A Moderate Survey of Sketching Techniques Comparison for Randomized Numerical Linear Algebra Under Machine Learning Setting

Yuqi Liu[*]
University of California, Berkeley
Berkeley, CA 94720
liuyuqi@berkeley.edu

Leon Mikulinsky
University of California, Berkeley
Berkeley, CA 94720
mikulinsky@berkeley.edu

Konstantin Zörner
University of California, Berkeley
Berkeley, CA 94720
konstantin.zoerner@berkeley.edu

James Demmel
University of California, Berkeley
Berkeley, CA 94720
demmel@berkeley.edu

Abstract

In our work, we explore different kinds of sketching methods'efficiency and accuracy with some common randomized numerical linear algebra tasks. It could be seen from our results that sparse operators and dense operators both perform well while differing a lot when matrices are ill-conditioned. SRTTs are especially unstable and perform poorly sometimes compared with the other two kinds.

Keywords Randomized Linear Algebra · Low-rank Approximation · Machine Learning · Sketching Techniques

1 Introduction

Due to the ubiquity of linear algebra in applied mathematics, dimension reduction and memory saving have been perpetual topics as there is an ever-increasing demand for solving larger problems faster. In 1984, it was proven that projecting onto a random basis approximately preserves pairwise distances with high probability [8], thereby opening the doors to using randomized techniques.

Sketching – projecting matrices of interest to a lower-dimensional subspace – forms the backbone of aptly named sketch-and-solve RLA algorithms, especially useful for matrix approximation and compression to speed up computations. It mainly has two steps. First, reduce the problem to one of a smaller dimension, and secondly, apply the deterministic algorithm to the reduced problem [12, 10]. The difference between those algorithms is mainly demonstrated in the first step where different sketching operators are employed. A large variety of different techniques have been proposed to construct random sketching operators both recently and historically[1, 13].

These years, randomized algorithms have gained a lot of attention within the scientific computation domain as well as improving machine learning algorithms for their efficiency [15, 6]. In our work, we examine the performances of those different sketching matrices solving SVD and least squares, which can be the building blocks for classification and regression. Our research aims at giving us a glimpse into how randomized linear algebra could help machine learning. Specifically, our work focuses on investigating the accuracy and efficiency of different sketching matrix structures both from the theoretical and experimental perspectives. Our code is avaliable at `https://github.com/KonstantinZoerner/Math221-Project`

2 Sketching Matrices

Here we define the size of the original matrix A as $m \times n$, and the size of the sketching operator S as $k \times m$, where $k \leq m$. Sketching matrices are linear maps defined as $(1 \pm \varepsilon)\ell_2$ embeddings, such that equation 1 is satisfied.

$$(1-\varepsilon)\|Ax\|_2^2 \leq \|SAx\|_2^2 \leq (1+\varepsilon)\|Ax\|_2^2 \qquad (1)$$

[*]Finished as a visiting student in UC Berkeley

with a certain probability depending on the structure of S, which represents how much SA deviates from being an isometry for A. The distortion ε can be explicitly calculated as follows [11]:

$$\varepsilon = \|I - (A^T A)^{-1/2}(SA)^T(SA)(A^T A)^{-1/2}\|_2 \quad (2)$$

We introduce several basic types of sketching matrices as follows.

2.1 Random Orthogonal Matrices

Arguably, one of the simplest sketch matrix types is the random orthogonal matrix. It is defined by the Johnson–Lindenstrauss lemma [3, 8], which states the following:

Lemma 1 *For $0 < \varepsilon < 1$ and any integer n, for $k \geq 8\varepsilon^{-2}\log n$, then for any set X of n vectors in $\mathbb{R}^m$, there is a random orthogonal matrix scaled by $\sqrt{m/k}$, S, such that for all $x_i \in X$,$(1-\varepsilon)||x_i - x_j||_2^2 \leq ||S(x_i - x_j)||_2^2 \leq (1+\varepsilon)||x_i - x_j||_2^2$ for all $1 \leq i, j \leq n, i \neq j$ with probability greater than or equal to $1/n$.*

The random orthogonal matrix defined by the Johnson-Lindenstrauss lemma is impractical due to the fact that the embedding dimension is proportional to ε^{-2} [12], making it difficult to realize the dimension-reducing capabilities of RLA.

2.2 Dense sketching operators

Dense sketching operators are matrix operators with entry-wise i.i.d. entries drawn from particular distributions. We introduce three kinds of operators for this survey, and their introduction could be checked in the supplementary materials.

Universality principles in high-dimensional probability [14] guarantee that these sketching operators are practically equivalent. [2]

Because it may take a prohibitively large amount of time conducting matrix multiplication ($\mathcal{O}(kmn)$ time) involved in sketching, the intended use case for dense sketching operators is mainly cases where the sketching operator is far smaller than the data to be sketched(e.g. low-rank approximation).In any case, for S a $k \times m$ sketching operator, the distortion for a sketched matrix $\varepsilon \in \Theta(\sqrt{n/k})$, where A is $m \times n$ [11].

2.3 Subsampled Random Trigonometric Transformations

Subsampled random trigonometric transformations (SRTTs) are structured sketching operators based on trigonometric transformations like the discrete Fourier transform, which are used to help embed the target matrix into a lower-dimensional subspace.

These sketching operators are defined as follows [5]:

$$S = \sqrt{\frac{m}{k}} RTD \quad (3)$$

Where D is a $m \times m$ diagonal matrix whose diagonal entries are uniformly distributed around the unit circle in the complex case (and in the real case, ± 1), T is a trigonometric transform, and $R \in \mathbb{R}^{k \times m}$ is a sampling matrix, selecting k rows from the $m \times m$ matrix it multiplies.

Depending on the initial data or other problem parameters, T can be a discrete Fourier transform or the discrete Hadamard transform.The Hadamard transform functions as a sort of analogue for the Fourier transform for real data, having a distortion $\varepsilon \in \Theta(\sqrt{(n\log n)/d})$ [11]. In practice, the sketching operator can be applied to each column vector x of the initial matrix A, thereby reducing the sketching operation to be of order at most $\mathcal{O}(mk\log m)$.

For a positive constant C and for $k \gtrapprox C(d + \log m)\log d$, then with high probability, for an d-dimensional subspace, the SRFT (subsampled random Fourier transform) matrix S is a subspace embedding with distortion $\frac{1}{2}$[16].

2.4 Sparse sketching operators

Sparse sketching operators are usually constructed by independently generating the rows or columns of a sketching operator such that the final S is sparse. There exists another type of sparse sketching operator – the i.i.d. sparse sketching operator – constructed by randomly setting many of a dense sketching operator's entries to 0. However, because of their (comparatively) more random structure, their theoretical guarantees are not as robust as the aforementioned row-by-row or column-by-column sketching operator [13].

A major advantage of these operators is the fact that sketching only takes $\mathcal{O}(\texttt{nnz}(S))$ time because of S's sparsity structure, allowing them to be the fastest sketching operators presented. Nevertheless, this speed comes with a tradeoff, namely, their distortion, because they inherently "see" less of the target matrix A than the previous two sketching architectures.

A prototypical example of a sparse sketching operator is the Clarkson-Woodruff transform (CWT), also known as the CountSketch matrix [2]. This matrix is generated by randomly choosing one element in each of its columns to be equal to ± 1 with equal probability, and setting the rest to 0. Using this as a sketching matrix, we have that the distortion $\varepsilon \in \Theta(\sqrt{n^2/k})$ [11].More examples could be checked in the supplementary materials.

[2]This applies to any such matrix when each of the entries are independent random variables, have mean 0 and variance 1, are drawn from a symmetric distribution, and have uniformly bounded moments [14].

2.5 Properties of Sketching Matrices and Sketch Quality

Now that we have introduced all of the sketching methods that we will use, we can look back at the two properties (1) embedding property and (2) distortion introduced in Section 2 which serve as our criterion for sketch quality. In practice, it oftentimes suffices to require a relaxed version of the embedding property (1), which requires a sketching matrix preserves relative norms [13], i.e.

$$\frac{\|Su\|_2}{\|Su\|_2} \approx \frac{\|u\|_2}{\|v\|_2} \quad \text{for } u, v \in \text{col}\{A\}. \tag{4}$$

To illustrate that the introduced sketching methods fulfill this property, we depict how they impact the relative norm in the supplementary materials alongside the computed distortion of some of the sketching methods for a matrix $A \in \mathbb{R}^{1024\times 50}$ with entries i.i.d. sampled from a standard normal distribution, and all these sketching matrices are of the same size.

3 Test Problems

3.1 Low-Rank Approximation

The low-rank approximation problem can be stated as follows: *Given an $m \times n$ matrix A, find a matrix A_k such that* $\text{rank}(A_k) = k \ll \min(m, n)$ *and* $\|A - A_k\|_2$ *satisfy some requirements.*

According to the two purposes of low-rank approximation, it could be classified into a fixed-rank one and a fixed-precision one. In our work, we mainly discuss the fixed-rank one. There are two primary approaches to solving this problem, which either focus on the spectrum of the target matrix or on its submatrices. Here, we focus on the former one, and it is best exemplified through SVD. This approach gives the best possible rank k approximation by the Eckart-Young-Mirsky theorem [4], though infeasible as it takes $O(mn^2)$.

With randomized algorithms, though, the problem could be significantly reduced; the illustration of SVD and randomized SVD could be checked in the supplementary materials.

3.2 Overdetermined Least Squares

The overdetermined least squares problem is defined as follows:

For $A \in \mathbb{R}^{m\times n}, x \in \mathbb{R}^n, b \in \mathbb{R}^m$ such that $m > n$, minimize $\|Ax - b\|_2^2$.

This can be solved by a number of deterministic algorithms with various trade-offs between speed, accuracy, and stability.With randomized algorithms, we follow the sketch-and-solve routine.

4 Experiments

4.1 Datasets

We use the `breastMNIST` training dataset from the `MedMNIST` database [17] for low-rank approximation, which has 546 data points[3]. This data is almost of full rank and is poorly conditioned. In addition to that, we use scikit-learn's dataset on Californian housing[4], which is suitable for least squares. As the dataset contains 20640 data points, we sample a smaller, easier-to-handle number of 1024 data points.

Our synthetically generated data are split into three different categories. First, we use matrices A with all entries sampled i.i.d. from a standard normal distribution, which behave nicely numerically and serve as a good base case. Secondly, we use matrices with singular values that span a wide range, a fact which results in a high condition number. In particular, we select $A = U\Sigma V^T$ with U, V orthogonal and Σ diagonal with $\sigma_{ii} = e^{k_i}$, where k_i are equidistantly spread in $\{-10, 10\}$. Last, we utilize multicollinear matrices that we generate by sampling a vector a from an i.i.d. standard normal distribution and then setting A's i^{th} column to $a_i = a + 10^{-6}\theta_i$ where θ_i are standard normal random vectors with independent components.

4.2 Results

4.2.1 Low Rank Approximation

To align with the metric that we use in least squares, here we use the relative error compared with the optimal one [13] on numerical experiments and the relative error compared with the original data on a real-world dataset, defined as follows

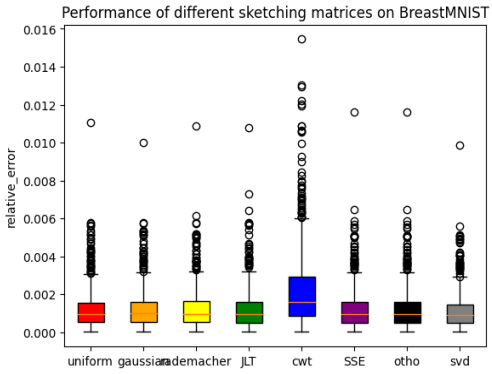
$$\text{Error} = \frac{||\hat{A} - A_k||_F}{A_k} \tag{5}$$

Note that for the `breastMNIST` experiments, we are using the squared error, and for numerical experiments, we are using the non-squared ones.

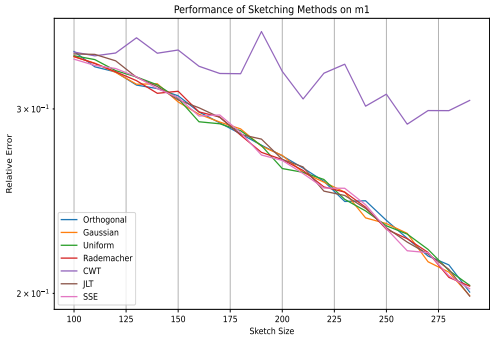
Our results for BreastMNIST are shown in Figure 1aWe also conduct numerical experiments solving the low-rank approximation problem. The results are shown in Figure 1b and Figure 2. On `breastMNIST`, we could see that the result is almost the same. It is worth attention that the three matrices we used for numerical experiments vary in their condition number. For the `m1` matrix, a relatively well-conditioned random Gaussian matrix was used; the other two are poorly conditioned. A little bit different from the ill-conditioned matrices that we chose in least

[3] `https://zenodo.org/records/10519652`

[4] `https://scikit-learn.org/1.5/modules/generated/sklearn.datasets.fetch_california_housing.html`

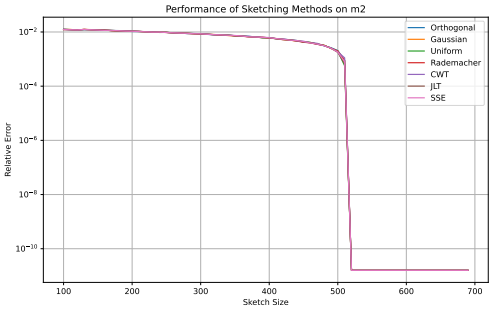


(a) Randomized SVD results on `BreastMMNIST` dataset



(b) 300-rank approximation for $A \in \mathbb{R}^{512\times 1024}$

Figure 1: Results for low-rank approximation



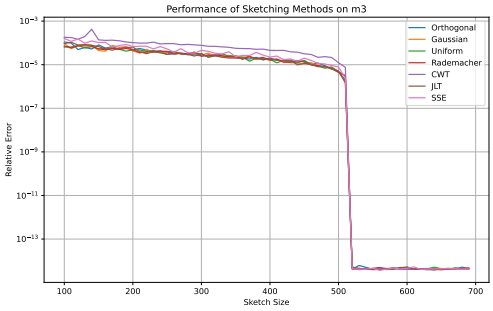


Figure 2: Numerical Results of Different kinds of Sketching Matrices on Some Very Ill-conditioned Matrices

squares approximation. The first ill-conditioned one is the matrix with singular values over a large range with noises; the second one is a Hilbert matrix. It could be seen from those figures that there is an 'optimal' rank where the relative errors drop dramatically, especially for ill-conditioned matrices.

4.2.2 Least Squares

We solved the least squares problem $\min_x \|Ax - b\|_2$ for the three types of matrices described in Section 4.1 using all sketching methods introduced so far using both the QR and the SVD algorithm that is outlined in the supplementary materials. For all choices of A, we chose b as a standard normal random vector with independent components. We found both methods to yield the same accuracy, so we only depict the results for QR here. In order to measure the accuracy of the different sketching methods, we used the following two metrics. First, the relative norm of the residual, i.e.,

$$\frac{\|Ax_{\text{opt}} - b\|_2 - \|Ax_{\text{sketch}} - b\|_2}{\|Ax_{\text{opt}} - b\|_2}, \tag{6}$$

and second, the relative error of the found x_{sketch}, i.e.,

$$\frac{\|x_{\text{opt}} - x_{\text{sketch}}\|_2}{\|x_{\text{opt}}\|_2}. \tag{7}$$

Repeating these experiments for many different sizes of A yields genuinely similar behavior. Thus, in Figure 6 in the supplementary materials we only chose to depict the results for $A \in \mathbb{R}^{256\times 20}$.

We repeated the same process for the Californian housing dataset introduced in Section 4.1. Here we found that the trigonometric transforms performed very poorly, so we decided to exclude them from the plot. The plot can be checked in the supplementary materials.

5 Conclusion

First, we verified that the orthogonal transform achieves the best performance, as shown in Figure 6 and that all dense i.i.d. sketching operators behave equivalently and robustly. However, sparse operators with poorly conditioned matrices having spectra covering a wide range, we observe different behavior. In this case, the CountSketch operator (CWT) performs the worst. Their performances also vary more when relative errors computed are larger, which could also be due to their ill-conditionedness. At the same time, results from SRFTs sometimes perform very poorly, which we leave to research in the future.

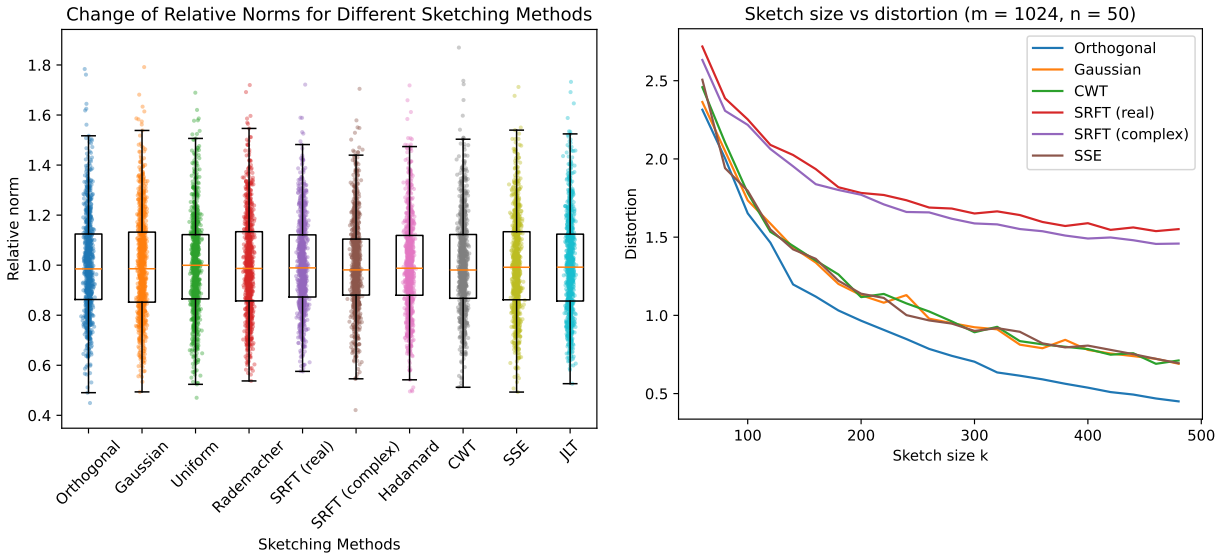


Figure 4: Illustration of the embedding property for different sketching methods (left). Distortion for a matrix $A \in \mathbb{R}^{1024\times 50}$ with i.i.d. standard normal entries for different sketching methods (right).

6 supplementary materials

6.1 Sketching Properties

The sketching properties could be checked in Figure 4

6.2 SVD

- **Singular Value Decomposition**: Factorizing the target matrix as $A = U\Sigma V^T$ with U, V orthogonal and Σ diagonal. By denoting u_i and v_i to be the column vectors of U and V respectively and $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n)$ and using the convention that $\sigma_1 \geq \cdots \geq \sigma_n = 0$, we can also write A as follows $A = \sum_{i=1}^{n} \sigma_i u_i v_i^T$ and form A_k by truncating the sum at some $k \leq n$.

The implementation of randomized SVD, see 1.

6.3 Dense Operators

- **Rademacher sketching operators**: entries are ± 1 with equal probability.
- **Uniform sketching operators**: entries are sampled from a uniform distribution over a symmetric interval.
- **Gaussian sketching operators**: entries are sampled from a normal distribution with mean 0.

6.4 Sparse Operators

The one we mentioned in the main text can be generalized to a Sparse Sign Embedding (SSE) [7], which can contain more than one non-zero entry per column. Then with high probability, for any d-dimensional subspace, S is an embedding with constant distortion $\frac{1}{2}$ [16].

These can also be generated by specifying a sparsity parameter ζ to represent the number of non-zero elements per column. Analogously, it has been shown that a sparse sign matrix serves as a subspace embedding with high probability with constant distortion for an arbitrary l-dimensional subspace of $\mathbb{R}^m$ when the embedding dimension grows $\mathcal{O}(d \log d)$ and the sparsity parameter ζ as $\mathcal{O}(\log d)$ [12].

Another scheme labeled JLT [9] demonstrated in our numerical experiments is a sparse random Rademacher matrix, also according to the lemma 1. Here we use the implementation from `https://github.com/dell/jlt/blob/main/linearMapping.py`.

6.5 Solving Overdetermined Least Squares

- **Normal equations**: Conceptually the simplest, it solves the problem by letting $x = (A^T A)^{-1} A^T b$. Despite requiring the least amount of floating point operations of all the following methods, this method method is not very suitable for practical applications because it is unstable for poorly-conditioned A.
- QR **decomposition**: This method performs the following factorization: $A = QR, Q \in \mathbb{R}^{m\times n}, R \in \mathbb{R}^{n\times n}$ such that Q is orthogonal and R is upper triangular for the solution $x = R^{-1}Q^T$. QR factorization can be achieved through a number of algorithms such as Gram-Schmidt or modified Gram-Schmidt, or those which incrementally construct the result using Householder reflections or Givens rotations. In practice, while all of these algorithms have complexity $\mathcal{O}(mn^2)$, Householder rotations are the most efficient among all four, while still being stable and preserving the orthogonality of Q's columns.
- **Singular Value Decomposition**: This method performs the following factorization $A = U\Sigma V^T$, where U, V are

Algorithm 1 Sampling and Projection of the Target Matrix

Input:A, k, s ▷ Target Matrix, Desired Rank, Oversampling Parameter
Output: B
1: $S = \text{sketch\_matrix}(m, k+s)$ ▷ Generates a $m \times k+s$ sketching matrix
2: $Y = SA$
3: $Q, \_ = \text{qr}(Y)$ ▷ Finds the orthogonal component of the sketched matrix
4: $B = Q^T A$ ▷ Projects the target matrix to the lower dimensional subspace

orthogonal and Σ is diagonal. The SVD van be used to solve the problem by setting $x = V\Sigma^+ U^T b$ where Σ^+ is the psuedoinverse of Σ. While the most numerically stable, the SVD takes $\mathcal{O}(mn^2)$, not to mention the matrix multiplication required to form x.

6.6 California Housing Dataset

See 5 for reference.

6.7 Numerical Results from Least Squares

See 6 for reference.

References

[1] Oleg Balabanov, Matthias Beaupere, Laura Grigori, and Victor Lederer. Block subsampled randomized hadamard transform for low-rank approximation on distributed architectures, 2022.

[2] Kenneth L. Clarkson and David P. Woodruff. Low-rank approximation and regression in input sparsity time. *J. ACM*, 63(6), January 2017.

[3] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

[4] Carl Eckart and G. Marion Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1:211–218, 1936.

[5] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, 2010.

[6] Filip Hanzely. *Optimization for Supervised Machine Learning: Randomized Algorithms for Data and Parameters*. PhD thesis, 2020.

[7] Dong Hu, Shashanka Ubaru, Alex Gittens, Kenneth L. Clarkson, Lior Horesh, and Vassilis Kalantzis. Sparse graph based sketching for fast numerical linear algebra. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3255–3259, 2021.

[8] William B. Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into hilbert space. *Contemporary mathematics*, 26:189–206, 1984.

[9] Daniel M. Kane and Jelani Nelson. A sparser johnson-lindenstrauss transform. *CoRR*, abs/1012.1577, 2010.

[10] Alex Lavaee. Sketch 'n solve: An efficient python package for large-scale least squares using randomized numerical linear algebra, 2024.

[11] Malik Magdon-Ismail and Alex Gittens. Fast fixed dimension l2-subspace embeddings of arbitrary accuracy, with application to l1 and l2 tasks, 2019.

[12] Per-Gunnar Martinsson and Joel Tropp. Randomized numerical linear algebra: Foundations & algorithms, 2021.

[13] Riley Murray, James Demmel, Michael W. Mahoney, N. Benjamin Erichson, Maksim Melnichenko, Osman Asif Malik, Laura Grigori, Piotr Luszczek, Michał Dereziński, Miles E. Lopes, Tianyu Liang, Hengrui Luo, and Jack Dongarra. Randomized numerical linear algebra : A perspective on the field with an eye to software, 2023.

[14] Samet Oymak and Joel A Tropp. Universality laws for randomized dimension reduction, with applications. *Information and Inference: A Journal of the IMA*, 7(3):337–446, 2018.

[15] Zaid Bin Tariq, Jayson P Van Marter, Anand G Dabak, Naofal Al-Dhahir, and Murat Torlak. A data-driven signal subspace approach for indoor bluetooth ranging. *IEEE Journal of Indoor and Seamless Positioning and Navigation*, 2024.

[16] Joel A. Tropp. ACM 204: Randomized Algorithms for Matrix Computations, August 2023.

[17] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

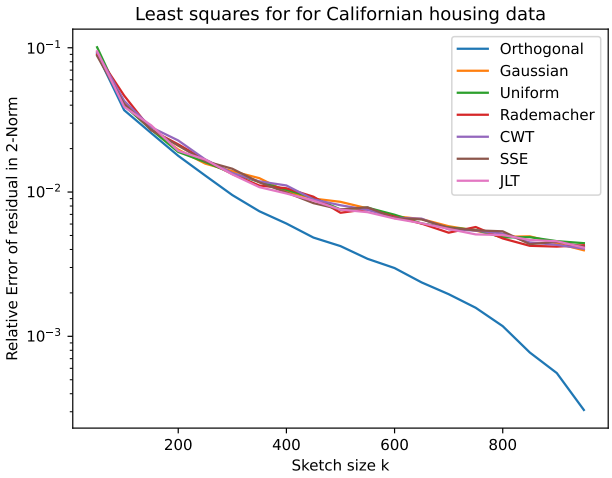


Figure 5: California Housing Dataset with Least Squares

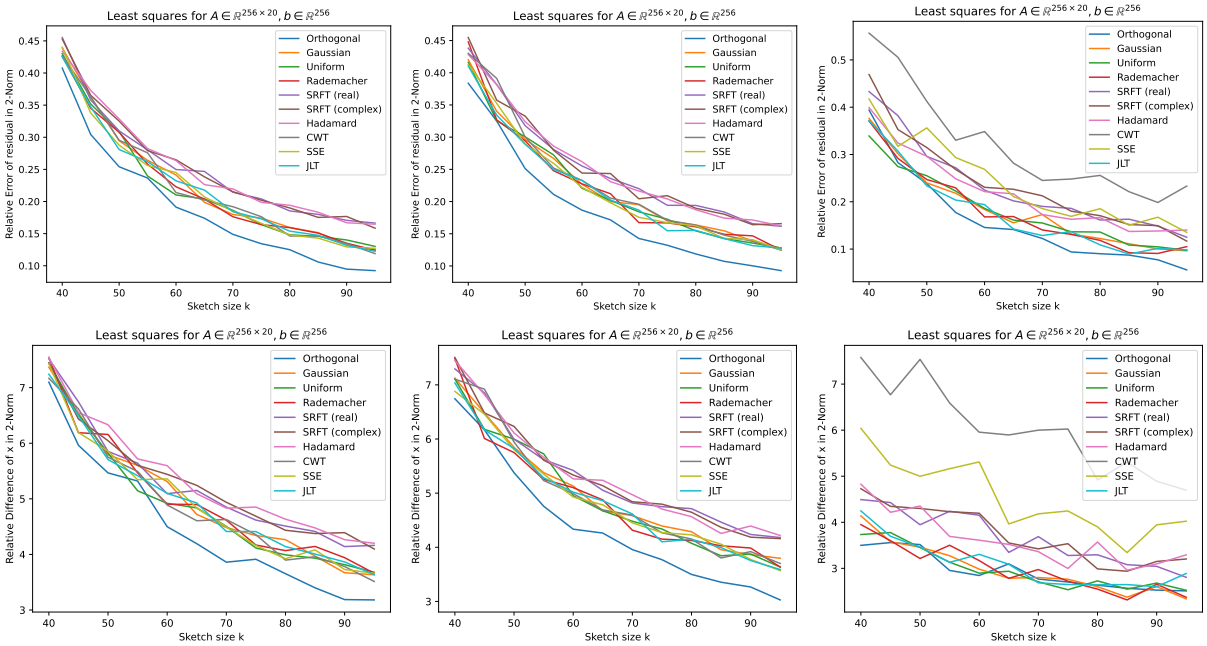


Figure 6: Accuracy of least squares using different matrices