Optimal Randomized Approximations for Matrix-Based Rényi's Entropy

Yuxin Dong, Tieliang Gong, Shujian Yu, and Chen Li

***Abstract*— The Matrix-based Rényi's entropy enables us to directly measure information quantities from given data without the costly probability density estimation of underlying distributions, thus has been widely adopted in numerous statistical learning and inference tasks. However, exactly calculating this new information quantity requires access to the eigenspectrum of a semi-positive definite (SPD) matrix A which grows linearly with the number of samples n, resulting in a $O(n^3)$ time complexity that is prohibitive for large-scale applications. To address this issue, this paper takes advantage of stochastic trace approximations for matrix-based Rényi's entropy with arbitrary $\alpha \in \mathbb{R}^+$ orders, lowering the complexity by converting the entropy approximation to a matrix-vector multiplication problem. Specifically, we develop random approximations for integer-order α cases and polynomial series approximations (Taylor and Chebyshev) for fractional α cases, leading to a $O(n^2sm)$ overall time complexity, where $s, m \ll n$ denote the number of vector queries and the polynomial order respectively. We theoretically establish statistical guarantees for all approximation algorithms and give explicit order of s and m with respect to the approximation error ϵ, showing optimal convergence rate for both parameters up to a logarithmic factor. Large-scale simulations and real-world applications validate the effectiveness of the developed approximations, demonstrating remarkable speedup with negligible loss in accuracy.**

***Index Terms*— Matrix-based Rényi's entropy, randomized numerical linear algebra, trace estimation, polynomial approximation, mutual information.**

Manuscript received 22 May 2022; revised 19 January 2023; accepted 17 March 2023. Date of publication 21 March 2023; date of current version 16 June 2023. This work was supported in part by the Key Research and Development Program of China under Grant 2021ZD0110700; in part the National Natural Science Foundation of China under Grant 62106191; in part by the Key Research and Development Program of Shaanxi Province under Grant 2021GXLH-Z-095; in part by the Innovative Research Group of the National Natural Science Foundation of China under Grant 61721002; in part by the Consulting Research Project of the Chinese Academy of Engineering, The Online and Offline Mixed Educational Service System for The Belt and Road Training in MOOC China; in part by the Project of China Knowledge Centre for Engineering Science and Technology; and in part by the Innovation Team from the Ministry of Education under Grant IRT_17R86. *(Corresponding author: Tieliang Gong.)*

Yuxin Dong, Tieliang Gong, and Chen Li are with the Key Laboratory of Intelligent Networks and Network Security and School of Computer Science and Technology, Ministry of Education, Xi'an 710049, China (e-mail: dongyuxin@stu.xjtu.edu.cn; adidasgtl@gmail.com; Cli@xjtu.edu.cn).

Shujian Yu is with the Department of Computer Science, Vrije Universiteit Amsterdam, 1081 HV Amsterdam, The Netherlands, and also with the Machine Learning Group, UiT—The Arctic University of Norway, 9019 Tromsø, Norway (e-mail: yusj9011@gmail.com).

Communicated by A. Mazumdar, Associate Editor for Machine Learning and Statistics.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIT.2023.3260122.

Digital Object Identifier 10.1109/TIT.2023.3260122

I. Introduction

THE Rényi's α-order entropy, introduced by Alfred Rényi [1], serves as a one-parameter generalization of the well-known Shannon's entropy. Following Rényi's work, extensive studies have been conducted in machine learning and statistical inference tasks, demonstrating elegant properties and impressive scalability [2], [3], [4], [5]. However, its heavy dependence on the underlying data distributions makes the estimation of high-dimensional probability density functions (PDF) inevitable, which is especially expensive or even intractable due to the curse of high-dimensionality [6].

Recently, the matrix-based Rényi's entropy [4], [7] is introduced as a substitution that can be quantified directly from given data samples. Inspired by the quantum generalization of Rényi's definition [8], this new family of information measures is defined on the eigenspectrum of a normalized Hermitian matrix constructed by projecting data points in reproducing kernel Hilbert space (RKHS), thus avoiding explicit estimation of underlying data distributions. Because of its intriguing property in high-dimensional scenarios, the matrix-based Rényi's entropy, and mutual information have been successfully applied in various data science applications, ranging from classical dimensionality reduction [9] and feature selection [10] problems to advanced deep learning problems such as robust learning against covariant shift [5], network pruning [11] and knowledge distillation [12].

Nevertheless, calculating this new information measure requires complete knowledge about the eigenspectrum of a Gram matrix, whose size grows linearly with the number of samples n, resulting in a $O(n^3)$ time complexity with traditional eigenvalue algorithms including eigenvalue decomposition, singular value decomposition, CUR decomposition, and QR factorization [13], [14], greatly hampering its practical applications on large-scale datasets.

To address this issue, we develop efficient approximations for matrix-based Rényi's entropy from the perspective of randomized numerical linear algebra. Motivated by the recent advancement of a variance-reduced stochastic trace estimator named Hutchinson++ (Hutch++) [15], we decompose the kernel matrix A by randomly projecting it into an orthogonal subspace which holds the largest eigenvalues with high probability, and the counterpart that holds smaller eigenvalues. Their traces are then exactly calculated and approximated by the original Hutchinson algorithm respectively, leading to an optimal $O(1/\epsilon)$ convergence rate in terms of the number

of vector queries. We further develop polynomial expansion techniques including Taylor and Chebyshev series to approximate arbitrary matrix power functional in Rényi's entropy. We theoretically analyze the quality-of-approximation results and conduct large-scale experiments to validate the effectiveness of the proposed approaches. Our main contributions in this work are threefold:

- We develop efficient approximations for matrix-based Rényi's entropy with randomized trace estimation and polynomial approximation techniques. Our algorithms reduce the overall time complexity from $O(n^3)$ to $O(n^2sm)$ $(s, m \ll n)$ and support arbitrary α values.
- We theoretically establish both upper and lower bounds for approximation accuracy, showing that the convergence rates $O(1/\epsilon)$ and $O(\sqrt{\kappa})$ (κ is the condition number of A) w.r.t s and m respectively are nearly optimal up to a logarithmic factor in terms of approximation error.
- We evaluate our algorithms on large-scale simulation datasets and real-world information-theoretic machine learning tasks, demonstrating promising speedup with negligible loss in validation accuracy.

II. Preliminaries

Shannon's entropy is one of the most commonly used measures of the randomness given the PDF $p(\mathbf{x})$ for a given continuous random variable X that values in a finite set $\mathcal{X}$:

$$H(X) = -\int_{\mathcal{X}} p(\mathbf{x}) \log p(\mathbf{x})\, \mathrm{d}\mathbf{x}.$$

A popular generalization is the Rényi's α-order entropy $H_\alpha(X)$ of order $\alpha > 0$ and $\alpha \neq 1$:

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \int_{\mathcal{X}} p^\alpha(\mathbf{x})\, \mathrm{d}\mathbf{x}, \tag{1}$$

Rényi entropy covers a family of different entropy measures through the hyper-parameter α, including Shannon entropy ($\alpha \to 1$), Min entropy ($\alpha \to \infty$), and Collision entropy ($\alpha = 2$), making it widely adopted in machine learning and statistical inference tasks. It is easy to see that both Shannon's entropy and Rényi's entropy require knowledge about data distributions, which hampers its application in high-dimensional scenarios. To solve this issue, Giraldo et al. proposed an alternative entropy measure that enables direct quantification from given data:

Definition 1 [7]: Let $\phi : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ be a real-valued positive kernel that is also infinitely divisible [16]. Given $\{\mathbf{x}_i\}_{i=1}^n \subset \mathcal{X}$, each $\mathbf{x}_i$ being a real-valued scalar or vector, and the Gram matrix K obtained from $K_{ij} = \phi(\mathbf{x}_i, \mathbf{x}_j)$, a matrix-based analogue to Rényi's α-entropy can be defined as:

$$S_\alpha(A) = \frac{1}{1-\alpha} \log(\mathrm{tr}(A^\alpha)) = \frac{1}{1-\alpha} \log \left[\sum_{i=1}^n \lambda_i^\alpha(A) \right],$$

where $A_{ij} = \frac{1}{n} \frac{K_{ij}}{\sqrt{K_{ii} K_{jj}}}$ is a normalized kernel matrix and $\lambda_i(A)$ denotes the i-th eigenvalue of A.

It is worth noting that the infinitely divisible condition imposed on kernel function is more strict than semi-definite positive. In practice, one can select Gaussian kernel (or polynomial kernel with even-order) to calculate matrix-based Rényi's entropy. The normalization ensures that the symmetric semi-positive definite (SPD) kernel matrix A has unit trace, then its eigenvalues are in $[0, 1]$ and satisfies $\sum_{i=1}^n \lambda_i(A) = \mathrm{tr}(A) = 1$. Therefore, the eigenvalues of A form a discrete probability distribution and serves as a natural density estimator for the random variable used to generate the samples $\{\mathbf{x}_i\}$. We denote the minimum and maximum eigenvalue of A as $u \in [0, 1/n]$ and $v \in [1/n, 1]$ respectively, and the corresponding condition number is then $\kappa = v/u$. In numerical scenarios, the power iteration and Lanczos iteration are effective algorithms for calculating u and v in $O(d \cdot \mathrm{nnz}(A))$, where $\mathrm{nnz}(\cdot)$ denotes the number of non-zero elements in a matrix and d is the number of iterations.

Definition 2 [10]: Let $\phi_1 : \mathcal{X}^1 \times \mathcal{X}^1 \mapsto \mathbb{R}$, $\cdots$, $\phi_L : \mathcal{X}^L \times \mathcal{X}^L \mapsto \mathbb{R}$ be positive infinitely divisible kernels and $\{\mathbf{x}_i^1, \cdots, \mathbf{x}_i^L\}_{i=1}^n \subset \mathcal{X}^1 \times \cdots \times \mathcal{X}^L$ be a collection of n samples, a matrix-based analogue to Rényi's α-order joint entropy among L variables can be defined as:

$$S_\alpha(A_1, \cdots, A_L) = S_\alpha\left(\frac{A_1 \circ \cdots \circ A_L}{\mathrm{tr}(A_1 \circ \cdots \circ A_L)} \right), \tag{2}$$

where $A_1, \cdots, A_L$ are normalized kernel matrices and $\circ$ denotes the Hadamard product.

Within these settings, the matrix-based Rényi's α-order conditional entropy $S_\alpha(A_1, \cdots, A_k|B)$ and mutual information $I_\alpha(\{A_1, \cdots, A_k\}; B)$ between variables $\mathbf{x}^1, \cdots, \mathbf{x}^k$ and $\mathbf{y}$ can be defined as:

$$S_\alpha(A_1, \cdots, A_k|B) = S_\alpha(A_1, \cdots, A_k, B) - S_\alpha(B), \tag{3}$$

$$I_\alpha(\{A_1, \cdots, A_k\}; B) = S_\alpha(A_1, \cdots, A_k) - S_\alpha(A_1, \cdots, A_k|B), \tag{4}$$

where $A_1, \cdots, A_k$ and B are corresponding kernel matrices constructed from $\mathbf{x}^1, \cdots, \mathbf{x}^k$ and $\mathbf{y}$. As we can see, the matrix-based Rényi's entropy functionals above avoid the estimation of underlying data distributions, which makes them easily applicable in high-dimensional scenarios. Moreover, it is simple to verify that they are permutation invariant to the ordering of variables $A_1, \cdots, A_k$. The matrix-based Rényi's mutual information has been successfully applied in feature selection tasks [10] by maximizing the multivariate mutual information $I(S_{sub}; Y)$, where S_{sub} is a subset of all features and Y is the target label. Also, the matrix-based entropy functional has recently been demonstrated to be differentiable, which makes it suitable to be used to train neural networks combining with the information bottleneck objective [5]:

$$\mathcal{L}_{IB} = I(Y; T) - \beta \cdot I(X; T), \tag{5}$$

where X, Y, and T are the input, the output, and an intermediate representation of the neural network.

The scalability to high-dimensional space and the differentiable property also make matrix-based Rényi's entropy functional to be used in other challenging applications involving deep neural networks. For example, in terms of knowledge distillation, [12] directly applies matrix-based Rényi's mutual information (i.e., Eq. (4)) as a new regularization term to

maximize the dependence between the student and teacher representations from samples in a mini-batch. In terms of network pruning (i.e., removing redundant filters in a large convolutional neural network), [11] uses matrix-based Rényi's mutual information to quantify the relevance between the responses of each filter and class labels, and then straightforwardly prunes filters with the least mutual information values.

Despite the empirical success of the matrix-based Rényi's entropy functional in the above-mentioned applications, the high computational complexity of this new information measure severely impedes its wider range of applications, especially in the cases that the number of samples or the mini-batch size is large. This limitation motivates our work to speed up its computation with theoretical guarantees. Stochastic trace estimation techniques have been previously explored to accelerate the calculation of the trace of large matrices, where the Hutchinson estimator is one of the commonly adopted trace estimation techniques. By generating a series of random vectors $\{\mathbf{g}_i\}_{i=1}^{s}$ with i.i.d. $\{\pm 1\}$ entries, [17] proved that by taking $s = O(\log(1/\epsilon)/\epsilon^2)$, with probability at least $1-\delta$:

$$\left|\mathrm{tr}(A) - \frac{1}{s}\sum_{i=1}^{s}\mathbf{g}_i^\top A\mathbf{g}_i\right| \le \epsilon \cdot \mathrm{tr}(A).$$

III. Approximation Algorithms

In this section, we aim to develop efficient approximations for matrix-based Rényi's entropy from the perspective of randomized numerical linear algebra. Intuitively, the entropy approximation problem is closely related with the well-known trace estimation problem, as stated in the following proposition:

Proposition 3: For any $\epsilon_0 \in (0,1)$ and sufficient large n, if a randomized algorithm $\mathcal{A}$ produces a $(1\pm\epsilon_0)$-approximation for $\mathrm{tr}(A)$ (where A is $n\times n$ SPD matrix) using s queries with probability at least $1-\delta$, then $\mathcal{A}$ produces a $(1\pm\epsilon)$-approximation for $S_\alpha(A) = \frac{1}{1-\alpha}\log \mathrm{tr}(A^\alpha)$ using s queries with the same probability, where $\epsilon_0 = 1 - \min(\mu, 1/\mu)^\epsilon$ and

$$\mu = \frac{1-un}{v-u}\cdot v^\alpha + \frac{vn-1}{v-u}\cdot u^\alpha.$$

Vice versa for $\epsilon_0 = \max(n^{\alpha-1}, n^{1-\alpha})^\epsilon - 1$.

Indeed, Proposition 3 shows that the trace estimation problem is equivalent to matrix-based Rényi's entropy approximation. Typical choices of $\mathcal{A}$ include the Gaussian trace estimator and Hutchinson estimator [18], which can generate unbiased estimates to $\mathrm{tr}(A)$. Compared to these methods, the recently developed randomized trace estimator Hutch++ [15] further utilizes the positive semi-definiteness property of A and achieves substantially lower estimation variance. By decomposing the kernel matrix A into a randomized orthogonal subspace Q and its complement $I-QQ^\top$, Hutch++ achieves nearly optimal convergence rate in terms of the number of vector queries s as shown in Algorithm 1.

In particular, by taking $f(A) = A^\alpha$, Algorithm 1 generates a $(1\pm\epsilon)$-approximation for $S_\alpha(A)$ with high probability, converting the entropy estimation problem into matrix-vector multiplications operations and reduces the overall complexity to $O(s\cdot \mathrm{nnz}(A))$, substantially lower than traditional $O(n^3)$ eigenvalue based approaches.

Algorithm 1 Hutch++ Algorithm for Implicit Matrix Trace Estimation [15]

1: **Input:** Kernel matrix $A \in \mathbb{R}^{n\times n}$, number of random vectors $s (s \ll n)$, positive matrix function $f(A)$.
2: **Output:** Approximation to $\mathrm{tr}(f(A))$.
3: Sample $S \in \mathbb{R}^{n\times \frac{s}{4}}, G \in \mathbb{R}^{n\times \frac{s}{2}}$ from i.i.d. standard Gaussian distribution.
4: Compute an orthonormal basis $Q \in \mathbb{R}^{n\times \frac{s}{4}}$ for the span of AS via QR decomposition.
5: **Return:** $Z = \mathrm{tr}\big(Q^\top f(A)Q\big) + \frac{2}{s}\mathrm{tr}\big(G^\top(I-QQ^\top)f(A)(I-QQ^\top)G\big)$.

A. Integer Order Approach

When $\alpha \in \mathbb{N}$, for any real-valued vector g, $A^\alpha \cdot g$ could be directly calculated by multiplying A with a vector for α times. This observation gives Algorithm 2 for integer order Rényi's entropy estimation:

Algorithm 2 Integer Order Matrix-Based Rényi's Entropy Estimation

1: **Input:** Kernel matrix $A \in \mathbb{R}^{n\times n}$, number of random vectors s, integer order $\alpha \ge 2$.
2: **Output:** Approximation to $S_\alpha(A)$.
3: Run Hutch++ with $f(A) = A^\alpha$ and s random vectors.
4: **Return:** $\tilde{\mathbf{S}}_\alpha(A) = \frac{1}{1-\alpha}\log(\text{Hutch++}(A^\alpha))$.

Theorem 4: Let $\tilde{S}_\alpha(A)$ be the output of Algorithm 2 with $s = O\Big(\frac{1}{\epsilon}\sqrt{\log\big(\frac{1}{\delta}\big)} + \log\big(\frac{1}{\delta}\big)\Big)$, then with probability at least $1-\delta$:

$$\left|\tilde{S}_\alpha(A) - S_\alpha(A)\right| \le \epsilon \cdot S_\alpha(A).$$

Remark 3: Theorem 4 establishes the main quality-of-approximation result for Algorithm 2, that a s with order $O(1/\epsilon)$ is sufficient to guarantee the approximation error with high probability. Algorithm 2 finishes in $O(\alpha s\cdot \mathrm{nnz}(A))$, which is substantially lower than eigenvalue decomposition algorithms.

B. Taylor Series Approach

The fractional order of α may constantly be come across in real-world applications [10] depending on the specific tasks. For example, [5] and [19] recommend $\alpha = 1.01$ to approximate Shannon entropy. In this circumstance, obtaining an exact value of $A^\alpha \cdot g$ is not feasible for random vector g. An ideal workaround is to adopt a Taylor expansion on the power term A^α:

$$(1+x)^\alpha = \sum_{k=0}^{\infty}\binom{\alpha}{k}x^k,\ x\in[-1,1]$$

Taking v as the largest eigenvalue of A, eigenvalues of $A/v - I_n$ are in $[-1, 0]$. Then A^α can be expanded as:

$$A^\alpha = v^\alpha \sum_{k=0}^{\infty} \binom{\alpha}{k} (A/v - I_n)^k.$$

An approximation to $A^\alpha \cdot g$ is now available by calculating $A \cdot g$, $A^2 \cdot g$, $\cdots$ in sequence. By selecting the first m major terms in the polynomial expansion above, we have Algorithm 3 for fractional order Rényi's entropy estimation:

Algorithm 3 Fractional Order Matrix-Based Rényi's Entropy Estimation via Taylor Series

1: **Input:** Kernel matrix $A \in \mathbb{R}^{n\times n}$, number of random vectors s, fractional order α, polynomial order m, eigenvalue upper bound v.
2: **Output:** Approximation to $S_\alpha(A)$.
3: Run Hutch++ with $f(A) = v^\alpha \sum_{k=0}^{m} \binom{\alpha}{k} (A/v - I_n)^k$ and s random vectors.
4: **Return:** $\tilde{\mathbf{S}}_\alpha(A) = \frac{1}{1-\alpha} \log(\text{Hutch++}(A^\alpha))$.

Theorem 4: Let $\tilde{S}_\alpha(A)$ be the output of Algorithm 3 with

$$s = O\Big(\tfrac{1}{\epsilon|\alpha-1|}\sqrt{\log\big(\tfrac{1}{\delta}\big)} + \log\big(\tfrac{1}{\delta}\big)\Big),$$
$$m = O\Big(\kappa \log\Big(\tfrac{1}{\epsilon|\alpha-1|}\Big)\Big),$$

where $\kappa = v/u$ is the condition number of A, then for any normalized kernel matrix A with eigenvalues in [u, v], with probability at least $1 - \delta$:

$$\left|\tilde{S}_\alpha(A) - S_\alpha(A)\right| \le \epsilon \cdot S_\alpha(A).$$

Remark 5: Theorem 4 presents the relative error bound for Algorithm 3, where explicit order of s and m are given to guarantee the approximation accuracy. Specifically, s is scaled by a coefficient $1/|\alpha - 1|$ compared to Theorem 4, and m is positively related to the condition number κ. Algorithm 3 finishes in $O(ms \cdot \text{nnz}(A))$ with $m, s \ll n$.

The analysis above requires $u > 0$, i.e. the kernel matrix has full rank. However, this requirement is hard to be satisfied in some machine learning tasks e.g. RKHS transporting and dimension reduction [20], [21], where rank-deficient matrices are frequently encountered. To account for this, we establish the following theorem:

Theorem 6: Let $\tilde{S}_\alpha(A)$ be the output of Algorithm 3 with

$$s = O\Big(\tfrac{1}{\epsilon|\alpha-1|}\sqrt{\log\big(\tfrac{1}{\delta}\big)} + \log\big(\tfrac{1}{\delta}\big)\Big),$$
$$m = O\Big((vn)^{\frac{1}{\min(1,\alpha)}} \sqrt[\alpha]{\tfrac{1}{\epsilon|\alpha-1|}}\Big),$$

then for any normalized kernel matrix A with eigenvalues in [0, v], with probability at least $1 - \delta$:

$$\left|\tilde{S}_\alpha(A) - S_\alpha(A)\right| \le \epsilon \cdot S_\alpha(A).$$

Remark 7: When $u = 0$, due to the existence of a singular point in $f(x) = x^\alpha$ at $x = 0$, a logarithmic convergence rate is no longer achievable. The polynomial approximation error is now dominated by ϵ instead of κ. The coefficient vn corresponds to the rare worst case when the eigenvalues of A all equal $1/n$, or are all in $\{0, v\}$.

C. Chebyshev Series Approach

Chebyshev expansion is an advanced technique to approximate analytic functions and often enjoys better theoretical properties. For some continuous function $f : [-1, 1] \to \mathbb{R}$, it is defined as

$$f(x) = \frac{c_0}{2} + \sum_{k=1}^{\infty} c_k T_k(x), \ x \in [-1, 1]$$

where $T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$ for $k \ge 1$, $T_0(x) = 1$ and $T_1(x) = x$. By taking the first m terms, the coefficients $c_k, k = 0, \cdots, m$ could be calculated as

$$c_k = \frac{2}{m+1} \sum_{i=0}^{m} f(x_i) T_k(x_i),$$

where $x_i = \cos\big(\pi(i+1/2)/(m+1)\big)$. Through a combination with linear mapping g: $[-1, 1] \to [u, v]$, we can now approximate $f(\lambda) = \lambda^\alpha$ for any $\lambda \in [u, v]$ with $\hat{T}_k = T_k \circ g^{-1}$, $k = 0, \cdots, m$, as shown in Algorithm 4.

Algorithm 4 Fractional Order Matrix-Based Rényi's Entropy Estimation via Chebyshev Series

1: **Input:** Kernel matrix $A \in \mathbb{R}^{n\times n}$, number of random vectors s, fractional order α, polynomial order m, eigenvalue lower & upper bounds u, v.
2: **Output:** Approximation to $S_\alpha(A)$.
3: Run Hutch++ with $f(A) = c_0/2 + \sum_{k=1}^{m} c_k \hat{T}_k(A)$ and s random vectors.
4: **Return:** $\tilde{\mathbf{S}}_\alpha(A) = \frac{1}{1-\alpha} \log(\text{Hutch++}(A^\alpha))$.

Theorem 8: Let $\tilde{S}_\alpha(A)$ be the output of Algorithm 4 with

$$s = O\Big(\tfrac{1}{\epsilon|\alpha-1|}\sqrt{\log\big(\tfrac{1}{\delta}\big)} + \log\big(\tfrac{1}{\delta}\big)\Big),$$
$$m = O\Big(\sqrt{\kappa} \log\Big(\tfrac{\kappa}{\epsilon|\alpha-1|}\Big)\Big),$$

where $\kappa = v/u$ is the condition number of A, then for any normalized kernel matrix A with eigenvalues in [u, v], with probability at least $1 - \delta$:

$$\left|\tilde{S}_\alpha(A) - S_\alpha(A)\right| \le \epsilon \cdot S_\alpha(A).$$

Remark 9: Theorem 8 requires only $O(\sqrt{\kappa})$ polynomial terms to guarantee the approximation accuracy for Algorithm 4 in the case that A is well-conditional, comparing to Theorem 4 which require $O(\kappa)$ to achieve the same approximation accuracy. Moreover, Algorithm 4 requires estimation of u, which is generally more difficult than estimating v because of its small magnitude.

Similarly, we establish the error bound of the Chebyshev series for rank-deficient kernel matrices.

Theorem 10: Let $\tilde{S}_\alpha(A)$ be output of Algorithm 4 with

$$s = O\Big(\tfrac{1}{\epsilon|\alpha-1|}\sqrt{\log\big(\tfrac{1}{\delta}\big)} + \log\big(\tfrac{1}{\delta}\big)\Big),$$
$$m = O\Big((vn)^{\frac{1}{2\min(1,\alpha)}} \sqrt[2\alpha]{\tfrac{1}{\epsilon|\alpha-1|}}\Big),$$

then for any normalized kernel matrix A with eigenvalues in [0, v], with probability at least $1-\delta$:

$$\left|\tilde{S}_\alpha(A) - S_\alpha(A)\right| \leq \epsilon \cdot S_\alpha(A).$$

Remark 11: Compared with Theorem 6, Algorithm 4 within rank deficient case still achieves better theoretical guarantees from all perspectives.

D. Connection With the Lanczos Method

Besides polynomial approximation, an alternative approach for approximating matrix functions is the Lanczos method [22]: given implicit matrix $f(A)$ and arbitrary vector b, an approximation of $f(A)\cdot b$ is acquired by linear interpolation in the Krylov subspace $\{b, A\cdot b, \cdots, A^m\cdot b\}$. It could be regarded as an adaptive polynomial approximation technique, where the coefficients are chosen according to the given matrix $f(A)$ and vector b. However, this approach does not achieve any faster convergence rate than explicit polynomial approximation: as pointed out in [23], the block Lanczos method achieves exactly the same upper bound $O(\sqrt{\kappa}\log(\kappa/\epsilon))$ as Chebyshev series in terms of subspace dimension, while requiring additional $O(nms)$ memory to store the block vectors in each step [22]. Moreover, the lower bound of the Lanczos method is also closely related to the lower bound of polynomial approximation [24], which will be discussed in the next section. Nevertheless, elaborate descriptions of this topic are beyond the scope of the current research, we leave this problem for future research.

IV. Lower Bounds

So far, we have established approximation algorithms for matrix-based Rényi's entropy and evaluated their theoretical properties. A natural question is if the $O(1/\epsilon)$, $O(\sqrt{\kappa})$ or $O(\sqrt[2\alpha]{1/\epsilon})$ upper bounds in our previous analysis are tight. In this section, we will prove that up to a logarithmic factor, they are consistent with theoretical lower bounds.

In Proposition 3, we show that an effective trace approximator implies an effective approximator for matrix-based Renyi's entropy. Based on the lower bound of randomized implicit trace estimation in fixed precision model [15], we obtain the lower bound of required matrix-vector multiplication queries s by complexity reduction:

Theorem 11: Any algorithm that accesses a normalized $n\times n$ kernel matrix A via matrix-vector multiplication queries $Ar_1, \cdots, Ar_m$, where $r_1, \cdots, r_m$ are possibly adaptively chosen randomized vectors under limited precision computation model, requires $s = \Omega\left(\frac{1}{\epsilon|\alpha-1|\log n \log(1/\epsilon|\alpha-1|\log n)}\right)$ such queries to output an estimate Z so that, with probability at least $\frac{2}{3}$, $|Z - S_\alpha(A)| \leq \epsilon\cdot S_\alpha(A)$ for arbitrary $\alpha > 0$.

Remark 12: Note that the lower bound $s = \Omega(1/\epsilon)$ matches our previous results up to a $\log(1/\epsilon)$ factor, which means our error bounds are nearly-optimal. Moreover, the scaling term $1/(1-\alpha)$ implies that precise approximation is impossible when $\alpha \to 1$. This observation is confirmed in our simulation studies (Section V-A).

Next, by applying the theory of best uniform approximation error, we establish the lower bounds for the required number of terms m in polynomial approximation. Given a continuous real function f defined on $[-1, 1]$, denote the m-terms best uniform approximation of f by p_m, then:

$$\|f - p_m\| = \min_{p\in\mathbb{P}_m} \|f - p\|,$$

where $\|\cdot\|$ denotes the uniform norm and $\mathbb{P}_m$ is the linear space of all polynomials with a degree at most m. Based on previous theoretical analysis of function $f(x) = x^\alpha$ [25], [26], [27], [28], we obtain:

Theorem 13: There exists a positive decreasing function $\epsilon_0 : \mathbb{R}^+ \to \mathbb{R}^+$ such that for arbitrary $0 < u < v < 1$ and $0 < \epsilon < \epsilon_0(v/u)$, any polynomial p_m that approximates matrix function $f(A) = A^\alpha$, requires $m = \Omega\left(\sqrt{\kappa}\log\left(\frac{1}{\kappa\epsilon|\alpha-1|\log n}\right)\right)$ degree to achieve

$$\left|\frac{1}{1-\alpha}\log\left(\mathrm{tr}(p_m(A))\right) - S_\alpha(A)\right| \leq \epsilon\cdot S_\alpha(A),$$

for any positive definite matrix A with all eigenvalues in $[u, v]$ and $\mathrm{tr}(A)\in[1, 2]$, where $\kappa = v/u$.

Theorem 14: For arbitrary $v > 0$ and small enough ϵ, any polynomial p_m that approximates matrix function $f(A) = A^\alpha$, requires $m = \Omega\left(\sqrt[2\alpha]{\frac{1}{\epsilon|\alpha-1|\log n}}\right)$ degree to achieve

$$\left|\frac{1}{1-\alpha}\log\left(\mathrm{tr}(p_m(A))\right) - S_\alpha(A)\right| \leq \epsilon\cdot S_\alpha(A),$$

for any positive semi-definite matrix A with all eigenvalues in $[0, v]$ and $\mathrm{tr}(A)\in[1, 2]$.

Remark 15: Theorem 13 and 14 present the lower bound for polynomial approximation in fractional order Rényi's entropy estimation. Also, these bounds indicate the near-optimality of Algorithm 4 in consideration of the results in Theorem 8 and 10.

V. Experimental Results

In this section, we evaluate the performance of proposed approximations implemented in C++ using Eigen [29]. Numerical studies are conducted on an Intel i7-10700 (2.90GHz) CPU with 64GB of RAM, with deep learning models trained on an RTX 2080Ti GPU. We give comprehensive experimental results for both synthetic data and real-world information-related tasks.

A. Simulation Studies

In the following simulation experiments, we generate synthetic data points by a mixture of Gaussian distribution $\frac{1}{2}N(-1, I_d) + \frac{1}{2}N(1, I_d)$ with $n = 5{,}000$ and $d = 10$, where I_d is an identity matrix of size d, resulting in a $5{,}000\times 5{,}000$ kernel matrix size. Gaussian kernel $\phi(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/2\sigma^2)$ with $\sigma = 1$ is adopted in matrix-based Rényi's entropy quantification. For each benchmark, we report the mean relative error (MRE) and corresponding standard deviation (SD) of approximation results after $K = 100$ trials. The oracle $S_\alpha(A)$ is computed through the direct $O(n^3)$ eigenvalue approach.

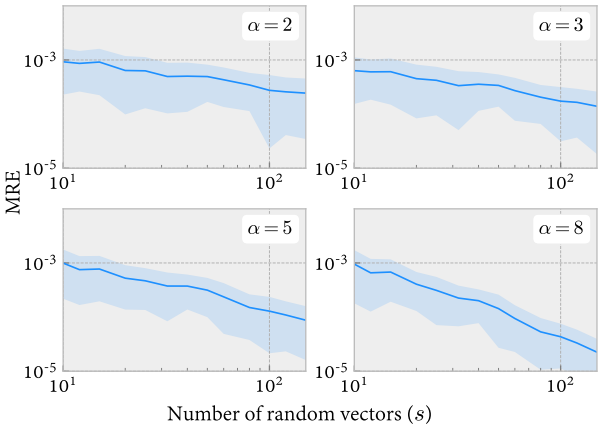

Fig. 1. Number of random vectors s versus MRE curves for integer α-order Rényi's entropy estimation.

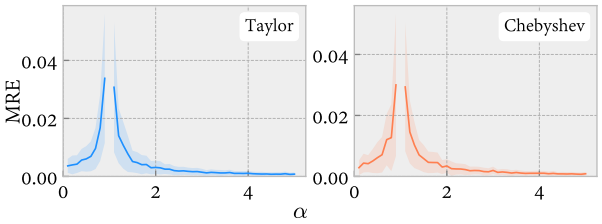

Fig. 2. α versus MRE curves for fractional α-order Rényi's entropy estimation algorithms.

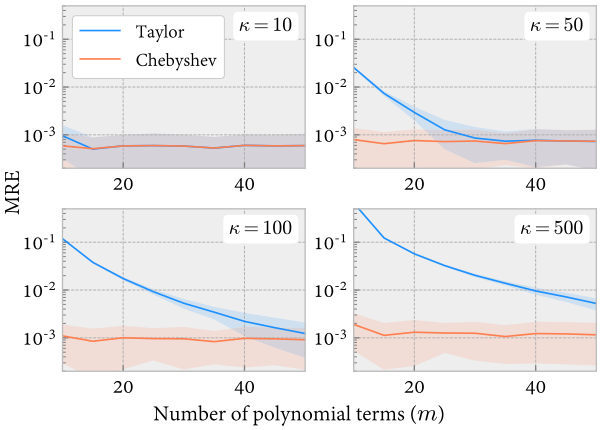

Fig. 3. Number of polynomial terms m versus MRE curves for different condition numbers κ.

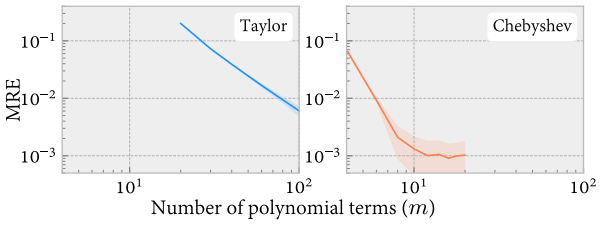

Fig. 4. Number of polynomial terms m versus MRE curves for rank deficient kernel matrices.

1) Integer Order Approximation: We first evaluate the performance of Algorithm 2 for integer-order entropy estimation. We report the s versus MRE curves for $\alpha \in \{2, 3, 5, 8\}$, where the number of random vectors s ranges from 10 to 150, as shown in Figure 1. The shaded area indicates the corresponding SD of MRE. We observe a linear relationship between s and MRE as expected. It is worth noting that we achieve a 0.1% relative error with only $s = 10$ random vectors, which costs roughly 1.2 seconds of running time for $\alpha = 2$. For comparison, the trivial eigenvalue approach takes 27 seconds to obtain a complete eigenvalue decomposition.

2) Fractional Order Approximation: We further evaluate the Taylor and Chebyshev algorithms for fractional α orders. The results on describing the impact of α on approximation MRE with $m = 20$ and $s = 100$ are reported in Figure 2. As expected, MRE curves grow with the increase of α for $\alpha < 1$ and decrease otherwise. This phenomenon is because of the $|\frac{1}{1-\alpha}|$ coefficient in our previous theoretical analysis. When α is close to 1, this term dominates the approximation error.

We next explore the influence of different condition numbers κ in polynomial approximation. Here we set $\alpha = 1.5$, $s = 100$, and m ranges from 10 to 50, with adjusted width parameter σ in Gaussian kernel to control the eigenspectrum. It can be seen from Figure 3 that the polynomial terms m required by Taylor approximation is larger than that of Chebyshev approximation for relatively large κ, which verifies our findings in Theorem 4 and 8. For smaller κ, the two approaches yield comparable results. Reminding that the Taylor series does not require estimation of u, it is thus more suitable for kernel matrices with flat eigenspectrum.

We further investigate the rank-deficient circumstances. We adopt the polynomial kernel $\phi(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + r)^p$ with $p = 2$, $r = 1$ to fulfill the infinitely divisible requirement, and set $d = 98$ to retain roughly 1% of the eigenvalues to be zero. We find that the Chebyshev approximation still outperforms the Taylor approach in terms of MRE with small m values, as shown in Figure 4.

Finally, we report the experimental results of both algorithms for different α values. We set $\sigma = 1$ in the Gaussian kernel and s varies from 10 to 150. For the Taylor approach, we set $m = 40$ and for Chebyshev, we set $m = 15$. From Figure 5, we can see that the two approaches achieve similar performance in this setting. Also, we get a relatively higher MRE for α near 1 ($\alpha = 0.8$), the same as we have discussed before. In this sense, we recommend a combination of $s = 50$ and $m = 15$ that takes 3 seconds to achieve a 10^{-3} relative error for most circumstances, leading to 9 times speedup compared with the trivial eigenvalue approaches. For larger kernel matrices, this advantage could be even more pronounced.

B. Real Data Studies

In real-world data-driven applications, the extended entropy measures including Rényi's α-order joint entropy (2), conditional entropy (3) and mutual information (4) enable much wider adoption of information-based machine learning tasks. By approximating the trace of the joint kernel matrix $A_1 \circ \cdots \circ A_L$ in (2), our approximations algorithms are immediately applicable to these extended information measures.

In this section, we will demonstrate the performance of our algorithms on these novel extensions by three representative

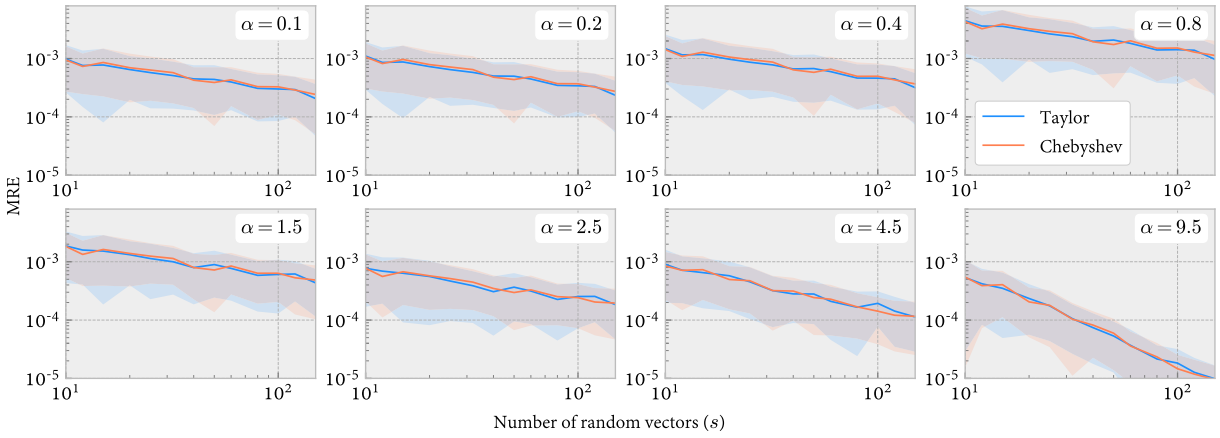

Fig. 5. Number of random vectors s versus MRE curves for fractional α-order Rényi's entropy estimation.

real-world applications, which accelerate, respectively, the computation of entropy (in neural network parameterization of information bottleneck), mutual information (in feature ranking), and multivariate mutual information (in feature selection). We select $\sigma = 1$ in the Gaussian kernel and $\alpha = 2$ for simplicity.

1) Parametrization of Information Bottleneck by Neural Networks: The Information Bottleneck (IB) objective was first introduced by [30] and has recently been adopted in deep network training to learn either stochastic or deterministic compressed yet meaningful representations [19], [31], [32]. Denoting X as the input and Y as the target label, the IB approach learns an intermediate representation T that balances the trade-off between the predictive performance of T on task Y (quantified by $I(Y;T)$) and the complexity of T (quantified by $I(X;T)$) (see eq. (5)), where β is a hyper-parameter that balances $I(Y;T)$ and $I(X;T)$. There are different ways to parameterize IB by neural networks. In general, the maximization of $I(Y;T)$ is equivalent to the minimization of cross-entropy (CE) loss [33], [34]. On the other hand, for a deterministic and feed-forward neural network, we have $I(X;T) = H(T)$,[1] the entropy of latent representation T [34], [35]. Therefore, the objective (5) can be transferred to minimization of CE loss with an additional regularization on the entropy of T[2]:

$$\mathcal{L}_{DIB} = \min \mathrm{CE} + \beta \cdot H(T), \tag{6}$$

which is also called the deterministic IB (DIB) [19], [38]. If we optimize DIB with the matrix-based entropy functional, it can be simply parameterized by a deep neural network with gradient-based optimization. The training can be significantly accelerated by approximating the trace of the kernel matrix constructed from T.

We term our method the Approximated DIB (ADIB) and compare it with the classic variational IB (VIB) [33] and the original DIB [19] that also uses the matrix-based entropy functional without any fast approximations. We follow the experiment settings in [19], where VGG16 [39] and CIFAR-10 are selected as the backbone network and classification dataset respectively. The last fully-connected layer in VGG before the softmax layer is selected as the bottleneck T. All models are trained with 400 epochs, 0.1 initial learning rate which is reduced by a factor of 10 every 100 epochs, and 100 batch size such that the kernel matrix in Rényi's entropy is 100×100. The performance of DIB and ADIB are evaluated with number of random vectors $s = 10$ and $\beta = 0.01$. The final classification accuracy and time spent on calculating IB / training networks are reported in Table I. Our ADIB significantly outperforms CE and VIB in terms of classification accuracy. It also achieves comparable performance to the original DIB with significantly

TABLE I

TEST ERROR AND TIME SPENT FOR DIFFERENT METHODS ON CIFAR-10. RIGHT IS THE TOTAL TIME OF NETWORK TRAINING, WHILE LEFT IS THE TIME SPENT SOLELY ON CALCULATING IB. THE NUMBER QUOTED INDICATES CORRESPONDING α VALUE IN RÉNYI'S ENTROPY

BACKBONE	OBJECTIVE	ERROR (%)	TIME (HOUR)
VGG16	CE	7.36	- / 2.27
	VIB	7.15	0.03 / 2.30
	DIB (1.01)	5.66	1.13 / 3.40
	DIB (2)	5.69	
	ADIB (2)	5.71	0.15 / 2.42

TABLE II

NUMBER OF INSTANCES (#I), FEATURES (#F), AND CLASSES (#C) OF CLASSIFICATION DATASETS USED IN FEATURE SELECTION AND RANKING EXPERIMENTS, AND THE CORRESPONDING RUNNING TIME OF RMI AND ARMI

DATASET	#I	#F	#C	SELECTION		RANKING	
				RMI	ARMI	RMI	ARMI
MADELON	2600	500	2	9.98	1.33	1.10	0.23
KRVSKP	3196	37	2	1.37	0.14	0.15	0.03
OPTDIGITS	5620	65	10	13.79	0.80	1.44	0.14
STATLOG	6435	37	6	11.71	0.59	1.22	0.10
SPAMBASE	4601	57	2	6.64	0.48	0.70	0.08
WAVEFORM	5000	40	3	6.17	0.39	0.65	0.07
GALAXY	9150	16	2	14.28	0.52	1.47	0.09
BEANS	13611	17	7	48.14	1.17	4.91	0.21

[1]This is just because for a deterministic mapping from X to T, there is no uncertainty about T given X. Hence, $H(T|X) = H(T) - I(X;T) = 0$.

[2]The same strategy has also been used in recent deep IB approaches, such as [36] and [37].

TABLE III
THE BEST CLASSIFICATION ERROR (%) ACHIEVED BY EACH FEATURE RANKING (THE UPPER HALF) AND FEATURE SELECTION (THE LOWER HALF) METHODS FOR $k = 10$ FEATURES. THE LAST COLUMN INDICATES THE AVERAGE RANKING OF DIFFERENT METHODS ON OUR TEST BENCHMARK

	MADELON	KRVSKP	OPTDIGITS	STATLOG	SPAMBASE	WAVEFORM	GALAXY	BEANS	AVERAGE RANK
ADC	15.00	5.88	9.77	13.33	9.11	18.04	0.73	8.38	3.25
NFIG	15.00	5.63	73.47	14.84	9.98	18.04	0.64	8.38	4.25
SU	15.00	5.82	12.31	14.55	9.13	18.04	0.63	8.38	3.25
DAS	48.81	38.96	88.67	15.70	24.82	18.04	0.81	8.38	5.88
WJE	15.23	5.07	16.35	18.20	11.39	16.32	0.86	8.38	4.75
RMI	13.23	5.48	7.31	14.83	9.78	16.14	0.51	8.29	1.63
ARMI	13.23	5.48	7.31	14.83	9.78	16.14	0.51	8.33	1.75
MIFS	47.00	5.88	8.90	12.37	21.73	26.28	0.96	7.02	5.75
FOU	34.42	4.79	10.28	11.83	18.04	19.70	12.77	7.05	5.25
MIM	14.65	5.88	9.77	13.07	9.26	17.92	0.74	8.40	6.00
MRMR	46.77	5.85	7.35	12.65	9.02	15.28	0.90	7.55	4.88
JMI	12.27	5.88	6.55	12.77	9.17	14.86	0.69	8.96	4.63
CMIM	16.65	5.79	5.64	12.65	8.91	15.28	5.98	7.08	3.88
RMI	10.62	2.57	5.50	12.34	8.87	15.62	0.51	7.11	2.00
ARMI	10.31	2.57	5.91	12.63	8.87	16.82	0.51	7.41	2.63

less computational time (which saves nearly 1 hour of training time). Note that, this speedup could be enhanced further by using a larger batch size, which is a common choice in modern fine-tuning techniques [40].

2) Application to Feature Ranking: Given a set of features $S = \{X_1, \cdots, X_n\}$, the feature selection task aims to find the smallest subset S_{sub} that maximizes the relevance of the label Y. From an information-theoretic perspective, an ultimate goal is to maximize the multivariate mutual information $I(S_{sub}; Y)$, which is however usually impractical in real-world scenarios because of the curse of dimensionality and the difficulty of global optimization.

Feature ranking is a simple yet effective alternative to feature selection. Using the weight $w_i = I(X_i; Y)$ for each feature $X_i \in S$, ranking methods quantify the effectiveness of each feature individually and select the most important features upon two-dimensional probability distributions, which is much easier to estimate in practice. However, they ignore information redundancy and synergy between different features, which usually yields suboptimal performances.

We compare matrix-based Rényi's mutual information (RMI, (4)) and our Approximated RMI (ARMI) with 5 state-of-the-art information-theoretic feature ranking methods, namely Asymmetric Dependency Coefficient (ADC) [41], Normalized First-Order Information Gain (NFIG) [42], Symmetrical Uncertainty (SU) [43], Distance-based Attribute Selection (DAS) [44] and Weighted Joint Entropy (WJE) [45]. 8 well-known classification datasets used in previous works constitute our experiment benchmark [46], [47], [48], [49], which covers a wide range of instance-feature ratios, number of classes, discreteness and data source domains as shown in Table II.

For non-Rényi methods, continuous features are discretized into 5 bins by the equal-width strategy used in [50]. We set the number of random vectors $s = 100$ in ARMI. The Support Vector Machine (SVM) algorithm with RBF kernel ($\sigma = 1$) is adopted as the classifier using 10-fold cross-validation. In our observation, classification accuracy tends to stabilize after selecting the top $k = 10$ features (shown in the Appendix), so we report the best accuracy achieved by each method for selecting at most 10 features in Table III. The comparison of running time between RMI and ARMI is also reported in Table II. As we can see, both RMI and ARMI outperform other Shannon's entropy-based methods. Moreover, ARMI achieves 5 to 20 times speedup, 11.11 times on average compared to the original RMI. For all datasets except "Beans", ARMI obtains exactly the same feature ranking orders as RMI.

3) Application to Feature Selection: We further explore the possibility of improving feature ranking methods by considering the interactions between different features, i.e. directly maximizing the final target $I(S_{sub}; Y)$ by matrix-based entropy functional. Before the definition of the matrix-based RMI in 4, a direct estimation of this quantity was thought to be extremely hard or even intractable due to the curse of high dimensionality [51]. Thus, enormous efforts have been made on approximation techniques that retain only the first or second-order interactions, including Mutual Information-based Feature Selection (MIFS) [52], First-Order Utility (FOU) [53], Mutual Information Maximization (MIM) [54], Maximum-Relevance Minimum-Redundancy (MRMR) [55], Joint Mutual Information (JMI) [56] and Conditional Mutual Information Maximization (CMIM) [57] that achieve the state-of-the-art performance.

We use the same datasets, discretization criterion and classifier settings. We follow a greedy optimization strategy (i.e., incrementally build the selected feature set, in each step) and select the first 10 features that maximize the target $I_\alpha(S_{sub}; Y)$. The results are shown in Table III and II. ARMI achieves 10 to 40 times speedup, 19.07 on average over the original RMI. For "Optdigits", "Spambase" and "Galaxy" datasets, RMI and ARMI select exactly the same features. Also, it is worth noting that in the Galaxy dataset, there is one feature named "redshift" that achieves higher than 95% classification accuracy solely in identifying the class

of a given star. In both selection and ranking experiments, RMI and ARMI successfully identified this feature in the first place, but other methods failed to find it out initially until the third feature is selected. This verifies the effectiveness of ARMI in both low and high-dimensional circumstances and demonstrates its great potential in a variety of information-theory-related tasks.

VI. Conclusion

In this paper, we develop computationally efficient approximations for matrix-based Rényi's entropy, which achieve substantially lower complexity compared to the trivial eigenvalue approach. Through further adoption of Taylor and Chebyshev expansions, we support arbitrary values of α. Statistical guarantees are established for all proposed algorithms and their optimality is proven by theoretical analysis. Large-scale simulations and real-world experiments are conducted to support our theoretical results. It is shown that our approximation algorithms bring tremendous speedup for a wide range of information-related tasks, while only introducing negligible loss in accuracy.

Appendix
Proof of Main Results

For simplicity, we ignore some trivial cases in the following analysis, including 1) $\kappa = 1$, i.e. all eigenvalues of A equal $1/n$; 2) $v = 1$, i.e. except for the largest eigenvalue $\lambda_0 = 1$, all other eigenvalues of A equal 0.

A. Properties of the Trace of A

Consider a special case that the eigenspectrum of A takes extreme value, the information potential $\mathrm{tr}(A^\alpha)$ can be expressed in terms of u and v:

$$\mathrm{tr}(A^\alpha) \in \begin{cases} [\mu, n^{1-\alpha}], & \text{for } \alpha < 1 \\ [n^{1-\alpha}, \mu], & \text{for } \alpha > 1 \end{cases},$$
$$where\ \ \mu = \frac{1-un}{v-u} \cdot v^\alpha + \frac{vn-1}{v-u} \cdot u^\alpha. \tag{7}$$

μ is the special case of $\mathrm{tr}(A^\alpha)$ when all eigenvalues of A belongs to $\{u, v\}$. Some properties about μ are worthy to address for our following analysis.

Proposition 15: Let μ be defined as in (7), then

$$|\log \mu| = \Omega(|\alpha - 1|), \quad |\log \mu| = O(|\alpha - 1| \log n). \tag{8}$$

Proof: If $u = 0$, the conclusion is obvious since we have $\mu = v^{\alpha-1}$ and $v \in (1/n, 1)$.

Otherwise, let $\kappa = v/u$ be the condition number of A, then

$$\mu = u^\alpha \frac{\kappa^\alpha - \kappa^\alpha un + \kappa un - 1}{\kappa u - u}$$
$$= u^{\alpha-1}\left(\frac{\kappa(\kappa^{\alpha-1} - 1)(1-un)}{\kappa - 1} + 1\right).$$

Ignoring the trivial case $\kappa = 1$, for any constant number $\gamma \in (0, \infty)$ that satisfies $\kappa > \gamma + 1$, we have:

$$\kappa^{\alpha-1} - 1 \le \frac{\kappa(\kappa^{\alpha-1} - 1)}{\kappa - 1} \le \left(1 + \frac{1}{\gamma}\right)(\kappa^{\alpha-1} - 1).$$

Thus, we have shown that for all $\kappa \in (1, \infty)$:

$$|\log \mu| = Theta\big(\big|(\alpha - 1)\log u + \log\big((\kappa^{\alpha-1} - 1)(1 - un) + 1\big)\big|\big).$$

When $u \in (1/2n, 1/n)$, $1 - un = O(1)$ and

$$|\log \mu| = \Omega(|(\alpha - 1)\log u|) = \Omega(|(\alpha - 1)\log n|),$$
$$|\log \mu| = O(|(\alpha - 1)(\log u + \log k)|) = O(|(\alpha - 1)\log v|).$$

Otherwise when $u <= 1/2n$, $|\log \mu| = \Theta(|(\alpha - 1)\log v|)$. Combining with $v \in (1/n, 1)$, we finally get:

$$|\log \mu| = \Omega(|\alpha - 1|), \quad |\log \mu| = O(|\alpha - 1| \log n).$$

□

B. Proof of Proposition 3

Proof: Let Z be the output of the algorithm $\mathcal{A}$ on approximating $\mathrm{tr}A$, then with probability at least $1 - \delta$,

$$-\epsilon_0 \cdot \mathrm{tr}(A^\alpha) \le Z - \mathrm{tr}(A^\alpha) \le \epsilon_0 \cdot \mathrm{tr}(A^\alpha).$$

When $\alpha < 1$, $1 < \mu \le \mathrm{tr}(A^\alpha)$ by (7), i.e.

$$1 - \epsilon_0 = \mu^{-\epsilon} \ge \mathrm{tr}^{-\epsilon}(A^\alpha),$$
$$1 + \epsilon_0 < \frac{1}{1 - \epsilon_0} = \mu^\epsilon \le \mathrm{tr}^\epsilon(A^\alpha),$$

it follows that

$$(\mathrm{tr}^{-\epsilon}(A^\alpha) - 1)\mathrm{tr}(A^\alpha) \le Z - \mathrm{tr}(A^\alpha) \le (\mathrm{tr}^{\epsilon}(A^\alpha) - 1)\mathrm{tr}(A^\alpha)$$
$$\mathrm{tr}^{1-\epsilon}(A^\alpha) \le Z \le \mathrm{tr}^{1+\epsilon}(A^\alpha)$$
$$\mathrm{tr}^{-\epsilon}(A^\alpha) \le \frac{Z}{\mathrm{tr}(A^\alpha)} \le \mathrm{tr}^{\epsilon}(A^\alpha).$$

Taking log on both sides, we have:

$$\left|\log \frac{Z}{\mathrm{tr}(A^\alpha)}\right| \le \epsilon |\log \mathrm{tr}(A^\alpha)|$$
$$\frac{1}{1-\alpha}|\log Z - \log \mathrm{tr}(A^\alpha)| \le \frac{\epsilon}{1-\alpha}|\log \mathrm{tr}(A^\alpha)|$$
$$\left|\tilde{S}_\alpha(A) - S_\alpha(A)\right| \le \epsilon \cdot S_\alpha(A),$$

where $\tilde{S}_\alpha(A) = \frac{1}{1-\alpha}\log Z$ is the estimate of $S_\alpha(A)$. Similarly, we can draw the same conclusion for $\alpha > 1$.

On the other hand, let $\bar{Z}$ be the output of the algorithm $\mathcal{A}$ on approximating $S_\alpha(A)$, then with probability at least $1 - \delta$

$$\left|\bar{Z} - S_\alpha(A)\right| \le \epsilon \cdot S_\alpha(A).$$

When $\alpha < 1$, with the same steps as above we can get:

$$(\mathrm{tr}^{-\epsilon}(A^\alpha) - 1)\mathrm{tr}(A^\alpha) \le \tilde{\mathrm{tr}}(A^\alpha) - \mathrm{tr}(A^\alpha) \le (\mathrm{tr}^{\epsilon}(A^\alpha) - 1)\mathrm{tr}(A^\alpha),$$

where $\tilde{\mathrm{tr}}(A^\alpha) = \exp((1-\alpha)\bar{Z})$ is the estimate of $\mathrm{tr}(A^\alpha)$. By (7) we have $n^{1-\alpha} \ge \mathrm{tr}(A^\alpha)$, then

$$\mathrm{tr}^{\epsilon}(A^\alpha) \le n^{\epsilon(1-\alpha)} = 1 + \epsilon_0,$$
$$\mathrm{tr}^{-\epsilon}(A^\alpha) \ge n^{-\epsilon(1-\alpha)} = \frac{1}{1+\epsilon_0} \ge 1 - \epsilon_0.$$

Combining the inequalities above yields:

$$-\epsilon_0 \cdot \mathrm{tr}(A^\alpha) \le \tilde{\mathrm{tr}}(A^\alpha) - \mathrm{tr}(A^\alpha) \le \epsilon_0 \cdot \mathrm{tr}(A^\alpha).$$

We can get the same result for $\alpha > 1$. This finishes the proof. □

C. Proof of Theorem 4

Lemma 16 Theorem 1 in [15]: If Algorithm 1 is implemented with $s = O\left(\frac{1}{\epsilon}\sqrt{\log\left(\frac{1}{\delta}\right)} + \log\left(\frac{1}{\delta}\right)\right)$ matrix-vector multiplication queries, then for any positive semi-definite matrix $f(A)$, with probability at least $1-\delta$, the output Z satisfies:

$$\left|Z - \mathrm{tr}\big(f(A)\big)\right| \le \epsilon \cdot \mathrm{tr}\big(f(A)\big).$$

Proof: Combining Proposition 3 and Lemma 16, algorithm 2 returns an estimate $\tilde{S}_\alpha(A)$ using

$$s = O\left(\frac{1}{\epsilon_0}\sqrt{\log\left(\frac{1}{\delta}\right)} + \log\left(\frac{1}{\delta}\right)\right)$$

queries so that with confidence at least $1-\delta$

$$\left|\tilde{S}_\alpha(A) - S_\alpha(A)\right| \le \epsilon \cdot S_\alpha(A),$$

where $\epsilon_0 = 1 - \min(\mu, 1/\mu)^\epsilon$. We can get the convergence rate of s by further applying Proposition 15:

$$s = O\left(\frac{1}{\epsilon|\alpha-1|}\sqrt{\log\left(\frac{1}{\delta}\right)} + \log\left(\frac{1}{\delta}\right)\right).$$

For integer $\alpha \ge 2$, $|\alpha - 1| = \Theta(1)$. □

D. Proof of Theorem 4

Proof: Let $B = A/v - I_n$, $p_m(A) = v^\alpha \sum_{k=0}^m \binom{\alpha}{k} B^k$ and Z be the estimate of $\mathrm{tr}(p_m(A))$ using Hutch++ algorithm.

$$\begin{aligned}
&|\mathrm{tr}(p_m(A)) - \mathrm{tr}(A^\alpha)| \\
&= v^\alpha \left|\sum_{k=m+1}^{\infty} \binom{\alpha}{k} \mathrm{tr}(B^k)\right| \\
&\le v^\alpha C \left|\sum_{k=0}^{\infty} \binom{\alpha}{k} \mathrm{tr}(B^{m+1}B^k)\right| \qquad (9)\\
&\le C\left|\left(\frac{u}{v} - 1\right)^{m+1}\right| \mathrm{tr}(A^\alpha), \qquad (10)
\end{aligned}$$

where $C = \max_{k\in[0,\infty]} \left|\binom{\alpha}{\lceil\alpha\rceil+k+1} \Big/ \binom{\alpha}{k}\right|$.

(9) follows by noticing that:

$$\begin{aligned}
\left|\binom{\alpha}{m+k+1} \Big/ \binom{\alpha}{k}\right| &= \left|\prod_{i=k}^{m+k} \frac{\alpha - i}{i+1}\right| \le \left|\prod_{i=k}^{\lceil\alpha\rceil+k} \frac{\alpha - i}{i+1}\right| \\
&= \left|\binom{\alpha}{\lceil\alpha\rceil+k+1} \Big/ \binom{\alpha}{k}\right|,
\end{aligned}$$

$$\lim_{k\to\infty} \left|\prod_{i=k}^{\lceil\alpha\rceil+k} \frac{\alpha - i}{i+1}\right| = 1.$$

Thus C is a constant that depends only on α. (10) follows by the von Neumann's trace inequality, that for any two positive semi-definite matrices A and B, $\mathrm{tr}(AB) \le \sum_i \lambda_i(A)\lambda_i(B)$, where $\lambda_i(A)$ denotes the i-th singular value of A (the same for $\lambda_i(B)$). By noticing that $\binom{\alpha}{k}\mathrm{tr}(B^k)$, $k \in [m, \infty]$ are either all positive semi-definite or all negative semi-definite by assuming $m > \alpha$, we have:

$$\begin{aligned}
|\mathrm{tr}(B^{m+1}B^k)| &\le |u/v - 1|^{m+1} \sum_i \lambda_i^k(B) \\
&= |u/v - 1|^{m+1} |\mathrm{tr}(B^k)|.
\end{aligned}$$

Let $\epsilon_0 = 1 - \min(\mu, 1/\mu)^\epsilon$ and $\epsilon_1 = \frac{\epsilon_0}{3}$. by taking $m = O\left(\kappa \log\left(\frac{1}{\epsilon_0}\right)\right)$, we have

$$\begin{aligned}
C\left|\frac{u}{v} - 1\right|^{m+1} &< \frac{\epsilon_0}{2}, \qquad (11)\\
|\mathrm{tr}(p_m(A)) - \mathrm{tr}(A^\alpha)| &\le \frac{\epsilon_0}{2}\mathrm{tr}(A^\alpha),\\
\mathrm{tr}(p_m(A)) &\le \frac{\epsilon_0}{2}\mathrm{tr}(A^\alpha) + \mathrm{tr}(A^\alpha) \le \frac{3}{2}\mathrm{tr}(A^\alpha).
\end{aligned}$$

Additionally, noticing that

$$\begin{aligned}
\min_{\lambda\in[u,v]} p_m(\lambda) &\ge \min_{\lambda\in[u,v]} (\lambda^\alpha - |p_m(\lambda) - \lambda^\alpha|) \\
&\ge \min_{\lambda\in[u,v]} \left(\lambda^\alpha - C\left|\left(\frac{u}{v} - 1\right)^{m+1}\right|\lambda^\alpha\right) \qquad (12)\\
&\ge \min_{\lambda\in[u,v]} \left(\left(1 - \frac{\epsilon_0}{2}\right)\lambda^\alpha\right) > 0. \qquad (13)
\end{aligned}$$

(12) follows by taking A as a 1×1 matrix with entry λ in (10). (13) follows by applying (11). Therefore, $p_m(A)$ is positive semi-definite when m is large enough. By taking $s = O\left(\frac{1}{\epsilon_1}\sqrt{\log\left(\frac{1}{\delta}\right)} + \log\left(\frac{1}{\delta}\right)\right)$ in Lemma 16 we have:

$$|Z - \mathrm{tr}(p_m(A))| \le \frac{\epsilon_0}{3}\mathrm{tr}(p_m(A)).$$

Combining the above results:

$$\begin{aligned}
|Z - \mathrm{tr}(A^\alpha)| &\le |Z - \mathrm{tr}(p_m(A))| \\
&\quad + |\mathrm{tr}(p_m(A)) - \mathrm{tr}(A^\alpha)| \\
&\le \frac{\epsilon_0}{3}\mathrm{tr}(p_m(A)) + \frac{\epsilon_0}{2}\mathrm{tr}(A^\alpha) \\
&\le \frac{\epsilon_0}{2}\mathrm{tr}(A^\alpha) + \frac{\epsilon_0}{2}\mathrm{tr}(A^\alpha) \\
&= \epsilon_0 \cdot \mathrm{tr}(A^\alpha).
\end{aligned}$$

According to Proposition 3 and 15, we finally have:

$$\begin{aligned}
s &= O\left(\frac{1}{\epsilon|\alpha-1|}\sqrt{\log\left(\frac{1}{\delta}\right)} + \log\left(\frac{1}{\delta}\right)\right),\\
m &= O\left(\kappa \log\left(\frac{1}{\epsilon|\alpha-1|}\right)\right).
\end{aligned}$$

□

E. Proof of Theorem 6

Lemma 17 Theorem 2 in [58]: Let $\Gamma(x)$ be the gamma function and let $R(x,y) = \Gamma(x+y)/\Gamma(x)$, then

$$\begin{aligned}
R(x,y) &\ge x(x+y)^{y-1} \qquad for\ \ 0 \le y \le 1,\\
R(x,y) &\ge x^y \qquad for\ \ 1 \le y \le 2,\\
R(x,y) &\ge x(x+1)^{y-1} \qquad for\ \ y \ge 2.
\end{aligned}$$

Lemma 18 Theorem 5 in [15]: If Algorithm 1 is implemented with $s = O\left(\frac{1}{\epsilon}\sqrt{\log\left(\frac{1}{\delta}\right)} + \log\left(\frac{1}{\delta}\right)\right)$ matrix-vector multiplication queries, then for any matrix $f(A)$, with probability at least $1-\delta$, the output Z satisfies:

$$\left|Z - \mathrm{tr}\left(f(A)\right)\right| \leq \epsilon \cdot \|f(A)\|_*,$$

where $\|\cdot\|_*$ is the nuclear norm.

Proof: Let $p_m(\lambda) = v^\alpha \sum_{k=0}^{m} \binom{\alpha}{k}(\lambda/v - 1)^k$ and Z be the estimate of $\mathrm{tr}(p_m(A))$ using Hutch++ algorithm. Denote $E(\lambda)$ as the polynomial approximation error at point λ: $E(\lambda) = |p_m(\lambda) - \lambda^\alpha|$. By noticing that $\binom{\alpha}{k}(\lambda/v-1)^k$, $k \in [m, \infty]$ are either all positive or all negative for $\lambda \in [0, v]$ by assuming $m > \alpha$, we have:

$$\begin{aligned} E(\lambda) &= v^\alpha \left| \sum_{k=m+1}^{\infty} \binom{\alpha}{k} \left(\frac{\lambda}{v} - 1\right)^k \right| \\ &\leq v^\alpha \left| \sum_{k=m+1}^{\infty} \binom{\alpha}{k} (-1)^k \right| = E(0). \end{aligned}$$

From the property of binomial terms, we have that for any $\alpha > 0$ and integer $k > 1$, $\binom{\alpha}{k-1} + \binom{\alpha}{k} = \binom{\alpha+1}{k}$. Then

$$\begin{aligned} &\left(1 + \frac{\lambda}{v} - 1\right) \sum_{k=m}^{\infty} \binom{\alpha}{k} \left(\frac{\lambda}{v} - 1\right)^k \\ &= \sum_{k=m}^{\infty} \binom{\alpha}{k} \left(\frac{\lambda}{v} - 1\right)^k + \sum_{k=m+1}^{\infty} \binom{\alpha}{k-1} \left(\frac{\lambda}{v} - 1\right)^k \\ &= \binom{\alpha}{m} \left(\frac{\lambda}{v} - 1\right)^m + \sum_{k=m+1}^{\infty} \binom{\alpha+1}{k} \left(\frac{\lambda}{v} - 1\right)^k. \end{aligned}$$

Setting $\lambda = 0$ in the equation above, we have:

$$\binom{\alpha}{m}(-1)^m + \sum_{k=m+1}^{\infty} \binom{\alpha+1}{k}(-1)^k = 0.$$

Therefore:

$$\begin{aligned} E(0) &= v^\alpha \left| \sum_{k=m+1}^{\infty} \binom{\alpha}{k}(-1)^k \right| = v^\alpha \left| -\binom{\alpha-1}{m}(-1)^m \right| \\ &= v^\alpha \left| \binom{\alpha-1}{m} \right| = v^\alpha \left| \frac{\Gamma(\alpha)}{\Gamma(m+1)\Gamma(\alpha-m)} \right| \\ &\leq v^\alpha \left| \frac{\Gamma(\alpha)}{\Gamma(m-\alpha+1)\Gamma(\alpha-m)} \right| \\ &\quad \cdot \begin{cases} \dfrac{1}{(m-\alpha+1)(m+1)^{\alpha-1}} & 0 < \alpha < 1 \\ \dfrac{1}{(m-\alpha+1)^\alpha} & 1 < \alpha < 2 \\ \dfrac{1}{(m-\alpha+1)(m-\alpha+2)^{\alpha-1}} & \alpha \geq 2 \end{cases} \qquad (14) \\ &\leq \frac{v^\alpha \Gamma(\alpha)}{\pi} \cdot \frac{2}{(m-\alpha+1)^\alpha}. \qquad (15) \end{aligned}$$

(14) follows by applying Lemma 17 on $R(m-\alpha+1, \alpha)$. (15) follows by Euler's reflection formula that for any fractional number z, $\Gamma(z)\Gamma(1-z) = \pi/\sin \pi z$. And by assuming that $m \geq 1$, $(m+1)^{1-\alpha} \leq 2(m-\alpha+1)^{1-\alpha}$ when $\alpha < 1$.

Let $\epsilon_0 = 1 - \min(\mu, 1/\mu)^\epsilon$ and $\epsilon_1 = \frac{\epsilon_0}{3}$. By choosing m as:

$$\begin{aligned} \frac{2v^\alpha \Gamma(\alpha)}{\pi(m-\alpha+1)^\alpha} &\leq \frac{\epsilon_0}{2n} \cdot \mathrm{tr}(A^\alpha), \\ m &\geq \alpha - 1 + \sqrt[\alpha]{\frac{4nv^\alpha \Gamma(\alpha)}{\epsilon_0 \pi \min(v^{\alpha-1}, n^{1-\alpha})}}. \end{aligned}$$

Let $\lambda_1, \cdots, \lambda_n$ be the eigenvalues of A, then

$$\begin{aligned} |\mathrm{tr}(p_m(A)) - \mathrm{tr}(A^\alpha)| &\leq \sum_{i=1}^{n} E(\lambda_i) \leq nE(0) \\ &\leq \frac{2nv^\alpha \Gamma(\alpha)}{\pi(m-\alpha+1)^\alpha} \\ &\leq \frac{\epsilon_0}{2} \cdot \mathrm{tr}(A^\alpha). \end{aligned}$$

Taking $s = O\left(\frac{1}{\epsilon_1}\sqrt{\log\left(\frac{1}{\delta}\right)} + \log\left(\frac{1}{\delta}\right)\right)$ in Lemma 18 we have:

$$\begin{aligned} |Z - \mathrm{tr}(p_m(A))| &\leq \frac{\epsilon_0}{3}\|p_m(A)\|_* = \frac{\epsilon_0}{3}\sum_{i=1}^{n}|p_m(\lambda_i)| \\ &\leq \frac{\epsilon_0}{3}\left(\sum_{i=1}^{n}\lambda_i^\alpha + \sum_{i=1}^{n}|p_m(\lambda_i) - \lambda_i^\alpha|\right) \\ &= \frac{\epsilon_0}{3}\left(\mathrm{tr}(A^\alpha) + \sum_{i=1}^{n} E(\lambda_i)\right) \\ &\leq \frac{\epsilon_0}{3}\left(\mathrm{tr}(A^\alpha) + \frac{\epsilon_0}{2}\mathrm{tr}(A^\alpha)\right) \\ &\leq \frac{\epsilon_0}{2} \cdot \mathrm{tr}(A^\alpha). \end{aligned}$$

Combining the results we get so far:

$$\begin{aligned} |Z - \mathrm{tr}(A^\alpha)| &\leq |Z - \mathrm{tr}(p_m(A))| \\ &\quad + |\mathrm{tr}(p_m(A)) - \mathrm{tr}(A^\alpha)| \\ &\leq \frac{\epsilon_0}{2}\mathrm{tr}(A^\alpha) + \frac{\epsilon_0}{2}\mathrm{tr}(A^\alpha) \\ &= \epsilon_0 \cdot \mathrm{tr}(A^\alpha). \end{aligned}$$

Applying Proposition 3 and 15, we finally have:

$$\begin{aligned} s &= O\left(\frac{1}{\epsilon|\alpha-1|}\sqrt{\log\left(\frac{1}{\delta}\right)} + \log\left(\frac{1}{\delta}\right)\right), \\ m &= \begin{cases} O\left(\sqrt[\alpha]{vn}\sqrt[\alpha]{\dfrac{1}{\epsilon|\alpha-1|}}\right) & \alpha < 1 \\ O\left(vn\sqrt[\alpha]{\dfrac{1}{\epsilon|\alpha-1|}}\right) & \alpha > 1 \end{cases}. \end{aligned}$$

□

F. Proof of Theorem 8

The following lemma gives the upper bound of Chebyshev series approximation.

Lemma 19 Theorem 2.1 in [59]: Suppose f is analytic with $|f(z)| \leq M$ in the region bounded by the ellipse with foci ± 1 and major and minor semi-axis lengths summing to $K > 1$.

Let p_m denote the Chebyshev polynomial approximation of f with degree m, then for any $m \in \mathbb{Z}^+$:

$$\max_{\lambda\in[-1,1]} |f(\lambda) - p_m(\lambda)| \leq \frac{4M}{(K-1)K^m}.$$

By selecting an appropriate analytic region, we are able to establish the error bound of approximating $f(\lambda) = \lambda^\alpha$.

Proposition 20: Let g be the linear mapping $[-1,1] \to [u,v]$, $f(\lambda) = \lambda^\alpha$ be the target function, $p_m(\lambda)$ be the Chebyshev series of degree $m = O\big(\sqrt{\frac{v}{u}}\log\big(\frac{v}{u\epsilon}\big)\big)$ for function $f \circ g$, then the following inequality holds:

$$\begin{aligned}&\max_{x\in[-1,1]} |(f\circ g)(x) - p_m(x)| \\ &\quad = \max_{\lambda\in[u,v]} |f(\lambda) - q_m(\lambda)| \leq \epsilon u^\alpha,\end{aligned}$$

where $q_m = p_m \circ g^{-1}$.

Proof: It is easy to show that the power function $f(z) = z^\alpha$ is analytic in $\mathbb{C}/\{-\infty, 0\}$. Combining with linear mapping g, the function $f \circ g$ is analytic in the region $\mathbb{C}/\{-\infty, -1 - \frac{2u}{v-u}\}$. Therefore we choose the ellipse region E_c with major semi-axis length $1 + \frac{2u}{v-u} = 1 + \beta$, minor semi-axis length $\sqrt{(\beta+1)^2 - 1} = \sqrt{\beta^2 + 2\beta}$ and foci at ± 1.

We then apply Lemma 19 with $K = 1 + \beta + \sqrt{\beta^2 + 2\beta}$ and $M = (1+\beta)^\alpha$. By noticing that $\log K = \Theta\big(\sqrt{\beta}\big)$, we get the upper bound of m:

$$m \geq \frac{\log\Big(\frac{4M}{(K-1)\epsilon u^\alpha}\Big)}{\log K} = O\bigg(\sqrt{\frac{v}{u}}\log\Big(\frac{v}{u\epsilon}\Big)\bigg).$$

□

Proof: By taking $m = O\Big(\sqrt{\frac{v}{u}}\log\Big(\frac{v}{u\epsilon_1}\Big)\Big)$ in Proposition 20, where $\epsilon_1 = \epsilon_0/2$ and $\epsilon_0 = 1 - \min(\mu, 1/\mu)^\epsilon$, we have:

$$\begin{aligned}&\max_{\lambda\in[u,v]} |f(\lambda) - q_m(\lambda)| \leq \epsilon_1 u^\alpha, \\ &|\mathrm{tr}(q_m(A)) - \mathrm{tr}(A^\alpha)| \leq \sum_{i=1}^n |f(\lambda_i) - q_m(\lambda_i)| \\ &\qquad\qquad \leq n\epsilon_1 u^\alpha \leq \frac{\epsilon_0}{2}\cdot \mathrm{tr}(A^\alpha).\end{aligned}$$

Additionally, noticing that

$$\begin{aligned}\min_{\lambda\in[u,v]} q_m(\lambda) &\geq \min_{\lambda\in[u,v]} \lambda^\alpha - \max_{\lambda\in[u,v]} |\lambda^\alpha - q_m(\lambda)| \\ &\geq u^\alpha - \frac{\epsilon_0}{2}u^\alpha \geq 0.\end{aligned}$$

Therefore, $q_m(A)$ is positive semi-definite when m is large enough. By taking $s = O\Big(\frac{1}{\epsilon_2}\sqrt{\log\big(\frac{1}{\delta}\big)} + \log\big(\frac{1}{\delta}\big)\Big)$ in Lemma 16 where $\epsilon_2 = \frac{\epsilon_0}{3}$ we have:

$$\begin{aligned}&|Z - \mathrm{tr}(q_m(A))| \leq \frac{\epsilon_0}{3}\mathrm{tr}(q_m(A)), \\ &\mathrm{tr}(q_m(A)) \leq \frac{\epsilon_0}{2}\mathrm{tr}(A^\alpha) + \mathrm{tr}(A^\alpha) \leq \frac{3}{2}\mathrm{tr}(A^\alpha),\end{aligned}$$

where Z is the estimate of $\mathrm{tr}(q_m(A))$ using Hutch++ algorithm. Combining the results we get so far:

$$\begin{aligned}|Z - \mathrm{tr}(A^\alpha)| &\leq |Z - \mathrm{tr}(q_m(A))| \\ &\quad + |\mathrm{tr}(q_m(A)) - \mathrm{tr}(A^\alpha)| \\ &\leq \frac{\epsilon_0}{3}\mathrm{tr}(q_m(A)) + \frac{\epsilon_0}{2}\mathrm{tr}(A^\alpha) \\ &\leq \frac{\epsilon_0}{2}\mathrm{tr}(A^\alpha) + \frac{\epsilon_0}{2}\mathrm{tr}(A^\alpha) \\ &\leq \epsilon_0 \cdot \mathrm{tr}(A^\alpha).\end{aligned}$$

Applying Proposition 3 and 15, we finally have:

$$\begin{aligned}s &= O\left(\frac{1}{\epsilon|\alpha-1|}\sqrt{\log\left(\frac{1}{\delta}\right)} + \log\left(\frac{1}{\delta}\right)\right), \\ m &= O\left(\sqrt{\kappa}\log\left(\frac{\kappa}{\epsilon|\alpha-1|}\right)\right).\end{aligned}$$

□

G. Proof of Theorem 10

Proof: When $u = 0$, the coefficients of Chebyshev series $\hat{T}_k$ have analytical expressions:

$$\begin{aligned}c_k &= \frac{2}{\pi}\int_0^\pi (q_m)^\alpha(cos\theta)\cos(k\theta)\,\mathrm{d}\theta \\ &= \frac{2}{\pi}\int_0^\pi \Big(\frac{v}{2}(\cos\theta + 1)\Big)^\alpha \cos(k\theta)\,\mathrm{d}\theta \\ &= \frac{2v^\alpha\Gamma(\alpha+\frac{1}{2})(\alpha)_k}{\sqrt{\pi}\Gamma(\alpha+1)(\alpha+k)_k}.\end{aligned}$$

where $(\alpha)_k$ is the falling factorial: $(\alpha)_k = \alpha\cdot(\alpha-1)\cdot\ldots\cdot(\alpha-k+1)$. Then for each eigenvalue λ of A:

$$\begin{aligned}&|\lambda^\alpha - q_m(\lambda)| = \Big|\sum_{i=m+1}^\infty c_i\hat{T}_i(\lambda)\Big| \\ &\leq \sum_{i=m+1}^\infty |c_i| = \sum_{i=m+1}^\infty \left|\frac{2v^\alpha\Gamma(\alpha+\frac{1}{2})(\alpha)_i}{\sqrt{\pi}\Gamma(\alpha+1)(\alpha+i)_i}\right| \qquad (16) \\ &= \frac{2v^\alpha}{\sqrt{\pi}}\sum_{i=m+1}^\infty \left|\frac{\Gamma(\alpha+\frac{1}{2})\Gamma(\alpha+1)}{\Gamma(\alpha+i+1)\Gamma(\alpha-i+1)}\right| \\ &\leq \frac{2v^\alpha}{\sqrt{\pi}}\sum_{i=m+1}^\infty \left|\frac{\Gamma(\alpha+\frac{1}{2})\Gamma(\alpha+1)}{\Gamma(i-\alpha)\Gamma(\alpha-i+1)(i-\alpha)^{2\alpha+1}}\right| \qquad (17) \\ &\leq \frac{2v^\alpha\Gamma(\alpha+\frac{1}{2})\Gamma(\alpha+1)}{\pi^{3/2}}\sum_{i=m+1}^\infty \left|\frac{1}{(i-\alpha)^{2\alpha+1}}\right| \qquad (18) \\ &\leq \frac{2v^\alpha\Gamma(\alpha+\frac{1}{2})\Gamma(\alpha+1)}{\pi^{3/2}}\int_m^\infty \frac{1}{(x-\alpha)^{2\alpha+1}}\,\mathrm{d}x \qquad (19) \\ &= \frac{2v^\alpha\Gamma(\alpha+\frac{1}{2})\Gamma(\alpha+1)}{\pi^{3/2}}\frac{1}{2\alpha(m-\alpha)^{2\alpha}} \\ &= \frac{v^\alpha\Gamma(\alpha+\frac{1}{2})\Gamma(\alpha)}{\pi^{3/2}(m-\alpha)^{2\alpha}}.\end{aligned}$$

(16) follows by noticing that $\hat{T}_n(x) \in [-1,1]$ for any $x \in [0,v]$. (17) follows by applying Lemma 17 on $R(i-\alpha, 2\alpha+1)$ similar to (14). (18) follows by Euler's reflection formula similar to (15). (19) follows by assuming

$m > \alpha$ and noticing that $n^{-k} \le \int_{n-1}^{n} x^{-k}\,\mathrm{d}x$ for $n > 1$ and $k > 1$.

Let $\epsilon_0 = 1 - \min(\mu, 1/\mu)^{\epsilon}$ and $\epsilon_1 = \frac{\epsilon_0}{3}$. By choosing m as:

$$\frac{nv^{\alpha}\Gamma(\alpha+\frac{1}{2})\Gamma(\alpha)}{\pi^{3/2}(m-\alpha)^{2\alpha}} \le \frac{\epsilon_0}{2}\cdot \mathrm{tr}(A^{\alpha}),$$

$$m \ge \alpha + \sqrt[2\alpha]{\frac{2nv^{\alpha}\Gamma(\alpha+\frac{1}{2})\Gamma(\alpha)}{\epsilon_0\pi^{3/2}\min(v^{\alpha-1}, n^{1-\alpha})}}.$$

Let $\lambda_1, \cdots, \lambda_n$ be the eigenvalues of A, then

$$|\mathrm{tr}(q_m(A)) - \mathrm{tr}(A^{\alpha})| \le \frac{\epsilon_0}{2}\cdot \mathrm{tr}(A^{\alpha}),$$

Taking $s = O\Big(\frac{1}{\epsilon_1}\sqrt{\log\big(\frac{1}{\delta}\big)} + \log\big(\frac{1}{\delta}\big)\Big)$ in Lemma 18 we have:

$$\begin{aligned}|Z - \mathrm{tr}(q_m(A))| &\le \frac{\epsilon_0}{3}\|q_m(A)\|_* = \frac{\epsilon_0}{3}\sum_{i=1}^{n}|q_m(\lambda_i)| \\ &\le \frac{\epsilon_0}{3}\left(\sum_{i=1}^{n}\lambda_i^{\alpha} + \sum_{i=1}^{n}|\lambda_i^{\alpha} - q_m(\lambda_i)|\right) \\ &\le \frac{\epsilon_0}{3}\Big(\mathrm{tr}(A^{\alpha}) + \frac{\epsilon_0}{2}\mathrm{tr}(A^{\alpha})\Big) \\ &\le \frac{\epsilon_0}{2}\cdot \mathrm{tr}(A^{\alpha}).\end{aligned}$$

Combining the results we get so far:

$$\begin{aligned}|Z - \mathrm{tr}(A^{\alpha})| &\le |Z - \mathrm{tr}(p_m(A))| \\ &\quad + |\mathrm{tr}(p_m(A)) - \mathrm{tr}(A^{\alpha})| \\ &\le \frac{\epsilon_0}{2}\mathrm{tr}(A^{\alpha}) + \frac{\epsilon_0}{2}\mathrm{tr}(A^{\alpha}) \\ &= \epsilon_0\cdot \mathrm{tr}(A^{\alpha}).\end{aligned}$$

Applying Proposition 3 and 15, we finally have:

$$s = O\left(\frac{1}{\epsilon|\alpha-1|}\sqrt{\log\left(\frac{1}{\delta}\right)} + \log\left(\frac{1}{\delta}\right)\right),$$

$$m = \begin{cases} O\left(\sqrt[2\alpha]{vn}\sqrt[2\alpha]{\dfrac{1}{\epsilon|\alpha-1|}}\right) & \alpha < 1 \\ O\left(\sqrt{vn}\sqrt[2\alpha]{\dfrac{1}{\epsilon|\alpha-1|}}\right) & \alpha > 1 \end{cases}.$$

□

H. Proof of Theorem 11

Lemma 21: Theorem III-B in [15] Any algorithm that accesses a positive semi-definite matrix A via matrix-vector multiplication queries $Ar_1, \cdots, Ar_m$, where $r_1, \cdots, r_m$ are possibly adaptively chosen vectors with integer entries in $[-2^b, \cdots, 2^b]$, requires $s = \Omega\Big(\frac{1}{\epsilon(b+\log(1/\epsilon))}\Big)$ such queries to output an estimate Z so that, with probability $> \frac{2}{3}$, $(1-\epsilon)\,\mathrm{tr}(A) \le Z \le (1+\epsilon)\,\mathrm{tr}(A)$.

Proof: Combining Lemma 21 and Proposition 3, we have that the lower bound of estimating $S_{\alpha}(A)$ to relative error $1\pm\epsilon$ with probability at least $\frac{2}{3}$ is

$$s = \Omega\Big(\frac{1}{\epsilon_0(b+\log(1/\epsilon_0))}\Big),$$

where $\epsilon_0 = n^{\epsilon|\alpha-1|} - 1 \le \epsilon|1-\alpha|\log n$.

In limited precision computation settings, b is some constant value, then we finally get

$$s = \Omega\left(\frac{1}{\epsilon|\alpha-1|\log n \log\Big(\frac{1}{\epsilon|\alpha-1|\log n}\Big)}\right).$$

□

I. Proof of Theorem 13

The following lemma gives the convergence rate of function $f(x) = (\gamma - x)^{-t}$ in best uniform approximation. It is proved in [25] pp.38-39 and [26] pp.102-103.

Lemma 22: Let $\|\cdot\|$ denote the L_{∞} norm of functions and $E_m(f) = \min_{p\in\mathbb{P}_m}\|f-p\|$ be the error of best uniform approximation of a given function $f(x)$ on the finite interval $[-1, 1]$. Then when $m \to \infty$,

$$E_m\big((\gamma-x)^{-t}\big) \sim \frac{m^{t-1}}{|\Gamma(t)|}\frac{\Big(\gamma - \sqrt{\gamma^2-1}\Big)^m}{\Big(\sqrt{\gamma^2-1}\Big)^{1+t}},$$

where $t, \gamma \in \mathbb{R}$ and $\gamma > 1$.

Proposition 23: There exists a positive decreasing function $\epsilon_0 : \mathbb{R}^+ \to \mathbb{R}^+$ such that for arbitrary $0 < u < v < 1$ and $\epsilon \in (0, \epsilon_0(v/u))$, any polynomial $q_m(\lambda)$ that approximates function $f(\lambda) = \lambda^{\alpha}$, requires $m = \Omega\big(\sqrt{\frac{v}{u}}\log\big(\frac{u}{v\epsilon}\big)\big)$ degree to achieve $|\sum_{i=1}^{n}(f(\lambda_i) - q_m(\lambda_i))| \le \epsilon$, for arbitrary real numbers $\lambda_1, \cdots, \lambda_n \in [u, v]$ that satisfy $\sum_{i=1}^{n}\lambda_i \in [b, b+v)$, where $b \ge v$ is a constant number.

Proof: Under the same assumptions, We list the following problems for polynomial approximation. We claim that each of them could be reduced to the next problem in sequence.

Problem 24: The minimum degree m required for any polynomial q_m to achieve $|\sum_{i=1}^{n}(f(\lambda_i) - q_m(\lambda_i))| \le \epsilon$ for any $\lambda_1, \cdots, \lambda_n \in [u, v]$ that satisfy $\sum_{i=1}^{n}\lambda_i \in [b, b+v)$.

Problem 25: The minimum degree m required for any polynomial q_m to achieve $\sum_{i=1}^{n}|f(\lambda_i) - q_m(\lambda_i)| \le \epsilon$ for any $\lambda_1, \cdots, \lambda_n \in [u, v]$ that satisfy $\sum_{i=1}^{n}\lambda_i = b$.

Problem 26: The minimum degree m required for any polynomial q_m to achieve $|f(\lambda) - q_m(\lambda)|\phi(\lambda) \le \epsilon$ for any $\lambda \in [u, v]$, where $\phi(\lambda) = \lfloor\min(\frac{nv-b}{\lambda-u}, \frac{b-nu}{v-\lambda})\rfloor$.

Problem 27: The minimum degree m required for any polynomial q_m to achieve $|f(\lambda) - q_m(\lambda)| \le \epsilon$ for any $\lambda \in [u, v]$.

Problem 28: The minimum degree m required for any polynomial q_m to achieve $|f(\lambda) - q_m(\lambda)| \le \epsilon$ for any $\lambda \in [u, u+2]$.

For Problem 28, approximating $(f \circ g)(x) = (x+u+1)^{\alpha}$ is equivalent to approximating $(\gamma - x)^{-t}$ with $\gamma = u+1$ and $t = -\alpha$, since they are symmetric about y-axis. Let $\epsilon = E_m(f \circ g)$, with the following property of the gamma function

$$|\Gamma(-\alpha)| = \frac{\pi}{|\sin\pi\alpha|\Gamma(\alpha+1)},$$

by applying Lemma 22, when m is large enough, we have:

$$\lim_{m\to\infty}\frac{\left(1+u-\sqrt{u^2+2u}\right)^m}{m^{\alpha+1}}=\frac{\epsilon|\Gamma(-\alpha)|}{\left(\sqrt{u^2+2u}\right)^{\alpha-1}}=\Theta\left(\frac{\epsilon}{(\sqrt{u})^{\alpha-1}|sin\pi\alpha|}\right).$$

Thus, for each $u\in(0,1)$, there is an $\epsilon_0\in(0,1)$, such that when $\epsilon<\epsilon_0$:

$$m=\Omega\left(\frac{1}{\sqrt{u}}\log\left(\frac{u|\sin\pi\alpha|}{\epsilon}\right)\right).$$

When $\alpha\notin\mathbb{N}$, we have $|\sin\pi\alpha|=\Theta(1)$.

Problem 27 $\rightarrow$ 28: Noticing that an approximation of $f_{[u,u+2]}(\lambda)$ could be acquired by approximating $\left(\frac{u+2}{v}\right)^{\alpha}f_{[\frac{uv}{u+2},v]}\left(\frac{v}{u+2}\lambda\right)$, The lower bound of m for approximating $f_{[\frac{uv}{u+2},v]}$ is then

$$m=\Omega\left(\frac{1}{\sqrt{u}}\log\left(\frac{u}{v\epsilon}\right)\right).$$

This is equivalent to approximating $f_{[u,v]}$ with

$$m=\Omega\left(\sqrt{\frac{v}{u}}\log\left(\frac{u}{v\epsilon}\right)\right).$$

Problem 26 $\rightarrow$ 27: We assume $u<\frac{b-v}{n-1}$ and $v>\frac{b-u}{n-1}$ without loss of generality, otherwise it means $\min_i\lambda_i>u$ or $\max_i\lambda_i<v$, and the interval $[u,v]$ could be make tighter to fit this assumption. Then for any $\lambda\in[u,v]$,

$$\phi(\lambda)=\left\lfloor\min\left(\frac{nv-b}{v-\lambda},\frac{b-nu}{\lambda-u}\right)\right\rfloor\geq\left\lfloor\min\left(\frac{nv-b}{v-u},\frac{b-nu}{v-u}\right)\right\rfloor\geq 1.$$

Problem 25 $\rightarrow$ 26: For any $\lambda\in[u,v]$, we construct a series of numbers $\lambda_1,\cdots,\lambda_n$ by setting $\lambda_1,\cdots,\lambda_{n_\lambda}=\lambda$ and $\lambda_{n_\lambda+1},\cdots,\lambda_n=\frac{b-\lambda n_\lambda}{n-n_\lambda}$, where $n_\lambda=\lfloor\min(\frac{nv-b}{\lambda-u},\frac{b-nu}{v-\lambda})\rfloor$. Then the series satisfies $\sum_{i=1}^n\lambda_i=b$ and

$$\sum_{i=1}^n|f(\lambda_i)-q_m(\lambda_i)|\leq\epsilon,$$
$$n_\lambda|f(\lambda)-q_m(\lambda)|\leq\epsilon,$$
$$\phi(\lambda)|f(\lambda)-q_m(\lambda)|\leq\epsilon.$$

Problem 24 $\rightarrow$ 25: Let q_m be the solution for Problem 24 that achieves $\epsilon/2$ approximation error. In the trivial case, we have $f(\lambda)\leq q_m(\lambda)$ for all $\lambda\in[u,v]$ (or $f(\lambda)\geq q_m(\lambda)$), then $\sum_{i=1}^n|f(\lambda_i)-q_m(\lambda_i)|=|\sum_{i=1}^n(f(\lambda_i)-q_m(\lambda_i))|$.

Otherwise there exists some $\rho\in(u,v)$ that satisfies $f(\rho)=q_m(\rho)$, since both $f(\lambda)$ and $q_m(\lambda)$ are continuous functions. Given arbitrary query $\lambda_1,\cdots,\lambda_n$, there exists a partition n_ρ of the reordered numbers λ_i so that for all $i\in[1,n_\rho]$, $f(\lambda_i)\leq q_m(\lambda_i)$ and for all $i\in[n_\rho+1,n]$, $f(\lambda_i)>q_m(\lambda_i)$.

$$\sum_{i=1}^{n_\rho}|f(\lambda_i)-q_m(\lambda_i)|=\left|\sum_{i=1}^{n_\rho}(f(\lambda_i)-q_m(\lambda_i))\right|,$$
$$\sum_{i=n_\rho+1}^{n}|f(\lambda_i)-q_m(\lambda_i)|=\left|\sum_{i=n_\rho+1}^{n}(f(\lambda_i)-q_m(\lambda_i))\right|.$$

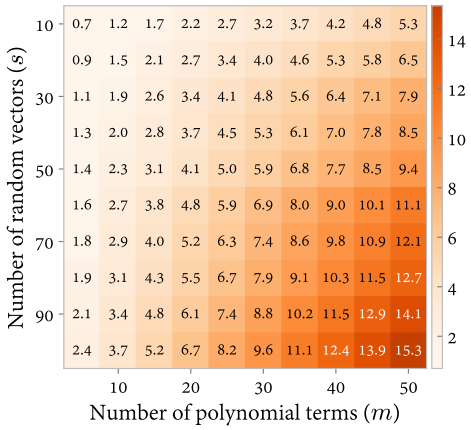

Fig. 6. Running time of different s and m combinations.

Construct two queries: $\lambda_1^1,\cdots,\lambda_{n_1}^1$ and $\lambda_1^2,\cdots,\lambda_{n_2}^2$:

$$\lambda_i^1=\begin{cases}\lambda_i & i\leq n_\rho\\ \rho & i>n_\rho\end{cases},\quad n_1=n_\rho+\left\lceil\sum_{i=n_\rho+1}^{n}\lambda_i/\rho\right\rceil.$$

$$\lambda_i^2=\begin{cases}\lambda_{i+n_\rho} & i\leq n-n_\rho\\ \rho & i>n-n_\rho\end{cases},\quad n_2=n-n_\rho+\left\lceil\sum_{i=1}^{n_\rho}\lambda_i/\rho\right\rceil.$$

Let $b=\sum_{i=1}^n\lambda_i$, then $\sum_{i=1}^{n_1}\lambda_i^1\in[b,b+v)$ and $\sum_{i=1}^{n_2}\lambda_i^2\in[b,b+v)$. Therefore

$$\begin{aligned}&\sum_{i=1}^n|f(\lambda_i)-q_m(\lambda_i)|\\&=\sum_{i=1}^{n_\rho}|f(\lambda_i)-q_m(\lambda_i)|+\sum_{i=n_\rho+1}^{n}|f(\lambda_i)-q_m(\lambda_i)|\\&=\left|\sum_{i=1}^{n_\rho}(f(\lambda_i)-q_m(\lambda_i))\right|+\left|\sum_{i=n_\rho+1}^{n}(f(\lambda_i)-q_m(\lambda_i))\right|\\&=\left|\sum_{i=1}^{n_1}(f(\lambda_i^1)-q_m(\lambda_i^1))\right|+\left|\sum_{i=1}^{n_2}(f(\lambda_i^2)-q_m(\lambda_i^2))\right|\\&\leq\frac{\epsilon}{2}+\frac{\epsilon}{2}=\epsilon.\end{aligned}$$

From the reductions above, the lower bound of m for solving Problem 24 is $m=\Omega(\sqrt{\frac{v}{u}}\log(\frac{u}{v\epsilon}))$. □

Proof: Let $Z=\mathrm{tr}(q_m(A))$ be the trace of the approximated matrix functional, then

$$|Z-\mathrm{tr}(A^\alpha)|=\left|\sum_{i=1}^{n}(q_m(\lambda_i)-\lambda_i^\alpha)\right|.$$

Let $\epsilon_0=n^{\epsilon|\alpha-1|}-1$, then by applying Proposition 23, we have that $m=\Omega\left(\sqrt{\frac{v}{u}}\log\left(\frac{u}{v\epsilon_0}\right)\right)$ is the lower bound to achieve $|Z-\mathrm{tr}(A^\alpha)|\leq\epsilon_0$ with $b=1$. Combining with Proposition 3, the lower bound for matrix function approximation is

$$\begin{aligned}m&=\Omega\left(\sqrt{\frac{v}{u}}\log\left(\frac{u}{v\epsilon_0}\right)\right)\\&=\Omega\left(\sqrt{\frac{v}{u}}\log\left(\frac{u}{v\epsilon|\alpha-1|\log n}\right)\right).\end{aligned}$$

□

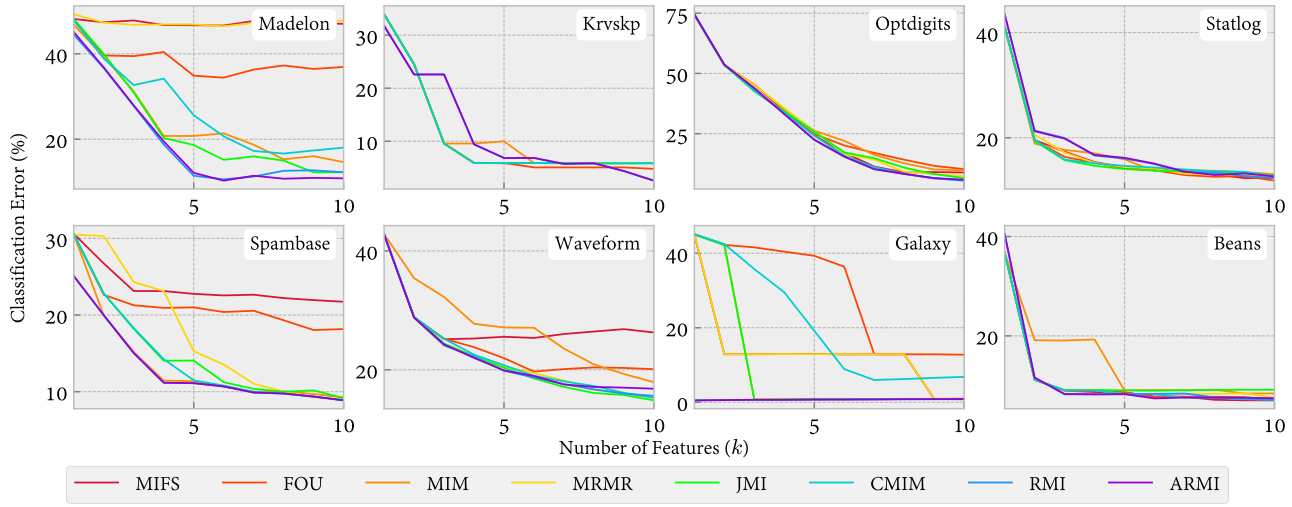

Fig. 7. Number of features (k) versus classification error (%) curves for different feature selection methods.

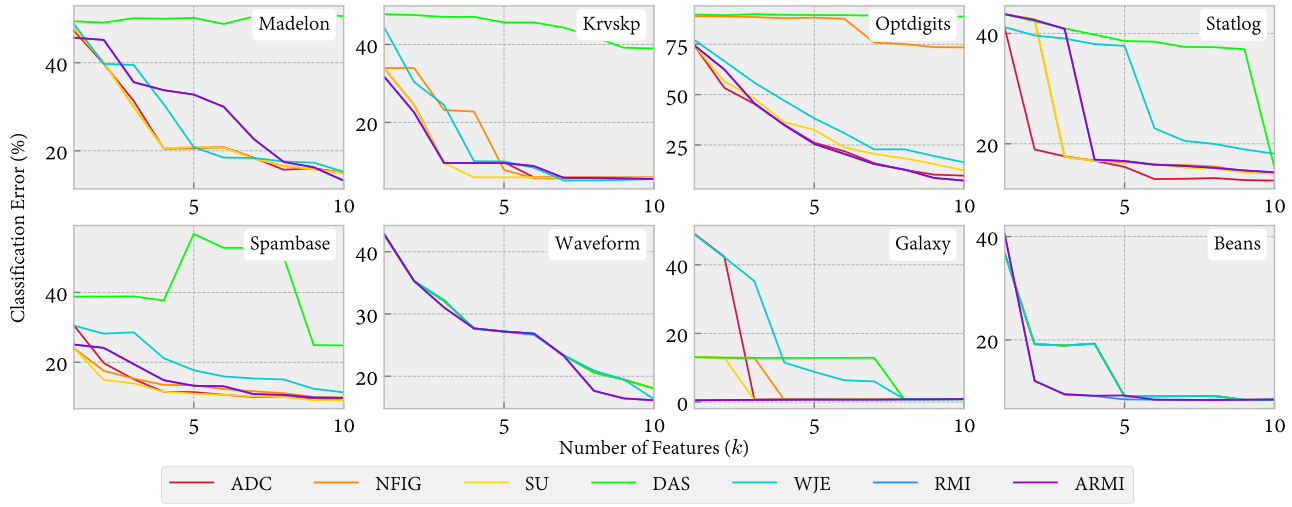

Fig. 8. Number of features (k) versus classification error (%) curves for different feature ranking methods.

J. Proof of Theorem 14

Similarly, the following lemma gives the convergence rate of function $f(x) = |x|^\alpha$ in the best uniform approximation. It is proved in [27] and [28].

Lemma 29: Let $\|\cdot\|$ denote the L_∞ norm of functions and $E_m(f) = \min_{p\in\mathbb{P}_m}\|f-p\|$ be the error of best uniform approximation of a given function $f(x)$ on the finite interval $[-1,1]$. Then when $m\to\infty$,

$$E_m(x^\alpha) \sim \delta(\alpha) m^{-2\alpha},$$

where $\alpha \in \mathbb{R}^+$, $\delta(\alpha)$ is a non-negative constant number depending only on α, and satisfies $\delta(\alpha) > 0$ when $\alpha \notin \mathbb{N}$.

Proposition 30: For arbitrary $v > 0$ and small enough ϵ, any polynomial $q_m(\lambda)$ that approximates $f(\lambda) = \lambda^\alpha$ requires $m = \Omega\left(\sqrt[2\alpha]{\frac{1}{\epsilon}}\right)$ degree to achieve $|\sum_{i=1}^n (f(\lambda_i) - q_m(\lambda_i))| \le \epsilon$ for arbitrary real numbers $\lambda_1, \cdots, \lambda_n \in [0, v]$ that satisfy $\sum_{i=1}^n \lambda_i \in [b, b+v)$, where $b \ge v$ is a constant number.

Proof: We can prove Proposition 30 through a similar procedure as the proof of 23. □

Proof: Let $Z = \mathrm{tr}(q_m(A))$ be the trace of the approximated matrix functional and $\epsilon_0 = n^{\epsilon|\alpha-1|} - 1$. From Proposition 30 we know that $m = \Omega\left(\sqrt[2\alpha]{\frac{1}{\epsilon_0}}\right)$ is the lower bound to achieve $|Z - \mathrm{tr}(A^\alpha)| \le \epsilon_0$ with $b = 1$. Then by applying Proposition 3, we get the lower bound

$$m = \Omega\left(\sqrt[2\alpha]{\frac{1}{\epsilon_0}}\right) = \Omega\left(\sqrt[2\alpha]{\frac{1}{\epsilon|\alpha-1|\log n}}\right).$$

Let $Z = \mathrm{tr}(q_m(A))$ be the trace of the approximated matrix functional, then

$$|Z - \mathrm{tr}(A^\alpha)| = \left|\sum_{i=1}^n (q_m(\lambda_i) - \lambda_i^\alpha)\right|.$$

Let $\epsilon_0 = n^{\epsilon|\alpha-1|} - 1$, then by applying Proposition 23, we have that $m = \Omega\left(\sqrt{\frac{v}{u}}\log\left(\frac{u}{v\epsilon_0}\right)\right)$ is the lower bound to achieve $|Z - \mathrm{tr}(A^\alpha)| \le \epsilon_0$ with $b = 1$. Combining with Proposition 3,

the lower bound for matrix function approximation is

$$m = \Omega\left(\sqrt{\frac{v}{u}}\log\left(\frac{u}{v\epsilon_0}\right)\right) = \Omega\left(\sqrt{\frac{v}{u}}\log\left(\frac{u}{v\epsilon|\alpha-1|\log n}\right)\right).$$

□

SUPPLEMENTARY EXPERIMENTAL RESULTS

A. Running Time of Fractional Approximations

An intuitive showcase of running time with different s and m combinations is shown in Figures 6, in which we can observe the linear increase in time complexity with s or m. For comparison, the trivial eigenvalue approach takes 27 seconds for a 5000×5000 matrix.

B. Accuracy Curves of Feature Selection

The classification accuracy achieved by each feature selection and feature ranking method after each incrementally selected feature is reported in Figure 7 and 8 respectively. It is easy to see that classification error stops decreasing after the first 10 most informative features are selected.

REFERENCES

[1] A. Rényi, "On measures of entropy and information," in *Proc. 4th Berkeley Symp. Math. Statist. Probab. Contrib. Theory Statist.*, vol. 1. Berkeley, CA, USA: Univ. California Press, 1961, pp. 547–561.

[2] J. C. Principe, *Information Theoretic Learning: Rényi's Entropy and Kernel Perspectives*. New York, NY, USA: Springer, 2010.

[3] A. Teixeira, A. Matos, and L. Antunes, "Conditional Rényi entropies," *IEEE Trans. Inf. Theory*, vol. 58, no. 7, pp. 4273–4277, Apr. 2012.

[4] L. G. S. Giraldo and J. C. Principe, "Information theoretic learning with infinitely divisible kernels," 2013, *arXiv:1301.3551*.

[5] S. Yu, F. Alesiani, X. Yu, R. Jenssen, and J. Principe, "Measuring dependence with matrix-based entropy functional," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 12, 2021, pp. 10781–10789.

[6] J. Fan and R. Li, "Statistical challenges with high dimensionality: Feature selection in knowledge discovery," in *Proc. Int. Congr. Mathematicians*, Madrid, Spain, Aug. 2006, pp. 595–622.

[7] L. G. S. Giraldo, M. Rao, and J. C. Principe, "Measures of entropy from data using infinitely divisible kernels," *IEEE Trans. Inf. Theory*, vol. 61, no. 1, pp. 535–548, Jan. 2015.

[8] M. Müller-Lennert, F. Dupuis, O. Szehr, S. Fehr, and M. Tomamichel, "On quantum Rényi entropies: A new generalization and some properties," *J. Math. Phys.*, vol. 54, no. 12, Dec. 2013, Art. no. 122203.

[9] A. J. Brockmeier, T. Mu, S. Ananiadou, and J. Y. Goulermas, "Quantifying the informativeness of similarity measurements," *J. Mach. Learn. Res.*, vol. 18, pp. 1–61, Jul. 2017.

[10] S. Yu, L. G. S. Giraldo, R. Jenssen, and J. C. Principe, "Multivariate extension of matrix-based Rényi's α-order entropy functional," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 11, pp. 2960–2966, Mar. 2019.

[11] C. H. Sarvani, M. Ghorai, S. R. Dubey, and S. H. S. Basha, "HRel: Filter pruning based on high relevance between activation maps and class labels," *Neural Netw.*, vol. 147, pp. 186–197, Mar. 2022.

[12] R. Miles, A. L. Rodríguez, and K. Mikolajczyk, "Information theoretic representation distillation," 2021, *arXiv:2112.00459*.

[13] M. W. Mahoney and P. Drineas, "CUR matrix decompositions for improved data analysis," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 3, pp. 697–702, 2009.

[14] D. S. Watkins, "The QR algorithm revisited," *SIAM Rev.*, vol. 50, no. 1, pp. 133–145, Jan. 2008.

[15] R. A. Meyer, C. Musco, C. Musco, and D. P. Woodruff, "Hutch++: Optimal stochastic trace estimation," in *Proc. Symp. Simplicity Algorithms (SOSA)*, 2021, pp. 142–155.

[16] R. Bhatia, "Infinitely divisible matrices," *Amer. Math. Monthly*, vol. 113, no. 3, pp. 221–235, Mar. 2006.

[17] F. Roosta-Khorasani and U. Ascher, "Improved bounds on sample size for implicit matrix trace estimators," *Found. Comput. Math.*, vol. 15, no. 5, pp. 1187–1212, Oct. 2015.

[18] H. Avron and S. Toledo, "Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix," *J. ACM*, vol. 58, no. 2, pp. 1–34, Apr. 2011.

[19] X. Yu, S. Yu, and J. C. Principe, "Deep deterministic information bottleneck with matrix-based entropy functional," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 3160–3164.

[20] Z. Zhang, M. Wang, and A. Nehorai, "Optimal transport in reproducing kernel Hilbert spaces: Theory and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 7, pp. 1741–1754, Jul. 2020.

[21] M. T. Harandi, M. Salzmann, and R. Hartley, "From manifold to manifold: Geometry-aware dimensionality reduction for SPD matrices," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 17–32.

[22] M. Bellalij, L. Reichel, G. Rodriguez, and H. Sadok, "Bounding matrix functionals via partial global block Lanczos decomposition," *Appl. Numer. Math.*, vol. 94, pp. 127–139, Aug. 2015.

[23] S. Ubaru, J. Chen, and Y. Saad, "Fast estimation of $tr(f(A))$ via stochastic Lanczos quadrature," *SIAM J. Matrix Anal. Appl.*, vol. 38, no. 4, pp. 1075–1099, Jan. 2017.

[24] C. Musco, C. Musco, and A. Sidford, "Stability of the Lanczos method for matrix function approximation," in *Proc. 29th Annu. ACM-SIAM Symp. Discrete Algorithms*. Philadelphia, PA, USA: SIAM, 2018, pp. 1605–1624.

[25] B. Lam, "Some exact and asymptotic results for best uniform approximation," Ph.D. dissertation, School Inf. Commun. Technol., Univ. Tasmania, Hobart, TAS, Australia, 1972.

[26] S. Bernstein, *Collected Works: Constructive Theory of Functions (1905–1930)* (Translation series), vol. 1. Washington, DC, USA: United States Atomic Energy Commission, 1952.

[27] R. S. Varga and A. J. Carpenter, "Some numerical results on best uniform rational approximation of xα on [0,1]," *Numer. Algorithms*, vol. 2, no. 2, pp. 171–185, Jun. 1992.

[28] S. Bernstein, "Sur la meilleure approximation de $|x|^p$ par des polynômes de degrés très élevés," *Izv. Akad. Nauk SSSR Ser. Mat.*, vol. 2, no. 2, pp. 169–190, 1938.

[29] G. Guennebaud et al. (2010). *Eigen v3*. [Online]. Available: http://eigen.tuxfamily.org

[30] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *Proc. 37th Annu. Allerton Conf. Commun., Control Comput.*, 1999, pp. 368–377.

[31] L. Ardizzone, R. Mackowiak, C. Rother, and U. Köthe, "Training normalizing flows with the information bottleneck for competitive generative classification," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2020, pp. 1–13.

[32] T. Wu, H. Ren, P. Li, and J. Leskovec, "Graph information bottleneck," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2020, pp. 20437–20448.

[33] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–19, 2017.

[34] R. A. Amjad and B. C. Geiger, "Learning representations for neural network-based classification using the information bottleneck principle," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 9, pp. 2225–2239, Sep. 2020.

[35] A. M. Saxe et al., "On the information bottleneck theory of deep learning," *J. Stat. Mechanics, Theory Exp.*, vol. 2019, no. 12, Dec. 2019, Art. no. 124020.

[36] A. Kirsch, C. Lyle, and Y. Gal, "Unpacking information bottlenecks: Unifying information-theoretic objectives in deep learning," 2020, *arXiv:2003.12537*.

[37] K. Ahuja et al., "Invariance principle meets information bottleneck for out-of-distribution generalization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 3438–3450.

[38] D. J. Strouse and D. J. Schwab, "The deterministic information bottleneck," *Neural Comput.*, vol. 29, no. 6, pp. 1611–1630, 2017.

[39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14, 2015.

[40] S. L. Smith, P.-J. Kindermans, and Q. V. Le, "Don't decay the learning rate, increase the batch size," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–11. [Online]. Available: https://openreview.net/forum?id=B1Yy1BxCZ

[41] D. V. Sridhar, E. B. Bartlett, and R. C. Seagrave, "Information theoretic subset selection for neural network models," *Comput. Chem. Eng.*, vol. 22, nos. 4–5, pp. 613–626, Jan. 1998.

[42] R. Setiono and H. Liu, "Improving backpropagation learning with feature selection," *Int. J. Speech Technol.*, vol. 6, no. 2, pp. 129–139, Apr. 1996.

[43] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[44] R. L. De Mántaras, "A distance-based attribute selection measure for decision tree induction," *Mach. Learn.*, vol. 6, no. 1, pp. 81–92, 1991.

[45] J. Chi and M. Jabri, "Entropy based feature evaluation and selection technique," in *Proc. 4th Austral. Conf. Neural Netw. (ACNN)*, 1993, pp. 181–196.

[46] D. Dua and C. Graff. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[47] N. X. Vinh, S. Zhou, J. Chan, and J. Bailey, "Can high-order dependencies improve mutual information based feature selection?" *Pattern Recognit.*, vol. 53, pp. 46–58, May 2016.

[48] M. Koklu and I. A. Ozkan, "Multiclass classification of dry beans using computer vision and machine learning techniques," *Comput. Electron. Agricult.*, vol. 174, Jul. 2020, Art. no. 105507.

[49] B. Abolfathi et al., "The fourteenth data release of the Sloan digital sky survey: First spectroscopic data from the extended baryon oscillation spectroscopic survey and from the second phase of the apache point observatory galactic evolution experiment," *Astrophysical J. Suppl.*, vol. 235, no. 2, p. 42, 2018.

[50] N. X. Vinh, J. Chan, and J. Bailey, "Reconsidering mutual information based feature selection: A statistical significance view," in *Proc. Nat. Conf. Artif. Intell.*, vol. 3, no. 1, 2014, pp. 2092–2098.

[51] G. Brown, A. Pocock, M.-J. Zhao, and M. Luján, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," *J. Mach. Learn. Res.*, vol. 13, pp. 27–66, Jun. 2012.

[52] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Netw.*, vol. 5, no. 4, pp. 537–550, Jul. 1994.

[53] G. Brown, "A new perspective for information theoretic feature selection," *J. Mach. Learn. Res.*, vol. 5, no. 1, pp. 49–56, 2009.

[54] D. D. Lewis, "Feature selection and feature extraction for text categorization," in *Proc. Workshop Speech Natural Lang. (HLT)*, 1992, p. 212.

[55] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[56] H. Yang, "Data visualization and feature selection: New algorithms for nongaussian data," in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, pp. 687–693.

[57] F. Fleuret, "Fast binary feature selection with conditional mutual information," *J. Mach. Learn. Res.*, vol. 5, no. 9, pp. 1–25, 2004.

[58] S. Das, "Inequalities for q-gamma function ratios," *Anal. Math. Phys.*, vol. 9, no. 1, pp. 313–321, Mar. 2019.

[59] S. Xiang, X. Chen, and H. Wang, "Error bounds for approximation in Chebyshev points," *Numerische Math.*, vol. 116, no. 3, pp. 463–491, Sep. 2010.

Yuxin Dong received the B.S. degree from Xi'an Jiaotong University, Xi'an, China, in 2019, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Technology. His research interests include information theory, statistical learning theory, and bioinformatics.

Tieliang Gong received the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China, in 2018. From September 2018 to October 2020, he was a Post-Doctoral Researcher with the Department of Mathematics and Statistics, University of Ottawa, Ottawa, ON, Canada. He is currently an Assistant Professor with the School of Computer Science and Technology, Xi'an Jiaotong University. His research interests include statistical learning theory, machine learning, and information theory.

Shujian Yu received the Ph.D. degree in electrical and computer engineering with minor in statistics from the University of Florida, Gainesville, FL, USA, in 2019. From 2019 to 2021, he was a Machine Learning Research Scientist with the NEC Laboratories Europe, Heidelberg, Germany. He is currently with the Department of Physics and Technology, UiT—The Arctic University of Norway, Tromsø, Norway. He will join the Department of Computer Science, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands, in 2023, as a tenure-track Assistant Professor. He was a recipient of the 2020 International Neural Networks Society Aharon Katzir Young Investigator Award for the contribution on the development of novel information theoretic measures for analysis and training of deep neural networks. He is also selected for the 2023 AAAI New Faculty Highlights.

Chen Li received the Ph.D. degree from the University of Cambridge, U.K., in 2014. From June 2014 to March 2016, he was a Post-Doctoral Researcher with the Massachusetts Institute of Technology, USA. He is currently a Professor with the School of Computer Science and Technology, Xi'an Jiaotong University, China. His research interests include natural language processing, biological text mining, digital pathology, and bioinformatics.