Applied Mathematics Letters 120 (2021) 107246

Contents lists available at ScienceDirect

Applied Mathematics Letters

www.elsevier.com/locate/aml

Faster stochastic trace estimation with a Chebyshev product identity

Eric Hallman[1]

2311 Stinson Dr, Raleigh, NC, United States of America

A R T I C L E I N F O

Article history:
Received 31 December 2020
Received in revised form 13 March 2021
Accepted 14 March 2021
Available online 24 March 2021

Keywords:
Stochastic trace estimation
Chebyshev polynomials
Spectral function
Hutchinson's method

A B S T R A C T

Methods for stochastic trace estimation often require the repeated evaluation of expressions of the form $z^T p_n(A)z$, where A is a symmetric matrix and p_n is a degree n polynomial written in the standard or Chebyshev basis. We show how to evaluate these expressions using only $\lceil n/2 \rceil$ matrix–vector products, thus substantially reducing the cost of existing trace estimation algorithms that use Chebyshev interpolation or Taylor series.

1. Introduction

Given a symmetric matrix $A \in \mathbb{R}^{d\times d}$ and a function $f : \mathbb{R} \to \mathbb{R}$, we consider the problem of estimating

$$\mathrm{tr}(f(A)) = \sum_{i=1}^{d} f(\lambda_i), \tag{1}$$

where $\lambda_1, \ldots, \lambda_d$ are the eigenvalues of A. When A is large enough to make computing its eigenvalues impractical, one common approach is to use *Hutchinson's method* [1]. This method samples m independent vectors $\{z^{(i)}\}_{i=1}^m$ from a Rademacher distribution (entries ± 1 with equal probability), and yields the estimate

$$\mathrm{tr}(f(A)) \approx \frac{1}{m}\sum_{i=1}^{m} \left(z^{(i)}\right)^T f(A) z^{(i)}. \tag{2}$$

For many functions f of interest (e.g., $\exp(x)$, x^{-1}, $x^{p/2}$, or $\log(x)$), the right hand side of (2) is further simplified by approximating f by a degree n polynomial p_n, most commonly through Chebyshev interpolation or a Taylor series.

E-mail address: erhallma@ncsu.edu.

[1] This research was supported in part by the National Science Foundation, United States of America through grant DMS-1745654.

https://doi.org/10.1016/j.aml.2021.107246

A common way to evaluate an expression of the form $z^T p_n(A)z$ is to compute $z_n = p_n(A)z$ and return $z^T z_n$ (see [2–4] for examples). We refer to this method as *one-sided evaluation*, and it will in general require n matrix–vector products (matvecs) with A. For polynomials written in the standard or Chebyshev bases we show how to reduce the number of matvecs to $\lceil n/2 \rceil$. Since the cost of many existing trace estimation algorithms is dominated by matvecs, our method will cut their runtime approximately in half.

1.1. Related work

A related idea is explored in [5] where the authors exploit the symmetry of A to reduce the cost of their estimator. Our proposal is more efficient and is more generally applicable, as their method applies only when $f(A)$ is positive definite.

The Chebyshev identity (6) that this paper relies on is also used in [6] for trace estimation problems. There, the authors show how to efficiently compute the products $\{p_n(A)z_j\}_{j=0}^m$ where $z_j = T_j(A)z_0$ and T_j is the jth Chebyshev polynomial.

For more general background on computing matrix polynomials, see [7, Ch. 4] or [8, Sec. 9.2]. Our method bears some resemblance to that of Paterson and Stockmeyer [9], but aims to compute $z^T p_n(A)z$ rather than $p_n(A)$ itself.

A few recent papers that use Chebyshev approximations for stochastic trace estimation are [2,3,5,10], and Taylor series are used similarly in [4]. For more applications of stochastic trace estimation, see [11].

One primary competitor to Chebyshev interpolation is stochastic Lanczos quadrature. For more information on this method and its pros and cons with respect to using Chebyshev polynomials, see [12]. In short, the authors suggest that Lanczos quadrature is generally superior since it converges at twice the rate of Chebyshev interpolation. If this is true, then our method (which does not to our knowledge extend to Lanczos quadrature) should put Chebyshev interpolation back on more or less equal footing. It may require an approximating polynomial with twice the degree of that needed by Lanczos, but can compute it with the same number of matvecs!

2. Standard basis

It is noted in [5] that expressions of the form $z^T A^n z$ can be evaluated with $\lceil n/2 \rceil$ matvecs by letting $k = \lfloor n/2 \rfloor$ and computing $z_k = A^k z$, then returning $z_k^T z_k$ if n is even and $z_k^T A z_k$ if n is odd. We first extend this idea to polynomials of the form $p_n(x) = \sum_{j=0}^n \alpha_j x^j$. Algorithm 1 requires $\lceil n/2 \rceil$ matvecs and at each step j needs to store only the two most recent vectors z_j, z_{j-1} in memory.

Algorithm 1 Two-sided evaluation (standard basis)

Input: Symmetric $A \in \mathbb{R}^{d \times d}$, $z_0 \in \mathbb{R}^d$, polynomial coefficients $a = [\alpha_0, \alpha_1, \ldots, \alpha_n]$
Output: $s = z_0^T p_n(A) z_0$

$s = \alpha_0 z_0^T z_0$
for $j = 1, 2, \ldots, \lceil n/2 \rceil$ **do**
  $z_j = A z_{j-1}$ $\{z_j = A^j z_0\}$
  $s = s + \alpha_{2j-1} z_{j-1}^T z_j$
  if $n = 2j - 1$ **then stop**
  $s = s + \alpha_{2j} z_j^T z_j$
end for

3. Chebyshev basis

We use Chebyshev polynomials of the first kind, which can be defined by the recurrence

$$T_{j+1}(x) = 2xT_j(x) - T_{j-1}(x), \tag{3}$$

where $T_0(x) = 1$ and $T_1(x) = x$. A function $f : [-1, 1] \to \mathbb{R}$ can then be approximated by a polynomial of the form

$$f(x) \approx p_n(x) = \sum_{j=0}^{n} \alpha_j T_j(x). \tag{4}$$

The polynomial p_n interpolates f at a set of Chebyshev nodes $\{x_j\}_{j=0}^{n}$. Several different choices for the nodes are available [13,14], but as one example Trefethen [13] uses the nodes

$$x_j = \cos \frac{j\pi}{n}, \quad 0 \le j \le n \tag{5}$$

and shows how to quickly compute the coefficients $\{\alpha_j\}_{j=0}^{n}$ by using an FFT. Our concern is only that the polynomial in (4) is expressed in the Chebyshev basis.

The key idea is to use the fact that Chebyshev polynomials follow the relation [15]

$$T_j(x)T_k(x) = \frac{1}{2}\left(T_{j+k}(x) + T_{|k-j|}(x)\right), \quad \forall j, k \ge 0. \tag{6}$$

By letting $k = j$ or $k = j + 1$ in the above equation and rearranging, it follows that for all $j \ge 0$,

$$T_{2j}(x) = 2T_j(x)^2 - T_0(x) = 2T_j(x)^2 - 1 \tag{7}$$

and

$$T_{2j+1}(x) = 2T_j(x)T_{j+1}(x) - T_1(x) = 2T_j(x)T_{j+1}(x) - x. \tag{8}$$

We can therefore evaluate terms of the form $z^T T_{2j}(A)z$ by computing $z_j = T_j(A)z$ and returning $2z_j^T z_j - z^T z$. Similarly, we can evaluate terms of the form $z^T T_{2j+1}(A)z$ by computing $z_{j+1} = T_{j+1}(A)z$ and returning $2z_j^T z_{j+1} - z^T Az$.

Our method is presented in Algorithm 2. It requires $\lceil n/2 \rceil$ matvecs and at each step j needs to store only the two most recent vectors z_j, z_{j-1} in memory. It should therefore take about half the time required by one-sided evaluation.

Algorithm 2 Two-sided evaluation (Chebyshev basis)

Input: Symmetric $A \in \mathbb{R}^{d \times d}$, $z_0 \in \mathbb{R}^d$, Chebyshev coefficients $a = [\alpha_0, \alpha_1, \dots, \alpha_n]$
Output: $s = z_0^T p_n(A) z_0$

$z_1 = Az_0$
$\zeta_0 = z_0^T z_0$
$\zeta_1 = z_0^T z_1$
$s = \alpha_0\zeta_0 + \alpha_1\zeta_1 + \alpha_2\left(2{z_1}^T z_1 - \zeta_0\right)$
for $j = 2, 3, \dots, \lceil n/2 \rceil$ **do**
  $z_j = 2(Az_{j-1}) - z_{j-2}$ $\{z_j = T_j(A)z_0\}$
  $s = s + \alpha_{2j-1}\left(2(z_{j-1}^T z_j) - \zeta_1\right)$
  if $n = 2j - 1$ **then stop**
  $s = s + \alpha_{2j}\left(2(z_j^T z_j) - \zeta_0\right)$
end for

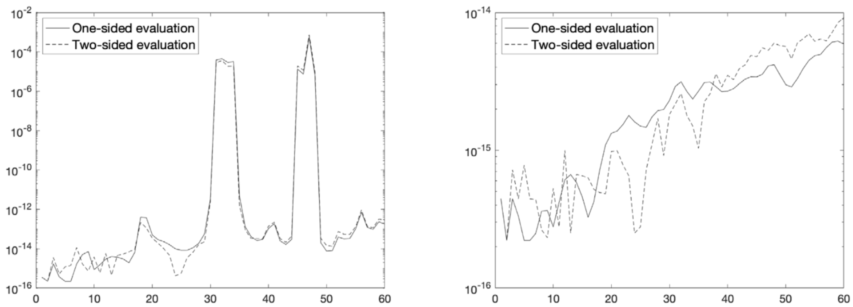

Fig. 1. Relative (left) and absolute (right) errors in the computed values of $z_0^T T_j(A) z_0$ from one-sided and two-sided evaluation.

3.1. Stability

The three-term recurrence of scalar Chebyshev polynomials is shown in [16] to satisfy a mixed forward–backward stability condition, and for vectors the Clenshaw algorithm is shown in [17] to be backward stable. The latter result implies that each individual vector $\{z_j\}_{j=0}^n$ produced by Algorithm 2 satisfies a backward stability condition, but it remains an open question whether Algorithm 2 is itself stable. In particular, the quantity computed in line 9 may be inaccurate if $2z_j^T z_j - \zeta_0$ is small compared to $2\|z_j\|_2^2 + \|z_0\|^2$, and similarly for line 7. We expect that the effect of these rounding errors will typically be minor, but leave a more rigorous analysis for future exploration.

4. Numerical experiments

In order to gain some insight into the behavior of Algorithm 2, we test it on two synthetic problems. In the first example, $A \in \mathbb{R}^{50\times 50}$ and $z_0 \in \mathbb{R}^{50}$ are chosen so that the quantities $z_0^T T_j(A) z_0$ are small for a few select values of j (31–34 and 45–48) even though z_0 and z_j have 2-norms close to 1. For each $j = 0 : 60$, we evaluate $z_0^T T_j(A) z_0$ using both one-sided and two-sided evaluation and compare these quantities to the results obtained using one-sided evaluation in extended precision.

Results are shown in Fig. 1. The two algorithms had more or less the same behavior, with the absolute error increasing slowly as a function of the degree j. As expected, the outputs from two-sided evaluation had a large relative error when $z_0^T T_j(A) z_0$ was small. More interestingly, the outputs from one-sided evaluation were just as inaccurate due to cancellation from computing the inner products $z_0^T z_j$.

In the second example, $A \in \mathbb{R}^{50\times 50}$ and $z_0 \in \mathbb{R}^{50}$ are chosen so that $\|T_j(A)z_0\|_2$ is small for $j = 35$ only—in particular, the matrix A is chosen to have several eigenvalues close to numbers of the form $\cos((k+1/2)\pi/35)$. One-sided and two-sided evaluation are again compared to one-sided evaluation in extended precision.

Results are shown in Fig. 2. Once again, the two algorithms had more or less the same behavior. This time, however, $|z_0^T z_j|$ was not much smaller than $\|z_0\|_2\|z_j\|_2$, so the error from one-sided evaluation did not come from cancellation in the inner product. Instead, the error arose from the computation of z_j itself: compared to the result produced using extended precision, the vector z_j computed in double precision had a relative 2-norm error of about 5×10^{-5}. By contrast, none of the other Chebyshev vectors had a relative error larger than 10^{-14}.

These experiments suggest that although it is possible to create cases where two-sided evaluation will have a large relative error, our algorithm appears to have roughly the same accuracy as one-sided evaluation. Although further investigation is required, we suspect that it is possible to show that Algorithm 2 satisfies a backward stability condition.

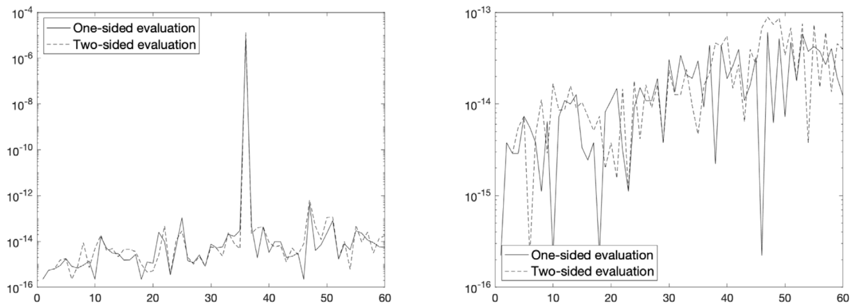

Fig. 2. Relative (left) and absolute (right) errors in the computed values of $z_0^T T_j(A) z_0$ from one-sided and two-sided evaluation.

5. Conclusion

We have shown how to evaluate the expression $z^T p_n(A) z$, where A is symmetric and p_n is a polynomial in the standard or Chebyshev basis, using no more than $\lceil n/2 \rceil$ matvecs with A. Our proposed method is simple to implement and can be used for any stochastic trace estimation technique that relies on Taylor expansions or Chebyshev interpolation. The stability of our method remains an open question, but numerical experiments suggest that its output will have accuracy comparable to that of standard one-sided evaluation. We therefore recommend that two-sided evaluation be incorporated into existing algorithms.

References

[1] M.F. Hutchinson, A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines, Comm. Statist. Simulation Comput. 18 (3) (1989) 1059–1076.
[2] I. Han, D. Malioutov, J. Shin, Large-scale log-determinant computation through stochastic Chebyshev expansions, in: International Conference on Machine Learning, 2015, pp. 908–917.
[3] I. Han, D. Malioutov, H. Avron, J. Shin, Approximating spectral sums of large-scale matrices using stochastic Chebyshev approximations, SIAM J. Sci. Comput. 39 (4) (2017) A1558–A1585.
[4] C. Boutsidis, P. Drineas, P. Kambadur, E.-M. Kontopoulou, A. Zouzias, A randomized algorithm for approximating the log determinant of a symmetric positive definite matrix, Linear Algebra Appl. 533 (2017) 95–117.
[5] E. Dudley, A.K. Saibaba, A. Alexanderian, Monte Carlo estimators for the Schatten p-norm of symmetric positive semidefinite matrices, arXiv preprint arXiv:2005.10174.
[6] C. Bekas, E. Kokiopoulou, Y. Saad, An estimator for the diagonal of a matrix, Appl. Numer. Math. 57 (11–12) (2007) 1214–1229.
[7] N.J. Higham, Functions of Matrices: Theory and Computation, SIAM, 2008.
[8] G.H. Golub, C.F. Van Loan, Matrix Computations, fourth ed., The Johns Hopkins University Press, Baltimore, 2013.
[9] M.S. Paterson, L.J. Stockmeyer, On the number of nonscalar multiplications necessary to evaluate polynomials, SIAM J. Comput. 2 (1) (1973) 60–66.
[10] E. Di Napoli, E. Polizzi, Y. Saad, Efficient estimation of eigenvalue counts in an interval, Numer. Linear Algebra Appl. 23 (4) (2016) 674–692.
[11] S. Ubaru, Y. Saad, Applications of trace estimation techniques, in: International Conference on High Performance Computing in Science and Engineering, Springer, 2017, pp. 19–33.
[12] S. Ubaru, J. Chen, Y. Saad, Fast estimation of $\mathrm{tr}(f(A))$ via stochastic Lanczos quadrature, SIAM J. Matrix Anal. Appl. 38 (4) (2017) 1075–1099.
[13] L.N. Trefethen, Is Gauss quadrature better than Clenshaw–Curtis? SIAM Rev. 50 (1) (2008) 67–87.
[14] S.E. Notaris, Interpolatory quadrature formulae with Chebyshev abscissae of the third or fourth kind, J. Comput. Appl. Math. 81 (1) (1997) 83–99.
[15] M. Abramowitz, I.A. Stegun, R.H. Romer, HandBook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, 1988.
[16] A. Smoktunowicz, A. Smoktunowicz, E. Pawelec, The three-term recursion for Chebyshev polynomials is mixed forward-backward stable, Numer. Algorithms 69 (4) (2015) 785–794.
[17] J.L. Aurentz, A.P. Austin, M. Benzi, V. Kalantzis, Stable computation of generalized matrix functions via polynomial interpolation, SIAM J. Matrix Anal. Appl. 40 (1) (2019) 210–234.