RESEARCH STATEMENT

Archan Ray

1. Introduction

The need for fast algorithms cannot be overstated in an age where the size of datasets as well as parameters required in learning algorithms has grown rapidly [21, 14, 15, 9]. There are several ways to accelerate processing of large data, and sublinear algorithms form an important cornerstone of such methods. Sublinear algorithms access only a small part of the input data, thus they scale well to very large datasets.

A major focus of my work is on development of sublinear algorithms for matrices which are ubiquitous in both computer science and in particular machine learning. Matrices are often used to represent data and parameters of learning models, and as such large datasets and complex learning models have lead to a requirement for efficient computational algorithms. The body of my work has been on pushing the boundaries of sublinear time or sublinear query algorithms in context of matrices and their applications. But since only a small part of the original matrix is observed, approximation is inherent to sublinear algorithms. One of my primary goals is to bound the error of approximation both theoretically and empirically. My research can be thought of being driven by two major themes:

Theoretical Analysis of Sublinear Methods. My work has helped develop fast algorithms for several core problems involving matrices. One of the most important primitives is to compute eigenvalues of a symmetric matrices. Eigenvalues provide information about low-rank structure of matrices, clusterability of data points, and are important in several problems in engineering, physics, and experimental chemistry. Using *random sampling* we have developed sublinear algorithms that can compute non-trivial approximations to all the eigenvalues of a symmetric matrix in our work appearing in ICALP 2023 [1]. On the other hand, the quality of matrix approximation is often measured by the spectral norm of the error matrix. The smaller the spectral norm of the error matrix, the closer are the full spectrum of the input and the approximation matrix. In [2], we develop a sublinear query *deterministic algorithm* that approximates bounded entry symmetric matrices in the spectral norm using element-wise sparsification. Similar results can be obtained with high probability using uniform sampling [12]. Surprisingly our work shows that there exists a fixed set of entries that can be sampled from all bounded entry symmetric matrices to achieve similar approximation guarantees. We further extend our algorithms to obtain the first $o(n^\omega)$ deterministic algorithms that can compute singular value and vector approximations [2], where $\omega \approx 2.37$ is the exponent of the matrix multiplication [8, 4].

Applications of Sublinear Methods. In parallel to the theoretical analysis, I am also interested in applications of theoretically motivated matrix approximation algorithms to various domains. I have demonstrated the efficacy of randomized algorithms in approximation of eigenvalues of symmetric matrices on several synthetic and real world matrices [1]. Moreover in our work appearing in AAAI 2022 [3], we have shown that low-rank approximation of matrices can be used to develop fast algorithms for machine learning with application in natural language processing (NLP). Specifically, we show that matrix approximations can maintain downstream task performance in three core NLP tasks – 1) document embedding, 2) approximating similarity matrices generated using cross-encoders [9] and 3) approximating the similarity function used to determine coreference relationships across documents. One of my long term goals is to develop a toolkit that can be used to maintain performance of learning algorithms while also significantly speeding up computation. This can then be applied to develop fast learning algorithms that are computationally efficient.

In the following paragraphs I summarize my research and how they tie into the general research themes mentioned above.

2. Theoretical Analysis of Sublinear Methods

Sublinear time or sublinear query algorithms can significantly improve computation complexity of various problems in computer science and linear algebra. Although significant work exists which uses sublinear algorithms to approximate matrices, several avenues remain open. We begin by describing our work on approximating the eigenvalues of a symmetric matrix.

Eigenvalue Approximation. Eigenvalues are extensively studied in various fields with applications in engineering, optimization, data analysis, spectral graph theory and beyond. Computing eigenvalues with high accuracy using traditional matrix multiplication methods for dense matrices requires $O(n^\omega)$ runtime; however in practice the runtime is $O(n^3)$. As n increases this computational complexity becomes intractable.

In [1] we propose a sublinear time randomized algorithm that approximates all the eigenvalues of a symmetric matrix with bounded entries by sampling a random principal submatrix of the input matrix. Our result can be viewed as a concentration bound on the complete eigenspectrum of a random submatrix, significantly extending known bounds on just the singular values (the magnitudes of the eigenvalues) [6, 20, 13]. Specifically for any matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ with maximum entry magnitude $\|\mathbf{A}\|_\infty \leq 1$, we show that using a simple algorithm *all* the eigenvalues of $\mathbf{A}$ can be approximated up to additive error $\pm \epsilon n$ by randomly sampling an $\tilde{O}\left(\frac{\log^3 n}{\epsilon^3}\right) \times \tilde{O}\left(\frac{\log^3 n}{\epsilon^3}\right)$ principal submatrix of $\mathbf{A}$ with good probability. We also give improved error bounds when the rows of the input matrix can be sampled with probabilities proportional to their sparsities or their squared ℓ_2 norms. Even for the strictly easier problems of approximating the singular values or testing the existence of large negative eigenvalues [5], our results are the first that take advantage of non-uniform sampling to give improved error bounds.

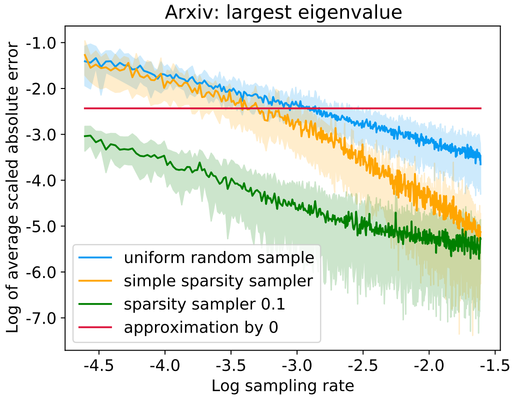

Figure 1. Log scale absolute approximation error vs. log sampling rate for our random sampling algorithms compared with approximation by 0, for approximating the largest eigenvalue of the adjacency matrix of the graph of co-references in condense matter papers in ArXiv [16].

A comparison of these algorithms can be seen in Figure 1. The plot demonstrates that our method gets very good approximations even at a low sampling rate. For the real world graphs, e.g., the adjacency matrix of ArXiv co-reference graph [16], which have power law degree distributions, sparsity based sampling techniques significantly outperforms other sampling algorithms.

Deterministic Spectral Approximation. Although randomized algorithms are prevalent in matrix approximation literature, no such algorithm had existed which can deterministically compute eigenvalues of symmetric matrices in sublinear time. Essentially all existing algorithms to approximate eigenvalues are based on random sampling. In [2] we develop the first deterministic algorithms that approximates *all* symmetric matrices with bounded entries in the spectral norm *using sublinear queries*. Specifically our work proposes deterministic sparisfiers that can approximate any symmetric matrix in the spectral norm using sublinear queries to the input matrix. By sublinear queries we mean $\tilde{O}(n/\epsilon^c)$ queries to the input matrix for $c = 2$ when the matrix is positive semidefinite (PSD) and $c = 4$ otherwise.

Moreover, we extend these algorithms to give the first $o(n^\omega)$ time deterministic algorithms for several central problems related to singular value and singular vector approximations. We also give the first $o(n^\omega)$ time deterministic algorithm to test if all the eigenvalues of a matrix is greater than 0 or if the smallest eigenvalue is at least $-\epsilon \max(n, \|\mathbf{A}\|_1)$, where $\|\mathbf{A}\|_1$ is the nuclear norm of $\mathbf{A}$. An optimal randomized algorithm for this problem with detection threshold $-\epsilon n$ was presented in [5]. Thus [2] significantly extends the boundaries of deterministic algorithms in these applications. The success of these applications thus opens up the possibility for new classes of fast deterministic algorithms for general matrices.

3. Applications of Sublinear Methods

Parallel to the theoretical analysis of sublinear algorithms to approximate several properties of matrices, my work also concentrates on applications of sublinear algorithms to practical problems. In [1] we demonstrate the effectiveness of sublinear time algorithms to approximate all the eigenvalues of several synthetic and real world matrices. The real world matrices include – 1) similarity matrix of random data points drawn from a binary image, 2) adjacency matrices of social networks and collaboration networks. Eigenvalues can be used to identify clusterability of graphs, and thus our approximation algorithm can help in developing fast algorithms for clustering nodes in a network. We observe relatively small error in approximating all eigenvalues, with the error decreasing as the number of samples increases. We also observe in Figure 1 that the algorithms that leverage sparsity information produces significant advantages over other randomized sampling algorithms for adjacency matrices corresponding to graphs as a direct result of the power law degree distribution.

Applications in NLP. Many machine learning tasks center around the computation of pairwise similarities between data points using an appropriately chosen similarity function. E.g., in kernel methods, a non-linear kernel inner product is used to measure similarity, and often to construct a pairwise kernel similarity matrix. Computing all pairwise similarities for a data set with n points requires $\Omega(n^2)$ similarity computations. This can be a major runtime bottleneck, especially when each computation requires the evaluation of a neural network or other expensive operation. One approach to avoid this bottleneck is to produce a compressed approximation to the $n \times n$ pairwise

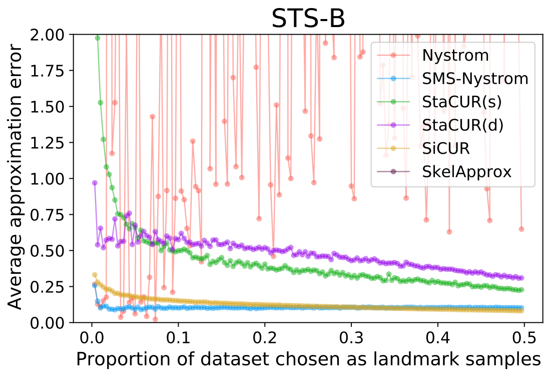

Figure 2. Evaluation of various sublinear time algorithms on the sentence similarity task. The x-axis is the proportion of the dataset sampled.

similarity matrix $\mathbf{K}$ for the data set, but avoid ever fully forming this matrix and run in sublinear time with respect to the size of $\mathbf{K}$. Nyström approximation [25] is often used to produce such compressed representation that can approximate PSD matrices, but is empirically unstable in approximation of indefinite symmetric matrices. In [3] we propose a simple modification to Nyström approximation (Submatrix-Shifted-Nyström) that stabilizes its application to any symmetric matrix. We also show that both Submatrix-Shifted-Nyström, and a simple variant of CUR decomposition [10, 11, 28] yield accurate approximations (see Figure 2) for a myriad of tasks in NLP like document embedding, document classification, document co-reference, and sentence similarity. Moreover the approximation algorithms also maintain downstream task performance in all these tasks while greatly reducing the time and space required as compared to the exact similarity matrix.

4. Future Work and Open Questions

Our work leaves several open questions and avenues for future work. I want to develop a toolbox that can approximate several properties of matrices using various algorithms especially using – randomized, deterministic and sketching algorithms. I outline some concrete directions below.

Spectrum Approximation using Matrix-Vector Algorithms. Matrix-vector algorithms queries a matrix by multiplying it with a vector [22]. Example applications include Lanczos or Krylov methods [17], testing if a matrix is PSD [18], and matrix sketching algorithms [26]. Moreover given the current advancements in hardware capabilities, matrix-vector products can be computed in a distributive and parallel setting, resulting in very fast algorithms. An important step in understanding the effectiveness of sublinear time eigenvalue approximation algorithms is to understand the quality of approximation vs the number of matrix-vector queries made to the input matrix. Recently, [23] proposed an optimal algorithm that can compute approximate eigenvalues using matrix sketching. We can also design eigenvalue approximation algorithms using Lanczos methods with varying degrees of adaptivity. An interesting idea will be to compare the quality of approximation of the eigenvalue approximation algorithms vs the number of matrix-vector queries made to the input matrix. This study can help design algorithms which can deployed in a distributed and parallel setting, leading to very fast algorithms which requires computing eigenvalues. We also plan to check how the matrix-vector products required to achieve a specific approximation factor varies across adaptive and non-adaptive sketching algorithms. This would allow us to understand if there exists any fundamental difference among the algorithmic approaches in practice.

Randomized Algorithms for Other Matrix Properties. Lanczos methods [17] has been successfully applied to several core problems including eigenvalue approximation and singular value approximations. In fact all eigenvalues can approximated up to additive error $\pm\epsilon n$ for any symmetric matrices using Lanczos methods. Recently in [7], for a symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, a sublinear algorithm via kernel polynomial method [24] is proposed which can approximate total ℓ_1 error of eigenvalue approximation up to $\epsilon\|\mathbf{A}\|_1$ using n/ϵ^2 matrix-vector queries. We conjecture that first using eigenvalue deflation via Lanczos methods and then combining with spectral density estimation, the matrix-vector queries required to achieve the said bound can be improved to $\sqrt{n}/\epsilon^2$. This would immediately improve the runtime of spectral density estimation applications including matrix multiplication using Hessian matrices [19, 27], and matrix inversion.

Model Compression and Efficient Learning. Large parametric models have achieved dramatic empirical success across many applications like object classification and language modelling. A better understanding of why these models require such large numbers of parameters could help answer how to reduce their computational costs. Understanding how to reduce the complexity of these models, could be aided with an understanding of how the parameters of the models are used. One simple way to reduce parameters is by model compression. But most linear algebraic compression techniques do not translate to applicable learning algorithms. My general goal here is to understand fundamentally how the parameter space can be compressed using algebraic tools and careful manipulation of the feature space. One can study the tradeoffs between parametric and non-parametric models in achieving a specific task, say non-linear data classification. I.e., given any dataset, can we first apply non-parametric learning to split up the input space into sets of data points, followed by a general parametric model to classify data points in all of the sets? Our preliminary experiments on synthetic and real world datasets suggests that using this two phased learning allows us to compress the parameter space while significantly improving runtime of the learning algorithms.

Co-authored Papers

[1] Rajarshi Bhattacharjee, Gregory Dexter, Petros Drineas, Cameron Musco, and **Archan Ray**. "Sublinear Time Eigenvalue Approximation via Random Sampling". In: *Proceedings of the 50th International Colloquium on Automata, Languages and Programming (ICALP)*. 2023.

[2] Rajarshi Bhattacharjee, Gregory Dexter, Cameron Musco, **Archan Ray**, Sushant Sachdeva, and David P Woodruff. "Universal Matrix Sparsifiers and Fast Deterministic Algorithms for Linear Algebra". In: *Proceedings of the 15th Conference on Innovations in Theoretical Computer Science (ITCS)*. 2024.

[3] **Archan Ray**, Nicholas Monath, Andrew McCallum, and Cameron Musco. "Sublinear Time Approximation of Text Similarity Matrices". In: *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)* (2022).

* The author listing for [1] and [2] is alphabetical.

Other References

[4] Josh Alman and Virginia Vassilevska Williams. "A Refined Laser Method and Faster Matrix Multiplication". In: *Proceedings of the 32nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2021.

[5] Ainesh Bakshi, Nadiia Chepurko, and Rajesh Jayaram. "Testing Positive Semi-Definiteness via Random Submatrices". In: *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science (FOCS)* (2020).

[6] Vladimir Braverman, Aditya Krishnan, and Christopher Musco. "Linear and Sublinear Time Spectral Density Estimation". In: *Proceedings of the 54th Annual ACM Symposium on Theory of Computing (STOC)* (2022).

[7] Vladimir Braverman, Aditya Krishnan, and Christopher Musco. "Sublinear Time Spectral Density Estimation". In: *Proceedings of the 54th Annual ACM Symposium on Theory of Computing (STOC)*. 2022, pp. 1144–1157.

[8] James Demmel, Ioana Dumitriu, Olga Holtz, and Robert Kleinberg. "Fast Matrix Multiplication is Stable". In: *Numerische Mathematik* (2007).

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 21st Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)* (2019).

[10] Petros Drineas, Michael W Mahoney, and Nello Cristianini. "On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning." In: *Journal of Machine Learning Research (JMLR)* (2005).

[11] Petros Drineas, Michael W Mahoney, and Shan Muthukrishnan. "Relative-Error CUR Matrix Decompositions". In: *SIAM Journal on Matrix Analysis and Applications* (2008).

[12] Petros Drineas and Anastasios Zouzias. "A note on element-wise matrix sparsification via a matrix-valued Bernstein inequality". In: *Inf. Process. Lett.* 111.8 (2011), pp. 385–389.

[13] Alex Gittens and Joel A Tropp. "Tail Bounds for All Eigenvalues of a Sum of Random Matrices". In: *http://arxiv.org/abs/1104.4513* (2011).

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *Proceedings of the 31st IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet Classification with Deep Convolutional Neural Networks". In: *Communications of the ACM* (2017).

[16] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. "Graph Evolution: Densification and Shrinking Diameters". In: *ACM transactions on Knowledge Discovery from Data (TKDD)* (2007).

[17] Cameron Musco and Christopher Musco. "Randomized Block Krylov Methods for Stronger and Faster Approximate Singular Value Decomposition". In: *Proceedings of the 28th Advances in Neural Information Processing Systems (NeurIPS)* (2015).

[18] Deanna Needell, William Swartworth, and David P Woodruff. "Testing Positive Semidefiniteness Using Linear Measurements". In: *Proceedings of the 63rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 2022.

[19] Barak A Pearlmutter. "Fast exact multiplication by the Hessian". In: *Neural computation* (1994).

[20] Mark Rudelson and Roman Vershynin. "Sampling from Large Matrices: An Approach Through Geometric Functional Analysis". In: *Journal of the ACM (JACM)* (2007).

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. "Imagenet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* (2015).

[22] Xiaoming Sun, David P Woodruff, Guang Yang, and Jialin Zhang. "Querying a Matrix Through Matrix-Vector Products". In: *ACM Transactions on Algorithms (TALG)* (2021).

[23] William Swartworth and David P Woodruff. "Optimal Eigenvalue Approximation via Sketching". In: *Proceedings of the 64th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 2023.

[24] Alexander Weiße, Gerhard Wellein, Andreas Alvermann, and Holger Fehske. "The kernel polynomial method". In: *Reviews of modern physics* (2006).

[25] Christopher Williams and Matthias Seeger. "Using the Nyström Method to Speed up Kernel Machines". In: *Proceedings of the 14th Advances in Neural Information Processing Systems (NeurIPS)*. 2001.

[26] David P Woodruff. "Sketching as a tool for numerical linear algebra". In: *http://arxiv.org/abs/1411.4357* (2014).

[27] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. “Pyhessian: Neural networks through the lens of the hessian”. In: *2020 IEEE international conference on big data (Big data)*. 2020.

[28] Kai Zhang, Ivor W Tsang, and James T Kwok. “Improved Nyström Low-Rank Approximation and Error Analysis”. In: *Proceedings of the 25th International Conference on Machine Learning (ICML)*. 2008.