Variance Reduction for Matrix Computations with Applications to Gaussian Processes

Anant Mathur[1(✉)], Sarat Moka[2], and Zdravko Botev[1]

[1] University of New South Wales,
High Street, Kensington Sydney, NSW 2052, Australia
anant.mathur@unsw.edu.au

[2] Macquarie University, 192 Balaclava Rd, Macquarie Park, NSW 2113, Australia

Abstract. In addition to recent developments in computing speed and memory, methodological advances have contributed to significant gains in the performance of stochastic simulation. In this paper we focus on variance reduction for matrix computations via matrix factorization. We provide insights into existing variance reduction methods for estimating the entries of large matrices. Popular methods do not exploit the reduction in variance that is possible when the matrix is factorized. We show how computing the square root factorization of the matrix can achieve in some important cases arbitrarily better stochastic performance. In addition, we detail a factorized estimator for the trace of a product of matrices and numerically demonstrate that the estimator can be up to 1,000 times more efficient on certain problems of estimating the log-likelihood of a Gaussian process. Additionally, we provide a new estimator of the log-determinant of a positive semi-definite matrix where the log-determinant is treated as a normalizing constant of a probability density.

Keywords: stochastic simulation · Variance reduction · Gaussian processes

1 Introduction

In many scientific computing and machine learning problems we are required to estimate the entries of a positive semi-definite (PSD) matrix $\mathbf{A} = (a_{i,j}) \in \mathbb{R}^{n\times n}$, where $\mathbf{A}$ is not explicitly available, but is rather only available as a black box function. That is to say, we assume we only have access to an oracle function that for any $\boldsymbol{x} \in \mathbb{R}^n$ can compute $\mathbf{A}\boldsymbol{x}$. Current methods [3] employ Monte Carlo simulations to produce unbiased estimates for $a_{i,j}$ via operations of the kind $\mathbf{A}\boldsymbol{Z}$ where $\boldsymbol{Z} \in \mathbb{R}^n$ is a random vector drawn from a pre-defined distribution. Hutchinson [12] was the first to use this methodology to approximate matrix traces, $\mathrm{tr}(\mathbf{A}) = \sum_i a_{i,i}$ and Bekas et al. [3] extended this method to estimate the actual diagonal values $a_{i,i}$. Significant innovations [1,2,8,16,18] in this area

Q. Zhao and L. Xia (Eds.): VALUETOOLS 2021, LNICST 404, pp. 243–261, 2021.
https://doi.org/10.1007/978-3-030-92511-6_16

of research have sought to improve the performance of these estimators through variance reduction and other means, effectively improving their scalability to larger matrices.

One major application of this kind of estimation is fitting Gaussian processes where one is required to evaluate the functions of the black box matrices: $\text{tr}(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \theta})$ and $\ln|\mathbf{K}| = \text{tr}(\ln(\mathbf{K}))$, where $\mathbf{K} \in \mathbb{R}^{n\times n}$ is a covariance matrix parameterized by θ, and is of dimension n - the size of the data-set. In these cases, explicitly computing the matrix functions roughly requires $O(n^3)$ operations, making their evaluation infeasible for any matrix of reasonable size whereas, $\mathbf{K}^{-1}\boldsymbol{x}$ and $\log(\mathbf{K})\boldsymbol{x}$ can be estimated in m iterations via a Krylov Subspace method that takes $O(mn^2)$ time. Generally, $m \ll n$ iterations are required with each iteration only requiring a matrix-vector-multiplication operation with $\mathbf{K}$.

The scope of black box matrix estimation, however, is not limited to Gaussian processes. For example, one can estimate statistical leverages for a sparse linear model by approximating the diagonals of the projection matrix $\mathbf{A} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$ as in [7] or one can precondition neural networks by estimating the diagonals of a Hessian matrix [6]. Furthermore, black box trace estimation has many applications including but not limited to evaluating the Cramer-Rao lower bound for the variance of an unbiased estimator [4] and approximating spectral densities [14].

Martens et al. [15] suggest that the variance for the Monte Carlo estimate of $a_{i,j}$ can depend on the matrix factorization of $\mathbf{A}$, and our work extends their ideas further. Our paper explores how taking the square root factorization of $\mathbf{A} = \mathbf{A}^{1/2}\mathbf{A}^{1/2}$ can reduce variance with little to no additional cost to CPU time.

In particular, our paper makes the following contributions:

1. We illustrate the advantages of black box matrix estimation with factorization over estimation without factorization. In particular, we consider the estimator of Bekas et al. [3] for the diagonal elements that uses no factorization and compare it with the factorized square root estimator to show that the former can exhibit an arbitrarily larger variance compared to the latter. No such analysis has been given in [15]. We further provide analysis on how the distribution of $\boldsymbol{Z}$ effects the variance of the factorized estimator. This includes variance results for when $\boldsymbol{Z}$ are simulated to be uniformly distributed on the surface of the unit radius n-dimensional sphere centered at the origin.
2. We introduce an estimator for the trace of a product of matrices, $\text{tr}(\mathbf{AW})$ where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a PSD matrix and $\mathbf{W} \in \mathbb{R}^{n\times n}$ is a symmetric matrix, see also [19] and [10]. This estimator exploits square root factorization to achieve variance reduction. We provide numerical examples for the problem of estimating the log-likelihood of a Gaussian process to suggest that the estimator can be up to 1,000 times more efficient in performance, compared to existing methods, for commonly used covariance matrices with dimension n up to 20,000.
3. We propose a novel Monte Carlo estimator for the log-determinant, $\ln|\mathbf{A}|$ where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is assumed to be a PSD matrix. Existing Monte Carlo

methods for approximating the log-determinant require evaluating expressions such as $\ln(\mathbf{K})\boldsymbol{Z}$ via the Lanczos algorithm [11,21]. Our proposed method instead relies on the more memory efficient operation of solving the linear-system $\mathbf{K}^{-1}\boldsymbol{Z}$ via the conjugate gradient method.

This paper is organized as follows. In Sect. 2 we review how one can estimate the entries of the PSD matrix $\mathbf{A}$ via matrix vector multiplications with the random vector $\boldsymbol{Z}$. We then explain how factorization can attain variance reduction and we provide three examples to illustrate when factorization provides greater performance. In this section, we also discuss how one can efficiently compute the factorized estimator. In Sect. 3, we introduce a Monte Carlo estimator for $\operatorname{tr}(\mathbf{AW})$ and we illustrate the performance improvement attained when the estimator is applied to the problem of estimating the log-likelihood of a Gaussian process. In Sect. 4 we discuss the existing Monte Carlo method for estimating $\ln|\mathbf{A}|$ via the Lanczos algorithm [21] and we introduce a new Monte Carlo estimator which treats the log-determinant as the normalizing constant of a probability density. Finally, we discuss implications of our research and areas of future work. All proofs are available in the Appendix unless stated otherwise.

2 Estimating Matrix Elements

Suppose we wish to estimate the entries of the PSD matrix $\mathbf{A} = (a_{i,j}) \in \mathbb{R}^{n\times n}$. Let $\mathbf{BC}^\top$ be an arbitrary matrix factorization of the matrix $\mathbf{A}$, with $\mathbf{B}, \mathbf{C} \in \mathbb{R}^{n\times p}$. Let $\boldsymbol{Z} \in \mathbb{R}^p$ be a random vector such that $\mathbb{E}\left[\boldsymbol{Z}\boldsymbol{Z}^\top\right] = \mathbf{I}_p$. Then, a rank-1 unbiased estimate of $\mathbf{A}$ is given in [15]:

$$\hat{\mathbf{A}} = \mathbf{B}\boldsymbol{Z}\boldsymbol{Z}^\top\mathbf{C}^\top = (\mathbf{B}\boldsymbol{Z})\,(\mathbf{C}\boldsymbol{Z})^\top. \tag{1}$$

To approximate $a_{i,j}$ we evaluate $[\mathbf{B}\boldsymbol{Z}]_i \times [\mathbf{C}\boldsymbol{Z}]_j$, and to approximate the diagonal of $\mathbf{A}$ we evaluate $(\mathbf{B}\boldsymbol{Z}) \odot (\mathbf{C}\boldsymbol{Z})$, where $\odot$ is defined as the element-wise matrix product. If we select the trivial factorization, $\mathbf{A} = \mathbf{A}\mathbf{I}_n$, that is when $\mathbf{B} = \mathbf{A}$ and $\mathbf{C} = \mathbf{I}_n$, we obtain the well known Bekas et al. [3] estimator $\hat{\ell}_{\text{Bekas}}$ for the diagonal,

$$\hat{\ell}_{\text{Bekas}} = \boldsymbol{Z} \odot \mathbf{A}\boldsymbol{Z}. \tag{2}$$

Various sampling distributions for $\boldsymbol{Z}$ have been proposed, including the Rademacher distribution, whose entries are either -1 or 1 each with probability 0.5. One can also sample $\boldsymbol{Z}$ from the standard Gaussian, uniformly from the set of standard basis vectors or uniformly from the surface of the unit radius n-dimensional sphere centered at the origin.

We note that Bekas et al. [3] suggest scaling the diagonal estimate $\hat{\ell}_{\text{Bekas}}$ element-wise by the vector $\boldsymbol{Z} \odot \boldsymbol{Z}$. This scaling has no effect when $\boldsymbol{Z}$ is Rademacher as $\boldsymbol{Z} \odot \boldsymbol{Z}$ is equal to the vector of ones. When $\boldsymbol{Z}$ is Gaussian, the performance with scaling is improved, as proven by Kaperick [13]. However, even with scaling the variance of the Gaussian estimator is still marginally higher than that of the Rademacher estimator.

The estimator $\hat{\ell}_{\text{Bekas}}$ does not guarantee non-negative diagonal entries when $\mathbf{A}$ is a PSD matrix. To mitigate this, we can use a decomposition of the form $\mathbf{A} = \mathbf{B}\mathbf{B}^\top$, that is, when $\mathbf{C} = \mathbf{B}$. The entries of the ensuing estimate $\mathbf{B}\boldsymbol{Z} \odot \mathbf{B}\boldsymbol{Z}$ are now guaranteed to be non-negative. We note that this decomposition is not unique and can for example be the Cholesky decomposition, or the symmetric square root factorization $\mathbf{A} = \mathbf{A}^{1/2}\mathbf{A}^{1/2}$. Moreover, if the functional form of $\mathbf{A}$ is known, it may be exploitable and the factorization $\mathbf{A} = \mathbf{B}\mathbf{B}^\top$ can be constructed without computing the symmetric square root or Cholesky factorization.

Example 1 (Diagonals of Inverse Matrices). Suppose we wish to estimate the variances for the OLS estimator $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{y}$, where $\mathbf{X} \in \mathbb{R}^{n\times p}$, and $\boldsymbol{y} \in \mathbb{R}^n$. Assume our observations are independent and $\mathbb{V}\text{ar}(y_i) = \sigma^2$ for all $i = 1, \ldots, n$. Then, the covariance matrix $\mathbf{A} = \mathbb{V}\text{ar}\left[\hat{\boldsymbol{\beta}}\right]$ is,

$$\mathbf{A} = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{I}_n\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1} = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}.$$

Our goal is to estimate the diagonal entries of $\mathbf{A}$. In this example, we do not need to compute the square root of $\mathbf{A}$ directly (e.g., $\mathbf{B} = \mathbf{A}^{1/2}$). Instead, we can use conjugate gradient methods, since $\mathbf{A} = \mathbf{B}\mathbf{B}^\top$, where,

$$\mathbf{B} = \sigma(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top.$$

Thus, the factorized diagonal estimator is $\hat{\ell} = \mathbf{B}\boldsymbol{Z} \odot \mathbf{B}\boldsymbol{Z}$ and the computation $(\mathbf{X}^\top\mathbf{X})^{-1}\boldsymbol{c}$, where $\boldsymbol{c} = \mathbf{X}^\top\boldsymbol{Z}$ can be solved via the conjugate gradient method.

2.1 Variance Analysis

Martens et al. [15] provides variance formulas for estimator (1) when $\boldsymbol{Z}$ is drawn from the standard Gaussian and Rademacher distribution. In this section we state these formulas along with the variance formula for when $\boldsymbol{Z}$ is drawn uniformly on the surface of the unit radius n-dimensional sphere centered at the origin. For the following analysis we assume that $\hat{\mathbf{A}} = (\hat{a}_{i,j}) = (\mathbf{B}\boldsymbol{Z})\,(\mathbf{C}\boldsymbol{Z})^\top$ and $\mathbf{B}, \mathbf{C} \in \mathbb{R}^{n\times n}$. Furthermore, we denote $\boldsymbol{b}_{i,:}$ and $\boldsymbol{c}_{i,:}$ as the column vectors containing the i-th rows of the matrices $\mathbf{B}$ and $\mathbf{C}$.

Gaussian: When $\boldsymbol{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$, the variance for $\hat{a}_{i,j}$ is,

$$\mathbb{V}\text{ar}_G[\hat{a}_{i,j}] = \|\boldsymbol{b}_{i,:}\|^2\|\boldsymbol{c}_{j,:}\|^2 + a_{i,j}^2, \tag{3}$$

and when $\mathbf{B} = \mathbf{A}$, $\mathbf{C} = \mathbf{I}_n$, and $i = j$, we obtain the variance for the diagonal estimator of Bekas et al. [3], that is,

$$\mathbb{V}\text{ar}_G^{\mathbf{AI}}[\hat{a}_{i,i}] = 2a_{i,i}^2 + \sum_{j\neq i} a_{i,j}^2. \tag{4}$$

Theorem 1 (Theorem 4.1 in Martens et al. [15]). *For the factorization* $\mathbf{A} = \mathbf{B}\mathbf{C}^\top$, $\mathbb{V}\mathrm{ar}_G[\hat{a}_{i,i}]$ *is minimized when* $\mathbf{C} = \mathbf{B}$ *with resulting variance,*

$$\mathbb{V}\mathrm{ar}_G^{\mathbf{B}\mathbf{B}^\top}[\hat{a}_{i,i}] = 2a_{i,i}^2. \tag{5}$$

Our next result contributes to our understanding of using a spherical probing vector.

Uniform Spherical: Suppose $\boldsymbol{Y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$ then $\boldsymbol{Z} = \boldsymbol{Y}/\|\boldsymbol{Y}\|$ is uniformly distributed on the surface of the unit radius n-dimensional sphere centered at the origin.

Theorem 2 (Spherical Estimator for Matrix Elements). *Suppose* $\boldsymbol{Z}$ *is uniformly distributed on the surface of the unit radius* n*-dimensional sphere centered at the origin and* $\mathbf{A} = \mathbf{B}\mathbf{C}^\top$. *Then,*

$$\mathbb{E}\left[n\left(\mathbf{B}\boldsymbol{Z}\right)\left(\mathbf{C}\boldsymbol{Z}\right)^\top\right] = \mathbf{A},$$

and $\hat{a}_{i,j} = n\left[\mathbf{B}\boldsymbol{Z}\right]_i \times \left[\mathbf{C}\boldsymbol{Z}\right]_j$ *has variance,*

$$\mathbb{V}\mathrm{ar}_S[\hat{a}_{i,j}] = \frac{n}{n+2}\|\boldsymbol{b}_{i,:}\|^2\|\boldsymbol{c}_{j,:}\|^2 + \frac{n-2}{n+2}a_{i,j}^2.$$

As we can see, when n is large, $\mathbb{V}\mathrm{ar}_S[\hat{a}_{i,j}] \approx \mathbb{V}\mathrm{ar}_G[\hat{a}_{i,j}]$.

Rademacher: Martens et al. [15] give the following variance for $\hat{a}_{i,j}$ when $\boldsymbol{Z} \in \mathbb{R}^n$ is Rademacher,

$$\begin{aligned}\mathbb{V}\mathrm{ar}_R[\hat{a}_{i,j}] &= \mathbb{V}\mathrm{ar}_G[\hat{a}_{i,j}] - 2\sum_k c_{i,k}^2 b_{j,k}^2. \\ &= \|\boldsymbol{b}_{i,:}\|^2\|\boldsymbol{c}_{j,:}\|^2 + a_{i,j}^2 - 2\sum_k c_{i,k}^2 b_{j,k}^2.\end{aligned}$$

For the two cases where 1) $\mathbf{B} = \mathbf{A}$, $\mathbf{C} = \mathbf{I}$ (unfactorized) and 2) $\mathbf{C} = \mathbf{B}$ (factorized), we have the variances for the diagonal entry estimates,

$$\mathbb{V}\mathrm{ar}_R^{\mathbf{A}\mathbf{I}}[\hat{a}_{i,i}] = \sum_{j \neq i} a_{i,j}^2. \tag{6}$$

$$\mathbb{V}\mathrm{ar}_R^{\mathbf{B}\mathbf{B}^\top}[\hat{a}_{i,i}] = 2a_{i,i}^2 - 2\sum_k b_{i,k}^4. \tag{7}$$

These formulas indicate $\mathbb{V}\mathrm{ar}_R^{\mathbf{B}\mathbf{B}^\top} \leq \mathbb{V}\mathrm{ar}_G^{\mathbf{B}\mathbf{B}^\top}$ and $\mathbb{V}\mathrm{ar}_R^{\mathbf{A}\mathbf{I}}[\hat{a}_{i,i}] \leq \mathbb{V}\mathrm{ar}_G^{\mathbf{A}\mathbf{I}}[\hat{a}_{i,i}]$; therefore we recommend the Rademacher as the best choice for $\boldsymbol{Z}$.

Extending Results on Rademacher: Theorem 1 shows that $\mathbf{A} = \mathbf{B}\mathbf{B}^\top$ is the most optimal factorization when $\boldsymbol{Z}$ is Gaussian. This is not guaranteed for the Rademacher case and our contribution is to analyze the effect of the square root factorization in the Rademacher case. For example, for PSD matrices with growing off-diagonals with increasing size, $\mathbb{V}\mathrm{ar}_R^{\mathbf{AI}}[\hat{a}_{i,i}]$ can be made arbitrarily large, while $\mathbb{V}\mathrm{ar}_R^{\mathbf{B}\mathbf{B}^\top}[\hat{a}_{i,i}]$ remains bounded above by $2a_{i,i}^2$. We now show an example matrix that behaves this way.

Example 2 (Stochastic matrices). Suppose we have x, y such that $x > y \geq 0$. Consider the symmetric matrix $\mathbf{A}$ of size $n \times n$ with

$$a_{i,j} = \begin{cases} x, & \text{if } i = j, \\ y, & \text{if } i \neq j. \end{cases}$$

For this matrix $\mathbf{A}$, we can show that the eigenvalues (or singular values) are $x+py$ with multiplicity 1 and $x - y$ with multiplicity $n - 1$. Since all the eigenvalues are positive, $\mathbf{A}$ is a positive definite matrix. In particular,

$$\mathbb{V}\mathrm{ar}_R^{\mathbf{AI}}[\hat{a}_{i,i}] - \mathbb{V}\mathrm{ar}_R^{\mathbf{B}\mathbf{B}^\top}[\hat{a}_{i,i}] \geq \sum_{j \neq i} a_{i,j}^2 - 2a_{i,i}^2 = (n-1)y^2 - 2x^2.$$

Therefore, $\mathbb{V}\mathrm{ar}_R^{\mathbf{AI}}[\hat{a}_{i,i}] \geq \mathbb{V}\mathrm{ar}_R^{\mathbf{B}\mathbf{B}^\top}[\hat{a}_{i,i}]$ for all $i = 1, \ldots, n$ if $y \geq x\sqrt{2/(n-1)}$. Furthermore, the factorized estimator is substantially better than the unfactorized estimator when the ratio $\sqrt{(n-1)}\, y/x \to \infty$ as $n \to \infty$.

For instance, let $\varepsilon_n = \frac{1}{(n-1)^\gamma}$ for some constant $1/2 < \gamma < 1$. Then, with $x = \varepsilon_n$ and $y = \frac{1-\varepsilon_n}{n-1}$, the matrix $\mathbf{A}$ is a doubly stochastic matrix for all n, because each row and each column sum to $x + (n-1)y = 1$. Note that $x > y$ for sufficiently large values of n. Furthermore,

$$\frac{\sqrt{(n-1)}\, y}{x} = \frac{\sqrt{n-1}}{(n-1)^{-\gamma}} \frac{1-\varepsilon_n}{n-1} = (1-\varepsilon_n)\,(n-1)^{\gamma - 1/2},$$

which goes to ∞ as $n \to \infty$. In other words, the square root estimator can be arbitrarily better than the naive one.

We now show an example where the unfactorized estimator exhibits an arbitrary larger variance compared to the factorized estimator when estimating an arbitrary matrix element.

Example 3 (Toeplitz matrices with exponential decay). Consider the Gram matrix $\mathbf{A} = \{k(x_i, x_j)\}_{i,j=1,\ldots,n}$, where k is the Squared Exponential kernel parameterized by $\boldsymbol{\theta} = (\theta_1, \theta_2)^\top$,

$$k(x_i, x_j) = \theta_1 \exp\left[\frac{-(x_i - x_j)^2}{2\theta_2^2}\right],$$

for $x_1, \ldots, x_n \in \mathbb{R}$. For simplicity let $\theta_1 = 1$, forcing the diagonal entries to be unitary. Suppose $\{x_i\}_{i=1,\ldots,n}$ are equally separated with spacing $\frac{1}{n-1}$, then $\mathbf{A}$ is

Toeplitz. The entries of $\mathbf{A}$ can be written as,

$$a_{i,j} = \exp\left[\frac{-(i-j)^2}{2(n-1)^2\theta_2^2}\right].$$

For the estimate $\hat{a}_{i,j}$, the variance of the unfactorized estimator has the following lower bound,

$$\begin{aligned}\mathbb{V}\mathrm{ar}_R^{\mathbf{AI}}[\hat{a}_{i,j}] &= \sum_{k=1}^{n} \exp\left[\frac{-(i-k)^2}{(n-1)^2\theta_2^2}\right] - \exp\left[\frac{-(i-j)^2}{(n-1)^2\theta_2^2}\right] \\ &\geq n\exp(-\theta_2^{-2}) - 1.\end{aligned}$$

If we let $\mathbf{B} = (b_{i,j}) \in \mathbb{R}^{n\times n}$ be the square root matrix of $\mathbf{A}$, then we obtain the upper bound for the variance of the factorized estimator,

$$\mathbb{V}\mathrm{ar}_R^{\mathbf{BB}^\top}[\hat{a}_{i,j}] = 2 - 2\sum_m b_{i,m}^2 b_{j,m}^2 \leq 2.$$

In this example, $\mathbb{V}\mathrm{ar}_R^{\mathbf{AI}}[\hat{a}_{i,j}]$ can be made arbitrary larger than the variance for the factorized estimator. That is to say, for all $\delta > 0$ there exists n' such that $\mathbb{V}\mathrm{ar}_R^{\mathbf{AI}}[\hat{a}_{i,j}] - \mathbb{V}\mathrm{ar}_R^{\mathbf{BB}^\top}[\hat{a}_{i,j}] > \delta$ for all $n > n' = \lceil(\delta+3)\exp(\theta_2^{-2})\rceil$.

We now show an example of the matrix $\mathbf{A}$ where the unfactorized Rademacher estimator exhibits lower variance than the factorized estimator.

Example 4 (Tridiagonal matrices). It is clear that a PSD tridiagonal matrix should be an example where we expect the unfactorized estimator to perform on par or better than the symmetrized estimator due to its minimal off-diagonal entries. While it is safe to assume that this case is rare in application, it is still a worthwhile example to show when factorizing $\mathbf{A}$ may not be advantageous. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be the symmetric tridiagonal matrix,

$$\mathbf{A} = \begin{bmatrix} 1 & c & & & \\ c & 1 & c & & \\ & c & \ddots & \ddots & \\ & & \ddots & \ddots & c \\ & & & c & 1 \end{bmatrix}.$$

Then, the Cholesky decomposition of $\mathbf{A}$ is $\mathbf{A} = \mathbf{B}\mathbf{B}^\top$, where $\mathbf{B} \in \mathbb{R}^{n\times n}$ is a lower bi-diagonal matrix with entries given by the following recursive formulas,

$$b_{i,i} = \sqrt{1 - b_{i,i-1}^2}, \quad \text{and} \quad b_{i,i-1} = \frac{c}{b_{i-1,i-1}}, \tag{8}$$

where $b_{1,1} = 1$ and $b_{2,1} = c$. By using (6), (7) and (8) we can show,

$$\mathbb{V}\mathrm{ar}_R^{\mathbf{BB}^\top}[\hat{a}_{i,i}] = 4c^2\left[\frac{1}{b_{i-1,i-1}^2} - \frac{c^2}{b_{i-1,i-1}^4}\right] = 4c^2\frac{b_{i,i}^2}{b_{i-1,i-1}^2}, \quad \text{and} \tag{9}$$

$$\mathbb{V}\mathrm{ar}_R^{\mathbf{AI}^\top}[\hat{a}_{i,i}] = \begin{cases} c^2, & \text{if } i = 1 \text{ or } n, \\ 2c^2, & \text{otherwise.} \end{cases}$$

By numerically evaluating (8) we observe that $\frac{b_{i,i}^2}{b_{i-1,i-1}^2}$ is non-decreasing in i, hence $\mathbb{V}\mathrm{ar}_R^{\mathbf{BB}^\top}[\hat{a}_{i,i}]$ is non-decreasing in i. We further observe that as the limit $n \to \infty$,

$$\mathbb{V}\mathrm{ar}_R^{\mathbf{BB}^\top}[\hat{a}_{n,n}] \uparrow 4c^2.$$

Based on these numerical observations,

$$\frac{\mathbb{V}\mathrm{ar}_R^{\mathbf{BB}^\top}[\hat{a}_{i,i}]}{\mathbb{V}\mathrm{ar}_R^{\mathbf{AI}}[\hat{a}_{i,i}]} \leq \frac{\mathbb{V}\mathrm{ar}_R^{\mathbf{BB}^\top}[\hat{a}_{n,n}]}{\mathbb{V}\mathrm{ar}_R^{\mathbf{AI}}[\hat{a}_{n,n}]},$$

and as $n \to \infty$,

$$\frac{\mathbb{V}\mathrm{ar}_R^{\mathbf{BB}^\top}[\hat{a}_{n,n}]}{\mathbb{V}\mathrm{ar}_R^{\mathbf{AI}}[\hat{a}_{n,n}]} \uparrow 4.$$

Therefore, $\mathbb{V}\mathrm{ar}_R^{\mathbf{BB}^\top}[\hat{a}_{i,i}]$ will be at most $4 \times \mathbb{V}\mathrm{ar}_R^{\mathbf{AI}}[\hat{a}_{i,i}]$ for large values of n.

Numerical simulations with the square root factorization $\mathbf{B} = \mathbf{A}^{1/2}$ indicate a similar conclusion. In particular, for certain values of c, $\mathbb{V}\mathrm{ar}_R^{\mathbf{AI}}[\hat{a}_{i,i}] > \mathbb{V}\mathrm{ar}_R^{\mathbf{BB}^\top}[\hat{a}_{i,i}]$, with a relative difference of at most 0.4. This relative difference remains bounded above by 0.4 as $n \to \infty$, whereas in Examples 2 and 3 the relative difference, when the factorized estimator outperforms the unfactorized estimator, goes to ∞ as $n \to \infty$. In plain English, while the downside of using the factorized estimator is limited, the upside can be unlimited as the size of the matrix grows.

2.2 Computing $\mathbf{A}^{1/2}\boldsymbol{Z}$

We briefly review well-known methods for efficiently computing with the square root factorization. As discussed in the introduction, black box estimation relies on the ability to efficiently evaluate matrix-vector operations. Thus, when using the factorization $\mathbf{A} = \mathbf{B}\mathbf{B}^\top$ we must consider how to evaluate $\mathbf{B}\boldsymbol{Z}$. If the functional form of $\mathbf{A}$ is not exploitable or not known, the only practical choice for $\mathbf{B}$ is $\mathbf{B} = \mathbf{A}^{1/2}$, the unique symmetric square root of $\mathbf{A}$.

Chow and Saad [5] outline how to compute $\mathbf{A}^{1/2}\boldsymbol{Z}$ via the well-known Lanczos tridiagonalization algorithm. The Lanczos algorithm takes the black box matrix $\mathbf{A}$ and the initial vector $\boldsymbol{Z}$ and after $m \leq n$ iterations outputs the matrices $\mathbf{T}_m$ and $\mathbf{V}_m$ that satisfy,

$$\mathbf{T}_m = \mathbf{V}_m^\top \mathbf{A} \mathbf{V}_m,$$

where $\mathbf{V}_m \in \mathbb{R}^{n \times m}$ contains orthonormal columns, and $\mathbf{T}_m \in \mathbb{R}^{m \times m}$ is a symmetric tridiagonal matrix. We can then use the approximation $\mathbf{V}_m^\top f(\mathbf{A}) \mathbf{V}_m \approx f\left(\mathbf{V}_m^\top \mathbf{A} \mathbf{V}_m\right) = f(\mathbf{T}_m)$ and take $f(\mathbf{A}) = \mathbf{A}^{1/2}$ to obtain the approximation,

$$\boldsymbol{x}_m^* = \|\boldsymbol{Z}\| \mathbf{V}_m \mathbf{T}_m^{1/2} \boldsymbol{e}_1. \tag{10}$$

Generally $\|\boldsymbol{x}_m^* - \mathbf{A}^{1/2}\boldsymbol{Z}\|$ can be made small when $m \ll n$, thus allowing efficient computation and storing of $\mathbf{T}_m^{1/2}$. A discussion on the convergence rates for the Lanczos approximation can be found in Chow and Saad [5]. In general, this method has $O(mn^2)$ time complexity.

Alternatively, we can evaluate $\mathbf{A}^{1/2}\boldsymbol{Z}$ using the contour integral quadrature (CIQ) of Pleiss et al. [17]. This method also offers $O(mn^2)$ cost where m, the number of iterations is often less than 100 even for matrices of size up to 50,000. This method can be implemented via the GPyTorch framework [9].

3 Estimating the Trace of a Product of Matrices

In this section we introduce an estimator for the trace of a product of matrices, $\operatorname{tr}(\mathbf{AW})$, where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a PSD matrix and $\mathbf{W} \in \mathbb{R}^{n\times n}$ is a symmetric matrix. Applications of such traces abound and an example with Gaussian processes is given later on. If the goal is to estimate $\operatorname{tr}(\mathbf{A})$, we can take the sum of the estimator $\hat{\ell}_{\text{Bekas}}$ to obtain the well known Hutchinson estimator,

$$\operatorname{tr}(\hat{\mathbf{A}}) = \boldsymbol{Z}^\top \mathbf{A}\boldsymbol{Z}, \tag{11}$$

where $\mathbb{E}[\boldsymbol{Z}\boldsymbol{Z}^\top] = \mathbf{I}_n$. We note that there is no difference in the estimator when using the factorization $\mathbf{A} = \mathbf{B}\mathbf{B}^\top$ as,

$$\sum_i [\mathbf{B}\boldsymbol{Z}]_i \times [\mathbf{B}\boldsymbol{Z}]_i = \boldsymbol{Z}^\top \mathbf{B}\mathbf{B}^\top \boldsymbol{Z} = \boldsymbol{Z}^\top \mathbf{A}\boldsymbol{Z}.$$

However, suppose we want to estimate $\operatorname{tr}(\mathbf{AW})$, where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a PSD matrix and $\mathbf{W} \in \mathbb{R}^{n\times n}$ is a symmetric matrix (not necessarily PSD). We can then use the following estimator,

$$\hat{\psi}_{\text{sqrt}} = \boldsymbol{Z}^\top \mathbf{B}^\top \mathbf{W}\mathbf{B}\boldsymbol{Z}, \tag{12}$$

where $\mathbf{B} = \mathbf{A}^{1/2}$, see also [19] and [10]. To see this is unbiased, observe that due to the invariant cyclic nature of the trace,

$$\mathbb{E}\left[\boldsymbol{Z}^\top \mathbf{B}^\top \mathbf{W}\mathbf{B}\boldsymbol{Z}\right] = \mathbb{E}\left[\operatorname{tr}(\mathbf{Z}\mathbf{Z}^\top \mathbf{B}^\top \mathbf{W}\mathbf{B})\right] = \operatorname{tr}(\mathbf{AW}).$$

3.1 Variance Analysis

We now provide comparative variance analysis of the square root estimator (12) when $\boldsymbol{Z}$ is drawn from the standard Gaussian, from the Rademacher distribution, and uniformly on the surface of the unit radius n-dimensional sphere centered at the origin. For the following analysis we refer to the square root estimator as,

$$\hat{\psi}_{\text{sqrt}} = \boldsymbol{Z}^\top \mathbf{B}^\top \mathbf{W}\mathbf{B}\boldsymbol{Z},$$

where $\mathbf{B} = \mathbf{A}^{1/2}$, and the standard Bekas et al. [3] estimator as,

$$\hat{\psi} = \boldsymbol{Z}^\top \mathbf{A}\mathbf{W}\boldsymbol{Z}.$$

Gaussian

Theorem 3 (Lemma 9 in Avron and Toledo [2]). *Suppose* $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$, *and* $\boldsymbol{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$. *Let* $\mathbf{\Sigma}' = (\mathbf{\Sigma} + \mathbf{\Sigma}^\top)/2$. *Then,*

$$\mathbb{V}\mathrm{ar}\left[\boldsymbol{Z}^\top \mathbf{\Sigma} \boldsymbol{Z}\right] = 2\|\mathbf{\Sigma}'\|_{\mathrm{F}}^2, \tag{13}$$

where $\|\mathbf{X}\|_{\mathrm{F}} := \sqrt{\mathrm{tr}(\mathbf{X}^\top \mathbf{X})} = \sqrt{\sum_{i,j} (\mathbf{X}_{i,j})^2}$.

Following from this theorem we have the following.

Lemma 1 (Comparing the Variance of the Gaussian Trace Estimator). *If* $\boldsymbol{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$, *then* $\mathbb{E}[\hat{\psi}_{\mathrm{sqrt}}] = \mathbb{E}[\hat{\psi}] = \mathrm{tr}(\mathbf{AW})$,

$$\mathbb{V}\mathrm{ar}_G[\hat{\psi}] = \mathrm{tr}\left((\mathbf{AW})^\top (\mathbf{AW})\right) + \mathrm{tr}\left((\mathbf{WA})^\top (\mathbf{AW})\right), \quad \textit{and} \tag{14}$$

$$\mathbb{V}\mathrm{ar}_G[\hat{\psi}_{\mathrm{sqrt}}] = 2\,\mathrm{tr}\left((\mathbf{WA})^\top (\mathbf{AW})\right). \tag{15}$$

Furthermore,

$$\mathbb{V}\mathrm{ar}_G[\hat{\psi}_{\mathrm{sqrt}}] \leq \mathbb{V}\mathrm{ar}_G[\hat{\psi}].$$

Proof Using Theorem 3, we obtain the variance formulas (14) and (15). To prove $\mathbb{V}\mathrm{ar}[\ell_{\mathrm{sqrt}}] \leq \mathbb{V}\mathrm{ar}[\hat{\psi}]$ we need to show,

$$\mathrm{tr}\left((\mathbf{WA})^\top (\mathbf{AW})\right) \leq \mathrm{tr}\left((\mathbf{AW})^\top (\mathbf{AW})\right).$$

As $\langle \mathbf{X}, \mathbf{Y} \rangle_{\mathrm{F}} := \mathrm{tr}(\mathbf{X}^\top \mathbf{Y})$ defines the Frobenius inner-product, the inequality holds true from the Cauchy-Schwarz inequality.

Uniform Spherical

Theorem 4 (Variance of the Spherical Quadratic Form). *Suppose* $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ *and* $\boldsymbol{Z}$ *is uniformly distributed on the surface of the unit radius* n*-dimensional sphere centered at the origin. Let* $\mathbf{\Sigma}' = (\mathbf{\Sigma} + \mathbf{\Sigma}^\top)/2$. *Then,*

$$\mathbb{V}\mathrm{ar}\left[n\boldsymbol{Z}^\top \mathbf{\Sigma}' \boldsymbol{Z}\right] = \frac{n}{n+2} \times 2\left(\|\mathbf{\Sigma}'\|_{\mathrm{F}}^2 - \frac{(\sum_i \mathbf{\Sigma}'_{i,i})^2}{n}\right). \tag{16}$$

Following from this theorem we have the following.

Lemma 2 (Variance of the Spherical Trace Estimator). *If* $\boldsymbol{Z}$ *is uniformly distributed on the surface of the unit radius* n*-dimensional sphere centered at the origin, then* $\mathbb{E}[n\hat{\psi}_{\mathrm{sqrt}}] = \mathbb{E}[n\hat{\psi}] = \mathrm{tr}(\mathbf{AW})$,

$$\mathbb{V}\mathrm{ar}_S\left[n\hat{\psi}\right] = \frac{n}{n+2}\mathbb{V}\mathrm{ar}_G\left[\hat{\psi}\right] - \frac{2}{n+2}\left(\sum_i \left[\frac{\mathbf{AW}+\mathbf{WA}}{2}\right]_{i,i}\right)^2, \quad \textit{and} \tag{17}$$

$$\mathbb{V}\mathrm{ar}_S\left[n\hat{\psi}_{\mathrm{sqrt}}\right] = \frac{n}{n+2}\mathbb{V}\mathrm{ar}_G\left[\hat{\psi}_{\mathrm{sqrt}}\right] - \frac{2}{n+2}\left(\sum_i [\mathbf{B}^\top \mathbf{WB}]_{i,i}\right)^2. \tag{18}$$

As we can see, $\mathbb{V}\mathrm{ar}_S \approx \mathbb{V}\mathrm{ar}_G$ for large values of n.

Rademacher

Theorem 5 (Lemma 1 in Avron and Toledo [2]). *Suppose* $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$, *and* $\boldsymbol{Z} \in \mathbb{R}^n$ *follows a Rademacher distribution. Let* $\mathbf{\Sigma}' = (\mathbf{\Sigma} + \mathbf{\Sigma}^\top)/2$. *Then,*

$$\mathbb{V}\mathrm{ar}\left[\boldsymbol{Z}^\top \mathbf{\Sigma} \boldsymbol{Z}\right] = 2\left[\|\mathbf{\Sigma}'\|_{\mathrm{F}}^2 - \sum_i \mathbf{\Sigma}'^2_{i,i}\right]. \tag{19}$$

Following from this theorem we have the following.

Lemma 3 (Variance of the Rademacher Trace Estimator). *If* $\boldsymbol{Z}$ *is a* n*-dimensional Rademacher random vector, then* $\mathbb{E}[\hat{\psi}_{\mathrm{sqrt}}] = \mathbb{E}[\hat{\psi}] = \mathrm{tr}(\mathbf{AW})$,

$$\mathbb{V}\mathrm{ar}_R\left[\hat{\psi}\right] = \mathbb{V}\mathrm{ar}_G\left[\hat{\psi}\right] - 2\sum_i \left[\frac{\mathbf{AW} + \mathbf{WA}}{2}\right]^2_{i,i}, \quad \textit{and} \tag{20}$$

$$\mathbb{V}\mathrm{ar}_R\left[\hat{\psi}_{\mathrm{sqrt}}\right] = \mathbb{V}\mathrm{ar}_G\left[\hat{\psi}_{\mathrm{sqrt}}\right] - 2\sum_i [\mathbf{B}^\top \mathbf{WB}]^2_{i,i}. \tag{21}$$

Analogous to our conclusions for estimating arbitrary matrix elements, the above results indicate that the Rademacher estimator is guaranteed to be better than Gaussian estimator for estimating the trace of $\mathbf{AW}$. Moreover, when $\boldsymbol{Z}$ is Gaussian, we have proven $\hat{\psi}_{\mathrm{sqrt}}$ always performs better than $\hat{\psi}$. However, this is not always true when $\boldsymbol{Z}$ is Rademacher.

Let $\mathbf{\Lambda} = (\mathbf{AW} + \mathbf{WA})/2$, and $\mathbf{\Omega} = \mathbf{B}^\top \mathbf{WB}$. Then $\mathbb{V}\mathrm{ar}_R\left[\hat{\psi}_{\mathrm{sqrt}}\right] > \mathbb{V}\mathrm{ar}_R\left[\hat{\psi}\right]$ if and only if,

$$2\sum_i \left[(\mathbf{\Lambda}_{i,i})^2 - (\mathbf{\Omega}_{i,i})^2\right] > \sum_{i,j}\left[(\mathbf{\Lambda}_{i,j})^2 - (\mathbf{\Omega}_{i,j})^2\right].$$

Numerical experiments indicate this inequality can hold when the contribution of $\|\mathbf{\Lambda}\|_{\mathrm{F}}$ and $\|\mathbf{\Omega}\|_{\mathrm{F}}$ is concentrated on the diagonal. In these cases both $\hat{\psi}_{\mathrm{sqrt}}$ and $\hat{\psi}$ are very accurate when $\boldsymbol{Z}$ is Rademacher. However, when most of the contribution of the Frobenius norm stems from the off-diagonals we observe cases where $\|\mathbf{\Lambda}\|_2^2 \gg \|\mathbf{\Omega}\|_2^2$ and $\mathbb{V}\mathrm{ar}_R\left[\hat{\psi}\right] \gg \mathbb{V}\mathrm{ar}_R\left[\hat{\psi}_{\mathrm{sqrt}}\right]$. We now show such an example in the context of maximum likelihood estimation for Gaussian processes.

3.2 Gaussian Processes

Gaussian processes (GP's) are used in a variety of modelling applications mainly through the following regression setting. Let $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ be a set of n data points and let $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$ be the vector of regression values under the model,

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i, \quad \text{where } \epsilon_i \sim \mathcal{N}(0, \sigma^2).$$

In GP regression, we assume the following prior on the function values $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))^\top$,

$$p(\boldsymbol{f}|\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}) = \mathcal{N}(\mathbf{0}, \mathbf{K}).$$

The covariance matrix $\mathbf{K}$ has entries $\mathbf{K}_{i,j} = \{k(\boldsymbol{x}_i, \boldsymbol{x}_j|\boldsymbol{\theta})\}$, where k is a covariance function with hyper-parameters $\boldsymbol{\theta}$. A popular choice for k is the Squared Exponential covariance function,

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \theta_1 \exp\left[\frac{-\|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2}{2\theta_2^2}\right].$$

To fit a GP we must select the hyper-parameters $\boldsymbol{\theta} = (\theta_1, \theta_2)^\top$ by maximizing the likelihood. Therefore, we need to evaluate the log-likelihood function,

$$\ln[\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y})] = -\frac{1}{2}\ln|\mathbf{K}_y| - \frac{1}{2}\boldsymbol{y}^\top \mathbf{K}_y^{-1}\boldsymbol{y} + \text{constant}, \tag{22}$$

and the score function,

$$\frac{\partial \ln[\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y})]}{\partial \theta_i} = -\frac{1}{2}\operatorname{Tr}\left(\mathbf{K}_y^{-1}\frac{\partial \mathbf{K}_y}{\partial \theta_i}\right) + \frac{1}{2}\boldsymbol{y}^\top \mathbf{K}_y^{-1}\frac{\partial \mathbf{K}}{\partial \theta_i}\mathbf{K}_y^{-1}\boldsymbol{y}, \tag{23}$$

where $\mathbf{K}_y = \mathbf{K} + \sigma^2\mathbf{I}_n$. For large n explicitly evaluating $\operatorname{tr}\left(\mathbf{K}_y^{-1}\frac{\partial \mathbf{K}_y}{\partial \theta_i}\right)$ is infeasible. Instead, we must approximate the trace via simulation. Suppose we simulate M draws of the n-dimensional Rademacher random variable, $\{\boldsymbol{Z}_i\}_{i=1,\ldots,M}$. Then, the Hutchinson estimator for $\operatorname{tr}\left(\mathbf{K}_y^{-1}\frac{\partial \mathbf{K}_y}{\partial \theta_2}\right)$ is,

$$\hat{\psi} = \frac{1}{M}\sum_{i=1}^{M}\left(\mathbf{K}_y^{-1}\boldsymbol{Z}_i\right)^\top\left(\frac{\partial \mathbf{K}_y}{\partial \theta_i}\boldsymbol{Z}_i\right).$$

Instead, we can use the square root estimator,

$$\hat{\psi}_{\text{sqrt}} = \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{c}_i^\top\frac{\partial \mathbf{K}_y}{\partial \theta_i}\boldsymbol{c}_i, \quad \text{where } \boldsymbol{c}_i = \mathbf{K}_y^{-1/2}\boldsymbol{Z}_i.$$

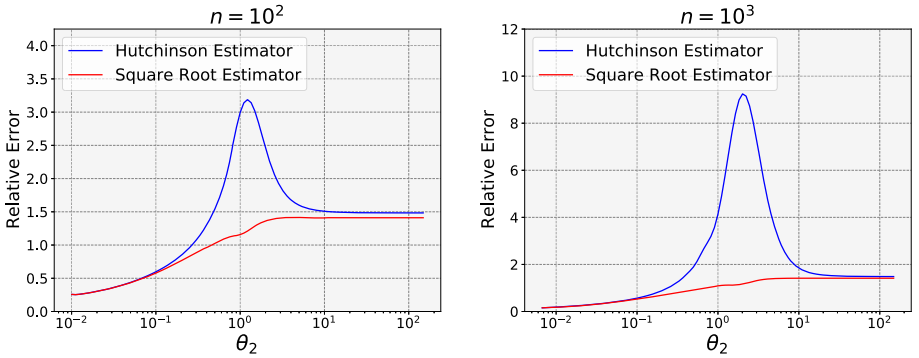

Fig. 1. True relative error as a function of θ_2 for $M = 1$ (a single replication), when $n = 10^2$ and $n = 10^3$. Relative Error $= \sqrt{\mathbb{V}\text{ar}(\hat{\psi})}/|\psi|$, where $\psi = \operatorname{tr}\left(\mathbf{K}_y^{-1}\frac{\partial \mathbf{K}_y}{\partial \theta_2}\right)$.

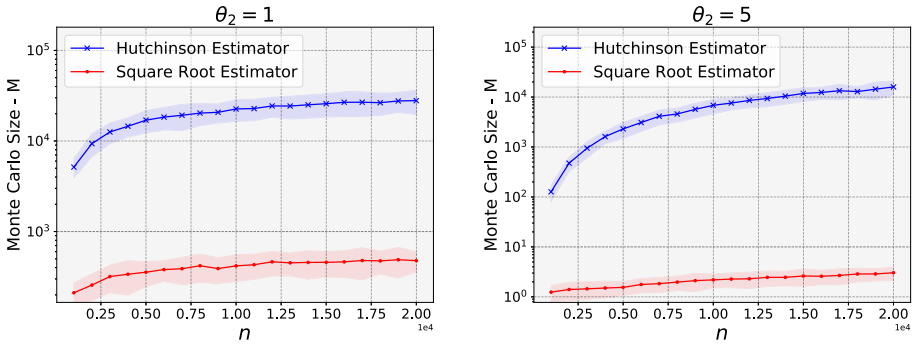

Fig. 2. Mean Monte Carlo size M required in simulation to obtain a mean squared error of 0.25 as n increases, when $\theta_2 = 1$ and $\theta_2 = 5$.

We can evaluate $\mathbf{K}_y^{-1/2}\boldsymbol{Z}_i$ with the Pleiss et al. [17] CIQ method. We observe that on average $\mathbf{K}_y^{-1/2}\boldsymbol{Z}_i$ takes double the CPU time to evaluate as compared to $\mathbf{K}_y^{-1}\boldsymbol{Z}_i$ via the conjugate gradient method; the latter being the required evaluation for the Hutchinson estimator. Importantly, both operations have $O(mn^2)$ complexity, where $m \ll n$.

To compare the performance of the estimators numerically, we fix $\sigma^2 = 0.1$ and $\theta_1 = 1$ and we let $\boldsymbol{X}$ be the n equidistant points spanning the interval $[0, 1]$. In Fig. 1 we evaluate the theoretical relative error of the two estimators when $M = 1$ and observe that the square root estimator always performs better than or equal to the Hutchinson estimator. For $\theta_2 \in [10^{-1}, 10^1]$ the relative error for Hutchinson estimator significantly deviates from the relative error of the square root estimator. In Fig. 2 we illustrate the differing performance as the size of the matrix increases in size. We observe that when $\theta_2 = 5$ the Hutchinson estimator can require up-to $1,000$ times more Monte Carlo replications to obtain the same accuracy (mean squared error) than the square root estimator.

4 Estimating the Log-Determinant

In this section we propose a conditional Monte Carlo estimator for the log-determinant of a positive semi-definite matrix by treating the log-determinant as a normalizing constant of a probability density. As seen from (22), training GP's require the evaluation of the log-determinant of $\mathbf{K}_y$. Suppose $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a PSD definite matrix. Then, $\ln|\mathbf{A}|$ can be evaluated in $O(n^3)$ time with the Cholesky decomposition. However, as previously stated this is infeasible for matrices with dimension greater than a few thousand. Existing log-determinant estimation techniques [9,21], utilize the Hutchinson estimator via the following approximation,

$$\ln|\mathbf{A}| = \operatorname{tr}(\ln(\mathbf{A})) = \mathbb{E}[\boldsymbol{Z}^\top \ln(\mathbf{A})\boldsymbol{Z}] \approx \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{Z}_i^\top \ln(\mathbf{A})\boldsymbol{Z}_i, \tag{24}$$

where $\boldsymbol{Z}_i \in \mathbb{R}^n$ is a Rademacher random vector and $\ln(\mathbf{A})\boldsymbol{Z}_i$, like $\mathbf{A}^{1/2}\boldsymbol{Z}_i$ is evaluated via the Lanczos tridiagonalization algorithm (see Sect. 2.2). Instead, we propose the following estimator,

$$\widehat{\ln|\mathbf{A}|} = -n\,c + 2\ln\left[\frac{1}{M}\sum_{i=1}^{M}\exp(-n(Y_i - c)/2)\right], \tag{25}$$

where $Y_i = \ln(\boldsymbol{\Theta}_i^\top \mathbf{A}^{-1}\boldsymbol{\Theta}_i)$, $c = \min_i Y_i$ and $\{\boldsymbol{\Theta}_i\}_{i=1,\ldots,M}$ are drawn i.i.d uniformly from the surface of the unit radius n-dimensional sphere centered at the origin. In this case, $\mathbf{A}^{-1}\boldsymbol{\Theta}_i$ is evaluated via the conjugate gradient method [11]. To show that this is a valid estimator, we first note that,

$$\ln|\mathbf{A}| = 2\ln\int (2\pi)^{-\frac{n}{2}}\exp\left(-\boldsymbol{z}^\top\mathbf{A}^{-1}\boldsymbol{z}/2\right)\mathbf{d}\boldsymbol{z}.$$

This equation can be derived by noting the integral of the multivariate normal density is equal to one. Thus, our goal is to produce an unbiased estimate of the integral on the RHS.

To estimate $\int \exp(-\boldsymbol{z}^\top\mathbf{A}^{-1}\boldsymbol{z}/2)\mathbf{d}\boldsymbol{z}$ we switch to spherical coordinates. Let $\boldsymbol{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$, then, $\boldsymbol{Z}$ can be represented as $\boldsymbol{Z} = R\boldsymbol{\Theta}$ with $\boldsymbol{\Theta}$ being uniformly distributed on the surface of the unit radius n-dimensional sphere centered at the origin and $R \sim \chi_n$, independently. Then, we have the unbiased estimator,

$$\frac{1}{(2\pi)^{n/2}}\int \exp(-\boldsymbol{z}^\top\mathbf{A}^{-1}\boldsymbol{z}/2)\mathbf{d}\boldsymbol{z} = \mathbb{E}[(\boldsymbol{\Theta}^\top\mathbf{A}^{-1}\boldsymbol{\Theta})^{-n/2}] \approx \frac{1}{M}\sum_{i=1}^{M}(\boldsymbol{\Theta}_i^\top\mathbf{A}^{-1}\boldsymbol{\Theta}_i)^{-n/2}.$$

The details of this derivation are provided in the Appendix. We therefore obtain the conditional Monte Carlo estimate,

$$\widehat{\ln|\mathbf{A}|} = 2\ln\left[\frac{1}{M}\sum_{i=1}^{M}(\boldsymbol{\Theta}_i^\top\mathbf{A}^{-1}\boldsymbol{\Theta}_i)^{-n/2}\right].$$

However, for large n, accurate evaluations of $(\boldsymbol{\Theta}_i^\top\mathbf{A}^{-1}\boldsymbol{\Theta}_i)^{-n/2}$ are unlikely due to round-off errors. Instead, we let,

$$Y_i = \ln(\boldsymbol{\Theta}_i^\top\mathbf{A}^{-1}\boldsymbol{\Theta}_i).$$

We then evaluate the numerically stable approximation,

$$\begin{aligned}\widehat{\ln|\mathbf{A}|} &= 2\ln\frac{1}{M}\sum_{i=1}^{M}\exp(-nY_i/2)\\ &= -n\,c + 2\ln\left[\frac{1}{M}\sum_{i=1}^{M}\exp(-n(Y_i - c)/2)\right],\end{aligned}$$

where $c = \min_i Y_i$. By evaluating the terms $\exp(-n(Y_i - c)/2)$ we can alleviate further round-off errors.

We note that as $\ln|\mathbf{A}| = -\ln|\mathbf{A}^{-1}|$, one can go about estimating $\ln|\mathbf{A}^{-1}|$ and therefore evaluate the terms $Y_i = \ln(\boldsymbol{\Theta}_i^\top \mathbf{A}\boldsymbol{\Theta}_i)$, which only require matrix vector multiplications with $\mathbf{A}$. As a numerical example, consider Fig. 3, where we estimate the log-determinant of the matrix $\mathbf{K}$, which is a squared exponential Gram matrix with $\theta_1 = 1$ and $\theta_2 = 1$.

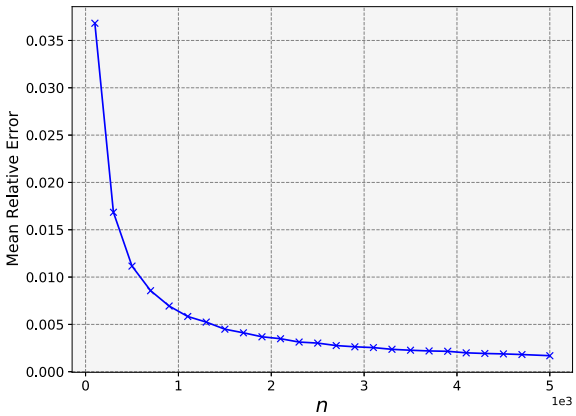

Fig. 3. Mean relative error of $\widehat{\ln|\mathbf{K}|}$, as n increases, where $\mathbf{K}$ is the squared exponential Gram matrix with $\theta_1 = 1$, $\theta_2 = 1$. Monte Carlo size: $M = 10$.

5 Conclusion

In this paper, we discuss how factorizing the PSD matrix $\mathbf{A}$ can reduce variance when trying to estimate its entries using only matrix vector operations of the form $\mathbf{A}\boldsymbol{Z}$ where $\boldsymbol{Z}$ is a random vector generated from a pre-defined distribution. We show in some examples that the unfactorized estimator can exhibit an arbitrary larger variance when compared to the factorized square root estimator. However, we also observe examples where the unfactorized estimator performs better than the factorized estimator, but the underperformance is relatively small and remains bounded from above as the size of the matrix grows. Whether this behavior of the underperformance holds true in general for all matrices $\mathbf{A}$ is an open question.

Furthermore, we show how factorization can achieve variance reduction when estimating $\mathrm{tr}(\mathbf{AW})$, where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a PSD matrix and $\mathbf{W} \in \mathbb{R}^{n\times n}$ is a symmetric matrix. Similar to arbitrary matrix entry estimation, we observe numerical examples where the factorized estimator significantly outperforms the unfactorized estimator. However, when $\boldsymbol{Z}$ is Rademacher, this outperformance is not theoretically guaranteed for all choices of $\mathbf{A}$ and $\mathbf{W}$. Ideally, we'd like a theoretical framework for identifying when the factorized estimator will work most in our favor.

Finally, we provide a conditional Monte Carlo estimate for the log-determinant of a PSD matrix $\mathbf{A}$ by treating it as the normalizing constant of a probability density. We note that it is possible to implement Markov chain Monte Carlo (MCMC) techniques such as particle filters to provide better estimates for the quantity $\mathbb{E}\left[(\boldsymbol{\Theta}^\top \mathbf{A}^{-1}\boldsymbol{\Theta})^{-n/2}\right]$.

A Proof of Theorem 1

The following proof is not explicitly stated in [15] and so we include it here for completeness.

Proof. To see this, first observe that when $\mathbf{C} = \mathbf{B}$, $\sum_j b_{i,j}^2 = a_{i,i}$. Therefore, from (3) we can deduce $\mathbb{V}\mathrm{ar}_G[a_{i,i}] = 2a_{i,i}^2$. Then, applying Cauchy-Schwarz inequality to (3), we obtain the inequality,

$$\mathbb{V}\mathrm{ar}_G[a_{i,i}] \geq (\boldsymbol{b}_{i,:})^\top (\boldsymbol{c}_{i,:}) + a_{i,i}^2.$$

Since $a_{i,i}^2$ remains the same for every decomposition of the form $\mathbf{A} = \mathbf{B}\mathbf{C}^\top$ and Cauchy-Schwarz inequality holds with equality if and only if $\boldsymbol{b}_{i,:}$ and $\boldsymbol{c}_{i,:}$ are linearly dependent, we conclude that $\mathbb{V}\mathrm{ar}_G[a_{i,i}]$ is minimized when $\mathbf{C} = \mathbf{B}$.

B Uniform Spherical Estimator

B.1 Proof of Theorem 2

Proof. Suppose $\boldsymbol{Z} = (Z_1, \ldots, Z_n)^\top$ is uniformly distributed on the surface of the unit radius n-dimensional sphere centered at the origin. Using spherical coordinates, it is not difficult to show that the vector $\boldsymbol{X} = \boldsymbol{Z} \odot \boldsymbol{Z}$ follows a Dirichlet distribution over the simplex:

$$f(\boldsymbol{x}) = \frac{\pi^{n/2}}{\Gamma(n/2)} \prod_{i=1}^{n} x_i^{-1/2}, \quad x_i \in [0,1], \quad \sum_{i=1}^{n} x_i = 1.$$

Therefore, using the well-known mean and variance formula for the Dirichlet distribution we have,

$$\mathbb{E}[\boldsymbol{X}] = \mathbf{1}/n, \quad \mathbb{V}\mathrm{ar}[\boldsymbol{X}] = \frac{1}{n(n/2+1)}\left[\mathbf{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\right]. \tag{26}$$

Using spherical coordinates, we also know that $\mathbb{E}[Z_i Z_j] = 0$ for $i \neq j$. Therefore,

$$\mathbb{E}[\boldsymbol{Z}\boldsymbol{Z}^\top] = \frac{1}{n}\mathbf{I}_n,$$

and,

$$\mathbb{E}\left[n\,(\mathbf{B}\boldsymbol{Z})\,(\mathbf{C}\boldsymbol{Z})^\top\right] = n\mathbb{E}\left[\mathbf{B}\boldsymbol{Z}\boldsymbol{Z}^\top\mathbf{C}^\top\right] = \mathbf{A}.$$

To prove the variance, we first note that,

$$\mathbb{E}[(\hat{a}_{i,j}/n)^2] = \mathbb{E}[([\mathbf{B}\boldsymbol{Z}]_i \times [\mathbf{C}\boldsymbol{Z}]_j)^2] = \sum_{k,l,m,n} b_{i,k}c_{j,l}b_{i,m}c_{j,n}\mathbb{E}[Z_k Z_l Z_m Z_n].$$

Representing $\boldsymbol{Z}$ in spherical coordinates, we obtain the formula,

$$\mathbb{E}[Z_k Z_l Z_m Z_n] = c_1[\delta_{kl}\delta_{mn} + \delta_{km}\delta_{ln} + \delta_{kn}\delta_{lm}] + (c_2 - 3c_1)[\delta_{kl}\delta_{lm}\delta_{mn}].$$

The constants c_1 and c_2 are given by,

$$c_1 = \mathbb{V}\mathrm{ar}[\boldsymbol{X}]_{p,q} + \mathbb{E}[\boldsymbol{X}]_p^2 = \frac{1}{n(n+2)}, \qquad p \neq q, \qquad \text{and}$$
$$c_2 = \mathbb{V}\mathrm{ar}[\boldsymbol{X}]_{p,p} + \mathbb{E}[\boldsymbol{X}]_p^2 = \frac{3}{n(n+2)}.$$

Thus $c_2 - 3c_1 = 0$, and,

$$\begin{aligned}\mathbb{E}[(a_{i,j}/n)^2] &= c_1 \sum_{k,l,m,n} b_{ik}c_{jl}b_{im}c_{jn}[\delta_{kl}\delta_{mn} + \delta_{km}\delta_{ln} + \delta_{kn}\delta_{lm}] \\ &= c_1\left[(\boldsymbol{b}_{i,:}^\top \boldsymbol{c}_{j,:})^2 + (\boldsymbol{b}_{i,:}^\top \boldsymbol{b}_{i,:})(\boldsymbol{c}_{j,:}^\top \boldsymbol{c}_{j,:}) + (\boldsymbol{b}_{i,:}^\top \boldsymbol{c}_{j,:})^2\right] \\ &= c_1\left[2a_{i,j}^2 + \|\boldsymbol{b}_{i,:}\|^2\|\boldsymbol{c}_{i,:}\|^2\right].\end{aligned}$$

Therefore,

$$\begin{aligned}\mathbb{V}\mathrm{ar}_S[(\hat{a}_{i,j}/n)] &= \mathbb{E}\left[(\hat{a}_{i,j}/n)^2\right] - a_{i,j}^2/n^2 \\ &= (2c_1 - 1/n^2)a_{i,j}^2 + c_1\|\boldsymbol{b}_{i,:}\|^2\|\boldsymbol{c}_{j,:}\|^2.\end{aligned}$$

Hence, the variance for $\hat{a}_{i,j}$ is,

$$\begin{aligned}\mathbb{V}\mathrm{ar}_S[(\hat{a}_{i,j})] &= n^2\mathbb{V}\mathrm{ar}_S[(\hat{a}_{i,j}/n)] \\ &= \frac{n-2}{n+2}a_{i,j}^2 + \frac{n}{n+2}\|\boldsymbol{b}_{i,:}\|^2\|\boldsymbol{c}_{j,:}\|^2.\end{aligned}$$

B.2 Proof of Theorem 4

Proof. Suppose $\boldsymbol{\Sigma}'$ has eigenvalues $\{\lambda_i\}$, and orthonormal eigendecomposition $\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top$. Then, as the random variable that is distributed uniformly on the surface of the n-dimensional sphere centered at the origin is invariant to orthogonal rotations, we obtain,

$$\mathbb{V}\mathrm{ar}[\boldsymbol{Z}^\top\boldsymbol{\Sigma}'\boldsymbol{Z}] = \mathbb{V}\mathrm{ar}[\boldsymbol{Z}^\top\boldsymbol{\Lambda}\boldsymbol{Z}] = \mathbb{V}\mathrm{ar}[\boldsymbol{\lambda}^\top\boldsymbol{X}] = \boldsymbol{\lambda}^\top\mathbb{V}\mathrm{ar}[\boldsymbol{X}]\boldsymbol{\lambda}.$$

We notice that $\mathrm{tr}(\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}) = \boldsymbol{\lambda}^\top\boldsymbol{\lambda}$. Therefore,

$$\mathbb{V}\mathrm{ar}[n\boldsymbol{Z}^\top\boldsymbol{\Sigma}'\boldsymbol{Z}] = \frac{n}{n+2} \times 2\left(\mathrm{tr}(\boldsymbol{\Sigma}'^\top\boldsymbol{\Sigma}') - \frac{(\sum_i \boldsymbol{\Sigma}'_{i,i})^2}{n}\right).$$

To see that this gives the variance for $n\boldsymbol{Z}^\top \boldsymbol{\Sigma} \boldsymbol{Z}$, we note,

$$\boldsymbol{Z}^\top \boldsymbol{\Sigma}' \boldsymbol{Z} = \boldsymbol{Z}^\top \left[(\boldsymbol{\Sigma} + \boldsymbol{\Sigma}^\top)/2 \right] \boldsymbol{Z} = \boldsymbol{Z}^\top \boldsymbol{\Sigma} \boldsymbol{Z}.$$

We note that when $\boldsymbol{Z}$ is distributed uniformly on the unit radius complex sphere instead, the variance formula is given in Tropp [20].

C Log-Determinant

Suppose $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$ is a PSD matrix. Let $\boldsymbol{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$. Then, $\boldsymbol{Z}$ can be represented as $\boldsymbol{Z} == R\boldsymbol{\Theta}$ with $\boldsymbol{\Theta}$ being uniformly distributed on the surface of the unit radius n-dimensional sphere centered at the origin and $R \sim \chi_n$, independently. Then using the standard normal as a change of measure we get the following,

$$\begin{aligned}
\frac{1}{(2\pi)^{n/2}} \int \exp(-\boldsymbol{z}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{z}/2) \mathbf{d}\boldsymbol{Z} &= \mathbb{E}\left[\exp(-\boldsymbol{Z}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{Z}/2 + \|\boldsymbol{Z}\|^2/2) \right] \\
&= \mathbb{E}\left[\exp(-R^2 \boldsymbol{\Theta}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Theta}/2 + R^2/2) \right] \\
&= \frac{1}{2^{n/2}\Gamma(n/2)} \mathbb{E}\left[\int_0^\infty r^{n/2-1} \exp(-r\boldsymbol{\Theta}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Theta}/2) \mathbf{d}r \right] \\
&= \mathbb{E}\left[(\boldsymbol{\Theta}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Theta})^{-n/2} \right] \\
&\approx \frac{1}{M} \sum_{i=1}^{M} (\boldsymbol{\Theta}_i^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Theta}_i)^{-n/2}.
\end{aligned}$$

We use the following integral formula to evaluate the integral on line 3,

$$\int_0^\infty r^{n/2-1} \exp(-r\alpha/2) \mathbf{d}r = \alpha^{-n/2} 2^{n/2} \Gamma(n/2).$$

Thus a conditional Monte Carlo estimate for the log-determinant is,

$$\widehat{\ln |\boldsymbol{\Sigma}|} = 2 \ln \left[\frac{1}{M} \sum_{i=1}^{M} (\boldsymbol{\Theta}_i^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\Theta}_i)^{-n/2} \right].$$

References

1. Adams, R.P., et al.: Estimating the spectral density of large implicit matrices. arXiv preprint arXiv:1802.03451 (2018)
2. Avron, H., Toledo, S.: Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. J. ACM (JACM) **58**(2), 1–34 (2011)
3. Bekas, C., Kokiopoulou, E., Saad, Y.: An estimator for the diagonal of a matrix. Appl. Numer. Math. **57**(11–12), 1214–1229 (2007)
4. Casella, G., Berger, R.L.: Statistical inference. Cengage Learning (2021)
5. Chow, E., Saad, Y.: Preconditioned krylov subspace methods for sampling multivariate gaussian distributions. SIAM J. Sci. Comput. **36**(2), A588–A608 (2014)

6. Dauphin, Y.N., De Vries, H., Bengio, Y.: Equilibrated adaptive learning rates for non-convex optimization. arXiv preprint arXiv:1502.04390 (2015)
7. Drineas, P., Magdon-Ismail, M., Mahoney, M., Woodruff, D.P.: Fast approximation of matrix coherence and statistical leverage. J. Mach. Learn. Res. **13**(1), 3475–3506 (2012)
8. Fitzsimons, J.K., Osborne, M.A., Roberts, S.J., Fitzsimons, J.F.: Improved stochastic trace estimation using mutually unbiased bases. arXiv preprint arXiv:1608.00117 (2016)
9. Gardner, J.R., Pleiss, G., Bindel, D., Weinberger, K.Q., Wilson, A.G.: Gpytorch: blackbox matrix-matrix gaussian process inference with gpu acceleration. arXiv preprint arXiv:1809.11165 (2018)
10. Geoga, C.J., Anitescu, M., Stein, M.L.: Scalable gaussian process computations using hierarchical matrices. J. Comput. Graph. Stat. **29**(2), 227–237 (2020)
11. Golub, G.H., Van Loan, C.F.: Matrix Computations, vol. 3. JHU Press (2013)
12. Hutchinson, M.F.: A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. Commun. Stat. Simul. Comput. **18**(3), 1059–1076 (1989)
13. Kaperick, B.J.: Diagonal Estimation with Probing Methods. Ph.D. thesis, Virginia Tech (2019)
14. Lin, L., Saad, Y., Yang, C.: Approximating spectral densities of large matrices. SIAM Rev. **58**(1), 34–65 (2016)
15. Martens, J., Sutskever, I., Swersky, K.: Estimating the hessian by back-propagating curvature. arXiv preprint arXiv:1206.6464 (2012)
16. Meyer, R.A., Musco, C., Musco, C., Woodruff, D.P.: Hutch++: optimal stochastic trace estimation. In: Symposium on Simplicity in Algorithms (SOSA), pp. 142–155. SIAM (2021)
17. Pleiss, G., Jankowiak, M., Eriksson, D., Damle, A., Gardner, J.R.: Fast matrix square roots with applications to gaussian processes and bayesian optimization. arXiv preprint arXiv:2006.11267 (2020)
18. Stathopoulos, A., Laeuchli, J., Orginos, K.: Hierarchical probing for estimating the trace of the matrix inverse on toroidal lattices. SIAM J. Sci. Comput. **35**(5), S299–S322 (2013)
19. Stein, M.L., Chen, J., Anitescu, M.: Stochastic approximation of score functions for gaussian processes. In: The Annals of Applied Statistics, pp. 1162–1191 (2013)
20. Tropp, J.A.: Randomized algorithms for matrix computations (2020)
21. Ubaru, S., Chen, J., Saad, Y.: Fast estimation of tr(f(a)) via stochastic lanczos quadrature. SIAM J. Matrix Anal. Appl. **38**(4), 1075–1099 (2017)