

Randomized Linear Algebra Approaches to Estimate the Von Neumann Entropy of Density Matrices

Eugenia-Maria Kontopoulou
Purdue University
Computer Science
West Lafayette, IN, USA
Email: ekontopo@purdue.edu

Ananth Grama
Purdue University
Computer Science
West Lafayette, IN, USA
Email: ayg@cs.purdue.edu

Wojciech Szpankowski
Purdue University
Computer Science
West Lafayette, IN, USA
Email: spa@cs.purdue.edu

Petros Drineas
Purdue University
Computer Science
West Lafayette, IN, USA
Email: pdrineas@purdue.edu

Abstract—The *von Neumann entropy*, named after John von Neumann, is the extension of classical entropy concepts to the field of quantum mechanics and, from a numerical perspective, can be computed simply by computing all the eigenvalues of a density matrix, an operation that could be prohibitively expensive for large-scale density matrices. We present and analyze two randomized algorithms to approximate the von Neumann entropy of density matrices: our algorithms leverage recent developments in the Randomized Numerical Linear Algebra (RandNLA) literature, such as randomized trace estimators, provable bounds for the power method, and the use of Taylor series and Chebyshev polynomials to approximate matrix functions. Both algorithms come with provable accuracy guarantees and our experimental evaluations support our theoretical findings showing considerable speedup with small accuracy loss.

I. INTRODUCTION

Entropy is a fundamental quantity in many areas of science and engineering. The *von Neumann entropy*, named after John von Neumann, is the extension of classical entropy concepts to the field of quantum mechanics, and its foundations can be traced to von Neumann’s work on *Mathematische Grundlagen der Quantenmechanik*¹. In his work, Von Neumann introduced the notion of the *density matrix*, which facilitated the extension of the tools of classical statistical mechanics to the quantum domain in order to develop a theory of quantum measurements.

From a mathematical perspective (see Section I-A for details) the density matrix \mathbf{R} is a symmetric positive semidefinite matrix in $\mathbb{R}^{n \times n}$ with unit trace. Let p_i , $i = 1 \dots n$ be the eigenvalues of \mathbf{R} , in decreasing order; then, the entropy of \mathbf{R} is defined as

$$\mathcal{H}(\mathbf{R}) = - \sum_{i=1}^n p_i \ln p_i. \quad (1)$$

The above definition is a proper extension of both the Gibbs entropy and the Shannon entropy to the quantum case and implies an obvious algorithm to compute $\mathcal{H}(\mathbf{R})$ by first computing the eigendecomposition of \mathbf{R} ; known algorithms for this task necessitate $\mathcal{O}(n^3)$ time [1]. Clearly, as n grows, such running times are impractical. For example, [2] describes an entangled two-photon state generated by spontaneous parametric down-conversion, which can result in a density matrix with $n \approx 10^8$.

¹Originally published in German in 1932; published in English under the title *Mathematical Foundations of Quantum Mechanics* in 1955.

Motivated by the above discussion, we seek numerical algorithms that approximate the von Neumann entropy of large density matrices, e.g., symmetric positive definite matrices with unit trace, faster than the trivial $\mathcal{O}(n^3)$ approach. Our algorithms build upon recent developments in the field of Randomized Numerical Linear Algebra (RandNLA), an interdisciplinary research area that exploits randomization as a computational resource to develop improved algorithms for large-scale linear algebra problems. Indeed, our work here focuses at the intersection of RandNLA and information theory, delivering novel randomized linear algebra algorithms and related quality-of-approximation results for a fundamental information-theoretic metric.

A. Background

We will focus on finite-dimensional function (state) spaces. In this setting, the density matrix \mathbf{R} represents the statistical mixture of pure states, and from a linear algebraic perspective can be written as

$$\mathbf{R} = \Psi \Sigma_p \Psi^T \in \mathbb{R}^{n \times n}, \quad (2)$$

where $\Psi \in \mathbb{R}^{n \times k}$ is the orthonormal matrix whose columns are the k pure states and $\Sigma_p \in \mathbb{R}^{k \times k}$ is a diagonal matrix whose entries are the (positive) p_i ’s. Let $h(x) = x \ln x$ for any $x > 0$ and let $h(0) = 0$. Then, using the cyclical property of the trace and the definition of $h(x)$ eqn. (1) becomes

$$- \sum_{i, p_i > 0} p_i \ln p_i = -\text{tr}(\Psi h(\Sigma_p) \Psi^T) = -\text{tr}(h(\mathbf{R})) = -\text{tr}(\mathbf{R} \ln \mathbf{R}). \quad (3)$$

The second equality follows from the definition of matrix functions [3].

We conclude the section by noting that our algorithms will use two tools that appeared in prior work. The first tool is the power method, with a provable analysis that first appeared in [4]. The second tool is a provably accurate trace estimation algorithm for symmetric positive semidefinite matrices that appeared in [5].

B. Our contributions

We present and analyze two randomized algorithms to approximate the von Neumann entropy of density matrices, leveraging two different polynomial approximations of the matrix function $\mathcal{H}(\mathbf{R}) = -\text{tr}(\mathbf{R} \ln \mathbf{R})$: the first approximation uses a

Taylor series expansion while the second approximation uses Chebyshev polynomials. Both algorithms return, with high probability, relative-error approximations to the true entropy of the input density matrix, under certain assumptions. More specifically, in both cases, we need to assume that the input density matrix has n non-zero eigenvalues, or, equivalently, that the probabilities p_i , $i = 1 \dots n$, corresponding to the underlying n pure states are non-zero. The running time of both algorithms is proportional to the sparsity of the input density matrix and depends (see Theorems 1 and 3 for precise statements) on, roughly, the ratio of the largest to the smallest probability p_1/p_n , as well as the desired accuracy.

From a technical perspective, the theoretical analysis of both algorithms proceeds by combining the power of polynomial approximations, either using Taylor series or Chebyshev polynomials, to matrix functions, combined with randomized trace estimators. A provably accurate variant of the power method is used to estimate the largest probability p_1 . If this estimate is significantly smaller than one, it can improve the running times of the proposed algorithms (see discussion after Theorem 1). We note that while the power method introduces an additional $\text{nnz}(\mathbf{R}) \log(n)$ term in the running time, in practice, its computational overhead is negligible.

Finally, in Section IV, we present a highlight of our experimental evaluations of our algorithms on large-scale synthetic density matrices, generated using Matlab's QETLAB toolbox [6]. For a 30,000 by 30,000 matrix that was used in our evaluations, the exact computation of the entropy takes hours, whereas our algorithms return approximations with relative errors well below 0.5% in only a few minutes.

We conclude by noting that a longer version of this short abstract is available in [7]. Indeed, we will often reference [7] for omitted proofs, omitted experimental evaluations, and detailed discussions that are omitted from this version due to space considerations.

C. Prior work

To the best of our knowledge, prior to this work, the only non-trivial algorithm to approximate the von Neumann entropy of a density matrix appeared in [2]. Their approach is based on the Chebyshev polynomial expansion that we also analyze in Section III. However, our analysis is somewhat different and, overall, much tighter, leveraging a provably accurate variant of the power method as well as provably accurate trace estimators to derive a relative error approximation to the entropy of a density matrix, under appropriate assumptions. A brief technical comparison between our results in Section III and the work of [2] appears in Section III-A, while a detailed comparison can be found in Section 3.3 of [7].

Independently and in parallel with our work, [8] presented a multipoint interpolation algorithm (building upon [9]) to compute a relative error approximation for the entropy of a real matrix with bounded condition number. The proposed running time of Theorem 35 of [8] does not depend on the condition number of the input matrix (i.e., the ratio of the largest to the smallest probability), which is a clear advantage in the case

of ill-conditioned matrices. However, the dependency of the algorithm of Theorem 35 of [8] on terms like $(\log n/\epsilon)^6$ or $n^{1/3} \text{nnz}(\mathbf{A}) + n\sqrt{\text{nnz}(\mathbf{A})}$ could blow up the running time of the proposed algorithm for reasonably conditioned matrices.

We also note the recent work in [10], which used Taylor approximations to matrix functions to estimate the log determinant of symmetric positive definite matrices (see also Section 1.2 of [10] for an overview of prior work on approximating matrix functions via Taylor series). Finally, the work of [11] used a Chebyshev polynomial approximation to estimate the log determinant of a matrix and is reminiscent of our approach in Section III and, of course, the work of [2].

II. AN APPROACH VIA TAYLOR SERIES

Our first approach to approximate the von Neumann entropy of a density matrix uses a Taylor series expansion to approximate the logarithm of a matrix, combined with a relative-error trace estimator for symmetric positive semi-definite matrices and the power method to upper bound the largest singular value of a matrix.

A. Algorithm and Main Theorem

Our main result is an analysis of Algorithm 1 (see below) that guarantees relative error approximation to the entropy of the density matrix \mathbf{R} , under the assumption that $\mathbf{R} = \sum_{i=1}^n p_i \psi_i \psi_i^T \in \mathbb{R}^{n \times n}$ has n pure states with $0 < \ell \leq p_i$ for all $i = 1 \dots n$. The following theorem is our main quality-

Algorithm 1 A Taylor series approach to estimate the entropy.

- 1: **INPUT:** $\mathbf{R} \in \mathbb{R}^{n \times n}$, accuracy parameter $\epsilon > 0$, failure probability δ , and integer $m > 0$.
- 2: Estimate \tilde{p}_1 using the power method of [4] with $t = \mathcal{O}(\ln n)$ and $q = \mathcal{O}(\ln(1/\delta))$.
- 3: Set $u = \min\{1, 6\tilde{p}_1\}$.
- 4: Set $s = \lceil 20 \ln(2/\delta)/\epsilon^2 \rceil$.
- 5: Let $\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_s \in \mathbb{R}^n$ be i.i.d. random Gaussian vectors.
- 6: **OUTPUT:** return

$$\widehat{\mathcal{H}}(\mathbf{R}) = \ln u^{-1} + \frac{1}{s} \sum_{i=1}^s \sum_{k=1}^m \frac{\mathbf{g}_i^T \mathbf{R} (\mathbf{I}_n - u^{-1} \mathbf{R})^k \mathbf{g}_i}{k}.$$

of-approximation result for Algorithm 1.

Theorem 1. *Let \mathbf{R} be a density matrix such that all probabilities p_i , $i = 1 \dots n$ satisfy $0 < \ell \leq p_i$. Let u be computed as in Algorithm 1 and let $\widehat{\mathcal{H}}(\mathbf{R})$ be the output of Algorithm 1 on inputs \mathbf{R} , m , and $\epsilon < 1$; Then, with probability at least $1 - 2\delta$,*

$$\left| \widehat{\mathcal{H}}(\mathbf{R}) - \mathcal{H}(\mathbf{R}) \right| \leq 2\epsilon \mathcal{H}(\mathbf{R}),$$

by setting $m = \lceil \frac{u}{\ell} \ln \frac{1}{\epsilon} \rceil$. The algorithm runs in time

$$\mathcal{O}\left(\frac{u}{\ell} \cdot \frac{\ln(1/\epsilon) \ln(1/\delta)}{\epsilon^2} \cdot \text{nnz}(\mathbf{R}) + \ln(n) \cdot \ln(1/\delta) \cdot \text{nnz}(\mathbf{R})\right).$$

A few remarks are necessary to better understand the above theorem. First, ℓ could be set to p_n , the smallest of the

probabilities corresponding to the n pure states of the density matrix \mathbf{R} . Second, it should be obvious that u in Algorithm 1 could be simply set to one and thus we could avoid calling the power method to estimate p_1 by \tilde{p}_1 and thus compute u . However, if p_1 is small, then u could be significantly smaller than one, thus reducing the running time of Algorithm 1, which depends on the ratio u/ℓ . Third, ideally, if both p_1 and p_n were used instead of u and ℓ , respectively, the running time of the algorithm would scale with the ratio p_1/p_n . Fourth, even though the overhead of the power method (the second term in the running time expression) appears to dominate, at least asymptotically, it is negligible in practice.

B. Proof of Theorem 1

We now prove Theorem 1, which analyzes the performance of Algorithm 1. Our first lemma presents a simple expression for $\mathcal{H}(\mathbf{R})$ using a Taylor series expansion.

Lemma 2. *Let $\mathbf{R} \in \mathbb{R}^{n \times n}$ be a symmetric positive definite matrix with unit trace and whose eigenvalues lie in the interval $[\ell, u]$, for some $0 < \ell \leq u \leq 1$. Then,*

$$\mathcal{H}(\mathbf{R}) = \ln u^{-1} + \sum_{k=1}^{\infty} \frac{\text{tr}(\mathbf{R}(\mathbf{I}_n - u^{-1}\mathbf{R})^k)}{k}.$$

Proof. The proof follows by further manipulating eqn. (3) and then applying the Taylor expansion of $\ln(\mathbf{I}_n - u^{-1}\mathbf{R}) = -\sum_{k=1}^{\infty} u^{-1}\mathbf{R}^k/k$. See [7] for the full proof. \square

We now proceed to prove Theorem 1. We will condition our analysis on power method algorithm being successful, which happens with probability at least $1 - \delta$. In this case, $u = \min\{1, 6\tilde{p}_1\}$ is an upper bound for all probabilities p_i . For notational convenience, set $\mathbf{C} = \mathbf{I}_n - u^{-1}\mathbf{R}$. We start by manipulating $\Delta = \left| \widehat{\mathcal{H}}(\mathbf{R}) - \mathcal{H}(\mathbf{R}) \right|$ as follows:

$$\begin{aligned} \Delta &= \left| \sum_{k=1}^m \frac{1}{k} \cdot \frac{1}{s} \sum_{i=1}^s \mathbf{g}_i^\top \mathbf{R}^k \mathbf{g}_i - \sum_{k=1}^{\infty} \frac{1}{k} \text{tr}(\mathbf{R}^k) \right| \\ &\leq \left| \sum_{k=1}^m \frac{1}{k} \cdot \frac{1}{s} \sum_{i=1}^s \mathbf{g}_i^\top \mathbf{R}^k \mathbf{g}_i - \sum_{k=1}^m \frac{1}{k} \text{tr}(\mathbf{R}^k) \right| + \left| \sum_{k=m+1}^{\infty} \frac{1}{k} \text{tr}(\mathbf{R}^k) \right| \\ &= \underbrace{\left| \frac{1}{s} \sum_{i=1}^s \mathbf{g}_i^\top \left(\sum_{k=1}^m \mathbf{R}^k / k \right) \mathbf{g}_i - \text{tr} \left(\sum_{k=1}^m \frac{1}{k} \mathbf{R}^k \right) \right|}_{\Delta_1} + \underbrace{\left| \sum_{k=m+1}^{\infty} \text{tr}(\mathbf{R}^k) / k \right|}_{\Delta_2}. \end{aligned}$$

We now bound the two terms Δ_1 and Δ_2 separately. We start with Δ_1 : the idea is to apply Theorem 5.2 from [5] on the matrix $\sum_{k=1}^m \mathbf{R}^k/k$ with $s = \lceil 20 \ln(2/\delta)/\epsilon^2 \rceil$. Hence, with probability at least $1 - \delta$:

$$\Delta_1 \leq \epsilon \cdot \text{tr} \left(\sum_{k=1}^m \mathbf{R}^k / k \right) \leq \epsilon \cdot \text{tr} \left(\sum_{k=1}^{\infty} \mathbf{R}^k / k \right). \quad (4)$$

A subtle point in applying Theorem 5.2 from [5] is that the matrix $\sum_{k=1}^m \mathbf{R}^k/k$ must be symmetric positive semidefinite. To prove this, let the SVD of \mathbf{R} be $\mathbf{R} = \Psi \Sigma_p \Psi^T$, where all three matrices are in $\mathbb{R}^{n \times n}$ and the diagonal entries of Σ_p are in the interval $[\ell, u]$. Then, it is easy to see that $\mathbf{C} = \mathbf{I}_n - u^{-1}\mathbf{R} = \Psi(\mathbf{I}_n - u^{-1}\Sigma_p)\Psi^T$ and $\mathbf{R}^k = \Psi \Sigma_p (\mathbf{I}_n - u^{-1}\Sigma_p)^k \Psi^T$, where the diagonal entries of $\mathbf{I}_n - u^{-1}\Sigma_p$ are non-negative, since the largest entry in Σ_p is upper bounded by u . This proves that \mathbf{R}^k is symmetric

positive semidefinite for any k , a fact which will be useful throughout the proof. Now,

$$\sum_{k=1}^m \mathbf{R}^k / k = \Psi \left(\Sigma_p \sum_{k=1}^m (\mathbf{I}_n - u^{-1}\Sigma_p)^k / k \right) \Psi^T,$$

which shows that the matrix of interest is symmetric positive semidefinite. Additionally, since \mathbf{R}^k is symmetric positive semidefinite, its trace is non-negative, which proves the second inequality in eqn. (4) as well. We proceed to bound Δ_2 as follows:

$$\begin{aligned} \Delta_2 &= \left| \sum_{k=m+1}^{\infty} \text{tr}(\mathbf{R}^k) / k \right| = \left| \sum_{k=m+1}^{\infty} \text{tr}(\mathbf{R}^m \mathbf{C}^{k-m}) / k \right| \\ &= \left| \sum_{k=m+1}^{\infty} \text{tr}(\mathbf{C}^m \mathbf{C}^{k-m} \mathbf{R}) / k \right| \leq \left| \sum_{k=m+1}^{\infty} \|\mathbf{C}^m\|_2 \cdot \text{tr}(\mathbf{C}^{k-m} \mathbf{R}) / k \right| \quad (5) \\ &= \|\mathbf{C}^m\|_2 \cdot \left| \sum_{k=m+1}^{\infty} \text{tr}(\mathbf{R}^k) / k \right| \leq \|\mathbf{C}^m\|_2 \cdot \left| \sum_{k=1}^{\infty} \text{tr}(\mathbf{R}^k) / k \right| \quad (6) \\ &\leq \left(1 - \frac{\ell}{u}\right)^m \sum_{k=1}^{\infty} \text{tr}(\mathbf{R}^k) / k. \quad (7) \end{aligned}$$

To prove eqn. (5), we used von Neumann's trace inequality². Eqn. (5) now follows since $\mathbf{C}^{k-m}\mathbf{R}$ is symmetric positive semidefinite³. To prove eqn. (6), we used the fact that $\text{tr}(\mathbf{R}^k) / k \geq 0$ for any $k \geq 1$. Finally, to prove eqn. (7), we used the fact that $\|\mathbf{C}\|_2 = \|\mathbf{I}_n - u^{-1}\Sigma_p\|_2 \leq 1 - \ell/u$ since the smallest entry in Σ_p is at least ℓ by our assumptions. We also removed unnecessary absolute values since $\text{tr}(\mathbf{R}^k) / k$ is non-negative for any positive integer k .

Combining the bounds for Δ_1 and Δ_2 gives

$$\left| \widehat{\mathcal{H}}(\mathbf{R}) - \mathcal{H}(\mathbf{R}) \right| \leq \left(\epsilon + \left(1 - \frac{\ell}{u}\right)^m \right) \sum_{k=1}^{\infty} \frac{\text{tr}(\mathbf{R}^k)}{k}.$$

We have already proven in Lemma 2 that

$$\sum_{k=1}^{\infty} \frac{\text{tr}(\mathbf{R}^k)}{k} \leq \mathcal{H}(\mathbf{R}) - \ln u^{-1} \leq \mathcal{H}(\mathbf{R}),$$

where the last inequality follows since $u \leq 1$. Collecting our results, we get

$$\left| \widehat{\mathcal{H}}(\mathbf{R}) - \mathcal{H}(\mathbf{R}) \right| \leq \left(\epsilon + \left(1 - \frac{\ell}{u}\right)^m \right) \mathcal{H}(\mathbf{R}).$$

Setting

$$m = \left\lceil \frac{u}{\ell} \ln \frac{1}{\epsilon} \right\rceil$$

and using $(1 - x^{-1})^x \leq e^{-1}$ ($x > 0$), guarantees that $(1 - \ell/u)^m \leq \epsilon$ and concludes the proof of the theorem. We note that the failure probability of the algorithm is at most 2δ (the sum of the failure probabilities of the power method and the trace estimation algorithm). Finally, we discuss the running time of Algorithm 1, which is equal to $\mathcal{O}(s \cdot m \cdot \text{nnz}(\mathbf{R}))$.

Since $s = \mathcal{O}\left(\frac{\ln(1/\delta)}{\epsilon^2}\right)$ and $m = \mathcal{O}\left(\frac{u \ln(1/\epsilon)}{\ell}\right)$, the running

²Indeed, for any two matrices \mathbf{A} and \mathbf{B} , $\text{tr}(\mathbf{A}\mathbf{B}) \leq \sum_i \sigma_i(\mathbf{A})\sigma_i(\mathbf{B})$, where $\sigma_i(\mathbf{A})$ (respectively $\sigma_i(\mathbf{B})$) denotes the i -th singular value of \mathbf{A} (respectively \mathbf{B}). Since $\|\mathbf{A}\|_2 = \sigma_1(\mathbf{A})$ (its largest singular value), this implies that $\text{tr}(\mathbf{A}\mathbf{B}) \leq \|\mathbf{A}\|_2 \sum_i \sigma_i(\mathbf{B})$; if \mathbf{B} is symmetric positive semidefinite, $\text{tr}(\mathbf{B}) = \sum_i \sigma_i(\mathbf{B})$.

³This can be proven using an argument similar to the one used to prove eqn. (4).

time becomes (after accounting for the running time of the power method)

$$\mathcal{O}\left(\frac{u}{\ell} \cdot \frac{\ln(1/\epsilon) \ln(1/\delta)}{\epsilon^2} \cdot \text{nnz}(\mathbf{R}) + \ln(n) \cdot \ln(1/\delta) \cdot \text{nnz}(\mathbf{R})\right).$$

III. AN APPROACH VIA CHEBYSHEV POLYNOMIALS

Our second approach is to use a Chebyshev polynomial-based approximation scheme to estimate the entropy of a density matrix. Our approach follows the work of [2], but our analysis uses the trace estimators of [5] and the power method of [4] and its analysis. Importantly, we present conditions under which the proposed approach is competitive with the approach of Section II.

A. Algorithm and Main Theorem

The proposed algorithm leverages the fact that the von Neumann entropy of a density matrix \mathbf{R} is equal to the (negative) trace of the matrix function $\mathbf{R} \ln \mathbf{R}$ and approximates the function $\mathbf{R} \ln \mathbf{R}$ by a sum of Chebyshev polynomials; then, the trace of the resulting matrix is estimated using the trace estimator of [5].

Let $f_m(x) = \sum_{w=0}^m \alpha_w \mathcal{T}_w(x)$ with $\alpha_0 = \frac{u}{2} (\ln \frac{u}{4} + 1)$, $\alpha_1 = \frac{u}{4} (2 \ln \frac{u}{4} + 3)$, and $\alpha_w = \frac{(-1)^w u}{w^3 - w}$ for $w \geq 2$. Let $\mathcal{T}_w(x) = \cos(w \cdot \arccos((2/u)x - 1))$ and $x \in [0, u]$ be the Chebyshev polynomials of the first kind for any integer $w > 0$. Algorithm 2 computes u (an upper bound estimate for the largest probability p_1 of the density matrix \mathbf{R}) and then computes $f_m(\mathbf{R})$ and estimates its trace. We note that the computation $\mathbf{g}_i^\top f_m(\mathbf{R}) \mathbf{g}_i$ can be done efficiently using Clenshaw's algorithm; see Appendix C of [7] for the well-known approach.

Algorithm 2 A Chebyshev polynomial-based approach to estimate the entropy.

- 1: **INPUT:** $\mathbf{R} \in \mathbb{R}^{n \times n}$, accuracy parameter $\epsilon > 0$, failure probability δ and integer $m > 0$.
 - 2: Estimate \tilde{p}_1 using the power method of [4] with $t = \mathcal{O}(\ln n)$ and $q = \mathcal{O}(\ln(1/\delta))$.
 - 3: Set $u = \min\{1, 6\tilde{p}_1\}$.
 - 4: Set $s = \lceil 20 \ln(2/\delta) / \epsilon^2 \rceil$.
 - 5: Let $\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_s \in \mathbb{R}^n$ be i.i.d. random Gaussian vectors.
 - 6: **OUTPUT:** $\hat{\mathcal{H}}(\mathbf{R}) = -\frac{1}{s} \sum_{i=1}^s \mathbf{g}_i^\top f_m(\mathbf{R}) \mathbf{g}_i$.
-

Our main result is an analysis of Algorithm 2 that guarantees a relative error approximation to the entropy of the density matrix \mathbf{R} , under the assumption that $\mathbf{R} = \sum_{i=1}^n p_i \psi_i \psi_i^\top \in \mathbb{R}^{n \times n}$ has n pure states with $0 < \ell \leq p_i$ for all $i = 1 \dots n$. The following theorem is our main quality-of-approximation result for Algorithm 2; a detailed proof of Theorem 3 may be found in [7].

Theorem 3. *Let \mathbf{R} be a density matrix such that all probabilities p_i , $i = 1 \dots n$ satisfy $0 < \ell \leq p_i$. Let u be computed as in Algorithm 1 and let $\hat{\mathcal{H}}(\mathbf{R})$ be the output of Algorithm 2*

on inputs \mathbf{R} , m , and $\epsilon < 1$; Then, with probability at least $1 - 2\delta$,

$$\left| \hat{\mathcal{H}}(\mathbf{R}) - \mathcal{H}(\mathbf{R}) \right| \leq 3\epsilon \mathcal{H}(\mathbf{R}),$$

by setting $m = \sqrt{\frac{u}{2\epsilon \ell \ln(1/(1-\ell))}}$. The algorithm runs in time

$$\mathcal{O}\left(\sqrt{\frac{u}{\ell \ln(1/(1-\ell))}} \cdot \frac{\ln(1/\delta)}{\epsilon^{2.5}} \cdot \text{nnz}(\mathbf{R}) + \ln(n) \cdot \ln(1/\delta) \cdot \text{nnz}(\mathbf{R})\right).$$

The similarities between Theorems 1 and 3 are obvious: same assumptions and directly comparable accuracy guarantees. The only difference is in the running times: the Taylor series approach has a milder dependency on ϵ , while the Chebyshev-based approximation has a milder dependency on the ratio u/ℓ , which controls the behavior of the probabilities p_i . We also note that the discussion following Theorem 1 is also applicable here.

We conclude the section with a brief comparison of our bound in Theorem 3 with the results of [2]; a detailed comparison of the two bounds may be found in Section 3.3 of [7]. The work of [2] culminates to the error bounds described in Theorem 4.3 (and the ensuing discussion). Unfortunately, without imposing a lower bound assumption on the p_i s it is difficult to get a meaningful error bound and an efficient algorithm. Indeed, [2] needs an efficient trace estimation procedure for the matrix $-f_m(\mathbf{R})$. However, while this matrix is always symmetric, it is not necessarily positive or negative semidefinite (unless additional assumptions are imposed on the p_i s, like we did in Theorem 3). Unfortunately, we are not aware of any provably accurate, relative error approximation algorithms for the trace of just *symmetric* matrices: the results of [5], [12] only apply to symmetric positive (or negative) semidefinite matrices. The work of [2] does provide an analysis of a trace estimator for general symmetric matrices (pioneered by Hutchinson in [13]). However, in our notation, in order to achieve a relative error bound, the final error bound of [2] (see eqns. (19) and (20) in [2]), could necessitate setting s (the number of random vectors in Algorithm 2) to a prohibitively large value, thus leading to running times that blow up to $\mathcal{O}(n^2 \text{nnz}(\mathbf{R}))$, which could easily exceed the trivial $\mathcal{O}(n^3)$ running time to exactly compute $\mathcal{H}(\mathbf{R})$. See Section 3.3 of [7] for details.

IV. EXPERIMENTS

A detailed experimental evaluation of the proposed algorithms may be found in Section 4 of [7]. Here, we only highlight one of our observations on a random density matrix of size $30,000 \times 30,000$ in order to prove the practical efficiency of our algorithms. Our algorithms were implemented in MatLab; to be precise, we used MATLAB R2016a on a (dedicated) node with two 10-Core Intel Xeon-E5 processors (2.60GHz) and 512 GBs of RAM. We generated our random density matrices using the QETLAB Matlab toolbox [6]. We computed the exact singular values of the matrix (and thus the entropy) using the `svd` function of MatLab. The accuracy of our algorithms was evaluated by measuring the relative error.

We set the parameters m and s to take values in the sets $\{5, 10, 15, 20\}$ and $\{50, 100, 200\}$, respectively. The parameter u was set to λ_{\max} , and was computed using the power method (see [7] for details) in approximately 3.6 minutes. We report the relative error (out of 100%) in Figure 1.

We observe that the relative error is always less than 1% for both methods, with the Chebyshev approximation yielding almost always slightly better results. Note that our Chebyshev-polynomial-based approximation algorithm significantly outperformed the exact computation: e.g., for $m = 5$ and $s = 50$, our estimate was computed in less than ten minutes and achieved less than .2% relative error, whereas the exact computations of the Von-Neumann entropy took approximately 5.6 hours. Finally, we report the wall-clock times (in minutes) in Figure 2. We note that for both algorithms and all combinations of the parameters, the approximation of the Von-Neumann entropy was computed in less than one hour.

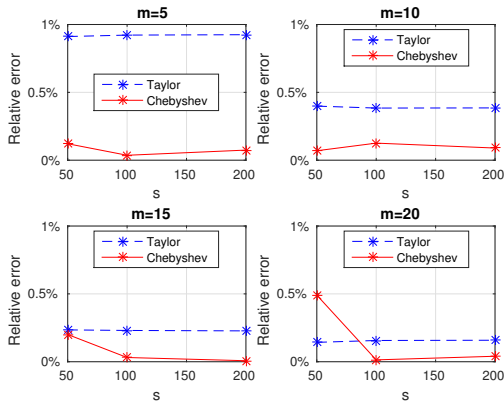


Fig. 1. Relative error for $30,000 \times 30,000$ density matrix using the Taylor and the Chebyshev approximation algorithms with $u = \lambda_{\max}$.

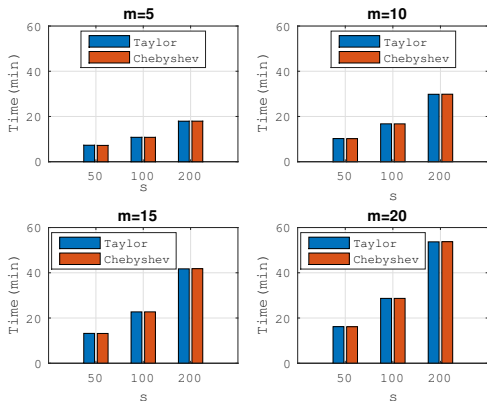


Fig. 2. Wall-clock times: Taylor approximation (blue) and Chebyshev approximation (red) for $u = \lambda_{\max}$. Exact computation needed approximately 5.6 hours.

V. CONCLUSIONS AND OPEN PROBLEMS

We presented and analyzed two randomized algorithms to approximate the von Neumann entropy of density ma-

trices. Our algorithms leverage recent developments in the RandNLA literature: randomized trace estimators, provable bounds for the power method, and the use of Taylor series and Chebyshev polynomials to approximate matrix functions. Both algorithms come with provable accuracy guarantees under assumptions on the spectrum of the density matrix. Empirical evaluations on $30,000 \times 30,000$ synthetic density matrices support our theoretical findings and demonstrate that we can efficiently approximate the von Neumann entropy in a few minutes with minimal loss in accuracy, whereas the exact computation takes over 5.5 hours. An important open problem is to relax (or eliminate) the assumptions associated with our key technical results. It would be critical to understand whether our assumptions are, for example, necessary to achieve relative error approximations and either provide algorithmic results that relax or eliminate our assumptions or provide matching lower bounds and counterexamples.

ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their suggestions and for bringing Theorem 35 of [8] to our attention.

REFERENCES

- [1] G. H. Golub and C. F. Van Loan, *Matrix Computations (3rd Ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 1996.
- [2] T. P. Wihler, B. Bessire, and A. Stefanov, "Computing the Entropy of a Large Matrix," *Journal of Physics A: Mathematical and Theoretical*, vol. 47, no. 24, p. 245201, 2014.
- [3] N. J. Higham, *Functions of Matrices: Theory and Computation*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2008.
- [4] L. Trevisan, "Graph Partitioning and Expanders," 2011, handout 7.
- [5] H. Avron and S. Toledo, "Randomized Algorithms for Estimating the Trace of an Implicit Symmetric Positive Semi-definite Matrix," *Journal of the ACM*, vol. 58, no. 2, p. 8, 2011.
- [6] N. Johnston, "QETLAB: A Matlab toolbox for quantum entanglement, version 0.9," <http://qetlab.com>, 2016.
- [7] E.-M. Kontopoulou, A. Grama, W. Szpankowski, and P. Drineas, "Randomized Linear Algebra Approaches to Estimate the Von Neumann Entropy of Density Matrices," 2018. [Online]. Available: <http://arxiv.org/abs/1801.01072>
- [8] C. Musco, P. Netrapalli, A. Sidford, S. Ubaru, and D. P. Woodruff, "Spectrum Approximation Beyond Fast Matrix Multiplication: Algorithms and Hardness," 2018. [Online]. Available: <http://arxiv.org/abs/1704.04163>
- [9] N. J. Harvey, J. Nelson, and K. Onak, "Sketching and streaming entropy via approximation theory," in *IEEE Annual Symposium on Foundations of Computer Science*, 2008, pp. 489–498.
- [10] C. Boutsidis, P. Drineas, P. Kambadur, E.-M. Kontopoulou, and A. Zouzias, "A Randomized Algorithm for Approximating the Log Determinant of a Symmetric Positive Definite Matrix," *Linear Algebra and its Applications*, vol. 533, pp. 95–117, 2017.
- [11] I. Han, D. Malioutov, and J. Shin, "Large-scale Log-determinant Computation through Stochastic Chebyshev Expansions," *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 908–917, 2015.
- [12] F. Roosta-Khorasani and U. Ascher, "Improved Bounds on Sample Size for Implicit Matrix Trace Estimators," *Foundations of Computational Mathematics*, vol. 15, pp. 1187–1212, 2015.
- [13] M. F. Hutchinson, "A Stochastic Estimator of the Trace of the Influence Matrix for Laplacian Smoothing Splines," *Communications in Statistics: Simulation and Computation*, vol. 19, no. 2, pp. 433–450, 1990.