# A Very Sketchy Talk

David P. Woodruff
Carnegie Mellon University, Pittsburgh, PA, USA

## Abstract

We give an overview of dimensionality reduction methods, or sketching, for a number of problems in optimization, first surveying work using these methods for classical problems, which gives near optimal algorithms for regression, low rank approximation, and natural variants. We then survey recent work applying sketching to column subset selection, kernel methods, sublinear algorithms for structured matrices, tensors, trace estimation, and so on. The focus is on fast algorithms. This is a short survey accompanying an invited talk at ICALP, 2021.

**2012 ACM Subject Classification** Theory of computation → Streaming, sublinear and near linear time algorithms

**Keywords and phrases** dimensionality reduction, optimization, randomized numerical linear algebra, sketching

**Digital Object Identifier** 10.4230/LIPIcs.ICALP.2021.6

**Category** Invited Talk

**Funding** *David P. Woodruff*: Supported by the Office of Naval Research (ONR) grant N00014-18-1-2562, the National Science Foundation (NSF) under Grant No. CCF-1815840, and a Simons Investigator Award.

## 1 Introduction

*Sketching*, or data dimensionality reduction, is a popular tool for speeding up algorithms in machine learning, optimization, and randomized numerical linear algebra.

In the overconstrained least squares regression problem one is given an $n \times d$ matrix $A$, $n > d$, together with an $n \times 1$ vector $b$, and one is tasked with finding an $x \in \mathbb{R}^d$ for which $\|Ax - b\|_2^2 = \sum_{i=1}^n (\langle A_i, x\rangle - b_i)^2$ is as small as possible, where $A_i$ is the $i$-th row of $A$, $b_i$ is the $i$-th entry of $b$, and $\langle A_i, x\rangle$ denotes the inner product between $A_i$ and $x$. Geometrically, in $\mathbb{R}^n$ one can view this as finding the vector $Ax$ which is closest to $b$ in the column span of $A$ (which is a $d$-dimensional subspace) in terms of Euclidean distance, that is, $Ax$ is just the projection of $b$ onto the column span of $A$. Alternatively, in $\mathbb{R}^{d+1}$ one can think of having $n$ points, the $i$-th of which is $(A_i, b_i)$, and one is trying to find a hyperplane defined by $x$ so as to minimize the sum of squares of distances between the points $(A_i, \langle A_i, x\rangle)$ and the points $(A_i, b_i)$. There is a closed-form solution to this problem of the form $x = A^- b$, where $A^-$ is the Moore-Penrose pseudoinverse of $A$, and the optimal $x$ can be computed in $O(nd^2)$ time by computing the singular value decomposition (SVD) [1] of $A$. While this is an exact solution, the $O(nd^2)$ running time is prohibitive for large values of $n$ and moderate values of $d$.

The sketch-and-solve paradigm instead solves this problem by first choosing a random matrix $S \in \mathbb{R}^{k\times n}$, where $k \ll n$. One then computes $S \cdot A$, which is a small $k \times n$ matrix, as well as $S \cdot b$, which is a small $k \times 1$ vector. One then solves the much smaller regression problem $\min_x \|SAx - Sb\|_2$ by computing its minimizer $x' = (SA)^- Sb$, and the hope is that $\|Ax' - b\|_2^2 \leq (1+\epsilon)\|Ax^* - b\|_2$, where $x^* = A^- b$ is the minimizer of $\|Ax - b\|_2^2$.

[1] This can be sped up using theoretical algorithms for fast matrix multiplication, giving $O(n \cdot d^{\omega-1})$ time, where $\omega \approx 2.376$ is the exponent of fast matrix multiplication.

48th International Colloquium on Automata, Languages, and Programming (ICALP 2021).
Editors: Nikhil Bansal, Emanuela Merelli, and James Worrell; Article No. 6; pp. 6:1–6:8
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A natural question is how to choose the sketching matrix $S$. One could choose $S$ to be a $k \times n$ matrix of independent and identically distributed $N(0, 1/k)$ random variables (normal with mean 0 and variance $1/k$), where $k = O(d/\epsilon^2)$, which turns out to work (see, e.g., discussion in [39]), but then computing $S \cdot A$ would take at least $nd^2$ time naïvely, which although it can be sped up with fast matrix multiplication, is slower than just computing the exact solution $x^* = A^- b$. Sárlos [32] pioneered the sketch-and-solve paradigm and observed that one could instead choose $S$ to be a so-called Subsampled Randomized Hadamard Transform, that is, $S = P \cdot H \cdot D$, where $D$ is an $n \times n$ diagonal matrix with independent diagonal entries each chosen uniformly in $\{-1, 1\}$, $H$ is the $n \times n$ Hadamard matrix (assuming $n$ is a power of 2), and $P$ uniformly samples $d \operatorname{poly}(\log d)/\epsilon^2$ entries of whichever vector it is applied to (see [20] for optimizations to the logarithmic factors). Then $S \cdot A$ and $S \cdot b$ can now be computed in $O(nd \log n)$ time; indeed, this follows since $D$ and $P$ can be applied to a vector in $O(n)$ time, and using the recursive structure defining $H$ the matrix $H$ can be applied to a vector in $O(n \log n)$ time. Consequently, $SA = PHDA$ can be computed in $O(nd \log n)$ time. One can then solve $\min_x \|SAx - Sb\|_2$ in $d^3 \operatorname{poly}(\log d)/\epsilon^2$ time, and the solution $x'$ can be shown to, with large probability, satisfy $\|SAx' - Sb\|_2 \leq (1+\epsilon)\|Ax^* - b\|_2$. This gives a runtime of $\tilde{O}(nd \log n + d^3/\epsilon^2)$, where the notation $\tilde{O}(f)$ denotes $f \cdot \operatorname{poly}(\log f)$. The additive $d^3/\epsilon^2$ term can be further improved using fast matrix multiplication algorithms.

This was later improved by Clarkson and Woodruff [14] who showed that one could instead choose the so-called CountSketch matrix $S$ [12] as one's sketching matrix; the analysis was later simplified and improved in [24, 27, 8, 17]. Here $S$ is $k \times n$, where $k = O(d^2/\epsilon^2)$ is again independent of the large dimension $n$. The key property is that $S$ has a single randomly chosen non-zero entry per column, which is chosen at a uniformly random position and is uniform in $\{-1, 1\}$, and chosen independently across the columns. The key property is that now $SA$ and $Sb$ can be computed in $\operatorname{nnz}(A)$ time, where $\operatorname{nnz}(A)$ denotes the number of non-zero entries of $A$, and that the solution $x'$ to $\min_x \|SAx - Sb\|_2$ is such that $\|Ax' - b\|_2 \leq (1+\epsilon)\|Ax^* - b\|_2$. The proof in [14] departed from previous proofs which first argued that any fixed vector $y$ has its norm preserved up to a $(1 \pm \epsilon)$-multiplicative factor with probability $1 - 2^{-O(d)}$; after this, a standard net argument could then be used. In fact CountSketch does not have this property and [14] instead observed that the $2^{O(d)}$ vectors one is interested in preserving the norms of, all live in a low-dimensional subspace, and consequently, the number of "heavy coordinates" as one ranges over all vectors in the subspace is a small subset of all possible $n$ coordinates. This then enables CountSketch to work in the sketch-and-solve paradigm, and gives an overall algorithm for solving regression in $\operatorname{nnz}(A) + \operatorname{poly}(d/\epsilon)$ time.

While we have the property that $\|Ax' - b\|_2^2 \leq (1+\epsilon)\|Ax^* - b\|_2^2$, this guarantee is often not sufficient in machine learning and optimization tasks, and one would instead like to bound $\|x^* - x'\|_2^2$. Indeed, one could hope $x'$ is close to a "ground truth" hyperplane and therefore give good generalization error. To do so, note that

$$\begin{aligned}
\|x^* - x'\|_2^2 &\leq \frac{\|Ax^* - Ax'\|_2^2}{\sigma_{min}^2(A)} \\
&\leq \frac{\|Ax' - b\|_2^2 - \|Ax^* - b\|_2^2}{\sigma_{min}^2(A)} \\
&\leq \frac{((1+\epsilon)^2 - 1)\|Ax^* - b\|_2^2}{\sigma_{min}^2(A)} \\
&\leq \frac{O(\epsilon)\|Ax^* - b\|_2^2}{\sigma_{min}^2(A)},
\end{aligned}$$

where the first inequality follows from the definition of the minimum singular value, the second inequality follows since $Ax^* - b$ and $Ax^* - Ax'$ are orthogonal, and the third inequality follows from the objective function guarantee discussed above. We note that this simple guarantee was significantly improved in [30], where the authors bounded $\|x^* - x\|_\infty^2$, i.e., the difference on every single coordinate; this improved analysis holds for the Subsampled Randomized Hadamard Transform as well as Gaussian sketches, but not CountSketch.

Another unsatisfactory aspect of the above is that the dependence on the approximation factor $\epsilon$ is polynomial, rather than polylogarithmic. To achieve the latter, one can combine sketching and optimization techniques in a somewhat different way. One can first run the sketch-and-solve paradigm with CountSketch with a constant $\epsilon_0 = 1/2$ to find an $x'$ with $\|Ax' - b\|_2^2 \leq 3/2 \cdot \|Ax^* - b\|_2^2$. This step takes $\text{poly}(d)$ time independent of the value of $\epsilon$. Let $x_0 = x'$. In parallel, one could compute $S \cdot A$ for a CountSketch matrix $S$ with $\epsilon_0 = 1/2$. Then write $SA = QR$, where $Q$ is a matrix with orthonormal columns; one could find $QR$ by letting $SA = U\Sigma V^T$ be its SVD and setting $Q = U$. This step takes $\text{poly}(d)$ time independent of the value of $\epsilon$. Let

$$\kappa(AR^{-1}) = \frac{\sigma_{max}^2(AR^{-1})}{\sigma_{min}^2(AR^{-1})}.$$

It is not hard to see that $\kappa(AR^{-1}) \leq 3$. Indeed, by the so-called subspace embedding property of $S$, we have for all $x$:

$$\frac{1}{2}\|Ax\|_2^2 \leq \|SAx\|_2^2 \leq \frac{3}{2}\|Ax\|_2^2.$$

This means for all unit vectors $x$, $\|AR^{-1}x\|_2^2 \leq (3/2)\|SAR^{-1}x\|_2^2 = 3/2$, and similarly for all unit vectors $x$, $\|AR^{-1}x\|_2^2 \geq (1/2)\|SAR^{-1}x\|_2^2 = 1/2$. Here the equalities follow from the fact that $SAR^{-1} = Q$, which has orthonormal columns, so $\|Qx\|_2^2 = 1$ for all unit vectors $x$. Using that $\sigma_{max}(B) = \sup_{\text{unit } x} \|Bx\|_2$ and $\sigma_{min}(B) = \inf_{\text{unit } x} \|Bx\|_2$ for a matrix $B$ with more rows than columns, we have:

$$\sigma_{max}^2(AR^{-1}) \leq \frac{3}{2} \cdot \sigma_{max}^2(SAR^{-1}) = \frac{3}{2} \cdot 1 = \frac{3}{2}.$$

Here the equality follows from the fact that $SAR^{-1} = Q$, and $Q$ has orthonormal columns, and thus $\sigma_{max}(Q) = \sigma_{min}(Q) = 1$. Similarly,

$$\sigma_{min}^2(AR^{-1}) \geq \frac{1}{2} \cdot \sigma_{min}^2(SAR^{-1}) = \frac{1}{2}.$$

Consequently, $\kappa(AR^{-1}) \leq 3$. At this point, one can simply run gradient descent on the function $f(x) = \frac{1}{2}\|AR^{-1}x - b\|_2^2$ with initial solution $Rx_0$. By standard arguments, the number of iterations required to get $\epsilon$ error is $O(\kappa \log(1/\epsilon)) = O(\log(1/\epsilon))$. Moreover, one never needs to explicitly compute $A \cdot R^{-1}$. Indeed, given an iterate $x_t$ in some iteration $t$, one can compute $R^{-1}x_t$ and then $AR^{-1}x_t$, in $O(d^2 + \text{nnz}(A))$ time per iteration, and thus $O((\text{nnz}(A) + d^2)\log(1/\epsilon))$ time overall. This, together with the additive $O(\text{nnz}(A) + \text{poly}(d))$ time needed to find $R^{-1}$ and $x_0$, gives an overall running time of $O((\text{nnz}(A) + d^2)\log(1/\epsilon) + \text{poly}(d))$. We refer the reader to [14] for further details.

## 2 Extensions

There are many related problems to regression (and other problems!) for which sketching can be applied, and we outline only a few here.

## 2.1 Ridge Regression

There are regularized variants of regression, such as ridge regression, where one instead seeks to minimize $\|Ax-b\|_2^2+\lambda\|x\|_2^2$, for a parameter $\lambda>0$. Here the $\lambda\|x\|_2^2$ term is known as the ridge or regularization, and encourages low-norm solutions. This formulation is useful both when $n>d$ as well as when $n<d$; in the latter case $A$ is underconstrained, i.e., has more columns than rows, and without the regularization there are multiple solutions possible and one is often interested in a low-norm solution. We remark that low-norm solutions often have better properties for applications, e.g., may not overfit the data as much, and by setting $\lambda$ to be large one encourages low norm solutions. For ridge regression, one can sketch $A$ with a dimension that depends on the so-called statistical dimension $sd_\lambda(A)=\sum_i \frac{\sigma_i^2}{\lambda+\sigma_i^2}$, which is always bounded by the rank of $A$ though can be much smaller if a few values $\sigma_i$ are very large and $\lambda$ is set appropriately. Sketching is thus immediately useful in the overconstrained case when $n>d$, since $sd_\lambda(A)$ may be much less than $d$, and so the sketching dimension is much smaller. In the underconstrained case, one often instead *sketches on the right*, setting up the problem $\min_y\|ARy-b\|_2^2+\lambda\|Ry\|_2^2$, for a sketching matrix $R$. In this case sketching ultimately allows for solving the problem in $\text{nnz}(A)+\text{poly}(sd_\lambda\sigma_1(A)/(\epsilon\lambda))$ time, which although depends on $\sigma_1(A)$, can still be useful. We note that one can combine sketching on both the left and the right for ridge regression; we refer the reader to [2, 3] and the references therein for further details and the history of sketching as applied to this problem.

## 2.2 Kernel Regression

Another application is kernel regression. In the kernel setting one is given $n$ points $x^1,\ldots,x^n\in\mathbb{R}^d$ and one would like to apply an often non-linear mapping $\phi$ to "lift" them to a feature space. A notable example is the polynomial kernel of degree $q$, where $\phi:\mathbb{R}^d\to\mathbb{R}^{d^q}$ where $\phi(x)_{i_1,i_2,\ldots,i_q}=x_{i_1}\cdot x_{i_2}\cdots x_{i_q}$. One reason the polynomial kernel is so important is that one can often Taylor-expand other kernels, such as the Gaussian kernel, and approximate them by a polynomial kernel of large enough degree. Define the $d^q\times n$ matrix $A$ with $i$-th column equal to $\phi(x^i)$. One would never want to compute this matrix, as the number $d^q$ of rows is prohibitively large. Nevertheless, one would like to be able to solve optimization problems with respect to this matrix.

In particular, in the kernel ridge regression problem one seeks to find a vector $y\in\mathbb{R}^d$ so as to minimize $\|A^TAy-b\|_2^2+\lambda\|y\|_2^2$. Initial work [28] showed how, given vectors $x^1,\ldots,x^q$, each in $\mathbb{R}^d$, to compute a sketch $S(x^1\otimes x^2\otimes\cdots\otimes x^q)$ without first having to compute the tensor product $x^1\otimes x^2\otimes\cdots\otimes x^q$, which would require an unreasonable $d^q$ amount of time. The rough idea is to apply separate sketches $S^1,\ldots,S^q$ to each of the $q$ "modes", obtaining $S^1x^1,S^2x^2,\ldots,S^qx^q$, where each $S^i$ is a CountSketch matrix. If $S^i$ has $k$ rows, then the coordinates of each $S^ix^i$ are associated with the coefficients of a degree-$(k-1)$ polynomial in a formal variable $z$. One then multiplies these polynomial modulo $z^k-1$ using the Fast Fourier Transform to improve efficiency. Interestingly, one can show this corresponds to applying another CountSketch $S$ (which is a function of $S^1,\ldots,S^q$) to the vector $x^1\otimes x^2\otimes\cdots\otimes x^q$, with certain structural properties (so $S$ is not a truly random CountSketch matrix, but nevertheless is good enough). Applying this to each column of $A$ separately, which has the form $(x^i)^{\otimes q}$, one can then solve the sketched kernel regression problem:

$$\min_y \|A^TS^TSAy-b\|_2^2+\lambda\|Ay\|_2^2,$$

where now one has $SA$ without ever having to materialize the matrix $A$.

While the initial work was well-suited for low degree polynomial kernels (small $q$), their dependence on the sketching dimension is exponential in $q$, making them less suitable for tasks such as approximating the Gaussian kernel, where $q$ is chosen to be at least logarithmic in $n$. In recent work [1], a binary tree scheme was used, together with additional sketching at each internal node, to design a linear and oblivious (not dependent on the input) sketch which reduces the dependence on the degree $q$ to polynomial. This was then successfully applied to sketching the Gaussian kernel.

## 2.3 Structured Matrix Regression

In a number of real-world instances of regression, the design matrix $A$ is structured, e.g., it might be a Hankel, Toeplitz, Vandermonde, or more generally a low-displacement rank matrix. Such matrices $A$ come with fast matrix multiplication algorithms, meaning one can compute $A \cdot x$ in $O(n \log n)$ or $n \cdot \operatorname{polylog}(n)$ time, which is often significantly faster than the $nd$ time needed to multiply an arbitrary $n \times d$ matrix $A$ with a vector $x$. Notice that the running time is sublinear in the time to write down the matrix $A$ and this leads to the quest for obtaining sublinear time algorithms for a number of optimization problems. Using dimensionality reduction-based methods [19, 33], if $T(A)$ is the time to multiply a given matrix $A$ by an arbitrary vector $x$, it is possible to $(1+\epsilon)$-approximate least squares regression in $T(A) \log n + \operatorname{poly}(d \log n/\epsilon)$ time, yielding sublinear time (in $nd$) for a number of structured regression problem.

# 3 Wrapping Up

While our focus in this short survey was on variants of regression, there is also a large body of work on applying sketching to other optimization problems. A very small set of examples includes the following:

- low rank approximation [14], where one seeks to approximate an $n \times d$ matrix $A$ by a product of an $n \times k$ matrix $L$ and a $k \times d$ matrix $R$, where $k \ll \min(n, d)$, and thus one can store $A$ with only $(n + d)k$ parameters as opposed to $nd$ parameters
- CUR decomposition [9], which is a special kind of low rank approximation where one seeks to approximate $A$ by $CUR$, where $C$ is an $n \times c$ matrix and consists of an actual subset of columns of $A$, $R$ is an $r \times d$ matrix and consists of an actual subset of rows of $A$, and $U$ is a small $c \times r$ arbitrary matrix. Here the hope is that $r, c$ are small and that this provides a more "interpretable" low rank approximation
- clustering, for which low-rank approximation can quickly provide an initial dimensionality reduction, which can then be used to create a coreset for problems such as $k$-means [18], or $k$-median [35, 21]
- distributed, sliding window, and streaming computation (see, e.g., [13, 10, 11], and references therein): since sketches provide a form of compression and can be easily updated due to their linearity, they are naturally useful for providing communication-efficient distributed algorithms as well as space-efficient algorithms in the sliding window and streaming models
- optimization, where sketching can be used for example to compress gradients in first order optimization [23], as well as inside of each iteration in second order methods [29], which often involve solving a least squares regression problem
- finding a latent simplex is an important problem in topic models and community detection, for which one is given $n$ data points that are formed by randomly perturbing some points that come from a latent simplex. Sketching was recently used to obtain truly input sparsity time algorithms in [4]

- trace estimation, where one is given an $n \times n$ positive semidefinite matrix $A$ and would like to estimate $\sum_{i=1}^{n} A_{i,i}$ up to a multiplicative factor of $1 + \epsilon$. Recently, fast sketching algorithms for low rank approximation were used in part to do this with constant probability using $O(1/\epsilon)$ matrix-vector products [25], improving the long-standing Hutchinson's algorithm which uses $\Omega(1/\epsilon^2)$ matrix-vector products.
- tensor low rank approximation [37] and weighted low rank approximation [31, 7], where one is given a tensor and would like to find a low rank approximation in some norm; such problems are often NP-hard and bicriteria algorithms, as well as fixed parameter tractable algorithms, have been proposed to obtain provable guarantees.
- robust variants of regression and low rank approximation, where the standard sum of squares error measure is replaced with more robust loss functions such as sum of absolute values [34, 15, 16, 36, 38]
- sublinear time low rank approximation, where one uses the structure of the input matrix to devise algorithms that achieve relative error low rank approximation in sublinear time, e.g., for positive semidefinite matrices [26, 5] or distance matrices [6, 22].

Sketching and dimensionality reduction are rapidly expanding areas. Some of these topics are covered in my older monograph [39]. See also the course notes for the "Algorithms for Big Data" class I teach at CMU, which contains much of this material[2]. I would like to thank the ICALP program committee for giving me the opportunity to write this, and my apologies for only covering a small part of the vast body of work on sketching and for focusing on work that I am most familiar with, and even unfortunately omitting many of those references as well. Hopefully though, if the reader has not seen sketching before, this document can serve as a short and simple introduction to the area.

## References

1 Thomas D. Ahle, Michael Kapralov, Jakob Bæk Tejs Knudsen, Rasmus Pagh, Ameya Velingker, David P. Woodruff, and Amir Zandieh. Oblivious sketching of high-degree polynomial kernels. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 141–160, 2020.

2 Haim Avron, Kenneth L. Clarkson, and David P. Woodruff. Faster kernel ridge regression using sketching and preconditioning. *SIAM J. Matrix Anal. Appl.*, 38(4):1116–1138, 2017.

3 Haim Avron, Kenneth L. Clarkson, and David P. Woodruff. Sharper bounds for regularized data fitting. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, pages 27:1–27:22, 2017.

4 Ainesh Bakshi, Chiranjib Bhattacharyya, Ravi Kannan, David P. Woodruff, and Samson Zhou. Learning a latent simplex in input-sparsity time, 2021.

5 Ainesh Bakshi, Nadiia Chepurko, and David P. Woodruff. Robust and sample optimal algorithms for PSD low rank approximation. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 506–516, 2020.

6 Ainesh Bakshi and David P. Woodruff. Sublinear time low-rank approximation of distance matrices. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 3786–3796, 2018.

[2] http://www.cs.cmu.edu/~dwoodruf/teaching/15859-fall17/index.html
http://www.cs.cmu.edu/~dwoodruf/teaching/15859-fall19/index.html
http://www.cs.cmu.edu/~dwoodruf/teaching/15859-fall20/index.html

7 Frank Ban, David P. Woodruff, and Richard Zhang. Regularized weighted low rank approximation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4061–4071, 2019.

8 Jean Bourgain, Sjoerd Dirksen, and Jelani Nelson. Toward a unified theory of sparse dimensionality reduction in euclidean space. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 499–508, 2015.

9 Christos Boutsidis and David P. Woodruff. Optimal CUR matrix decompositions. *SIAM J. Comput.*, 46(2):543–589, 2017.

10 Christos Boutsidis, David P. Woodruff, and Peilin Zhong. Optimal principal component analysis in distributed and streaming models. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 236–249, 2016.

11 Vladimir Braverman, Petros Drineas, Cameron Musco, Christopher Musco, Jalaj Upadhyay, David P. Woodruff, and Samson Zhou. Near optimal linear algebra in the online and sliding window models, 2020. arXiv:1805.03765.

12 Moses Charikar, Kevin C. Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theor. Comput. Sci.*, 312(1):3–15, 2004.

13 Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 205–214, 2009.

14 Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 81–90, 2013.

15 Kenneth L. Clarkson and David P. Woodruff. Input sparsity and hardness for robust subspace approximation, 2015. arXiv:1510.06073.

16 Kenneth L. Clarkson and David P. Woodruff. Sketching for $M$-estimators: A unified approach to robust regression. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 921–939, 2015.

17 Michael B. Cohen. Nearly tight oblivious subspace embeddings by trace inequalities. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 278–287, 2016.

18 Michael B. Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 163–172, 2015.

19 Michael B. Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 181–190, 2015.

20 Michael B. Cohen, Jelani Nelson, and David P. Woodruff. Optimal approximate matrix product in terms of stable rank, 2016. arXiv:1507.02268.

21 Zhili Feng, Praneeth Kacham, and David P. Woodruff. Strong coresets for subspace approximation and k-median in nearly linear time. *CoRR*, abs/1912.12003, 2019.

22 Piotr Indyk, Ali Vakilian, Tal Wagner, and David P. Woodruff. Sample-optimal low-rank approximation of distance matrices. In *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, pages 1723–1751, 2019.

23 Nikita Ivkin, Daniel Rothchild, Enayat Ullah, Vladimir Braverman, Ion Stoica, and Raman Arora. Communication-efficient distributed sgd with sketching. *arXiv preprint*, 2019. arXiv:1903.04488.

24 Xiangrui Meng and Michael W. Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 91–100, 2013.

25 Raphael A. Meyer, Cameron Musco, Christopher Musco, and David P. Woodruff. Hutch++: Optimal stochastic trace estimation. In *4th Symposium on Simplicity in Algorithms, SOSA 2021, Virtual Conference, January 11-12, 2021*, pages 142–155, 2021.

26 Cameron Musco and David P. Woodruff. Sublinear time low-rank approximation of positive semidefinite matrices. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 672–683, 2017.

27 Jelani Nelson and Huy L. Nguyen. OSNAP: faster numerical linear algebra algorithms via sparser subspace embeddings. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 117–126, 2013.

28 Ninh Pham and Rasmus Pagh. Fast and scalable polynomial kernels via explicit feature maps. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, pages 239–247, 2013.

29 Mert Pilanci and Martin J. Wainwright. Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares, 2014. `arXiv:1411.0347`.

30 Eric Price, Zhao Song, and David P. Woodruff. Fast regression with an $ell_infty$ guarantee. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, pages 59:1–59:14, 2017.

31 Ilya P. Razenshteyn, Zhao Song, and David P. Woodruff. Weighted low rank approximations with provable guarantees. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 250–263, 2016.

32 Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 143–152, 2006.

33 Xiaofei Shi and David P. Woodruff. Sublinear time numerical linear algebra for structured matrices. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 4918–4925, 2019.

34 Christian Sohler and David P. Woodruff. Subspace embeddings for the $l_1$-norm with applications. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 755–764, 2011.

35 Christian Sohler and David P. Woodruff. Strong coresets for k-median and subspace approximation: Goodbye dimension. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 802–813, 2018.

36 Zhao Song, David P. Woodruff, and Peilin Zhong. Towards a zero-one law for entrywise low rank approximation. *CoRR*, abs/1811.01442, 2018. `arXiv:1811.01442`.

37 Zhao Song, David P. Woodruff, and Peilin Zhong. Relative error tensor low rank approximation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2772–2789, 2019.

38 Ruosong Wang and David P. Woodruff. Tight bounds for p oblivious subspace embeddings. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1825–1843, 2019.

39 David P. Woodruff. Sketching as a tool for numerical linear algebra. *Found. Trends Theor. Comput. Sci.*, 10(1-2):1–157, 2014.