RESEARCH STATEMENT

Archan Ray (ray@cs.umass.edu)

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 [47, 27, 32, 20]. 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 the development of fast and especially sublinear algorithms. My recent works have been on pushing the boundaries of sublinear time or sublinear query algorithms in the context of matrices and their applications. Matrices are ubiquitous mathematical structures in both computer science and, in particular, machine learning, and are often used to represent data and parameters of learning models. As such, very large datasets and complex learning models have led to a requirement for efficient computational algorithms. Since only a small part of the original matrix is observed, approximation of the the full matrix is inherent to sublinear algorithms. One of my primary goals is to establish theoretical and empirical bounds on the error of approximation. My research is driven by two major themes:

Theoretical Analysis of Sublinear Methods. My work has contributed to the development of fast algorithms for several core problems involving matrices. These include eigenspectrum approximation, singular value and vector approximation, testing whether all eigenvalues of a matrix are positive, and low-rank approximation of matrices. These properties provide valuable insights into the low-rank structure of matrices, the clusterability of data points, and play a pivotal role in various engineering and experimental problems.

Our work appearing in ICALP 2023 [1], gives the first sublinear algorithms that compute non-trivial approximations to all the eigenvalues of a symmetric matrix using various *random sampling* techniques. We extend these results to deterministic sampling algorithms in our work appearing at ITCS 2024 [2]. In our work, we develop an algorithm that approximates symmetric matrices in the spectral norm using element-wise sparsification. Similar results can be obtained with high probability using uniform sampling [24]. 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 deterministic sublinear query algorithms for eigenspectrum approximation, and 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 [19, 5].

Implicit matrices enhance efficiency in various applications, especially when the matrix is a function of another (e.g., covariance or Hessian matrices). Computing such functions can be expensive, but using algorithms that can query a matrix $\mathbf{A}$ with a vector $\mathbf{v}_i$ offers a more computationally efficient approach. These algorithms are called matrix-vector query algorithms. Considering that matrix-vector query algorithms can be effiently parallelized in distributed systems and can be significantly faster when a function of matrix is of interest, it is important to study matrix-vector query algorithms. Our recent work [3] explores eigenspectrum approximation using matrix-vector queries. Notably, research [49] shows that a query-optimal non-adaptive algorithm can approximate symmetric matrix eigenvalues with error $\epsilon\|\mathbf{A}\|_F$. We demonstrate existence of a wider class of adaptive matrix-vector query algorithms that are nearly query optimal. We also show that one of these adaptive algorithms can be converted to a non-adaptive matrix-vector query optimal algorithm. While these algorithms achieve the eigenvalue approximation error of $\epsilon\|\mathbf{A}\|_F$ with near-optimal matrix-vector query complexity, it is also important to understand how they perform empirically. To this end, our

experiments demonstrate that matrix-vector query algorithms that non-trivially approximate a larger number of eigenvalues in the eigenspectrum of a matrix perform better when minimizing the largest error of approximation of any eigenvalue is of interest. Note that for a fixed number of matrix-vector queries, adaptive methods non-trivially approximates a smaller number of eigenvalues as compared to the non-adaptive algorithms. Thus, when minimizing the largest error of approximation of any eigenvalue is of interest, the non-adaptive matrix-vector query algorithms are empirically superior. However, when the approximating the extremal eigenvalues of a matrix is of interest, the adaptive matrix-vector query algorithms significantly outperform the non-adaptive algorithms.

Applications of Sublinear Methods. In parallel to theoretical analysis, I am also interested in applications of theoretically motivated matrix approximation algorithms to various domains. We have demonstrated the empirical performance 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 [4], 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 [20] 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 that 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 other fields. Computing eigenvalues with high accuracy using traditional matrix multiplication methods for dense matrices requires $O(n^\omega)$ runtime. However, in practice, the runtime is closer to $O(n^3)$. As the dimension of the matrix (n) increases, this computational complexity becomes intractable.

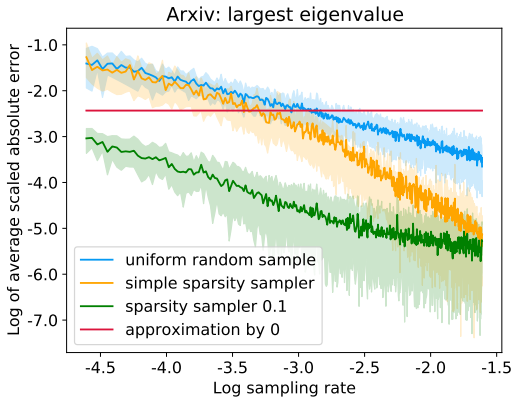


Figure 1. Log of the average scaled absolute approximation error vs. log of the sampling rate for our random sampling algorithms compared with approximation by 0, for approximating the largest magnitude eigenvalue of the adjacency matrix of the graph of co-references in condensed matter papers in arXiv [34].

In our work [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) [13, 46, 26]. Specifically, for any matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ with a maximum entry magnitude $\|\mathbf{A}\|_\infty \leq 1$, our work demonstrates that a

simple algorithm can approximate *all* the eigenvalues of $\mathbf{A}$ with additive error of up to $\pm\epsilon n$ 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 high 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 [6], our results are the first that take advantage of non-uniform sampling to give improved error bounds.

A comparison of these algorithms can be seen in Figure 1. The plot demonstrates that our methods achieve very good approximations even at a low sampling rate. For real world graphs, such as the adjacency matrix of the arXiv co-reference graph [34], which have a power-law degree distribution, sparsity based sampling techniques significantly outperform other sampling algorithms.

Deterministic Spectral Approximation. Although randomized algorithms are prevalent in the literature on matrix approximation, no deterministic algorithm had existed for computing eigenvalues of symmetric matrices in sublinear time. We observe that any algorithm that can approximate a matrix in the spectral norm, directly gives an eigenvalue approximation error bound via Weyl's inequality [53, 8]. In [2], we develop the first deterministic algorithms that approximate *all* symmetric matrices with bounded entries in the spectral norm *using sublinear queries*. Specifically, we observe that any matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$ that satisfies $\|\mathbf{1} - \mathbf{S}\|_2 \leq \epsilon n$, where $\mathbf{1}$ is the all-ones matrix, also yields *universal sparsifiers* for any bounded-entry positive semidefinite (PSD) matrix. That is, given a PSD matrix, $\mathbf{A}$ with $\|\mathbf{A}\|_\infty \leq 1$, $\|\mathbf{A} - \mathbf{A} \circ \mathbf{S}\|_2 \leq \epsilon n$. Our results also extend to non-PSD matrices, with a tighter error bound of $\epsilon \cdot \max(n, \|\mathbf{A}\|_1)$, where $\|\mathbf{A}\|_1$ is the nuclear norm of $\mathbf{A}$. Moreover, we demonstrate that the number of entries in $\mathbf{A}$ that needs to be read by such sparsifiers is near-optimal (tight up to logarithmic factors). A matrix $\mathbf{S}$ satisfying the bound on the all-ones matrix can be optimally constructed using the adjacency matrix of a Ramanujan graph with the appropriate number of non-zero entries.

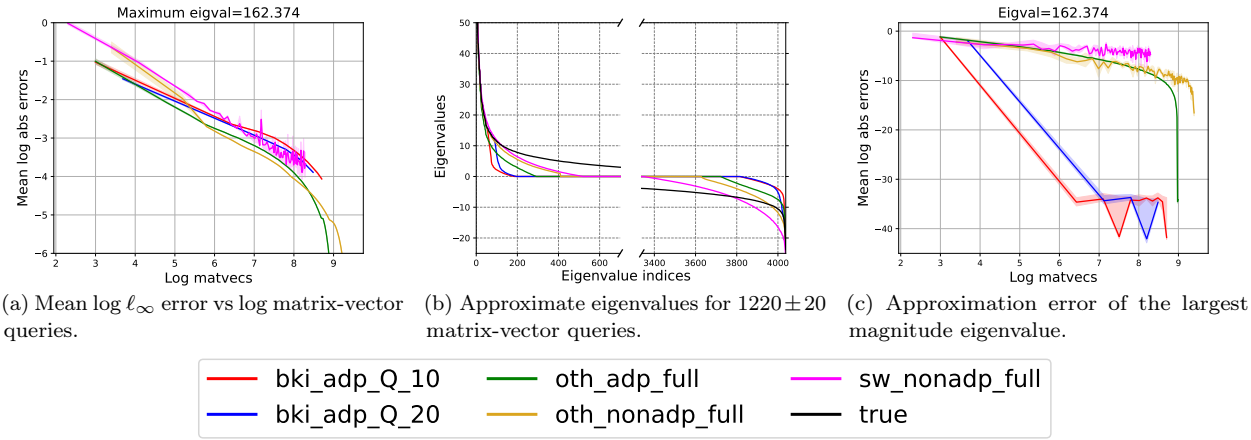
These results immediately yield the eigenvalue approximation error bound for all symmetric matrices. Furthermore, we extend these algorithms to give the first $o(n^\omega)$-time deterministic algorithms for several central problems related to singular value and singular vector approximations. Additionally, we present the first $o(n^\omega)$-time deterministic algorithm to test whether all the eigenvalues of a matrix is greater than 0 or if the smallest eigenvalue is at least $-\epsilon \max(n, \|\mathbf{A}\|_1)$. An optimal randomized algorithm for this problem with detection threshold $-\epsilon n$ was presented in [6]. Thus our work in [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.

Spectrum Approximation using Matrix-Vector Algorithms. We also study eigenvalue approximation in the matrix-vector query model [48, 45]. Within this model, the underlying matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is often implicit and can only be accessed by using matrix-vector queries of the form $\mathbf{Ax} \in \mathbb{R}^n$ where $\mathbf{x} \in \mathbb{R}^n$ is the query vector. $\mathbf{x}$ can be chosen randomly and possibly adaptively – i.e., at time t, $\mathbf{x}_t$ can be chosen based on the prior observations $\mathbf{Ax}_1, \mathbf{Ax}_2, \ldots, \mathbf{Ax}_{t-1} \in \mathbb{R}^n$ using query vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{t-1} \in \mathbb{R}^n$. When $\mathbf{x}_t$ is chosen non-adaptively, the matrix-vector query algorithms are often called linear sketching. Example applications of matrix-vector query algorithms include Lanczos or Krylov methods [40], testing if a matrix is PSD [43], and matrix sketching algorithms [55]. 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.

Given the matrix-vector query model, we theoretically and empirically investigate algorithms that approximate the eigenvalues of $\mathbf{A}$. In [50], the authors show that the query complexity of approximating each eigenvalue of a symmetric matrix $\mathbf{A}$ up to error $\epsilon\|\mathbf{A}\|_F$ is $\Omega(1/\epsilon^2)$ using a query optimal non-adaptive algorithm. Moreover, [50] also demonstrates that the lower bound query complexity for any matrix-vector query algorithm (both adaptive and non-adaptive) is $\Omega(1/\epsilon^2)$.

In our work [3], we empirically investigate the practical computational overhead for achieving eigenvalue approximation using several matrix-vector query algorithms. Particularly, we investigate the spectrum of matrix-vector algorithms ranging from non-adaptive to massively adaptive algorithms, and study how adaptivity affects practical performance of these algorithms to approximate the eigenspectrum of a matrix.

We show that there is a spectrum of adaptive matrix-vector query algorithm which uses $O(\log n)$-factor optimal matrix-vector queries to approximate all the eigenvalues of the input matrix. We also introduce a new non-adaptive algorithm that matches the sampling complexity lower bound given in [50]. Empirically, we observe that for a fixed number of matrix-vector queries, non-adaptive algorithms outperform adaptive algorithms when the error is measured in the ℓ_∞-norm. This is because a wider range of eigenvalues of the input matrix are approximated non-trivially using non-adaptive algorithms for any fixed number of matrix-vector queries by any adaptive algorithm. When approximating the largest magnitude eigenvalue of a symmetric matrix is of interest adaptive methods outperform non-adaptive algorithms. We summerize these observations in Figure 2. In the asymptotic limit, the query complexity bound is theoretically consistent (ignoring constant log factors) across all matrix-vector query algorithms. Our empirical observations help us understand their differences in performance under various error norms. This study helps in designing algorithms which can deployed in a distributed and parallel setting, leading to very fast algorithms which requires computing eigenvalues. Moreover, the interplay between number of matrix-vector queries and adaptivity would help in the choice of algorithms to approximate eigenvalues under specific constraints or requirements.



(a) Mean log ℓ_∞ error vs log matrix-vector queries.

(b) Approximate eigenvalues for 1220 ± 20 matrix-vector queries.

(c) Approximation error of the largest magnitude eigenvalue.

FIGURE 2. **Summary of observations for Facebook adjacency matrix [39].** Here we present some key observations of using various matrix-vector query algorithms to approximate eigenvalues of the Facebook adjacency matrix [39]. In Figure 2a we plot the mean log scaled absolute ℓ_∞-error vs the number of matrix-vector queries made by several matrix-vector algorithms. The maximum magnitude eigenvalue of each matrix is reported on top of Figure 2a. In Figure 2c we plot the eigenvalue approximates for the matrix-vector algorithms. Finally, in Figure 2c we plot the mean log scaled absolute error vs the number of matrix-vector queries made by the matrix-vector algorithms to approximate the largest magnitude eigenvalue of the Facebook adjacency matrix [39].

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 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 [54] is often used to produce such compressed representation that can approximate PSD matrices, but is empirically unstable in approximation of indefinite symmetric matrices. In [4] 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 [22, 23, 57] yield accurate approximations (see Figure 3) 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.

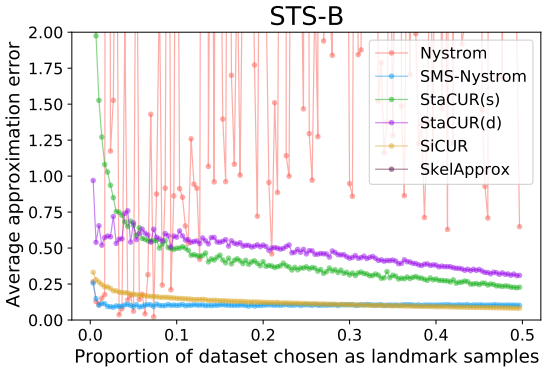


FIGURE 3. Evaluation of various sublinear time algorithms on the sentence similarity task. The x-axis is the proportion of the dataset sampled.

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.

Randomized Algorithms for Other Matrix Properties. Lanczos methods [40] 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 [14], for a symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, a sublinear algorithm via kernel polynomial method [52] is proposed which can approximate total ℓ_1 error of eigenvalue approximation up to $\epsilon\|\mathbf{A}\|_1$ using $O(n/\epsilon^2)$ 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 $O(\sqrt{n}/\epsilon^2)$. This would immediately improve the runtime of spectral density estimation applications including matrix multiplication using Hessian matrices [44, 56], and matrix inversion.

Deterministic Sublinear Algorithms. Our work in [2] demonstrates that PSD matrices with entries in $\{-1, 0, 1\}$ can be approximated in the spectral norm with error up to ϵn by querying near-optimal entries ($\tilde{O}(n/\epsilon)$). This improves the general query complexity by a factor of $O(1/\epsilon)$. However, it unknown if the improved query complexity bound can be applied to a wider class of PSD matrices. We conjecture that this bound extends to PSD matrices with entries in $\{-2, -1, 0, 1, 2\}$. A first step is to consider any PSD matrix $\mathbf{A} \in \{0, 1, 2\}^{n \times n}$, such that for all $i \in [n]$, $\mathbf{A}_{ii} = 2$ and $\mathbf{A}_{ij} \in \{0, 1\}$, for $i \neq j$. The least eigenvalue of such $\mathbf{A} - 2\mathbf{I}$ is at least -2. Any unweighted graph with a vertex set larger than 36 and smallest eigenvalue of the corresponding adjacency matrix greater than or equal to -2 is called a *generalized line graph* [15, 30]. A generalized line graph is made up to two kinds of graphs: a *line graph* and m disjoint *cocktail party graphs*. We finally restrict this class of matrices such that $\mathbf{A} - 2\mathbf{I}$ is the adjacency matrix of a line graph. We conjecture that using the adjacency matrix of an expander graph where any two ϵn sized vertex sets are connected by at least an edge, one can show that the adjacency matrix can be approximated in the spectral norm with error up to ϵn using $\tilde{O}(n/\epsilon)$ entries, near-optimally.

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. 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.

Consider the problem of network pruning [33, 11, 29], which removes hidden units from trained models in either a manner that is structured [28, 35, 36, 42] (e.g., remove entire row of matrix, remove channel of layer) or unstructured [31, 37, 25] (e.g., remove individual neurons). Given a weight matrix $\mathbf{W}^{i+1} \in \mathbb{R}^{m_i \times m_{i+1}}$ at the i^{th} layer and activation matrix $\mathbf{A}^i \in \mathbb{R}^{n \times m_i}$, for n data points, a general goal in model compression is to approximate the product $\mathbf{A}^i\mathbf{W}^{i+1}$. A plethora of sublinear sampling algorithms can be studied which minimizes error of the form $\|\mathbf{A}^i\mathbf{W}^{i+1} - \tilde{\mathbf{A}}^i\tilde{\mathbf{W}}^{i+1}\|_F^2$, including sublinear sampling methods [38, 51, 7, 41], approximate matrix multiplication [21, 18], and matrix sparsification [12]. Thus, there are several directions which we can take to find compressed activation and weight matrices. It will be interesting to see if these results with sublinear algorithms for matrices can be extended to large neural models. Moreover, studying how the approximation error of these sublinear algorithms affect the downstream task performance in neural networks can help in designing efficient pruning algorithms.

Finally, sublinear algorithms for model compression can also be used to study memorization in neural networks. Recent works [17, 16, 10, 9] demonstrate that memorization is prevalent among overparameterized networks. Memorizing training data leads to an increase in the number of parameters required in a learning model. This consequently leads to longer training and inference times while also causing the model to overfit to certain parts of the training data. Additionally, memorization renders the learning algorithm susceptible to adversarial data queries. Therefore, we seek to quantify the amount of training data that was memorized in the original model but lost due to the model compression. It is equally important that the performance of the learning model only minimally degrades as a result of model compression. consequently, we also aim to understand if reducing memorization impacts the downstream task performance of the learning model. Our objective is to study the rate at which the model *forgets* individual training samples without compromising the downstream task performance as a result of compression. Establishing a theoretical bound on the rate of data forgetting due to model compression has remained an open problem. We aim to establish a theoretical bound on the rate of data forgetting while also empirically demonstrating the proportion of data forgotten by the learning model as a result of memorization.

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] Cameron Musco and **Archan Ray**. "Eigenvalue Approximation using Matrix-Vector Query Algorithms". In: *In Preparation*. 2023.

[4] **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], [2], and [3] is alphabetical.

Other References

[5] 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.

[6] 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).

[7] Ainesh Bakshi, Kenneth L Clarkson, and David P Woodruff. "Low-Rank Approximation with $1/\epsilon^{1/3}$ Matrix-Vector Products". In: *Proceedings of the 54th Annual ACM Symposium on Theory of Computing (STOC)*. 2022.

[8] Rajendra Bhatia. *Matrix analysis*. Vol. 169. Springer Science & Business Media, 2013.

[9] Robi Bhattacharjee, Sanjoy Dasgupta, and Kamalika Chaudhuri. "Data-Copying in Generative Models: A Formal Framework". In: *http://arxiv.org/abs/2302.13181* (2023).

[10] Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raf. "Emergent and predictable memorization in large language models". In: *http://arxiv.org/abs/2304.11158* (2023).

[11] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. "What is the state of neural network pruning?" In: *Proceedings of machine learning and systems* (2020).

[12] Vladimir Braverman, Robert Krauthgamer, Aditya R Krishnan, and Shay Sapir. "Near-optimal entrywise sampling of numerically sparse matrices". In: *Conference on Learning Theory*. PMLR. 2021, pp. 759–773.

[13] 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).

[14] 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.

[15] Peter J Cameron, Jean-Marie Goethals, Johan Jacob Seidel, and Ernest E Shult. "Line graphs, root systems, and elliptic geometry". In: *Geometry and Combinatorics*. 1991.

[16] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. "Quantifying memorization across neural language models". In: *http://arxiv.org/abs/2202.07646* (2022).

[17] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. "Extracting training data from large language models". In: *30th USENIX Security Symposium (USENIX Security 21)*. 2021.

[18] Michael B Cohen, Jelani Nelson, and David P Woodruff. "Optimal approximate matrix product in terms of stable rank". In: *arXiv preprint arXiv:1507.02268* (2015).

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

[20] 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).

[21] Petros Drineas, Ravi Kannan, and Michael W Mahoney. "Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition". In: *SIAM Journal on Computing* (2006).

[22] 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).

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

[24] 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.

[25] Jonathan Frankle and Michael Carbin. "The lottery ticket hypothesis: Finding sparse, trainable neural networks". In: *http://arxiv.org/abs/1803.03635* (2018).

[26] 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).

[27] 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.

[28] Tianxing He, Yuchen Fan, Yanmin Qian, Tian Tan, and Kai Yu. "Reshaping deep neural network for fast decoding by node-pruning". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2014.

[29] Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. "Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks". In: *The Journal of Machine Learning Research* (2021).

[30] Alan J Hoffman. "On graphs whose least eigenvalue exceeds- $1-\sqrt{2}$". In: *Linear Algebra and its Applications* (1977).

[31] Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. "Network trimming: A data-driven neuron pruning approach towards efficient deep architectures". In: *http://arxiv.org/abs/1607.03250* (2016).

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

[33] Andrey Kuzmin, Markus Nagel, Saurabh Pitre, Sandeep Pendyam, Tijmen Blankevoort, and Max Welling. "Taxonomy and evaluation of structured compression of convolutional neural networks". In: *arXiv preprint arXiv:1912.09802* (2019).

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

[35] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. "Pruning filters for efficient convnets". In: *http://arxiv.org/abs/1608.08710* (2016).

[36] Lucas Liebenwein, Cenk Baykal, Harry Lang, Dan Feldman, and Daniela Rus. "Provable filter pruning for efficient neural networks". In: *http://arxiv.org/abs/1911.07412* (2019).

[37] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. "Learning efficient convolutional networks through network slimming". In: *Proceedings of the IEEE international conference on computer vision*. 2017.

[38] Michael W Mahoney and Petros Drineas. "CUR matrix decompositions for improved data analysis". In: *Proceedings of the National Academy of Sciences* (2009).

[39] Julian J McAuley and Jure Leskovec. "Learning to discover social circles in ego networks." In: *Proceedings of the 25th Advances in Neural Information Processing Systems (NeurIPS)*. 2012.

[40] 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).

[41] Cameron Musco and David P Woodruff. "Sublinear time low-rank approximation of positive semidefinite matrices". In: *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 2017.

[42] Ben Mussay, Margarita Osadchy, Vladimir Braverman, Samson Zhou, and Dan Feldman. "Data-independent neural pruning via coresets". In: *http://arxiv.org/abs/1907.04018* (2019).

[43] 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.

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

[45] Cyrus Rashtchian, David P Woodruff, and Hanlin Zhu. "Vector-matrix-vector queries for solving linear algebra, statistics, and graph problems". In: *http://arxiv.org/abs/2006.14015* (2020).

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

[47] 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).

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

[49] 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.

[50] 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).

[51] Shusen Wang and Zhihua Zhang. "Improving CUR matrix decomposition and the Nyström approximation via adaptive sampling". In: *The Journal of Machine Learning Research* (2013).

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

[53] Hermann Weyl. "The asymptotic distribution law of the eigenvalues of linear partial differential equations (with an application to the theory of cavity radiation)". In: *Mathematical Annals* (1912).

[54] 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.

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

[56] 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.

[57] 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.