{(1)}) & \cdots & \mathscr{P}(z_{t\text{-}1}, u^{(N)}) \end{bmatrix} \\ &= \begin{bmatrix} \tilde{z}^{(1)} & \cdots & \tilde{z}^{(N)} \end{bmatrix}\end{aligned}$$

By treating the joint distribution of these ensembles as Gaussian, we motivate an analogue of the Gaussian posterior update. We define the ensemble means,

$$\bar{m}_z := \frac{1}{N}\sum_{i=1}^{N} z^{\widetilde{(i)}}$$

$$\bar{m}_u := \frac{1}{N}\sum_{i=1}^{N} u^{\widetilde{(i)}}$$

and ensemble deviations

$$\begin{aligned}\breve{\mathrm{Z}} &= \frac{1}{\sqrt{N-1}}\begin{bmatrix} (\tilde{z}^{(1)} - \bar{m}_z) & \cdots & (\tilde{z}^{(N)} - \bar{m}_z) \end{bmatrix} \\ \breve{\mathrm{U}} &= \frac{1}{\sqrt{N-1}}\begin{bmatrix} (\tilde{u}^{(1)} - \bar{m}_z) & \cdots & (\tilde{u}^{(N)} - \bar{m}_u) \end{bmatrix}.\end{aligned}$$

If we assume they are jointly conditionally Gaussian (8), the empirical approximating distribution is defined

$$\left[\begin{array}{c} \tilde{\mathsf{z}}_t \\ \tilde{\mathsf{u}} \end{array}\right] \Bigg| \tilde{\mathsf{z}}_{t\text{-}1} = \tilde{z}_{t\text{-}1} \overset{\mathrm{d}}{\approx} \mathcal{N}\left(\left[\begin{array}{c} \bar{z} \\ \bar{u} \end{array}\right], \left[\begin{array}{cc} \breve{\mathrm{Z}}\breve{\mathrm{Z}}^\top + \sigma^2\mathrm{I} & \breve{\mathrm{U}}\breve{\mathrm{Z}}^\top \\ \breve{\mathrm{Z}}\breve{\mathrm{U}}^\top & \breve{\mathrm{U}}\breve{\mathrm{U}}^\top \end{array}\right]\right). \tag{17}$$

Similarly to the Gaussian update (9), we update the samples in our ensemble to the posterior distribution using the pathwise Matheron update [Wilson et al., 2021].

The Matheron update maps marginal Gaussian samples to conditional Gaussian samples. For details see Appendix A. In our case, the transformation iteratively approximates the following

$$(\tilde{\mathsf{u}} \mid \tilde{\mathsf{z}}_t{=}\tilde{z}_t, \tilde{\mathsf{z}}_{t\text{-}1}{=}\tilde{z}_{t\text{-}1}) \overset{\mathrm{d}}{\approx} \tilde{\mathsf{u}} + \mathrm{K}_{\tilde{z}\tilde{\mathsf{u}}}^\top \mathrm{K}_{\tilde{z},\tilde{z}}^{-1}\left(\tilde{z}_t - \mathscr{P}(\tilde{z}_t, \tilde{u})\right). \tag{18}$$

This needs to be written better. Also what tis this operator actually doing?

In the case that $\mathscr{P}$ is linear, this is exact. and such an update maps individual prior samples of u to individual posterior samples $\mathsf{u} \mid (\tilde{\mathsf{z}}_{t\text{-}1} = \tilde{z}_{t\text{-}1}, \tilde{\mathsf{z}}_t = \tilde{z}_t)$.

Our formulation generalises the classic Matheron update to incrementally construct locally linear updates to account for the nonlinear observation process induced by the forward prediction operator.. We set up a least-squares optimisation problem whose solution is equivalent to the classical Matheron update in the linear case, and iteratively solve that by a second-order method.

blather, and not correct

Plugging in sample covariances of (17) to the Matheron update (18), we need to solve an inner problem

$$\begin{aligned}&(\tilde{\mathsf{u}}^{(i)} \mid \tilde{\mathsf{z}}_t{=}\tilde{z}_t, \tilde{\mathsf{z}}_{t\text{-}1}) \\ &\overset{\mathrm{d}}{=} \tilde{\mathsf{u}}^{(i)} + \mathrm{K}_{\tilde{\mathsf{z}}_t\tilde{\mathsf{u}}}\mathrm{K}^{-1}_{\tilde{\mathsf{z}}_t^{(i)}\tilde{\mathsf{z}}_t^{(i)}}\left(\tilde{z}^{(i)} - \tilde{z}_t\right) \\ &= \tilde{\mathsf{u}}^{(i)} + \breve{\mathrm{U}}\breve{\mathrm{Z}}^\top(\breve{\mathrm{Z}}\breve{\mathrm{Z}}^\top + \sigma^2\mathrm{I})^{-1}\left(\tilde{z}^{(i)} - \tilde{z}_t\right) \\ &= \tilde{\mathsf{u}}^{(i)} + \underbrace{\breve{\mathrm{U}}}_{|X|\times N}\underbrace{\breve{\mathrm{Z}}^\top}_{N\times|S|}(\underbrace{\breve{\mathrm{Z}}}_{|S|\times N}\underbrace{\breve{\mathrm{Z}}^\top}_{N\times|S|} + \sigma^2\mathrm{I})^{-1}\left(\tilde{z}^{(i)} - \tilde{z}_t\right).\end{aligned} \tag{19}$$

At first glance, this method still requires an intractably expensive $\mathcal{O}(|\mathcal{X}|^3)$ cost when inverting the covariance $(\breve{\mathrm{Z}}\breve{\mathrm{Z}}^\top + \sigma^2\mathrm{I})$ in (19). However, the empirical representation has structure we can exploit to approximate its solution efficiently via Lanczos decomposition. That trick is known in the Gaussian process literature in the context of Lanczos Variance Estimates (LOVE) [Pleiss et al., 2018],although we exploit it in a different manner.

Given some rank k and an arbitrary starting vector $\boldsymbol{b}$, the Lanczos algorithm iteratively approximates $\mathrm{A} \in \mathbb{R}^{n\times n}$ by a low rank factorization $\mathrm{A} \approx \mathrm{QTQ}^\top$, where $\mathrm{T} \in \mathbb{R}^{k\times k}$ is tridiagonal and $\mathrm{Q} \in \mathbb{R}^{n\times k}$ has orthogonal columns. Crucially, we do not need to form A to evaluate matrix vector products $\mathrm{A}\boldsymbol{b}$ for arbitrary vector $\boldsymbol{b}$. Moreover, with a given Lanczos approximand Q, T we may estimate

$$\mathrm{A}^{-1}\boldsymbol{c} \approx \mathrm{QT}^{-1}\mathrm{Q}^\top\boldsymbol{c}. \tag{20}$$

accuracy of this update? Depends on condition number even for $\boldsymbol{b} \neq \boldsymbol{c}$.

In (19) we must calculate $\left(\breve{\mathrm{Z}}\breve{\mathrm{Z}} + \sigma^2\mathrm{I}\right)^{-1}\left(\tilde{z}^{(i)} - \tilde{z}_t\right)$. We approximate the solution to this linear system using the partial Lanczos decomposition starting with probe vector $\boldsymbol{b} = \tilde{m}_z$ and $\mathrm{A} = \left(\breve{\mathrm{U}}\breve{\mathrm{U}} + \sigma^2\mathrm{I}\right)$. This requires k matrix vector products of the form

citation chase for bounds on this approx quality.

$$\underbrace{\left(\underbrace{\breve{\mathrm{U}}\breve{\mathrm{U}}^\top}_{\mathcal{O}(N|\mathcal{X}|^2)} + \sigma^2\mathrm{I}\right)}_{\mathcal{O}(|\mathcal{X}|^2)}\boldsymbol{b} = \underbrace{\breve{\mathrm{U}}\ \underbrace{(\breve{\mathrm{U}}^\top\boldsymbol{b})}_{\mathcal{O}(N|\mathcal{X}|)}}_{\mathcal{O}(N|\mathcal{X}|)} + \sigma^2\boldsymbol{b}.$$

Using the latter representation, the required matrix-vector product may be found with a time complexity cost of $\mathcal{O}(N|\mathcal{X}|)$. Space complexity is also $\mathcal{O}(N|\mathcal{X}|)$. The output of the Lanczos decomposition is Q, T such that $\left(\tilde{\mathrm{K}} + \sigma^2\mathrm{I}\right)\boldsymbol{b} \approx \mathrm{Q}\mathrm{T}\mathrm{Q}^\top\boldsymbol{b}$, i.e. a low rank approximation of the covariance-matrix-vector product. Then, by (20), the solution to the inverse-covariance-matrix-vector product may be approximated by $\left(\tilde{\mathrm{K}} + \sigma^2\mathrm{I}\right)^{-1}\left(\tilde{z}^{(i)} - \tilde{z}_t\right) \approx \mathrm{Q}\mathrm{T}^{-1}\mathrm{Q}^\top\left(\tilde{z}^{(i)} - \tilde{z}_t\right)$, requiring the solution in x of the much smaller linear system $\mathrm{X}\mathrm{T} = \mathrm{Q}$. Exploiting the positive-definiteness of T we may use the Cholesky decomposition of $\mathrm{T} = \mathrm{L}^\top\mathrm{L}$ for a constant speedup over solving an arbitrary linear system. The time cost of the solution is $\mathcal{O}(|\mathcal{X}|k^3)$, for an overall cost to the matrix inversions of $\mathcal{O}(N|\mathcal{X}|k + |\mathcal{X}|k^3)$.

make more precise; i think we have missed some multiplies and Lanczos overhead, plus deviation calcs; also if we use iterative matheron, how many steps do we take in general?

overall cost

revisit stochastic operators

Algorithm 1: METHO algorithm

```
procedure METHO $(a, b)$                    ▷ The g.c.d. of a and b
    $r \leftarrow a \bmod b$
    while $r \neq 0$ do                     ▷ We have the answer if r is 0
        $a \leftarrow b$
        $b \leftarrow r$
        $r \leftarrow a \bmod b$
    return $b$                              ▷ The gcd is b
```

4 Hyperparameter selection

Optimize evidence by, er… No idea apart from brute-force search. I have an idea that would maybe help in the neural network case.

Update: in fact, both σ and τ can be selected by gradient descent, in principle, to maximise the marginal likelihood. I think in the Lanczos case this is tractable via a Woodbury identity.

5 Error analysis

Hah. No idea about analytic results. Empirical comparison?

6 Experiments

We use a Navier-Stokes flow model on a 2-dimensional square domain with Neuman boundary conditions, inhomogenous forcing. Darcy flow. etc. Tradeoff of time versus accuracy.

6.1 k

6.2 stochastic forward operator

6.3 τ

6.4 $|\mathcal{X}|$

relative accuracy

References

A. M. Stuart. Inverse problems: A Bayesian perspective. *Acta Numerica*, 19:451–559, 2010. doi:10.1017/S0962492910000061.

Masoumeh Dashti and Andrew M. Stuart. The Bayesian Approach To Inverse Problems. *arXiv:1302.6989 [math]*, July 2015.

Albert Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, January 2005. ISBN 978-0-89871-572-9.

Kailai Xu and Eric Darve. Adversarial Numerical Analysis for Inverse Problems, October 2019.

Gang Bao, Xiaojing Ye, Yaohua Zang, and Haomin Zhou. Numerical solution of inverse problems by weak adversarial networks. *Inverse Problems*, 36(11):115003, November 2020. doi:10.1088/1361-6420/abb447.

Mengyu Chu, Nils Thuerey, Hans-Peter Seidel, Christian Theobalt, and Rhaleb Zayer. Learning meaningful controls for fluids. *ACM Transactions on Graphics*, 40(4):1–13, August 2021. doi:10.1145/3476576.3476661.

Kailai Xu and Eric Darve. ADCME: Learning Spatially-varying Physical Fields using Deep Neural Networks. In *arXiv:2011.11955 [Cs, Math]*, November 2020.

Dan MacKinlay, Dan Pagendam, Petra M Kuhnert, Tao Cui, David Robertson, and Sreekanth Janardhanan. Model Inversion for Spatio-temporal Processes using the Fourier Neural Operator. In *Neurips Workshop on Machine Learning for the Physical Sciences*, page 7, 2021.

Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving Inverse Problems in Medical Imaging with Score-Based Generative Models. In *ICLR*. arXiv, June 2022.

Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, Alexandros G Dimakis, and Jon Tamir. Robust Compressed Sensing MRI with Deep Generative Priors. In *Advances in Neural Information Processing Systems*, volume 34, pages 14938–14954. Curran Associates, Inc., 2021.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural Operator: Graph Kernel Network for Partial Differential Equations. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*. arXiv, June 2020. doi:10.48550/arXiv.2003.03485.

Emilia Magnani, Nicholas Krämer, Runa Eschenhagen, Lorenzo Rosasco, and Philipp Hennig. Approximate Bayesian Neural Operators: Uncertainty Quantification for Parametric PDEs, August 2022.

Joe Watson, Jihao Andreas Lin, Pascal Klink, and Jan Peters. Neural Linear Models with Functional Gaussian Process Priors. page 10, 2020.

Matti Lassas, Eero Saksman, and Samuli Siltanen. Discretization-invariant Bayesian inversion and Besov space priors. *Inverse problems and imaging*, 3(1):87–122, 2009. doi:10.3934/ipi.2009.3.87.

Joseph Ortiz, Talfan Evans, and Andrew J. Davison. A visual introduction to Gaussian Belief Propagation. *arXiv:2107.02308 [cs]*, July 2021.

Geir Evensen. *Data Assimilation - The Ensemble Kalman Filter*. Springer, Berlin; Heidelberg, 2009. ISBN 978-3-642-03711-5 3-642-03711-9.

Marco A. Iglesias, Kody J. H. Law, and Andrew M. Stuart. Ensemble Kalman methods for inverse problems. *Inverse Problems*, 29(4):045001, March 2013. doi:10.1088/0266-5611/29/4/045001.

James T Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Pathwise Conditioning of Gaussian Processes. *Journal of Machine Learning Research*, 22(105):1–47, 2021.

Geoff Pleiss, Jacob R. Gardner, Kilian Q. Weinberger, and Andrew Gordon Wilson. Constant-Time Predictive Distributions for Gaussian Processes. In *ICML*. arXiv, June 2018. doi:10.48550/arXiv.1803.06058.

Yousef Saad. *Iterative Methods for Sparse Linear Systems: Second Edition*. SIAM, second edition, January 2003. ISBN 978-0-89871-800-3.

Michael Roth, Gustaf Hendeby, Carsten Fritsche, and Fredrik Gustafsson. The Ensemble Kalman filter: A signal processing perspective. *EURASIP Journal on Advances in Signal Processing*, 2017(1):56, August 2017. doi:10.1186/s13634-017-0492-x.

A Matheron updates

We recall the pathwise Gaussian process update [Wilson et al., 2021]: Suppose

$$\begin{bmatrix}\mathsf{y}\\ \mathsf{w}\end{bmatrix} \sim \mathcal{GP}\left(\begin{bmatrix}\boldsymbol{m}_{\mathsf{y}}\\ \boldsymbol{m}_{\mathsf{w}}\end{bmatrix}, \begin{bmatrix}\mathrm{K}_{\mathsf{y},\mathsf{y}} & \mathrm{K}_{\mathsf{y},\mathsf{w}}\\ \mathrm{K}_{\mathsf{w},\mathsf{y}} & \mathrm{K}_{\mathsf{w},\mathsf{w}}\end{bmatrix}\right)$$

form a jointly Gaussian process. Given some (function-valued) sample $y \sim \mathsf{y}$ of a Gaussian process and projected observation $\tilde{\mathsf{w}} = \tilde{w}$, the Matheron update to that sample is a new realisation of $\mathsf{y} \mid \tilde{\mathsf{w}}{=}\tilde{w}$ is given by

$$(\mathsf{y}(s) \mid \tilde{\mathsf{w}}{=}\tilde{w}) \overset{\mathrm{d}}{=} \mathsf{y}(s) + \boldsymbol{k}_{\tilde{w}\mathsf{y}}(s)^{\top}\mathrm{K}^{-1}_{\tilde{w},\tilde{w}}\left[\tilde{w} - m_{\tilde{\mathsf{w}}}\right]. \tag{21}$$

The Matheron update analog in the discretely-projected case, i.e. for $\tilde{\mathsf{y}}$, is

$$(\tilde{\mathsf{y}} \mid \tilde{\mathsf{w}}{=}\tilde{w}) \overset{\mathrm{d}}{=} \tilde{\mathsf{y}} + \mathrm{K}_{\tilde{\mathsf{w}}\tilde{\mathsf{y}}}\mathrm{K}^{-1}_{\tilde{\mathsf{w}}\tilde{\mathsf{w}}}\left[\tilde{w} - m_{\tilde{\mathsf{w}}}\right]. \tag{22}$$

Here, $\overset{\mathrm{d}}{=}$ denotes that both sides of the equation are identically distributed. That is to say, if we draw a sample $\tilde{y} \sim \mathrm{Law}(\tilde{\mathsf{y}})$, then this recipe gives us a means of transforming into a sample from $\tilde{y}' \sim \mathrm{Law}(\tilde{\mathsf{y}} \mid \tilde{\mathsf{w}}{=}\tilde{w})$. To wit,

$$\tilde{y}' = \tilde{y} + \mathrm{K}_{\tilde{\mathsf{w}}\tilde{\mathsf{y}}}\mathrm{K}^{-1}_{\tilde{\mathsf{w}}\tilde{\mathsf{w}}}\left[\tilde{w} - m_{\tilde{\mathsf{w}}}\right].$$

Suppose that this joint distribution arises from a nonlinear, possibly stochastic, operator $\mathscr{T}$

$$\mathscr{T}\tilde{\mathsf{y}} = \tilde{\mathsf{w}}$$

If the joint distribution of y, w is not exactly jointly Gaussian but arises from a nonlinear relationship, we can still construct an approximate analogue to the Matheron update based on an approximating jointly Gaussian distribution, e.g. derived by propagation of errors, or ensemble estimates or some other variational approximation. Without specifying the nature of the approximation, we assume that there is some means by which we may choose a "good" local approximation to the likelihood, In METHO we calculate the local approximation to a joint covariance in terms of ensemble statistics of the model propagated forward.

this is not quite right; rvs are deterministic here so need to talk about an approximating likelihood

$$\begin{aligned}\mathrm{K}_{\tilde{\mathsf{w}}\tilde{\mathsf{y}}}(\tilde{w},\tilde{y}) &\approx \mathrm{K}_{\tilde{\mathsf{w}}\tilde{\mathsf{y}}} \mid \tilde{\mathsf{w}} = \tilde{w}, \tilde{\mathsf{y}} = \tilde{y}\\ \mathrm{K}_{\tilde{\mathsf{w}}\tilde{\mathsf{w}}}(\tilde{w},\tilde{y}) &\approx \mathrm{K}_{\tilde{\mathsf{w}}\tilde{\mathsf{w}}} \mid \tilde{\mathsf{w}} = \tilde{w}, \tilde{\mathsf{y}} = \tilde{y}.\end{aligned}$$

We can use this to construct an incremental version of the Matheron update by interpreting this update as a nonlinear least-squares optimisation..

We define $\Delta\tilde{y} := [(\tilde{\mathsf{y}} \mid \tilde{\mathsf{w}}{=}\tilde{w}) - \tilde{\mathsf{y}}] \mid \tilde{\mathsf{y}} = \tilde{y}$, the ultimate Matheron step update applied to a given sample. We rewrite the update as the solution to a quadratic optimisation,

$$\begin{aligned}\Delta\tilde{y} &= \mathrm{K}_{\tilde{\mathsf{w}}\tilde{\mathsf{y}}}(\tilde{w},\tilde{y})[\mathrm{K}_{\tilde{\mathsf{w}}\tilde{\mathsf{w}}}(\tilde{w},\tilde{y})]^{-1}\left[\tilde{w} - m_{\tilde{\mathsf{w}}}\right]\\ &\Rightarrow\\ \Delta\tilde{y} &= \mathrm{argmin}_{\tilde{\delta}}\,\|\tilde{\delta} - \mathrm{K}_{\tilde{\mathsf{w}}\tilde{\mathsf{y}}}(\tilde{w},\tilde{y}+\tilde{\delta})\mathrm{K}^{-1}_{\tilde{w}\tilde{w}}(\tilde{w},\tilde{y}+\tilde{\delta})\left[\tilde{w} - m_{\tilde{\mathsf{w}}}\right]\|_2^2\\ &=: \mathrm{argmin}_{\tilde{\delta}}\, L^*(\tilde{\delta})\end{aligned}$$

Differentiating this loss function we find a gradient

$$\frac{\partial}{\partial\tilde{\delta}}L^*(\tilde{\delta}) = 2\Delta\tilde{\delta} - 2\mathrm{K}_{\tilde{\mathsf{w}}\tilde{\delta}}(\tilde{w}, \tilde{y}+\tilde{\delta})\mathrm{K}^{-1}_{\tilde{\mathsf{w}}\tilde{\mathsf{w}}}(\tilde{w}, \tilde{y}+\tilde{\delta})\left[\tilde{w} - m_{\tilde{\mathsf{w}}}\right] + \mathcal{O}(\text{high order terms}) \tag{23}$$

Disregarding the higher order terms, the Hessian is approximately

$$\frac{\partial^2}{\partial\tilde{\delta}\tilde{\delta}^\top}L^*(\tilde{\delta}) \approx 2\mathrm{I}.$$

This suggest we an approximate Newton update step

$$\begin{aligned}\tilde{\delta}^{(i+1)} &= \tilde{\delta}^{(i)} - \left(\frac{\partial^2}{\partial\tilde{\delta}\tilde{\delta}^\top}L^*(\tilde{\delta}^{(i)})\right)^{-1}\frac{\partial}{\partial\tilde{\delta}}L^*(\tilde{\tilde{\delta}}^{(i)})\\ &= \tilde{\delta}^{(i)} - \frac{1}{2}\frac{\partial}{\partial\tilde{\delta}}L^*(\tilde{\delta}^{(i)})\\ &= \mathrm{K}_{\tilde{\mathsf{w}}\tilde{y}}(\tilde{w}, \tilde{y}+\tilde{\delta}^{(i)})\mathrm{K}^{-1}_{\tilde{\mathsf{w}}\tilde{\mathsf{w}}}(\tilde{w}, \tilde{y}+\tilde{\delta}^{(i)})\left[\tilde{w} - m_{\tilde{\mathsf{w}}}\right].\end{aligned}$$

If w, y are linearly related and thus jointly Gaussian then this loss is exactly quadratic, does not depend upon $\tilde{\delta}$, and the update is exact in one step. In a nonlinear least squares setting, the update is no longer exact, but may converge subject to sufficiently nice behaviour as with any Newton-type method. Alternatively we can use line search direction for line search optimisation such as Levenberg–Marquardt.

mention non-differentiability of covariances; alternatively, can we differentiate those covariances?

.

B Lanczos approximations to the inverse covariance matrix

The Lanczos algorithm iteratively approximates $\mathrm{A} \in \mathbb{R}^{n\times n}$ by a sequence of factorizations in which, after the kth step, $\mathrm{A} \simeq \mathrm{Q}^{(k)}\mathrm{T}^{(k)}\mathrm{Q}^{(k)\top}$, where $\mathrm{T} \in \mathbb{R}^{k\times k}$ is positive-definite, tridiagonal and $\mathrm{Q} \in \mathbb{R}^{n\times k}$ has orthogonal columns. We do not need to form A, but simply to evaluate $\mathrm{A}\boldsymbol{b}$ for vectors $\boldsymbol{b}$. Hereafter we suppress the (k) superscript, assuming that the number of iterations is held fixed and sufficiently large.

Moreover, with a given Lanczos approximand Q, T we may estimate $\mathrm{A}^{-1}\boldsymbol{c} \simeq \mathrm{Q}\mathrm{T}^{-1}\mathrm{Q}^\top\boldsymbol{c}$. This is the essential trick in making $\left(\tilde{\mathrm{K}} + \sigma^2\mathrm{I}_N\right)^{-1}\boldsymbol{b} \approx \left(\breve{\mathrm{U}}\breve{\mathrm{U}} + \sigma^2\mathrm{I}\right)^{-1}\mathbf{b}$ which speeds the calculation of (19). Lanczos approximation of this quantity requires us to calculate products $(\breve{\mathrm{U}}\breve{\mathrm{U}}^\top + \sigma^2\mathrm{I})\boldsymbol{b} = \breve{\mathrm{U}}(\breve{\mathrm{U}}^\top\boldsymbol{b}) + \sigma^2\boldsymbol{b}$. Inspecting the latter from, we see that the necessary product may be calculated using the without calculating the full covariance matrix.

accuracy of this update? Depends on condition number

How good is our Lanczos approximation? Saad [2003] gives the following results the residual vector of the approximate solution x_m is such that

$$b - Ax_m = -\beta_{m+1}e_m^T y_m v_{m+1}.$$

citation chase for bounds on this approx quality?

C Stationary moves in an ensemble Gaussian posterior

Suppose we have a d-dimensional Gaussian RV whose variances are given by an empirical ensemble estimate of N samples,

$$\begin{aligned}\mathsf{x} &\sim \mathcal{N}(m_x, \mathrm{K}_{\mathsf{xx}})\\ \mathrm{K}_{\mathsf{xx}} &:= \breve{\mathrm{X}}\breve{\mathrm{X}}^\top\end{aligned}$$

just as in the Ensemble Kalman methods, so that

$$\begin{aligned}\bar{m}_{\mathsf{x}} &:= \frac{1}{N}\sum_{i=1}^N \mathsf{x}^{\tilde{(i)}}\\ \breve{\mathrm{X}} &= \frac{1}{\sqrt{N-1}}\left[(\mathsf{x}^{\tilde{(1)}} - \bar{m}_{\mathsf{x}}) \quad \cdots \quad (\mathsf{x}^{\tilde{(N)}} - \bar{m}_{\mathsf{x}})\right]\end{aligned}$$

Note that that the deviation matrix represents a square-root factorization for the covariance, we can simulate new realisations from the distribution of this variable using $\mathsf{x} \sim m_{\mathsf{x}} + \breve{\mathrm{X}}\xi$ for $\xi \sim \mathcal{N}(\mathbf{0}, \mathrm{I})$. A corollary to this is that this distribution and its updates comprise linear combinations of members of the ensemble, and so if $d > N$ it cannot span the space of all possible realisations.

not quite. what does it actually imply?

We could diversify the ensemble, in the sense of simulating realisations with the same moments but which did not come from the span of the ensemble if there existed a non-trivial random transform j which was not restricted to the span of this basis such that $\mathbf{x} \sim \mathcal{N}(m_{\mathbf{x}}, \Sigma_{\mathbf{x}}) \Rightarrow j(\mathbf{x}) \overset{\mathrm{d}}{=} \mathbf{x}$. Can we construct such a j?

Langevin dynamics might be a tractable approach. Given $\boldsymbol{\zeta} \sim \mathcal{N}(0, \mathrm{I}_d)$ and a step size ϵ, an Euler-Maruyama approximation to a Langevin move is

$$\begin{aligned} j(\boldsymbol{x}) &:= \boldsymbol{x} + \epsilon \nabla_{\boldsymbol{x}} \log p_{\mathbf{x}}(\boldsymbol{x}) + \sqrt{2\epsilon}\,\boldsymbol{\zeta}, \text{ and} \\ \nabla_{\boldsymbol{x}} \log p_{\mathbf{x}}(\boldsymbol{x}) &= -(\boldsymbol{x} - m_{\mathbf{x}})^\top \mathrm{K}_{\mathbf{xx}}^{-1} \end{aligned}$$

thus

$$\begin{aligned} \Delta\boldsymbol{x} - \sqrt{2\epsilon}\,\boldsymbol{\zeta} &= \epsilon(\boldsymbol{x} - m_{\mathbf{x}})^\top \mathrm{K}_{\mathbf{xx}}^{-1} \\ (\Delta\boldsymbol{x} - \sqrt{2\epsilon}\,\boldsymbol{\zeta})\mathrm{K}_{\mathbf{xx}} &= \epsilon(\boldsymbol{x} - m_{\mathbf{x}})^\top. \end{aligned}$$

Can we find any cool simplifications by plugging in ensemble deviations? Let us try:

$$\begin{aligned} \Delta\boldsymbol{x} - \sqrt{2\epsilon}\,\boldsymbol{\zeta} &= \epsilon(\boldsymbol{x} - m_{\mathbf{x}})^\top (\check{\mathrm{X}}\check{\mathrm{X}}^\top)^{-1} \\ (\Delta\boldsymbol{x} - \sqrt{2\epsilon}\,\boldsymbol{\zeta})\check{\mathrm{X}}\check{\mathrm{X}}^\top &= \epsilon(\boldsymbol{x} - m_{\mathbf{x}})^\top \\ \Delta\boldsymbol{x} &= \epsilon(\boldsymbol{x} - m_{\mathbf{x}})^\top (\check{\mathrm{X}}\check{\mathrm{X}}^\top)^{-1} + \sqrt{2\epsilon}\,\boldsymbol{\zeta} \end{aligned}$$

Hmph. That failed to do anything special. Once again a tricky inversion, but we can possibly use Lanczos approximations to get an approximate solution. If we have *already* factorised $\mathrm{K}_{\mathbf{xx}}$ then we can potentially recycle the Lanczos basis/pivots to get an approximate move. Although — presumably we already used that Lanczos approximation to make an update, so the covariance will be different and the update suspect.

This Euler-Maruyama approximation will clearly be bad if ϵ is not tiny; we could consider implicit steps or Metropolis adjustment? Both those look expensive.

We expect that iterating such steps will not be stationary, since the ensemble covariance

D Epistemic uncertainty in the posterior

Consider a jointly Gaussian ensemble covariance which includes a term to inflate the uncertainty of our estimate of the posterior, because, e.g., we would like to decrease the confidence of our estimate convinced that it is *truly* Gaussian, and quantifying that by adding Gaussian noise is the laziest option that might work. (Student-t distributions look hard and boring). This model will look similar to (17) but with the uncertainty term moved:

$$\begin{bmatrix} \tilde{\mathsf{z}}_t \\ \tilde{\mathsf{u}} \end{bmatrix} \Bigg| \mathsf{z}_{t\text{-}1} \simeq \mathcal{N}\left(\begin{bmatrix} \bar{z} \\ \bar{u} \end{bmatrix}, \begin{bmatrix} \check{\mathrm{Z}}\check{\mathrm{Z}}^\top & \check{\mathrm{U}}\check{\mathrm{Z}}^\top \\ \check{\mathrm{Z}}\check{\mathrm{U}}^\top & \check{\mathrm{U}}\check{\mathrm{U}}^\top + \mathrm{K}_\xi \end{bmatrix} \right). \tag{24}$$

What does that look like in ensemble inference?

D.1 Let us add independent noise to the samples

Simplest options: We can add some noise to each of the samples $\tilde{u}^{(i)} \leftarrow \boldsymbol{\xi}^{(i)} \sim \mathcal{N}(0, \mathrm{K}_\xi)$. Is this well posed? How about if $\mathrm{K}_\xi = \tau^2\mathrm{I}$? Connection to Langevin sampling above.

D.2 Inflate posterior covariance

Alternatively, suppose the posterior ensemble has covariance $\check{\mathrm{U}}\check{\mathrm{U}}^\top$ but we wish it had covariance $\check{\mathrm{U}}'\check{\mathrm{U}}'^\top \succ \check{\mathrm{U}}\check{\mathrm{U}}^\top$.

We could change that by scaling the ensemble, possibly randomly, about the mean by some scalar inflation factor s, $\mathbb{E}\mathsf{s} > 1$, $\tilde{u}^{(i)} \leftarrow \mathsf{s}(\tilde{u}^{(i)} - \boldsymbol{m}_{\mathsf{u}})$ or even $\tilde{u}^{(i)} \leftarrow \mathbf{S}(\tilde{u}^{(i)} - \boldsymbol{m}_{\mathsf{u}})$ from some matrix-valued $\mathbf{S}$.

Hmm, products of RVs is a complicated world to be in; I think I vote for deterministic scaling, or additive Gaussian noise, now that I think about it. According to [Roth et al., 2017] this is called variance inflation and is discussed in the literature.

E PDE systems

E.1 Navier-Stokes

E.2 Darcy

F notes

Todo list

G Refactored paper plan

1. code
   (a) Experiments with hyperparameters
   (b) ...
2. paper
   (a) in qhat metric are we approximating distributions when we say $\overset{\mathrm{d}}{\approx}$?
   (b) why does naive GD ensemble averaging work so well? Should write that up.
   (c) what is the implication of subsampling observation layer?
   (d) definitive $\mathcal{O}$ costing
   (e) observation noise