Contributed Discussion of "A Bayesian Conjugate Gradient Method"

F-X. Briol[1,2], F. A. DiazDelaO[3], P. O. Hristov[3]
[1]University College London, [2]The Alan Turing Institute
[3]Institute for Risk and Uncertainty, University of Liverpool

July 25, 2019

Brief intro acknowledging the value of the BCG algorithm in the growing field of Probabilistic Numerics and its potential. Maybe write this section-free.

1 Prior specification for BayesCG

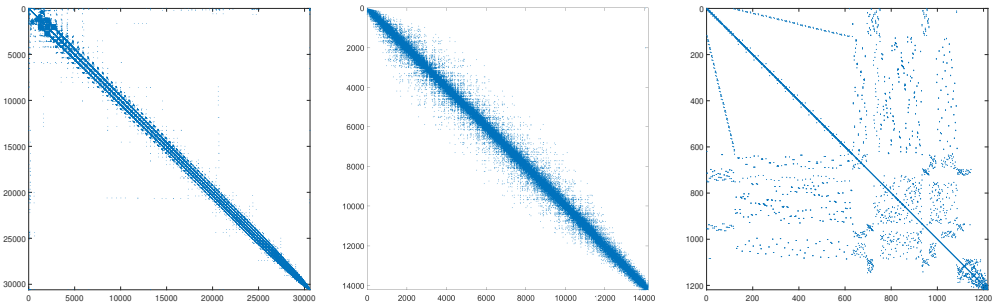
In Bayesian quadrature/cubature, the task is to estimate some integral $\Pi[f] = \int_{\mathcal{X}} f(x)\pi(x)\mathrm{d}x$, given evaluations of the integrand f at some locations on the domain $\mathcal{X}$. Clearly, the quantity of interest is $\Pi[f]$. Yet, it is common practice to instead put a prior on f, which then induces a prior on $\Pi[f]$. For differential equations, the problem is to find the solution u of some system of equations $\mathcal{A}u = g$ (where $\mathcal{A}$ is some known integro-differential operator), given evaluations of the right-hand side g. Again, several Bayesian methods [4, 3] propose to specify a prior on the right-hand side g instead of the quantity of interest u.

In both of these cases, the main motivation for placing priors on latent quantities is that this is more natural, or convenient, from a modelling point of view. At the same time, it is often possible to inspect the mathematical expression for the latent quantity, or we may at least have some additional information about it, such as smoothness or periodicity information. In such cases, encoding this information in the prior leads to algorithms with fast convergence rates and tighter credible intervals [4, 2].

We believe the same is true in the case of linear systems. In many applications, it is possible to know beforehand properties of the matrix A. This knowledge can be encoded in a prior. To illustrate this, consider some of the systems of differential equations used in engineering to describe fluid flow, structural response to loading or (increasingly) a combination of the two. For example, in computational structural mechanics the operator $\mathcal{A}$ can be used to describe the *stiffness* of an assembled finite element model (FEM). Similarly, in computational fluid dynamics (CFD) $\mathcal{A}$ can represent mesh coefficient matrices. Since both of these matrices describe physical properties of the object under study, their sparsity patterns will be governed largely by the object's geometry. It is therefore common that analysts have some prior knowledge about $\mathcal{A}$, based on experience with similar systems.

Figure 1 provides examples of the form of $\mathcal{A}$ for different structures. Figure 1(a) represents the stiffness matrix in an FEM of a notched cantilever beam. The sparsity pattern shown in Figure 1(b) encodes the coefficients of an unstructured mesh for a two dimensional airfoil in a CFD simulation [7]. Finally, the matrix in Figure 1(c) depicts the FEM

stiffness matrix of a jet engine compressor stage. All three geometries were meshed with two-dimensional triangular elements. An important note to make here is that the matrices shown in Figures 1(b) and 1(c) represent a typical coupled analysis. That is, the load on the compressor stage depends, among other factors, on the rotational speed and the force produced by its blades. This force in turn depends on the rotational speed of the compressor. Employing such chains of coupled models is not uncommon in design and analysis and can further complicate placing a prior on the solution, u.



(a) (b) (c)

Figure 1: Stiffness matrices exhibiting different degrees of sparsity and non-zero patterns. The systems described by these matrices are (a) notched beam; (b) a laminar airfoil; (c) jet engine compressor fan.

Properties of $\mathcal{A}$, such as its sparsity demonstrated above, will fully determine the difficulty of solving the linear system, and encoding them in a prior may lead to algorithms with faster convergence rate, and the potential for a more accurate representations of uncertainty.

2 A Generalisation to Multiple Linear Systems

Suppose we have several linear systems which need to be solved either simultaneously or sequentially, such that for $j \in \{1, \dots, J\}$, we want to solve:

$$x_j^* = A_j b_j,$$

where $x_j^* \in \mathbb{R}^d$, $A_j \in \mathbb{R}^{d \times d}$ and $b_j \in \mathbb{R}^d$ for some $d \in \mathbb{N}_{>0}$[1]. As discussed in [8], this is a common problem in statistics and machine learning. Take for example the issue of model selection for Gaussian processes: this includes calculating the log-marginal likelihood for several choices of covariance functions or covariance function hyperparameters, each requiring the solution of a linear system whose solutions will be closely related (atleast for similar choices of parameters). Similarly, for Bayesian inverse problems [6], the forward problem needs to be solved for several values of the parameters (perhaps some Markov Chain Monte Carlo samples), which will often boil down to solving several closely related linear systems.

As principled Bayesians, it would be natural to construct a joint estimator on the solutions of these J linear systems, rather than estimating the solutions independently. This is

[1]For simplicity of notation, we assume all systems are of the same size, but this could be generalised straightforwardly.

particularly the case if we know anything about how the solutions of these linear systems relate to one another, in which case information available through search directions in the j^{th} system may be informative about the solution $x^*_{j'}$ for $j \neq j'$. This idea is closely related to transfer learning, which was recently advocated for problems in numerical analysis by [9] (who focused on numerical integration). Although several methods exist to transfer information from one task to the other, such as recycled Krylov spaces [8], there are no existing Bayesian approach.

Interestingly, we show below that the Bayesian conjugate gradient algorithm of [5] may be generalised straightforwardly to this setting. All expressions below are given so as to mirror the notation of the one-system case closely. The main point to make is that all of these systems can be seen as a single, larger, linear system of the form $\underline{x}^* = \underline{A}\underline{b}$ where $\underline{x} = ((x^*_1)^\top, \ldots, (x^*_J)^\top)^\top \in \mathbb{R}^{dJ}$, $\underline{b} = (b_1^\top, \ldots, b_J^\top)^\top \in \mathbb{R}^{dJ}$ and $\underline{A} \in \mathbb{R}^{dJ \times dJ}$ is of the form

$$\underline{A} = \text{BlockDiag}\left[A_1, \ldots, A_J\right] = \begin{pmatrix} A_1 & & \\ & \ddots & \\ & & A_J \end{pmatrix}.$$

We define the data obtained by $y_i = s_i^\top A x^* = s_i^\top b$ for $i \in \{1, \ldots, m\}$. We will define $\underline{S}_m \in \mathbb{R}^{dJ \times m}$ to be the matrix consisting of columns given by m search directions. The data can therefore be expressed in vector form as $\underline{y}_m = \underline{S}_m^\top \underline{b}$. Taking a Bayesian approach, we select a prior of the form $\mathcal{N}(\underline{x}, \underline{x}_0, \underline{\Sigma}_0)$, for some $\underline{x}_0 \in \mathbb{R}^{dJ}$ and $\underline{\Sigma}_0 \in \mathbb{R}^{dJ \times dJ}$. Conditioning on the data $\underline{y}_m$, we obtain a posterior of the form $\mathcal{N}(\underline{x}; \underline{x}_m, \underline{\Sigma}_m)$ with $\underline{x}_m = \underline{x}_0 + \underline{\Sigma}_0 \underline{A}^\top \underline{S}_m \underline{\Lambda}_m^{-1} \underline{S}_m^\top \underline{r}_0$, $\underline{\Sigma}_m = \underline{\Sigma}_0 - \underline{\Sigma}_0 \underline{A}^\top \underline{S}_m \underline{\Lambda}_m^{-1} \underline{S}_m^\top \underline{A} \underline{\Sigma}_0$ where $\underline{r}_0 = \underline{b} - \underline{A}\underline{x}_0$ and $\underline{\Lambda}_m = \underline{S}_m^\top \underline{A} \underline{\Sigma}_0 \underline{A}^\top \underline{S}_m$.

The search directions which allow us to avoid the matrix inverse are $\underline{A}\underline{\Sigma}_0\underline{A}^\top$-orthogonal, and provide what we call the *multi-system Bayesian conjugate gradient algorithm*. Let $r_m = \underline{b} - \underline{A}\underline{x}_m$, $\tilde{\underline{s}}_1 = \underline{r}_0$ and $\underline{s}_m = \tilde{\underline{s}}_m / \|\tilde{\underline{s}}_m\|_{\underline{A}\underline{\Sigma}_0\underline{A}^\top}$ for all m, then for $m > 1$, assuming that $\tilde{\underline{s}}_m \neq \underline{0} = (0, \ldots, 0)$, these directions are:

$$\tilde{\underline{s}}_m = \underline{r}_{m-1} - \langle \underline{s}_{m-1}, \underline{r}_{m-1} \rangle_{\underline{A}\underline{\Sigma}_0\underline{A}^\top} \underline{s}_{m-1}.$$

At this point, most of the equations in the two paragraph above look identical to those in the paper, but include larger vectors and matrices. We now make several remarks on this algorithm:

1. The search directions obtained by the multi-system Bayesian conjugate gradient algorithm lead to some dependence across linear systems. That is, the estimator for x^*_j for some fixed j will be impacted by $A_{j'}, b_{j'}$ for some $j' \neq j$. This dependence will come from the matrix $\underline{\Sigma}_0$, the covariance matrix of our prior. This leads to a larger computational cost, due to the fact that we are now having to perform matrix-vector products of matrices of size $dJ \times dJ$, but this may be acceptable if it provides improved accuracy and uncertainty quantification.

2. Several special cases of prior matrix $\underline{\Sigma}_0$, inspired by vector-valued reproducing kernel Hilbert spaces [1] or multi-output Gaussian processes, can be more convenient to use in practice due to their intepretability. One example are separable covariance functions, which were previously explored by [9] for transfer learning in numerical integration. They take the form $\underline{\Sigma}_0 = B \otimes \Sigma_0$ where $\otimes$ denotes the Kronecker product, $B \in \mathbb{R}^{J \times J}$ and $\Sigma_0 \in \mathbb{R}^{d \times d}$. In this case, the matrix B can be seen as a covariance matrix across tasks (i.e. across linear systems), whilst Σ_0 is the covariance matrix which would

otherwise be used for a single linear system. In particular, this approach would allow us to combined the algorithm with alternative transfer learning approaches, such as the Krylov subspace recycling discussed in [8] which can be used to select Σ_0.

3. In the case where $\underline{\Sigma}_0$ has block-diagonal form BlockDiag $[\Sigma_{0,1}, \ldots, \Sigma_{0,J}]$ for $\Sigma_{0,1}, \ldots, \Sigma_{0,J} \in \mathbb{R}^{d\times d}$, the multi-system Bayesian conjugate gradient method reduces to J separate instances of the Bayesian conjugate gradient; it is therefore a strict generalisation.

4. The requirement that search directions are $\underline{A}\underline{\Sigma}_0\underline{A}^\top-$orthogonal forces us to solve the J linear systems simultaneously, obtaining one observation from each system at a given iteration. This prevents us from considering the sequential case where we first solve A_1, then solve A_2 and so on. However, we envisage that alternative algorithms could be developed for this case, and could help provide informative priors in a sequential manner.

References

[1] M. A. Álvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.

[2] F-X. Briol, C. J. Oates, M. Girolami, M. A. Osborne, and D. Sejdinovic. Probabilistic integration: A role in statistical computation? (with discussion). *Statistical Science*, 34 (1):1–22, 2019.

[3] O. A. Chkrebtii, D. A. Campbell, B. Calderhead, and M. Girolami. Bayesian solution uncertainty quantification for differential equations (with discussion). *Bayesian Analysis*, 11(4):1239–1267, 2016.

[4] J. Cockayne, C. J. Oates, T. Sullivan, and M. Girolami. Probabilistic meshless methods for partial differential equations and Bayesian inverse problems. *arXiv:1605.07811*, 2016.

[5] J. Cockayne, C. J. Oates, I. C. F. Ipsen, and M. Girolami. A Bayesian Conjugate Gradient Method. *Bayesian Analysis*, 2019.

[6] M. Dashti and A. M. Stuart. The Bayesian approach to inverse problems. In *Handbook of Uncertainty Quantification*, pages 311–428. Springer, 2017.

[7] Timothy A. Davis and Yifan Hu. The University of Florida sparse matrix collection. *ACM Trans. Math. Softw.*, 38(1), 2011.

[8] F. de Roos and P. Hennig. Krylov subspace recycling for fast iterative least-squares in machine learning. *arXiv:1706.00241*, 2017.

[9] X. Xi, F-X. Briol, and M. Girolami. Bayesian quadrature for multiple related integrals. In *International Conference on Machine Learning, PMLR 80*, pages 5369–5378, 2018.