CS395T: Continuous Algorithms, Part VIII
Linear regression

Kevin Tian

1 Preconditioning

In this lecture, we develop highly-efficient algorithms for one of the most basic optimization problems, linear (i.e. least-squares) regression. Specifically, throughout we fix a full-rank matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ with $n \geq d$ (which can be thought of as containing the d-dimensional features of n data points), and a vector $b \in \mathbb{R}^n$ of response variables.[1] Our goal is to solve the problem

$$\min_{x \in \mathbb{R}^d} \|\mathbf{A}x - b\|_2^2 . \tag{1}$$

Note that the Hessian of the objective (1) is $2\mathbf{A}^\top \mathbf{A} \in \mathbb{S}^{d \times d}_{\succ \mathbf{0}}$, so it is a convex optimization problem. When $\mathbf{A}^\top \mathbf{A}$ is well-conditioned, gradient descent (e.g. Theorem 4, Part II) achieves a highly-accurate solution to (1) in few iterations, each of which requires computing the gradient $2\mathbf{A}^\top(\mathbf{A}x - b)$ and performing a vector update. In the regime where $\mathbf{A}^\top \mathbf{A}$ has constant condition number, the runtime cost of an ϵ-accurate solution to (1) (in a sense we make precise later) scales as

$$O\left(\text{nnz}(\mathbf{A}) \cdot \log \frac{1}{\epsilon}\right). \tag{2}$$

On the other hand, we can directly compute the minimizer $x^\star$ of (1), since the first-order optimality condition implies that $2\mathbf{A}^\top \mathbf{A}x = 2\mathbf{A}^\top b$, so $x^\star = (\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top b$. The cost of computing $x^\star$ is dominated by the cost of computing $\mathbf{A}^\top \mathbf{A}$, which requires $O(nd^{\omega-1})$ time (Remark 1). Moreover, practical matrix multiplication algorithms have $\omega \approx 3$, and the resulting runtime of $\approx nd^2$ can be significantly more expensive than (2), which is at most $\approx nd$ and can be even smaller.

Remark 1. *We implicitly used two observations about matrix multiplication in our earlier discussion. First, given an algorithm which can multiply two $d \times d$ matrices in time $O(d^\omega)$, it is straightforward to multiply a $d \times n$ matrix $\mathbf{A}$ by a $n \times d$ matrix $\mathbf{B}$ in time $O(nd^{\omega-1})$ time, by tiling each $n \times d$ matrix (i.e. $\mathbf{A}^\top$ and $\mathbf{B}$) with $\approx \frac{n}{d}$ square matrices, and applying our d^ω time algorithm to each block separately. Summing over blocks takes time $\frac{n}{d} \cdot d^2$ which does not dominate. Second, we can use matrix multiplication to invert a full-rank $d \times d$ matrix $\mathbf{A}$ in time $O(d^\omega)$. To see this, let $\mathcal{I}(d)$ be the time it takes to invert a full-rank $d \times d$ matrix. Using the Schur complement formula*

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix} \implies \mathbf{A}^{-1} = \begin{pmatrix} \mathbf{S}^{-1} & -\mathbf{S}^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ -\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{S}^{-1} & \mathbf{A}_{22}^{-1} + \mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{S}^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \end{pmatrix},$$

when $\mathbf{A}_{22}$ is invertible, where $\mathbf{S} := \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}$, we obtain the recursion

$$\mathcal{I}(d) = 2\mathcal{I}\left(\frac{d}{2}\right) + O\left(\left(\frac{d}{2}\right)^\omega\right),$$

since the only matrices we need to invert are $\mathbf{A}_{22}$ and $\mathbf{S}$, and the remaining operations are $\frac{d}{2} \times \frac{d}{2}$ matrix multiplications or additions. This recursion yields $\mathcal{I}(d) = O(d^\omega)$ for $\omega > 2$, as claimed. More generally, a similar argument follows by applying pseudoinverses (see Definition 2) appropriately in place of inverses in the above formula, to account for non-full-rank submatrices.

[1] The assumption that $\mathbf{A}$ is full-rank, i.e. $\mathbf{A}^\top \mathbf{A} \in \mathbb{S}^{d \times d}_{\succ \mathbf{0}}$, is made only for expositional simplicity and all of the methods we discuss generalize to the case where $\mathbf{A}$ has a kernel as well.

Our focus is showing how, even when $\mathbf{A}$ may be arbitrarily poorly-conditioned, we can nonetheless achieve the goal runtime (2) up to an additive runtime term depending only on d. In moderately tall or dense instances, where $\mathrm{nnz}(\mathbf{A}) = \Omega(\mathrm{poly}(d))$, the dominant term is then still just the cost of matrix-vector products through $\mathbf{A}$, matching the well-conditioned setting.

Our approach is motivated by the following observation on *preconditioned gradient descent*.

Lemma 1. *Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ be full-rank with $n \geq d$, and let $\mathbf{P} \in \mathbb{S}^{d \times d}_{\succ \mathbf{0}}$ satisfy $\mathbf{K} \preceq \mathbf{P} \preceq \kappa \mathbf{K}$, where $\mathbf{K} := \mathbf{A}^\top \mathbf{A}$. Let $x \in \mathbb{R}^d$, and let $x^\star \in \mathbb{R}^d$ minimize (1). Then if $x' \leftarrow x - \mathbf{P}^{-1}(\mathbf{K}x - \mathbf{A}^\top b)$,*

$$\|x' - x^\star\|_{\mathbf{P}} \leq \left(1 - \frac{1}{\kappa}\right) \|x - x^\star\|_{\mathbf{P}}.$$

Proof. Recall that $x^\star = \mathbf{K}^{-1}\mathbf{A}^\top b$, so that

$$x^\star = x^\star - \mathbf{P}^{-1}(\mathbf{K}x^\star - \mathbf{A}^\top b).$$

In other words, $x = x^\star$ is the fixed point of the given update from x to x'. We continue:

$$\begin{aligned} x' - x^\star &= \left(x - \mathbf{P}^{-1}\left(\mathbf{K}x - \mathbf{A}^\top b\right)\right) - \left(x^\star - \mathbf{P}^{-1}(\mathbf{K}x^\star - \mathbf{A}^\top b)\right) \\ &= \left(\mathbf{I}_d - \mathbf{P}^{-1}\mathbf{K}\right)(x - x^\star). \end{aligned}$$

We have hence reduced our goal to showing $(\mathbf{I}_d - \mathbf{P}^{-1}\mathbf{K})^\top \mathbf{P}(\mathbf{I}_d - \mathbf{P}^{-1}\mathbf{K}) \preceq (1 - \frac{1}{\kappa})^2 \mathbf{P}$, since this would imply the claimed distance decrease. To see this, we expand:

$$\begin{aligned} \left(\mathbf{I}_d - \mathbf{K}\mathbf{P}^{-1}\right)\mathbf{P}\left(\mathbf{I}_d - \mathbf{P}^{-1}\mathbf{K}\right) &= \mathbf{P} - 2\mathbf{K} + \mathbf{K}\mathbf{P}^{-1}\mathbf{K} \\ &= \mathbf{P}^{\frac{1}{2}}\left(\mathbf{I}_d - \mathbf{P}^{-\frac{1}{2}}\mathbf{K}\mathbf{P}^{-\frac{1}{2}}\right)^2 \mathbf{P}^{\frac{1}{2}} \preceq \left(1 - \frac{1}{\kappa}\right)^2 \mathbf{P}. \end{aligned}$$

In the last inequality, we used that the assumption implies $\frac{1}{\kappa}\mathbf{I}_d \preceq \mathbf{P}^{-\frac{1}{2}}\mathbf{K}\mathbf{P}^{-\frac{1}{2}} \preceq \mathbf{I}_d$. □

Lemma 1 suggests a framework for an algorithm: even if $\mathbf{A}^\top \mathbf{A}$ is poorly-conditioned, if we can find a preconditioner matrix $\mathbf{P} \in \mathbb{S}^{d \times d}_{\succ \mathbf{0}}$ such that $\mathbf{P} \approx \mathbf{A}^\top \mathbf{A}$ spectrally, we can still solve (1) at a well-conditioned rate, provided we have computed $\mathbf{P}^{-1}$. Note that given $\mathbf{P}$, we can invert it in time $O(d^\omega)$ by Remark 1. In Sections 2 and 3, we will see different strategies for computing such a preconditioner $\mathbf{P}$, which vary in runtime and the types of structure afforded to $\mathbf{P}$.

In light of the progress guarantee given by Lemma 1, for a desired parameter $\epsilon \in (0, 1)$, in the rest of the lecture we focus on obtaining a vector $\hat{x} \in \mathbb{R}^d$ such that

$$\|\hat{x} - x^\star\|_{\mathbf{A}^\top \mathbf{A}} \leq \epsilon \|x^\star\|_{\mathbf{A}^\top \mathbf{A}}, \text{ for } x^\star := \left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top b. \tag{3}$$

We call such a $\hat{x}$ an *ϵ-approximate solution to* (1). Because the quadratic norm induced by a good preconditioner $\mathbf{P}$ is closely-approximated by the quadratic norm in $\mathbf{A}^\top \mathbf{A}$, we can transfer between the progress given by Lemma 1 and the distance in the $\mathbf{A}^\top \mathbf{A}$ norm up to a multiplicative $\sqrt{\kappa}$. In this case, (3) is the type of bound one would obtain after iterating the updates

$$x_0 \leftarrow \mathbb{0}_d, \; x_{t+1} \leftarrow x_t - \mathbf{P}^{-1}\left(\mathbf{A}^\top \mathbf{A} x_t - \mathbf{A}^\top b\right) \text{ for all } t \geq 0,$$

for $T = O(\kappa \log \frac{\kappa}{\epsilon})$ iterations, and then transferring the distance guarantee from $\|\cdot\|_{\mathbf{P}}$ to $\|\cdot\|_{\mathbf{A}^\top \mathbf{A}}$.

2 Oblivious subspace embeddings

In this section, we consider a strategy pioneered by the influential work [Sar06] and subsequently developed further by [CW13], the first to achieve a runtime of $\approx (\mathrm{nnz}(\mathbf{A}) + \mathrm{poly}(d)) \log \frac{1}{\epsilon}$ for achieving an ϵ-approximate solution to (1) in the sense of (3). The idea of [CW13] in particular was to efficiently construct a preconditioner via oblivious subspace embeddings.

Definition 1. *Let $m, n, d \in \mathbb{N}$ with $n \geq d$. We say that a random matrix $\mathbf{\Pi} \in \mathbb{R}^{m \times n}$ is a (d, ϵ, δ)-*oblivious subspace embedding *(OSE) if for every fixed $\mathbf{U} \in \mathbb{R}^{n \times d}$ with orthonormal columns,*

$$\left\|\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi} \mathbf{U} - \mathbf{I}_d\right\|_{\mathrm{op}} \leq \epsilon, \tag{4}$$

with probability $\geq 1 - \delta$ over the randomness of $\mathbf{\Pi}$, independent of $\mathbf{U}$.

Recall that the assumption that $\mathbf{U}$ has orthonormal columns means that $\mathbf{U}^\top\mathbf{U} = \mathbf{I}_d$. We can equivalently view $\mathbf{U}$ as giving a basis for the d-dimensional subspace $\text{Span}(\mathbf{U}) \subset \mathbb{R}^n$; as discussed in Section 2.2, Part V, $\mathbf{U}\mathbf{U}^\top \in \mathbb{S}^{n\times n}_{\succeq \mathbf{0}}$ is then the orthogonal projection matrix onto $\text{Span}(\mathbf{U})$.

To explain this naming choice, (4) is *oblivious* in the sense that for any holdout matrix $\mathbf{U}$ independent from the randomness of $\mathbf{\Pi}$, $\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U} \approx \mathbf{I}_d$ should hold with high probability. Additionally, (4) is an extremely strong condition: it means that for any $u = \mathbf{U}v \in \text{Span}(\mathbf{U})$,

$$\begin{aligned}(1-\epsilon)\left\|v\right\|_2^2 \le v^\top\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U}v \le (1+\epsilon)\left\|v\right\|_2^2 \\ \implies (1-\epsilon)\left\|u\right\|_2^2 \le \left\|\mathbf{\Pi}u\right\|_2^2 \le (1+\epsilon)\left\|u\right\|_2^2,\end{aligned}$$

simultaneously for all possible $v \in \mathbb{R}^d$, since $\left\|u\right\|_2 = \left\|\mathbf{U}v\right\|_2 = \left\|v\right\|_2$. In other words, $\mathbf{\Pi}$ approximately preserves the norms of every $u \in \text{Span}(\mathbf{U})$, so in this sense it is a *subspace embedding*.

For the condition in (4) to hold, it is clear that $\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U}$ must at least be full-rank, so $m \ge d$ necessarily. We mention one additional useful consequence of $\mathbf{\Pi}$ being an OSE.

Lemma 2. *Let $\mathbf{\Pi} \in \mathbb{R}^{m\times n}$ be a (d,ϵ,δ)-OSE. For all full-rank $\mathbf{A} \in \mathbb{R}^{n\times d}$ with $n \ge d$, with probability $\ge 1-\delta$, we have $(1-\epsilon)\mathbf{A}^\top\mathbf{A} \preceq \mathbf{A}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{A} \preceq (1+\epsilon)\mathbf{A}^\top\mathbf{A}$.*

Proof. Let the singular value decomposition of $\mathbf{A}$ be $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, for $\mathbf{U} \in \mathbb{R}^{n\times d}$, $\mathbf{V} \in \mathbb{R}^{d\times d}$ with orthonormal columns, and diagonal $\mathbf{\Sigma} \in \mathbb{S}^{d\times d}_{\succ \mathbf{0}}$. Assuming (4) holds for $\mathbf{U}$, the claim follows:

$$\begin{aligned}&(1-\epsilon)\mathbf{U}^\top\mathbf{U} \preceq \mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U} \preceq (1+\epsilon)\mathbf{U}^\top\mathbf{U} \\ \implies &(1-\epsilon)\mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \preceq \mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \preceq (1+\epsilon)\mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \\ \implies &(1-\epsilon)\mathbf{A}^\top\mathbf{A} \preceq \mathbf{A}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{A} \preceq (1+\epsilon)\mathbf{A}^\top\mathbf{A}.\end{aligned}$$

□

In other words, if we can find a OSE $\mathbf{\Pi}$ such that $\mathbf{\Pi}\mathbf{A}$ is easy to compute, Lemma 2 says that we can use $\mathbf{P} = \frac{1}{1-\epsilon}\mathbf{A}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{A}$ as our preconditioner in Lemma 1. Ideally, $\mathbf{\Pi}\mathbf{A}$ has few rows (say $m = \text{poly}(d)$), such that once we have $\mathbf{\Pi}\mathbf{A}$ we can compute $\mathbf{P}$ and invert it in $\text{poly}(d)$ time.

However, it is not even a priori clear that short OSEs $\mathbf{\Pi}$ exist, let alone that they are easy to apply to matrices. We begin by discussing OSE existence in Section 2.1. We then give an efficient construction of a *sparse* OSE in Section 2.2, and demonstrate its application to solving (1).

2.1 Dense OSEs

As a proof-of-concept, in this section we show $m = O((d+\log\frac{1}{\delta})\cdot\frac{1}{\epsilon^2})$ suffices for a (d,ϵ,δ)-OSE $\mathbf{\Pi}$ to exist. That is, for say $\epsilon = \frac{1}{2}$ in Lemma 2 (which leads to $\kappa = 3$ in Lemma 1), our existence proof will show that $\approx d$ rows in $\mathbf{\Pi}$ suffice for (4) to hold with high probability. Note that such a short $\mathbf{\Pi}$ need not be easy to apply, an issue we discuss in greater detail in the following Section 2.2.

Before giving our construction of an OSE, we require a few helper claims.

Lemma 3. *Let $S := \{x \in \mathbb{R}^d \mid \left\|x\right\|_2 = 1\}$ be the surface of the unit ball in $\mathbb{R}^d$, and let $\epsilon \in (0,1)$. There exists $N \subset S$ with $|N| \le (1+\frac{2}{\epsilon})^d$, such that*

$$\max_{x\in S}\min_{v\in N}\left\|v-x\right\|_2 \le \epsilon. \tag{5}$$

Proof. Consider a greedy iterative process, where $N \leftarrow \emptyset$ initially, and anytime there remains a point $x \in S$ where $\min_{v\in N}\left\|v-x\right\|_2 > \epsilon$, we add x to S. Suppose we have run this process for T iterations, producing $N = \{x_t\}_{t\in[T]}$. We claim $T \le (1+\frac{2}{\epsilon})^d$. To see this, by construction

$$\mathbb{B}\left(x_t, \frac{\epsilon}{2}\right) \cap \mathbb{B}\left(x_s, \frac{\epsilon}{2}\right) = \emptyset \text{ for all } s \ne t,$$

i.e. all of the radius-$\frac{\epsilon}{2}$ balls centered around points in N are disjoint. To see this, letting $s < t$, we would not have added x_t to N if it was at distance $\le \epsilon$ from x_s. Finally, we have T disjoint balls of volume $(\frac{\epsilon}{2})^d \cdot \text{Vol}(\mathbb{B}(1))$, all of which are contained in $\mathbb{B}(1+\frac{\epsilon}{2})$, so the claim follows:

$$T \le \frac{\text{Vol}\left(\mathbb{B}(1+\frac{\epsilon}{2})\right)}{\text{Vol}\left(\mathbb{B}\left(\frac{\epsilon}{2}\right)\right)} \le \left(1+\frac{2}{\epsilon}\right)^d.$$

□

Lemma 3 shows how to approximate S with a finite set N up to distance ϵ, so N is also sometimes called an ϵ-net. We next show that quadratic forms on a net approximate the operator norm.

Lemma 4. *In the notation of Lemma 3, let $N \subset S$ satisfy (5) for $\epsilon \leq \frac{1}{3}$. For all $\mathbf{M} \in \mathbb{S}^{d\times d}$,*

$$\|\mathbf{M}\|_{\text{op}} \leq \frac{1}{1-2\epsilon-\epsilon^2} \max_{v\in N} \left|v^\top \mathbf{M} v\right|.$$

Proof. Recall that $\|\mathbf{M}\|_{\text{op}} = \sup_{x,y\in S} x^\top \mathbf{M} y$, and also that $\|\mathbf{M}\|_{\text{op}} = |u^\top \mathbf{M} u|$ for some $u \in S$. Therefore, letting $u \in S$ achieve this latter equality, and letting $v \in N$ have $\|u - v\|_2 \leq \epsilon$,

$$\begin{aligned}
\|\mathbf{M}\|_{\text{op}} &= \left|u^\top \mathbf{M} u\right| \\
&\leq \left|v^\top \mathbf{M} v\right| + \left|(u-v)^\top \mathbf{M} v\right| + \left|v^\top \mathbf{M}(u-v)\right| + \left|(u-v)^\top \mathbf{M}(u-v)\right| \\
&\leq \left|v^\top \mathbf{M} v\right| + \left(2\epsilon + \epsilon^2\right) \|\mathbf{M}\|_{\text{op}}.
\end{aligned}$$

Rearranging and dividing by $1 - 2\epsilon - \epsilon^2$ yields the claim. □

We are now ready to give our first OSE existence proof.

Proposition 1. *Let $\mathbf{\Pi} \in \mathbb{R}^{m\times n}$ have i.i.d. entries $\sim \mathcal{N}(0, \frac{1}{m})$. Then, $\mathbf{\Pi}$ is a (d, ϵ, δ)-OSE for*

$$m = O\left(\frac{d + \log(\frac{1}{\delta})}{\epsilon^2}\right).$$

Proof. Throughout this proof, fix $\mathbf{U} \in \mathbb{R}^{n\times d}$ with orthonormal columns, and following the notation of Lemma 3, let N satisfy (5) with $\epsilon = \frac{1}{4}$, such that $|N| \leq 9^d$. Our goal is to show that with probability $\geq 1 - \delta$ over the randomness of $\mathbf{\Pi}$, we have for all $v \in N$ that

$$\left|v^\top \left(\mathbf{U}^\top \mathbf{\Pi}^\top \mathbf{\Pi} \mathbf{U} - \mathbf{I}_d\right) v\right| \leq \frac{\epsilon}{3},$$

since this would imply (4) by Lemma 4. By applying a union bound over all of the $\leq 9^d$ vectors $\mathbf{U}v \in \mathbb{R}^n$, for all $v \in N$, the Johnson-Lindenstrauss lemma (Corollary 1, Part V) with failure probability $\frac{\delta}{9^d}$ shows that if we take $m = O(\frac{d+\log(\frac{1}{\delta})}{\epsilon^2})$ for an appropriate constant, the above display indeed holds with probability $\geq 1 - \delta$, concluding the proof. □

While Proposition 1 is promising, it does not immediately let us use the framework suggested by Lemmas 1 and 2. This is because Lemma 2 requires us to compute $\mathbf{\Pi A}$, which for dense $\mathbf{\Pi}$ may be even more expensive than computing $\mathbf{A}^\top \mathbf{A}$, the original problem we were trying to avoid because it requires $\approx nd^{\omega-1}$ time. Moreover, this construction $\mathbf{\Pi A}$ seems to lose all structural information about $\mathbf{A}$, e.g. even if $\mathbf{A}$ is sparse we cannot say anything about the sparsity of $\mathbf{\Pi A}$.

2.2 Sparse OSEs

Motivated by the computational difficulties encountered at the end of last section, we next consider a strategy for choosing $\mathbf{\Pi}$ such that $\mathbf{\Pi A}$ is efficiently-computable, and retains structure present in the rows of $\mathbf{A}$. We analyze a matrix $\mathbf{\Pi}$ originally proposed by [TZ12], inspired by applications in streaming algorithms, which was studied in the context of OSEs by [NN13a]. Specifically, we let $\mathbf{\Pi} \in \{0, 1\}^{m\times n}$ have exactly one uniformly random nonzero entry per column. Letting the nonzero entry of the j^{th} column be denoted $h(j) \in [m]$,[2] for all $j \in [n]$, it is straightforward to check that

$$[\mathbf{\Pi A}]_{i:} = \sum_{\substack{j\in[n] \\ h(j)=i}} \mathbf{A}_{i:}, \text{ for all } i \in [m].$$

In other words, each row in $\mathbf{A}$ is added to a uniformly random row in $\mathbf{\Pi A}$ exactly one time. To see why this is desirable in applications, note that $\mathbf{\Pi A}$ can be computed in time $O(\text{nnz}(\mathbf{A}))$ using

[2] We choose this notation because the sketching and streaming community typically chooses h to be a hash function, where they analyze the limited independence of h required for space considerations.

one pass over $\mathbf{A}$, whereas if $\mathbf{\Pi}$ was an arbitrary $m \times n$ matrix (say with $m = d$), computing $\mathbf{\Pi A}$ would require time $\approx nd^{\omega-1}$ which can be potentially $\gg \text{nnz}(\mathbf{A})$. Additionally, observe that it is simple to apply $\mathbf{\Pi A}$ to a vector $v \in \mathbb{R}^d$ in $O(\text{nnz}(\mathbf{A}) + m)$ time, since it suffices to first compute $\mathbf{A}v$ and then sum relevant coordinates to each output entry. We next show that choosing $m \approx d^2$ in the [TZ12] matrix gives an OSE, with constant failure probability and accuracy.

Proposition 2. *Let $\{\tau_{ij}\}_{i\in[m],j\in[n]}$ be $\{0,1\}$-valued random variables such that for all $j \in [n]$, exactly one $\tau_{ij} = 1$ uniformly at random, and τ_{ij} is independent of $\tau_{i'j'}$ for all $j \neq j'$. Further, let $\{\sigma_{ij}\}_{i\in[m],j\in[n]} \sim_{\text{unif.}} \{-1,1\}$ i.i.d. Then for $\delta, \epsilon \in (0,1)$, if*

$$m \geq \frac{2d^2}{\delta\epsilon^2},$$

the random matrix $\mathbf{\Pi} \in \{-1,0,1\}^{m\times n}$ with $\mathbf{\Pi}_{ij} = \tau_{ij}\sigma_{ij}$ for all $i \in [n]$, $j \in [m]$ is a (d,ϵ,δ)-OSE.

Proof. Fix $\mathbf{U} \in \mathbb{R}^{n\times d}$ with orthonormal columns throughout the proof. We observe that

$$\mathbb{E}\left[\mathbf{\Pi}^\top\mathbf{\Pi}\right]_{ij} = \mathbb{E}\left\langle \mathbf{\Pi}_{:i}, \mathbf{\Pi}_{:j}\right\rangle = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases},$$

so $\mathbb{E}\mathbf{\Pi}^\top\mathbf{\Pi} = \mathbf{I}_n$ and therefore $\mathbb{E}\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U} = \mathbf{I}_d$. Further, for all $i, j \in [d]$, we have

$$\begin{aligned}\left[\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U}\right]_{ij} &= \left\langle [\mathbf{\Pi}\mathbf{U}]_{:i}, [\mathbf{\Pi}\mathbf{U}]_{:j}\right\rangle = \sum_{r\in[m]} [\mathbf{\Pi}\mathbf{U}]_{ri}\,[\mathbf{\Pi}\mathbf{U}]_{rj} \\ &= \sum_{r\in[m]}\left(\sum_{s\in[n]}\tau_{rs}\sigma_{rs}u_s^i\right)\left(\sum_{s\in[n]}\tau_{rs}\sigma_{rs}u_s^j\right) \\ &= \left\langle u^i, u^j\right\rangle + \sum_{r\in[m]}\sum_{\substack{s,t\in[n]\\ s\neq t}}\tau_{rs}\tau_{rt}\sigma_{rs}\sigma_{rt}u_s^iu_t^i,\end{aligned}$$

where we let $\{u^k\}_{k\in[d]}$ be the columns of $\mathbf{U}$. By orthonormality of $\mathbf{U}$, we thus have

$$\left[\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U} - \mathbf{I}_d\right]_{ij} = \sum_{r\in[m]}\sum_{\substack{s,t\in[n]\\ s\neq t}}\tau_{rs}\tau_{rt}\sigma_{rs}\sigma_{rt}u_s^iu_t^i. \tag{6}$$

Our strategy is to control $\|\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U} - \mathbf{I}_d\|_{\text{op}}$ using $\mathbb{E}\|\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U} - \mathbf{I}_d\|_{\text{F}}^2$. To this end, we bound the expected square of (6). First, for $i = j$, when expanding the square of (6), every entry vanishes in expectation except those which select the same (r,s,t) twice, by column independence. Hence,

$$\mathbb{E}\left[\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U} - \mathbf{I}_d\right]_{ii}^2 = \frac{2}{m}\sum_{\substack{s,t\in[n]\\ s\neq t}}\left(u_s^iu_t^i\right)^2 \leq \frac{2}{m}\left\|u^i\right\|_2^4. \tag{7}$$

The factor 2 arises since (s,t) can be matched with either (s,t) or (t,s), and we also used $\sum_{r\in[m]}\mathbb{E}\tau_{rs}^2\tau_{rt}^2 = m\cdot\frac{1}{m^2} = \frac{1}{m}$. Next, we consider the case of $i \neq j$:

$$\begin{aligned}\mathbb{E}\left[\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U} - \mathbf{I}_d\right]_{ij}^2 &= \frac{1}{m^2}\sum_{r\in[m]}\sum_{\substack{s,t\in[n]\\ s\neq t}}\left(\left(u_s^iu_t^j\right)^2 + \left(u_s^iu_t^iu_s^ju_t^j\right)\right) \\ &= \frac{1}{m}\sum_{\substack{s,t\in[n]\\ s\neq t}}\left(\left(u_s^iu_t^j\right)^2 + \left(u_s^iu_t^iu_s^ju_t^j\right)\right).\end{aligned} \tag{8}$$

Additionally, observe that

$$\sum_{\substack{s,t\in[n]\\ s\neq t}}u_s^iu_t^iu_s^ju_t^j = \left\langle u^i, u^j\right\rangle^2 - \sum_{s\in[n]}(u_s^iu_s^j)^2 = -\sum_{s\in[n]}(u_s^iu_s^j)^2 \leq 0.$$

Plugging this into (8), we hence have

$$\mathbb{E}\left[\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U}-\mathbf{I}_d\right]_{ij}^2 \le \frac{1}{m}\left\|u^i\right\|_2^2\left\|u^j\right\|_2^2.$$

Combining with (7), we finally conclude

$$\mathbb{E}\left\|\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U}-\mathbf{I}_d\right\|_{\text{op}}^2 \le \mathbb{E}\left\|\mathbf{U}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{U}-\mathbf{I}_d\right\|_{\text{F}}^2 \le \frac{2}{m}\left(\sum_{i\in[d]}\left\|u^i\right\|_2^2\right)^2 \le \frac{2d^2}{m},$$

since $\|\mathbf{U}\|_{\text{F}}^2 = d$ by column orthonormality. The conclusion follows from Markov's inequality. □

We conclude with the following consequence of Lemma 1 and Proposition 2.

Theorem 1 (Linear regression via OSEs)**.** *Let $\epsilon, \delta \in (0,1)$, $b \in \mathbb{R}^n$, and let $\mathbf{A} \in \mathbb{R}^{n\times d}$ be full-rank. There is an algorithm which computes $\hat{x} \in \mathbb{R}^d$ satisfying (3) with probability $\ge 1-\delta$ in time*

$$O\left(\left(\text{nnz}(\mathbf{A})+d^2\right)\log\left(\frac{1}{\epsilon}\right)\log\left(\frac{1}{\delta}\right)+d^{\omega+1}\log\left(\frac{1}{\delta}\right)\right).$$

Proof. First, let $\mathbf{\Pi}$ be the result of Proposition 2 with $\epsilon \leftarrow \frac{1}{2}$ and $\delta \leftarrow \frac{1}{2}$, i.e. with $m = O(d^2)$. Proposition 2 and Lemma 2 show that with probability $\ge \frac{1}{2}$, we have $\frac{1}{2}\mathbf{A}^\top\mathbf{A} \preceq \mathbf{A}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{A} \preceq \frac{3}{2}\mathbf{A}^\top\mathbf{A}$. Moreover, we can compute $\mathbf{\Pi}\mathbf{A}$ in time $O(\text{nnz}(\mathbf{A}))$, and using the tiling strategy in Remark 1, we can then compute $\mathbf{P} := 2\mathbf{A}^\top\mathbf{\Pi}^\top\mathbf{\Pi}\mathbf{A}$ and invert it in time $O(d^{\omega+1})$. Finally, running Lemma 1 with $\kappa \leftarrow 3$ for $O(\log\frac{1}{\epsilon})$ iterations gives the result, assuming our OSE succeeded. Each iteration of Lemma 1 takes time $O(\text{nnz}(\mathbf{A})+d^2)$ to run, since we have already precomputed $\mathbf{P}^{-1}$.

Our final algorithm runs the subroutine described above $O(\log\frac{1}{\delta})$ times, so that one of the runs constructs $\mathbf{P}$ satisfying Lemma 1's condition with $\kappa = 3$ with probability $\ge 1-\delta$. Finally, note

$$\|\mathbf{A}\hat{x}-b\|_2^2 - \|\mathbf{A}x^\star-b\|_2^2 = \|\mathbf{A}(\hat{x}-x^\star)\|_2^2 = \|\hat{x}-x^\star\|_{\mathbf{A}^\top\mathbf{A}}^2.$$

Hence, it suffices to evaluate $\|\mathbf{A}\hat{x}-b\|_2^2$ for each computed $\hat{x}$, and take the best such point. □

Remark 2. *Theorem 1 establishes a strong baseline for linear regression, as it shows that (omitting logarithmic factors) (3) can be obtained in time $\approx \text{nnz}(\mathbf{A})+d^{\omega+1}$. A natural question is whether it is possible to improve upon the row count in Proposition 2, i.e. from $\approx d^2$ to $\approx d$. In [NN13b], it was shown that for 1-sparse columns, $\Omega(d^2)$ rows are necessary. To overcome this, [CW13, NN13a] analyzed strategies where s random entries for each column were set to $\pm s^{-1/2}$, generalizing the $s=1$ strategy of Proposition 2, calling the resulting OSEs oblivious sparse norm-approximating projections (OSNAPs). This culminated in an analysis by [Coh16] who showed that the tradeoff*

$$s = O\left(\frac{\log_b(\frac{d}{\delta})}{\epsilon}\right),\ m = O\left(\frac{bd\log(\frac{d}{\delta})}{\epsilon^2}\right),$$

is achievable for any $b \ge 2$, e.g. taking $b = \log^c(\frac{d}{\delta})$ for a small constant c yields $s = O(1)$ and $m = O(d\log^{1+c}(\frac{d}{\delta}))$ for $\epsilon = \Theta(1)$. The resulting linear regression algorithm, following our earlier framework, improves Theorem 1's runtime to $\approx \text{nnz}(\mathbf{A})+d^\omega$. As seen in Section 3, a similar runtime follows using a different strategy. Notably, this line of research was essentially closed up to constant factors, assuming $\omega > 2$, by [CSWZ23], who gave a distribution over $\mathbf{\Pi}$ such that $\mathbf{\Pi}$ is a $(d, \Theta(1), \Theta(1))$-OSE, and $\mathbf{\Pi}$ has $O(d)$ rows and can be applied in time $O(\text{nnz}(\mathbf{A})+d^\omega)$.

3 Spectral sparsification

In this section, we adopt a different perspective on the preconditioning problem, where we ask that our preconditioner $\mathbf{P}$ (for use in Lemma 1) has the structured form

$$\mathbf{P} = \mathbf{A}^\top\mathbf{W}\mathbf{A} = \sum_{i\in[n]} w_i a_i a_i^\top, \tag{9}$$

for a diagonal matrix $\mathbf{W} = \mathbf{diag}(w) \in \mathbb{S}^{n\times n}_{\succeq \mathbf{0}}$, and where we let the rows of $\mathbf{A} \in \mathbb{R}^{n\times d}$ be denoted $\{a_i\}_{i\in[n]}$. In other words, we want $\mathbf{P}$ to be the Gram matrix of $\mathbf{W}^{\frac{1}{2}}\mathbf{A}$, which simply reweights the rows of $\mathbf{P}$. We are particularly interested in the case where $\mathrm{nnz}(w) \approx d$, i.e. almost all of the weights are zero, and therefore we have a compact representation of $\mathbf{W}^{\frac{1}{2}}\mathbf{A}$.

Remark 3. *One significant potential advantage of the preconditioner choice (9) is that it approximates $\mathbf{A}^\top\mathbf{A}$ with $\mathbf{B}^\top\mathbf{B}$, for a matrix $\mathbf{B} = \mathbf{W}^{\frac{1}{2}}\mathbf{A}$ which preserves the structures of rows of $\mathbf{A}$, e.g. sparsity patterns and more. This is desirable because a variety of linear system solvers can take advantage of sparsity, which can imply faster Gaussian elimination by careful choice of "pivot orders," i.e. the order in which rows are eliminated. For example, linear system solvers faster the naive d^ω runtime are known for tridiagonal matrices, banded matrices, circulant matrices, Hessenberg matrices, and more. Further, a successful line of research initiated by the breakthrough work of [ST14] has shown how to solve linear systems in matrices with combinatorial structure in nearly-linear time, including undirected graph Laplacians [ST14], connection Laplacians [KLP+16], directed graph Laplacians [CKK+18], M-matrices [AJSS19], and more [CFM+14, KPSZ18, BMNW22, JLM+23]. These results further motivate preconditioner constructions capable of preserving structure.*

3.1 Leverage score sampling

As a starting point, we analyze the following simple strategy in this section, known as *leverage score sampling*. Noting that $\mathbf{A}^\top\mathbf{A} = \sum_{i\in[n]} a_i a_i^\top$, consider the reweighting strategy where we average K independent draws to an unbiased rank-one estimate for $\mathbf{A}^\top\mathbf{A}$. That is, for some sampling probabilities $p = \{p_i\}_{i\in[n]} \in \Delta^n$, and some $K \in \mathbb{N}$, we let

$$\mathbf{M}_k = \frac{1}{p_{i_k}} a_{i_k} a_{i_k}^\top, \text{ for } i_k \sim p,$$

independently for $k \in [K]$, and we set $\mathbf{P} = \frac{1}{K}\sum_{k\in[K]} \mathbf{M}_k$. Clearly, this strategy produces a preconditioner $\mathbf{P}$ of the form in (9), since the output is a reweighted average of the row outer products of $\mathbf{A}$. However, we have substantial freedom in designing our sampling probabilities p.

Leverage score sampling proposes to use the following definition of row importances in choosing p.

Definition 2 (Leverage scores)**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ for $n \geq d$ have rows $\{a_i\}_{i\in[n]} \subset \mathbb{R}^d$. We let*

$$\tau_i(\mathbf{A}) := a_i^\top (\mathbf{A}^\top\mathbf{A})^\dagger a_i.$$

be the leverage score *of the i^{th} row of $\mathbf{A}$.*[3]

To gain some intuition for leverage scores, we summarize some of their properties in the following.

Lemma 5. *In the notation of Definition 2, the following properties hold.*

1. $\sum_{i\in[n]} \tau_i(\mathbf{A}) = \mathrm{rank}(\mathbf{A})$.
2. $\tau_i(\mathbf{A}) \in [0, 1]$ *for all* $i \in [n]$.

Proof. Item 1 follows since

$$\sum_{i\in[n]} \tau_i(\mathbf{A}) = \sum_{i\in[n]} \left\langle a_i a_i^\top, (\mathbf{A}^\top\mathbf{A})^\dagger \right\rangle = \left\langle \mathbf{A}^\top\mathbf{A}, (\mathbf{A}^\top\mathbf{A})^\dagger \right\rangle = \dim(\mathrm{Span}(\mathbf{A})).$$

Item 2 uses $\mathbf{A}^\top\mathbf{A} \succeq a_i a_i^\top \implies \mathbf{0}_d \preceq (\mathbf{A}^\top\mathbf{A})^\dagger \preceq (a_i a_i^\top)^\dagger$, and $\left\langle a_i a_i^\top, (a_i a_i^\top)^\dagger \right\rangle = 1$. □

Lemma 5 motivates using the leverage scores $\{\tau_i(\mathbf{A})\}_{i\in[n]}$ as measures of the relative importance of rows of $\mathbf{A}$, in composing the spectrum of $\mathbf{A}^\top\mathbf{A}$, bounded in $[0, 1]$. For example, a row contributes the maximum score of 1 if it is orthogonal to all other rows in $\mathbf{A}$. One convenient way to picture $\tau_i(\mathbf{A})$ geometrically is that it first normalizes the matrix $\mathbf{A}^\top\mathbf{A}$ to be in isotropic position (in

[3] Here, $\mathbf{M}^\dagger$ denotes the Moore-Penrose pseudoinverse of $\mathbf{M} \in \mathbb{S}^{d\times d}_{\succeq \mathbf{0}}$, which is the unique matrix in $\mathbb{S}^{d\times d}_{\succeq \mathbf{0}}$ such that $\mathbf{M}\mathbf{M}^\dagger = \mathbf{M}^\dagger\mathbf{M}$ is the identity matrix on the range of $\mathbf{M}$. In other words, if $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ is the eigendecomposition of $\mathbf{M}$, we let $\mathbf{M}^\dagger = \mathbf{U}\mathbf{\Lambda}^\dagger\mathbf{U}^\top$ where $\mathbf{\Lambda}^\dagger_{ii} = \mathbf{\Lambda}^{-1}_{ii}$ if $\mathbf{\Lambda}_{ii} > 0$, and otherwise $\mathbf{\Lambda}^\dagger_{ii} = 0$.

Span($\mathbf{A}$)) by pre- and post-multiplying it by $(\mathbf{A}^\top\mathbf{A})^{\frac{\dagger}{2}}$, and then measures the row length of a_i. That is, letting v_i be the i^{th} row of $\mathbf{A}(\mathbf{A}^\top\mathbf{A})^{\frac{\dagger}{2}}$ for all $i \in [n]$, we have

$$\|v_i\|_2^2 = \left\|(\mathbf{A}^\top\mathbf{A})^{\frac{\dagger}{2}}a_i\right\|_2^2 = \tau_i(\mathbf{A}),\ \sum_{i\in[n]} v_i v_i^\top = (\mathbf{A}^\top\mathbf{A})^{\frac{\dagger}{2}}\mathbf{A}^\top\mathbf{A}(\mathbf{A}^\top\mathbf{A})^{\frac{\dagger}{2}}, \tag{10}$$

which is the projection matrix onto Span($\mathbf{A}^\top$). We now formally analyze leverage score sampling.

Proposition 3. *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ have rows $\{a_i\}_{i\in[n]} \subset \mathbb{R}^d$, and let $\tau \in \mathbb{R}^n_{\geq 0}$ have $\tau_i = \tau_i(\mathbf{A})$ for all $i \in [n]$. Define $p = \frac{\tau}{\|\tau\|_1} \in \Delta^n$, and for $K \in \mathbb{N}$, let*

$$\mathbf{P} := \sum_{k\in[K]} \frac{1}{Kp_{i_k}} a_{i_k} a_{i_k}^\top,\ \textit{for } i_k \sim p, \tag{11}$$

i.i.d. for all $k \in [K]$. Then for $\epsilon, \delta \in (0,1)$, if $K \geq \frac{3\cdot\text{rank}(\mathbf{A})}{\epsilon^2}\log(\frac{2d}{\delta})$, with probability $\geq 1-\delta$,

$$(1-\epsilon)\mathbf{A}^\top\mathbf{A} \preceq \mathbf{P} \preceq (1+\epsilon)\mathbf{A}^\top\mathbf{A}. \tag{12}$$

Proof. For all $i \in [n]$, let $v_i := (\mathbf{A}^\top\mathbf{A})^{\frac{\dagger}{2}}a_i$, and let $\mathbf{\Pi} = (\mathbf{A}^\top\mathbf{A})^\dagger\mathbf{A}^\top\mathbf{A}$ be the projection matrix to Span($\mathbf{A}$). We observe that the condition (12) is equivalent to

$$(1-\epsilon)\mathbf{\Pi} \preceq (\mathbf{A}^\top\mathbf{A})^{\frac{\dagger}{2}}\mathbf{P}(\mathbf{A}^\top\mathbf{A})^{\frac{\dagger}{2}} = \sum_{k\in[K]} \frac{1}{Kp_{i_k}} v_{i_k} v_{i_k}^\top \preceq (1+\epsilon)\mathbf{\Pi}. \tag{13}$$

To prove (13), we appeal to the matrix Chernoff bound. Let $\mathbf{Z}_k$ be an independent random matrix set to $\frac{1}{Kp_i}v_i v_i^\top$ for $i \sim p$, and for all $k \in [K]$, so all the $\mathbb{E}\mathbf{Z}_k = \mathbf{\Pi}$ by (10). We have

$$\max_{i\in[n]} \frac{1}{Kp_i}\left\|v_i v_i^\top\right\|_{\text{op}} = \max_{i\in[n]} \frac{\|\tau\|_1}{K\tau_i}\|v_i\|_2^2 = \frac{\|\tau\|_1}{K}, \tag{14}$$

where the last equality used (10). Therefore, letting $R := \frac{\|\tau\|_1}{K} \leq \frac{\epsilon^2}{3}\log^{-1}(\frac{2d}{\delta})$, where we used Item 1 in Lemma 5 to obtain the inequality, the equivalent claim (13) follows by applying both the upper and lower bounds in Theorem 12, Part V,[4] and then taking a union bound. ☐

Proposition 3 yields a preconditioner which is the Gram matrix of an approximation to $\mathbf{A}$, whose row count matches Proposition 1 up to a low-order term due to its $\log(\frac{d}{\delta})$ dependence, rather than $\log\frac{1}{\delta}$. Moreover, it achieves this row count via a direct reweighting of $\mathbf{A}$'s rows of the form (9). This additional structure is not without a price; in particular, Proposition 3 is clearly non-oblivious, as it uses sampling probabilities which depend on the matrix $\mathbf{A}$ we are trying to approximate. It is also not clear how to implement Proposition 3 efficiently: it appears to require computation of $(\mathbf{A}^\top\mathbf{A})^\dagger$, which was the inversion we were trying to avoid in the first place in solving (1).

To break this chicken-and-egg problem, in the following Section 3.2 we analyze a strategy which takes advantage of an additional degree of flexibility in the proof of Proposition 3.

Corollary 1. *In the setting of Proposition 3, suppose instead that $\tau \in \mathbb{R}^n_{\geq 0}$ has $\tau_i \geq \tau_i(\mathbf{A})$ for all $i \in [n]$, and again define $p = \frac{\tau}{\|\tau\|_1}$. Then, defining $\mathbf{P}$ as in (11) with respect to the new sampling probabilities p, if $K \geq \frac{3\|\tau\|_1}{\epsilon^2}\log(\frac{2d}{\delta})$, the conclusion (12) holds with probability $\geq 1-\delta$.*

Proof. The proof is identical to Proposition 3, except (14) is an inequality instead of an equality. ☐

Corollary 1 states that as long as we have overestimates of the leverage scores of $\mathbf{A}$, as long as the overestimate quality is not too poor (i.e. $\|\tau\|_1$ is reasonable compared to rank($\mathbf{A}$)), we can still produce a preconditioner $\mathbf{P}$ with few sampled rows. This extra degree of freedom allows us to break the chicken-and-egg problem encountered in leverage score sampling mentioned earlier.

[4] Technically, we must take care because $\lambda_d(\mathbf{\Pi}) = 0$ if $\mathbf{A}$ is not full-rank. However, tracing through the proof of Theorem 12, Part V, we note that all matrices appearing in all inequalities have images in Span($\mathbf{\Pi}$) = Span($\mathbf{A}$), and therefore we obtain a concentration bound depending on the minimum eigenvalue of $\mathbf{\Pi}$ in its span.

Remark 4. *Matrices (9) which approximate $\mathbf{A}^\top\mathbf{A}$ in the sense of (12), where $\mathrm{nnz}(w) \ll n$, are often called* spectral sparsifiers. *This is because they preserve the spectrum of $\mathbf{A}^\top\mathbf{A}$ by using a sparse reweighting of its rank-one components. Proposition 3 shows that spectral sparsifiers exist for all matrices $\mathbf{A} \in \mathbb{R}^{n\times d}$ with $\mathrm{nnz}(w) = O(\frac{d\log d}{\epsilon^2})$; it is natural to ask if this is improvable. In a breakthrough result, [BSS14] proved that $\mathrm{nnz}(w) = O(\frac{d}{\epsilon^2})$ suffices using a careful analysis of an iterative process guided by potential functions, known to be optimal up to a constant factor.*

3.2 Uniform sampling

In this section, we summarize an elegant observation made by [CLM+15] which allows us to implement leverage score sampling significantly more efficiently, breaking the chicken-and-egg problem described in Section 3.1. To state this result more cleanly, for $S \subseteq [n]$, let $\mathbf{A}_{S:} \in \mathbb{R}^{|S|\times d}$ be the subset of $\mathbf{A}$'s rows indexed by S, and let

$$\tau_i^S(\mathbf{A}) := a_i^\top(\mathbf{A}_{S:}^\top\mathbf{A}_{S:})^\dagger a_i. \tag{15}$$

Lemma 6. *For $k \in [n]$, let S be a uniformly random subset of $[n]$ of size k. Then for $\mathbf{A} \in \mathbb{R}^{n\times d}$,*

$$\mathbb{E}\left[\sum_{i\in[n]} \tau_i^{S\cup\{i\}}(\mathbf{A})\right] \le \frac{nd}{k}.$$

Moreover, $\tau_i^S(\mathbf{A}) \ge \tau_i(\mathbf{A})$ for any $S \subseteq [n]$.

Proof. The second conclusion follows immediately from the definition (15) and $\mathbf{A}_{S:}^\top\mathbf{A}_{S:} \preceq \mathbf{A}^\top\mathbf{A}$. To see the first conclusion, note that for any $S \subseteq [n]$, $\sum_{i\in S}\tau_i^S(\mathbf{A}) = \mathrm{rank}(\mathbf{A}_{S:}) \le d$, so

$$\mathbb{E}\left[\sum_{i\in[n]} \tau_i^{S\cup\{i\}}(\mathbf{A})\right] = \mathbb{E}\left[\sum_{i\in S} \tau_i^{S}(\mathbf{A})\right] + \mathbb{E}\left[\sum_{i\notin S} \tau_i^{S\cup\{i\}}(\mathbf{A})\right] \le d + \mathbb{E}\left[\sum_{i\notin S} \tau_i^{S\cup\{i\}}(\mathbf{A})\right].$$

To bound the second term, note $\frac{1}{n-k}\mathbb{E}\sum_{i\notin S}\tau_i^{S\cup\{i\}}(\mathbf{A})$ is the expectation of a random variable which first selects $S \subseteq [n]$ and then returns $\tau_i^{S\cup\{i\}}(\mathbf{A})$ for a uniformly random $i \in [n]\setminus S$. However, sampling a uniformly random $S \subseteq [n]$ of size k is the same thing as sampling a uniformly random subset $T \subseteq [n]$ of size $k+1$,[5] and then dropping a random $i \in T$. Therefore,

$$\frac{1}{n-k}\mathbb{E}\left[\sum_{i\notin S} \tau_i^{S\cup\{i\}}(\mathbf{A})\right] = \mathbb{E}_T\left[\mathbb{E}_{i\sim_{\mathrm{unif.}}T}\left[\tau_i^T(\mathbf{A})\right]\right] \le \mathbb{E}_T\left[\frac{d}{k+1}\right] = \frac{d}{k+1}.$$

Combining the above two displays,

$$\mathbb{E}\left[\sum_{i\in[n]} \tau_i^{S\cup\{i\}}(\mathbf{A})\right] \le d + \frac{d(n-k)}{k+1} \le \frac{nd}{k}.$$

□

We mention one fact which simplifies the computation in Lemma 6.

Lemma 7. *Let $S \subseteq [n]$ and $i \notin S$. Then, $\tau_i^{S\cup\{i\}}(\mathbf{A}) = \frac{\tau_i^S(\mathbf{A})}{1+\tau_i^S(\mathbf{A})}$. Moreover, given a value $\tau \in [\tau_i^S(\mathbf{A}), (1+\epsilon)\tau_i^S(\mathbf{A})]$ for $\epsilon \in (0,1)$, we have $\frac{\tau}{1+\tau} \in [\tau_i^{S\cup\{i\}}(\mathbf{A}), (1+\epsilon)\tau_i^{S\cup\{i\}}(\mathbf{A})]$.*

Proof. The Sherman-Morrision formula states that for any matrix $\mathbf{M} \in \mathbb{S}^{d\times d}$ and $u \in \mathbb{R}^d$, we have

$$\left(\mathbf{M} + uu^\top\right)^\dagger = \mathbf{M}^\dagger - \frac{\mathbf{M}^\dagger uu^\top\mathbf{M}^\dagger}{1+u^\top\mathbf{M}^\dagger u}, \tag{16}$$

[5] The conclusion of the lemma is straightforward if $k = n$.

which can be verified by direct expansion. Therefore, letting $\mathbf{M} := \mathbf{A}_{S:}^\top \mathbf{A}_{S:}$,

$$a_i^\top \left(\mathbf{M} + a_i a_i^\top\right)^\dagger a_i = a_i^\top \mathbf{M}^\dagger a_i - \frac{(a_i^\top \mathbf{M}^\dagger a_i)^2}{1 + a_i^\top \mathbf{M}^\dagger a_i} = \frac{a_i^\top \mathbf{M}^\dagger a_i}{1 + a_i^\top \mathbf{M}^\dagger a_i}.$$

This yields the first claim, and the second claim then follows straightforwardly. □

Lemmas 6 and 7 show that if we can compute a good approximation to $(\mathbf{A}_{S:}^\top \mathbf{A}_{S:})^\dagger$ for a uniformly random subset $S \subseteq [n]$, we can use it to provide reasonable overestimates of the leverage scores of $\mathbf{A}$. These overestimates may not let us sparsify $\mathbf{A}$ in one shot, but we will show they let us significantly reduce the number of rows in $\mathbf{A}$, yielding a favorable recursion.

Theorem 2 (Linear regression via recursive row sampling)**.** *Let $\epsilon, \delta \in (0,1)$, $b \in \mathbb{R}^n$, and let $\mathbf{A} \in \mathbb{R}^{n \times d}$ be full-rank. There is an algorithm which computes $\hat{x} \in \mathbb{R}^d$ satisfying (3) with probability $\geq 1 - \delta$ in time*

$$O\left((\mathrm{nnz}(\mathbf{A}) + d^\omega)\left(\log^2(n) + \log\left(\frac{1}{\epsilon}\right)\right)\log\left(\frac{1}{\delta}\right)\right).$$

Proof. Throughout this proof, fix a parameter $R = O(\log\log \frac{n}{d}) \in \mathbb{N}$, which will be a bound on the number of rounds of recursion we perform. Moreover, for each $r \in [R]$ and $k \in [n]$, consider the following computational task. For $\mathbf{W} = \mathbf{diag}\,(w) \in \mathbb{R}^n_{\geq 0}$ with $\mathrm{nnz}(\mathbf{W}) \leq k$, we let $\mathcal{T}_r(k)$ be the time required to produce a matrix $\mathbf{P} \in \mathbb{R}^{d \times d}$ such that

$$\left(\mathbf{A}^\top \mathbf{W} \mathbf{A}\right)^\dagger \preceq \mathbf{P}^2 \preceq \exp\left(\frac{r}{R}\right)\left(\mathbf{A}^\top \mathbf{W} \mathbf{A}\right)^\dagger, \tag{17}$$

conditioned on an event $\mathcal{E}$ which will be described. We bound the cost $\mathcal{T}_r(n)$ using the following algorithmic framework, which consists of 3 steps at each layer of recursion.

1. For a parameter $k \in [n]$ to be chosen, we uniformly sample a subset $S \subseteq [n]$ and in $\mathcal{T}_{r-1}(k)$ time, we produce a matrix $\mathbf{P}$ satisfying (17) where $w = \mathbb{1}_S$ is set to the 0-1 indicator for S.
2. Using $\mathbf{P}$ and Lemma 7, we approximate $\tau_i^{S \cup \{i\}}(\mathbf{A})$ for all $i \in [n]$, by sampling a $k \times d$ $\mathbf{G}$ with entries i.i.d. $\sim \mathcal{N}(0, \frac{1}{k})$ for an appropriate k, and estimating $2a_i^\top \mathbf{P}\mathbf{G}^\top \mathbf{G}\mathbf{P}a_i \approx \tau_i^S(\mathbf{A})$.
3. Finally, we use our leverage score overestimates through Corollary 1 to produce a sparser approximation to $\mathbf{A}$, which allows us to recurse once again.

The event $\mathcal{E}$ we condition on is that Items 1, 2, and 3 succeed in each of our $\leq 2^R$ calls to each of the three randomized procedures described above, i.e. Lemma 6, the Johnson-Lindenstrauss lemma (Corollary 1, Part V), and Corollary 1. We set the failure probability for each of these to be $\frac{1}{9 \cdot 2^R}$. When we say we condition on Lemma 6 succeeding, we mean the sum of leverage scores through $S \cup \{i\}$ is within a $9 \cdot 2^R$ factor of its expectation, which is correct by Markov's inequality. Therefore, by a union bound, we condition on $\mathcal{E}$ in the proof which occurs with probability $\geq \frac{2}{3}$.

We now formalize each of these steps and discuss their runtimes, under $\mathcal{E}$. Given a matrix $\mathbf{P}$ satisfying (17) for $w = \mathbb{1}_S$, taking $k = O(\log(n) + R) = O(\log(n))$ in Corollary 1, Part V guarantees all $a_i^\top \mathbf{P}\mathbf{G}^\top \mathbf{G}\mathbf{P}a_i$ for $i \in [n]$ are within a 2 multiplicative factor of their expectations, i.e.

$$a_i^\top \left(\mathbf{A}^\top \mathbf{W} \mathbf{A}\right)^\dagger a_i \leq 2a_i^\top \mathbf{P}\mathbf{G}^\top \mathbf{G}\mathbf{P}a_i \leq 12 a_i^\top \left(\mathbf{A}^\top \mathbf{W} \mathbf{A}\right)^\dagger a_i, \tag{18}$$

where we also used (17). Note that computing all $2a_i^\top \mathbf{P}\mathbf{G}^\top \mathbf{G}\mathbf{P}a_i$ takes time $O((\mathrm{nnz}(\mathbf{A}) + d^2)\log(n))$, since we can first compute $\mathbf{P}\mathbf{G}^\top$ in time $O(d^2 \log(n))$, and then multiply all rows of $\mathbf{G}\mathbf{P}$ by $\{a_i\}_{i \in [n]}$. Next, assuming Lemma 6 succeeds in the sense described earlier, the sum of leverage score overestimates $\tau_i^{S \cup \{i\}}(\mathbf{A})$ is $\leq 9 \cdot 2^R \cdot \frac{nd}{k}$, so using (18) through Lemma 7 with $\epsilon \leftarrow 11$ produces overestimates $\tau \in \mathbb{R}^n_{\geq 0}$ which, combining (18) with our earlier bound, satisfy

$$\sum_{i \in [n]} \tau_i \leq 108 \cdot 2^R \cdot \frac{nd}{k}.$$

Using these overestimates we can apply Corollary 1 with error parameter $\epsilon \leftarrow \frac{1}{3R}$, which increases the approximation factor in (17) by an $\exp(\frac{1}{R})$ factor, accounting for this layer of recursion, using

$1296 \cdot 2^R R^2 \cdot \frac{nd}{k}(\log(18d) + R)$ sampled rows. Therefore, putting together all the pieces and using that $1296 \cdot 2^R R^2 \cdot \frac{nd}{k}(\log(18d) + R) \leq \frac{Cnd}{k} \log^3(n)$ for a constant C,

$$\mathcal{T}_r(n) = O\left(\left(\text{nnz}(\mathbf{A}) + d^2\right)\log(n)\right) + \mathcal{T}_{r-1}(k) + \mathcal{T}_{r-1}\left(\frac{Cnd}{k}\log^3(n)\right).$$

Choosing k optimally, we therefore have

$$\mathcal{T}_r(n) = O\left(\left(\text{nnz}(\mathbf{A}) + d^2\right)\log(n)\right) + 2\mathcal{T}_{r-1}\left(\sqrt{Cnd\log^3(n)}\right).$$

We let the recursion proceed, setting $r \leftarrow r - 1$ and $n \leftarrow \sqrt{Cnd\log^3(n)}$, until n becomes $2n_0$ for some $n_0 := O(d\log^3(d))$, at which point we can also bound[6] $\mathcal{T}_1(n) = O(d^\omega)$ because we can explicitly compute $\mathbf{A}^\top\mathbf{W}\mathbf{A}$, and invert and square root it in this time.[7] It is straightforward to check this process does indeed terminate in $R = O(\log\log\frac{n}{d})$ rounds of recursion, as claimed in the beginning of the proof, since $\log\frac{n}{n_0}$ halves at each round. Finally, the overall runtime is

$$\mathcal{T}_R(n) = O\left(\left(\text{nnz}(\mathbf{A}) + d^2\right)\log^2(n) + d^\omega\log(n)\right),$$

attaining failure probability $\frac{1}{3}$. Assuming the top layer of recursion succeeded, the remainder of the proof follows identically to Theorem 1. We boost the result to fail with probability $\leq \delta$ by again taking $O(\log\frac{1}{\delta})$ independent runs and outputting the point with best function value in (1). □

Notably, up to polylogarithmic factors, Theorem 2 gives a runtime which matches what Theorem 1 would have obtained, if the sparse OSEs in Proposition 2 had as few rows as the dense OSEs in Proposition 1, but were still just as efficient to apply. Moreover, it does so by using a preconditioner of the form (9), which preserves the structure of $\mathbf{A}$. As mentioned in Remark 2, these logarithmic factors have since been removed by [CSWZ23]. Interestingly, [CSWZ23] does so using embeddings that are not of the form (9), instead using properties of subsampled Hadamard transforms.

[6] We let exponentiation by ω suppress polylogarithmic factors for readability, i.e. we assume we can multiply $d \cdot \text{polylog}(d)$-dimension matrices in d^ω time, which affects ω by a $o(1)$ factor.

[7] This is assuming exact arithmetic, via the eigendecomposition algorithm in [PC99]. In finite-precision arithmetic, one can use polynomial approximations to the square root, see e.g. Fact 4, [JLM+23].

Source material

Portions of this lecture are based on reference material in [Woo14, LV23], as well as the author's own experience working in the field.

References

[AJSS19] AmirMahdi Ahmadinejad, Arun Jambulapati, Amin Saberi, and Aaron Sidford. Perron-frobenius theory in nearly linear time: Positive eigenvectors, m-matrices, graph kernels, and other applications. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019*, pages 1387–1404. SIAM, 2019.

[BMNW22] Mitchell Black, William Maxwell, Amir Nayyeri, and Eli Winkelman. Computational topology in a collapsing universe: Laplacians, homology, cohomology. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022*, pages 226–251. SIAM, 2022.

[BSS14] Joshua D. Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-ramanujan sparsifiers. *SIAM Rev.*, 56(2):315–334, 2014.

[CFM+14] Michael B. Cohen, Brittany Terese Fasy, Gary L. Miller, Amir Nayyeri, Richard Peng, and Noel Walkington. Solving 1-laplacians in nearly linear time: Collapsing and expanding a topological ball. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*, pages 204–216. SIAM, 2014.

[CKK+18] Michael B. Cohen, Jonathan A. Kelner, Rasmus Kyng, John Peebles, Richard Peng, Anup B. Rao, and Aaron Sidford. Solving directed laplacian systems in nearly-linear time through sparse LU factorizations. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018*, pages 898–909. IEEE Computer Society, 2018.

[CLM+15] 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*, pages 181–190. ACM, 2015.

[Coh16] 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*, pages 278–287. SIAM, 2016.

[CSWZ23] Yeshwanth Cherapanamjeri, Sandeep Silwal, David P. Woodruff, and Samson Zhou. Optimal algorithms for linear algebra in the current matrix multiplication time. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023*, pages 4026–4049. SIAM, 2023.

[CW13] 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, 2013*, pages 81–90. ACM, 2013.

[JLM+23] Arun Jambulapati, Jerry Li, Christopher Musco, Kirankumar Shiragur, Aaron Sidford, and Kevin Tian. Structured semidefinite programming for recovering structured preconditioners. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023*, 2023.

[KLP+16] Rasmus Kyng, Yin Tat Lee, Richard Peng, Sushant Sachdeva, and Daniel A. Spielman. Sparsified cholesky and multigrid solvers for connection laplacians. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016*, pages 842–850. ACM, 2016.

[KPSZ18] Rasmus Kyng, Richard Peng, Robert Schwieterman, and Peng Zhang. Incomplete nested dissection. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018*, pages 404–417. ACM, 2018.

[LV23] Yin Tat Lee and Santosh Vempala. *Techniques in Optimization and Sampling*. 2023.

[NN13a] 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*, pages 117–126. IEEE Computer Society, 2013.

[NN13b] Jelani Nelson and Huy L. Nguyen. Sparsity lower bounds for dimensionality reducing maps. In *Symposium on Theory of Computing Conference, STOC'13, 2013*, pages 101–110. ACM, 2013.

[PC99] Victor Y. Pan and Zhao Q. Chen. The complexity of the matrix eigenproblem. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, 1999*, pages 507–516. ACM, 1999.

[Sar06] Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006)*, pages 143–152. IEEE Computer Society, 2006.

[ST14] Daniel A. Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM J. Matrix Anal. Appl.*, 35(3):835–885, 2014.

[TZ12] Mikkel Thorup and Yin Zhang. Tabulation-based 5-independent hashing with applications to linear probing and second moment estimation. *SIAM J. Comput.*, 41(2):293–331, 2012.

[Woo14] David P. Woodruff. Sketching as a tool for numerical linear algebra. *Found. Trends Theor. Comput. Sci.*, 10(1-2):1–157, 2014.