2013 IEEE 54th Annual Symposium on Foundations of Computer Science

OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings

Jelani Nelson
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138
minilek@seas.harvard.edu

Huy L. Nguyễn
Department of Computer Science
Princeton University
Princeton, NJ 08540
hlnguyen@princeton.edu

***Abstract*—An** *oblivious subspace embedding (OSE)* **given some parameters ε, d is a distribution $\mathcal{D}$ over matrices $\Pi \in \mathbb{R}^{m\times n}$ such that for any linear subspace $W \subseteq \mathbb{R}^n$ with $\dim(W) = d$,**

$$\mathbb{P}_{\Pi\sim\mathcal{D}}(\forall x \in W \ \|\Pi x\|_2 \in (1\pm\varepsilon)\|x\|_2) > 2/3.$$

We show that a certain class of distributions, *Oblivious Sparse Norm-Approximating Projections* (OSNAPs), provides OSE's with $m = O(d^{1+\gamma}/\varepsilon^2)$, and where every matrix Π in the support of the OSE has only $s = O_\gamma(1/\varepsilon)$ non-zero entries per column, for $\gamma > 0$ any desired constant. Plugging OSNAPs into known algorithms for approximate least squares regression, ℓ_p regression, low rank approximation, and approximating leverage scores implies faster algorithms for all these problems.

Our main result is essentially a Bai-Yin type theorem in random matrix theory and is likely to be of independent interest: we show that for any fixed $U \in \mathbb{R}^{n\times d}$ with orthonormal columns and random sparse Π, all singular values of ΠU lie in $[1-\varepsilon, 1+\varepsilon]$ with good probability. This can be seen as a generalization of the sparse Johnson-Lindenstrauss lemma, which was concerned with $d = 1$. Our methods also recover a slightly sharper version of a main result of [Clarkson-Woodruff, STOC 2013], with a much simpler proof. That is, we show that OSNAPs give an OSE with $m = O(d^2/\varepsilon^2)$, $s = 1$.

***Keywords*-subspace embedding; numerical linear algebra; Johnson-Lindenstrauss lemma**

I. Introduction

There has been much recent work on applications of dimensionality reduction to handling large datasets. Typically special features of the data such as low "intrinsic" dimensionality, or sparsity, are exploited to reduce the volume of data before processing, thus speeding up analysis time. One success story of this approach is the applications of fast algorithms for the Johnson-Lindenstrauss (JL) lemma [20], which allows one to reduce the dimensionality of a set of vectors while preserving all pairwise distances. There have been two popular lines of work in this area: one focusing on fast embeddings for all vectors [2]–[4], [19], [24], [25], [37], and one focusing on fast embeddings specifically for sparse vectors [1], [6], [13], [21], [22].

In this work we focus on the problem of constructing an *oblivious subspace embedding (OSE)* [32] and on applications of these embeddings. Roughly speaking, the problem is to design a data-independent distribution over linear mappings such that when data come from an *unknown* low-dimensional subspace, they are reduced to roughly their true dimension while their structure (all distances in the subspace in this case) is preserved at the same time. It can be seen as a continuation of the approach based on the JL lemma to subspaces, and these embeddings have found applications in numerical linear algebra problems such as least squares regression, ℓ_p regression, low rank approximation, and approximating leverage scores [9]–[11], [15], [31], [32], [35]. We refer the interested reader to the surveys [17], [27] for an overview. Here we focus on the setting of sparse inputs, where it is important that the algorithms take time proportional to the input sparsity.

Throughout this document we use $\|\cdot\|$ to denote ℓ_2 norm in the case of vector arguments, and $\ell_{2\to2}$ operator norm for matrix arguments. Recall the definition of an OSE.

Definition 1. *An* oblivious subspace embedding (OSE) *is a distribution over $m\times n$ matrices Π such that for any d-dimensional subspace $W \subset \mathbb{R}^n$, $\mathbb{P}_{\Pi\sim\mathcal{D}}(\forall x \in W \ \|\Pi x\|_2 \in (1\pm\varepsilon)\|x\|_2) > 2/3$. Here n, d, ε are given parameters of the problem and we would like m as small as possible.*

OSE's were first introduced in [32] as a means to obtain fast randomized algorithms for several numerical linear algebra problems. To see the connection, consider for example the least squares regression problem of computing $\operatorname{argmin}_{x\in\mathbb{R}^d}\|Ax-b\|$ for some $A \in \mathbb{R}^{n\times d}$. Suppose $\Pi \in \mathbb{R}^{m\times n}$ preserves the ℓ_2 norm up to $1\pm\varepsilon$ of all vectors in the subspace spanned by b and the columns of A. Let $\tilde{x} = \operatorname{argmin}_x\|\Pi Ax - \Pi b\|$, $x^* = \operatorname{argmin}_x\|Ax-b\|$. Then

$$\begin{aligned}(1-\varepsilon)\|A\tilde{x}-b\| &\le \|\Pi A\tilde{x}-\Pi b\| \\ &\le \|\Pi Ax^*-\Pi b\| \\ &\le (1+\varepsilon)\|Ax^*-b\|.\end{aligned}$$

Thus $\tilde{x}$ provides a solution within $(1+\varepsilon)/(1-\varepsilon) = 1+O(\varepsilon)$ of optimal. Since this subspace has dimension at most $d+1$, one only needs m being some function of ε, d. Thus the running time for approximate $n\times d$ regression becomes that

0272-5428/13 $26.00 © 2013 IEEE
DOI 10.1109/FOCS.2013.21

for $m \times d$ regression, plus an additive term for the time required to compute $\Pi A, \Pi b$. This is a gain for instances with $n \gg d$. Also, the $2/3$ success probability guaranteed by Definition 1 can be amplified to $1-\delta$ by running this procedure $O(\log(1/\delta))$ times with independent randomness and taking the best $\tilde{x}$ found in any run. We furthermore point out that another reduction from $(1+\varepsilon)$-approximate least squares regression to OSE's via preconditioning followed by gradient descent actually only needs an OSE with constant distortion independent of ε (see [11]), so that $\varepsilon = \Theta(1)$ in an OSE is of primary interest.

It is known that a random matrix with independent subgaussian entries and $m = O(d/\varepsilon^2)$ provides an OSE with $1+\varepsilon$ distortion (see for example [11]). Unfortunately, the time to compute ΠA is then larger than the known $\tilde{O}(nd^{\omega-1})$ time bound to solve the exact regression problem itself, where $\omega < 2.373\ldots$ [39] is the exponent of square matrix multiplication. In fact, since $m \geq d$ in any OSE, dividing Π, A into $d \times d$ blocks and using fast square matrix multiplication to then multiply ΠA would yield time $\Theta(mnd^{\omega-2})$, which is $\Omega(nd^{\omega-1})$. Thus implementing the approach of the previous paragraph naively provides no gains. The work of [32] overcame this by choosing a special Π so that ΠA can be computed in time $O(nd\log n)$ (see also [35]). This matrix Π was the Fast JL Transform of [2], which has the property that Πx can be computed in roughly $O(n\log n)$ time for any $x \in \mathbb{R}^n$. Thus, multiplying ΠA by iterating over columns of A gives the desired speedup.

The $O(nd\log n)$ running time of the above scheme to compute ΠA seems almost linear, and thus nearly optimal, since the input size to describe A is nd. While this is true for dense A, in many applications one expects A to be sparse, in which case linear in the input description actually means $O(\mathrm{nnz}(A))$, where $\mathrm{nnz}(\cdot)$ counts non-zero entries. For example, one numerical linear algebra problem of wide interest is matrix completion, where one assumes that some small number of entries in a low rank matrix A have been revealed, and the goal is to then recover A. This problem arises in recommendation systems, where for example the rows of A represent users and the columns represent products, and $A_{i,j}$ is the rating of product j by customer i. One wants to infer "hidden ratings" to then make product recommendations, based on the few ratings that customers have actually made. Such matrices are usually very sparse; when for example $A_{i,j}$ is user i's score for movie j in the Netflix matrix, only roughly 1% of the entries of A are known [41]. Some matrix completion algorithms work by iteratively computing singular value decompositions (SVDs) of various matrices that have the same sparsity as the initial A, then thresholding the result to only contain the large singular values then re-sparsifying [7]. Furthermore it was empirically observed that the matrix iterates were low rank, so that a fast low rank approximation algorithm for sparse matrices, as what is provided in this work, could replace full SVD computation to give speedup.

In a recent beautiful and surprising work, [11] showed that there exist OSE's with $m = \mathrm{poly}(d/\varepsilon)$, and where every matrix Π in the support of the distribution is *very* sparse: even with only $s = 1$ non-zero entry per column! Thus one can transform, for example, an $n \times d$ least squares regression problem into a $\mathrm{poly}(\mathrm{d}/\varepsilon) \times d$ regression problem by multiplying ΠA in $\mathrm{nnz}(A) \cdot s = \mathrm{nnz}(A)$ time. The work [11] gave two sparse OSE's: one with $m = O(d^2\log^6(d/\varepsilon)/\varepsilon^2), s = 1$, and another with $m = \tilde{O}(d^2\log(1/\delta)/\varepsilon^4 + d\log^2(1/\delta)/\varepsilon^4), s = O(\log(d/\delta)/\varepsilon)$. The second construction has the benefit of providing a subspace embedding with success probability $1-\delta$ and not just $2/3$, which is important e.g. in a known reduction from ℓ_p regression to OSE's [9].

Our Main Contribution: We give OSE's with $m = O(d^{1+\gamma}/\varepsilon^2), s = O_\gamma(1/\varepsilon)$, where $\gamma > 0$ can be any constant. Note s does not depend on d. The constant hidden in the O_γ is $\mathrm{poly}(1/\gamma)$. The success probability is $1 - 1/d^c$ for any desired constant c. One can also set $m = O(d \cdot \mathrm{polylog}(d/\delta)/\varepsilon^2), s = \mathrm{polylog}(d/\delta)/\varepsilon$ for success probability $1-\delta$. Ours are the first analyses to give OSE's having $m = o(d^2)$ with $s = o(d)$. Observe that in both our parameter settings m is nearly linear in d, which is nearly optimal since any OSE must have $m = \Omega(d/\varepsilon^2)$ [29]. We also show that a simpler instantiation of our approach gives $m = O(d^2/\varepsilon^2), s = 1$, recovering a sharpening of a main result of [11] with a much simpler proof. Our quadratic dependence on d is optimal for $s = 1$ [30].

Plugging our improved OSE's into previous work implies faster algorithms for several numerical linear algebra problems, such as approximate least squares regression, ℓ_p regression, low rank approximation, and approximating leverage scores. In fact for all these problems, except approximating leverage scores, known algorithms only make use of OSE's with distortion $\Theta(1)$ independent of the desired $1+\varepsilon$ approximation guarantee, in which case our matrices have $m = O(d^{1+\gamma}), s = O_\gamma(1)$, i.e. constant column sparsity and a near-optimal number of rows.

We also remark that the analyses of [11] require $\Omega(d)$-wise independent hash functions, so that from the seed used to generate Π naively one needs an additive $\Omega(d)$ time to identify the non-zero entries in each column just to evaluate the hash function. In streaming applications this can be improved to additive $\tilde{O}(\log^2 d)$ time using fast multipoint evaluation of polynomials (see [23, Remark 16]), though ideally if $s = 1$ one could hope for a construction that allows one to find, for any column, the non-zero entry in that column in constant time given only a short seed that specifies Π (i.e. without writing down Π explicitly in memory, which could be prohibitively expensive for n large in applications such as streaming and out-of-core numerical linear algebra).

Recall that in the entry-wise turnstile streaming model, A receives entry-wise updates of the form $((i,j),v)$, which cause the change $A_{i,j} \leftarrow A_{i,j} + v$. Updating the embedding thus amounts to adding v times the jth row of Π to ΠA, which should ideally take $O(s)$ time and not $O(s) + \tilde{O}(\log^2 d)$. Our analyses only use 4-wise independent hash functions when $s = 1$ and $O(\log d)$-wise independent hash functions for larger s, thus allowing fast computation of any column of Π from a short seed.

A. Problem Statements and Bounds

We now formally define all numerical linear algebra problems we consider. Plugging our new OSE's into previous algorithms provides speedup for all these problems (see Figure 1; the consequences for ℓ_p regression are also given in Section III). The value r used in bounds denotes $\mathrm{rank}(A)$. In what follows, $b \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n\times d}$.

Leverage Scores: Let $A = U\Sigma V^*$ be the SVD. Output the row ℓ_2 norms of U up to $1 \pm \varepsilon$.

Least Squares Regression: Compute $\tilde{x} \in \mathbb{R}^d$ so that $\|A\tilde{x} - b\| \leq (1+\varepsilon) \cdot \min_{x\in\mathbb{R}^d} \|Ax - b\|$.

ℓ_p Regression ($p \in [1,\infty)$): Compute $\tilde{x} \in \mathbb{R}^d$ so that $\|A\tilde{x} - b\|_p \leq (1+\varepsilon) \cdot \min_{x\in\mathbb{R}^d} \|Ax - b\|_p$.

Low Rank Approximation: Given integer $k > 0$, compute $\tilde{A}_k \in \mathbb{R}^{n\times d}$ with $\mathrm{rank}(\tilde{A}) \leq k$ s.t. $\|A - \tilde{A}_k\|_F \leq (1+\varepsilon) \cdot \min_{\mathrm{rank}(A_k)\leq k} \|A - A_k\|_F$, where $\|\cdot\|_F$ is Frobenius norm.

B. Our Approach

Let $\Pi \in \mathbb{R}^{m\times n}$ be a sparse JL matrix as in [22]. For example, one such construction is to choose each column of Π independently, and within a column we pick exactly s random locations (without replacement) and set the corresponding entries to $\pm 1/\sqrt{s}$ at random with all other entries in the column then set to zero. Observe any d-dimensional subspace $W \subseteq \mathbb{R}^n$ satisfies $W = \{x : \exists y \in \mathbb{R}^d, x = Uy\}$ for some $U \in \mathbb{R}^{n\times d}$ whose columns form an orthonormal basis for W. A matrix Π preserving ℓ_2 norms of all $x \in W$ up to $1 \pm \varepsilon$ is thus equivalent to the statement $\|\Pi Uy\| = (1\pm\varepsilon)\|Uy\|$ simultaneously for all $y \in \mathbb{R}^d$. This is equivalent to $\|\Pi Uy\| = (1\pm\varepsilon)\|y\|$ since $\|Uy\| = \|y\|$. This in turn is equivalent to all singular values of ΠU lying in the interval $[1-\varepsilon, 1+\varepsilon]$.[1] Write $S = (\Pi U)^*\Pi U$, so that we want to show all eigenvalues of S lie in $[(1-\varepsilon)^2, (1+\varepsilon)^2]$. That is, we want to show

$$(1-\varepsilon)^2 \leq \inf_{\|y\|=1} \|Sy\| \leq \sup_{\|y\|=1} \|Sy\| \leq (1+\varepsilon)^2.$$

[1]Recall singular values of a (possibly rectangular) matrix B are the square roots of eigenvalues of B^*B; $(\cdot)^*$ denotes conjugate transpose.

By the triangle inequality we have $\|Sy\| = \|y\| \pm \|(S-I)y\|$. Thus, it suffices to show $\|S - I\| \leq \min\{1-(1-\varepsilon)^2, (1+\varepsilon)^2 - 1\} = 2\varepsilon - \varepsilon^2$. By Markov's inequality

$$\mathbb{P}(\|S-I\| > t) < t^{-\ell} \cdot \mathbb{E}\|S-I\|^\ell \leq t^{-\ell} \cdot \mathbb{E}\mathrm{tr}((S-I)^\ell) \quad (1)$$

for any even integer $\ell \geq 2$. This is because if the eigenvalues of $S-I$ are $\lambda_1, \ldots, \lambda_d$, then those of $(S-I)^\ell$ are $\lambda_1^\ell, \ldots, \lambda_d^\ell$. Thus $\mathrm{tr}((S-I)^\ell) = \sum_i \lambda_i^\ell \geq \max_i |\lambda_i|^\ell = \|S-I\|^\ell$, since ℓ is even so that the λ_i^ℓ are nonnegative. Setting $\ell = 2$ allows $m = O(d^2/\varepsilon^2), s = 1$ with a simple proof (Theorem 2), and $\ell = \Theta(\log d)$ yields the main result with $s > 1$ and $m \approx d/\varepsilon^2$ (Theorem 5 and Theorem 8).

We remark that this method of bounding the range of singular values of a random matrix by computing the expectation of traces of large powers is a classical approach in random matrix theory (see the work of Bai and Yin [5]). Such bounds were also used in bounding operator norms of random matrices in work of Füredi and Komlós [16], and in computing the limiting spectral distribution by Wigner [38]. See also the surveys [33], [36]. We also remark that this work can be seen as a natural extension of the work on the sparse JL lemma itself. Indeed, if one imagines that $d = 1$ so that $U = u \in \mathbb{R}^{n\times 1}$ is a "1-dimensional matrix" with orthonormal columns (i.e. a unit vector), then preserving the 1-dimensional subspace spanned by u with probability $1-\delta$ is equivalent to preserving the ℓ_2 norm of u with probability $1-\delta$. Indeed, in this case the expression $\|S - I\|$ in Eq. (1) is simply $|\|\Pi u\|^2 - 1|$. This is *exactly* the JL lemma, where one achieves $m = O(1/(\varepsilon^2\delta)), s = 1$ by a computation of the second moment [34], and $m = O(\log(1/\delta)/\varepsilon^2), s = O(\log(1/\delta)/\varepsilon)$ by a computation of the $O(\log(1/\delta))$th moment [22].

Our approach is very different from that of Clarkson and Woodruff [11]. For example, take the $s = 1$ construction so that Π is specified by a random hash function $h : [n] \to [m]$ and a random $\sigma \in \{-1,1\}^n$, where $[n] \stackrel{\mathrm{def}}{=} \{1, \ldots, n\}$. For each $i \in [n]$ we set $\Pi_{h(i),i} = \sigma_i$, and every other entry in Π is set to zero. The analysis in [11] then worked roughly as follows: let $\mathcal{I} \subset [n]$ denote the set of "heavy" rows, i.e. those rows u_i of U where $\|u_i\|$ is "large". We write $x = x_{\mathcal{I}} + x_{[n]\setminus\mathcal{I}}$, where x_S for a set S denotes x with all coordinates in $[n]\setminus S$ zeroed out. Then $\|x\|^2 = \|x_{\mathcal{I}}\|^2 + \|x_{[n]\setminus\mathcal{I}}\|^2 + 2\langle x_{\mathcal{I}}, x_{[n]\setminus\mathcal{I}}\rangle$. The argument in [11] conditioned on $\mathcal{I}$ being perfectly hashed by h so that $\|x_{\mathcal{I}}\|^2$ is preserved exactly. Using an approach in [21], [22] based on the Hanson-Wright inequality [18] together with a net argument, [11] argued that $\|x_{[n]\setminus\mathcal{I}}\|^2$ is preserved simultaneously for all $x \in W$; this step required $\Omega(d)$-wise independence to union bound over the net. A simpler concentration argument handled $\langle x_{\mathcal{I}}, x_{[n]\setminus\mathcal{I}}\rangle$. This type of analysis led to $m = \tilde{O}(d^4/\varepsilon^4), s = 1$. A more involved refinement, where one partitions the rows of U into multiple levels of "heaviness", led to the bound $m =$

reference	regression	leverage scores	low rank approximation
[11]	$O(\text{nnz}(A)) + \tilde{O}(d^3)$ $\tilde{O}(\text{nnz}(A) + r^3)$	$\tilde{O}(\text{nnz}(A) + r^3)$	$O(\text{nnz}(A))$ + $\tilde{O}(nk^2)$
this work	$O_\gamma(\text{nnz}(A)) + O(d^{\omega+\gamma})$ $\tilde{O}(\text{nnz}(A) + r^\omega)$	$\tilde{O}(\text{nnz}(A) + r^\omega)$	$O_\gamma(\text{nnz}(A)) + \tilde{O}(nk^{\omega-1+\gamma} + k^{\omega+\gamma})$

Figure 1. The improvement gained in running times by using our OSE's, where $\gamma > 0$ is an arbitrary constant. Dependence on ε suppressed for readability; see Section III for dependence.

$O(d^2 \log^6(d/\varepsilon)/\varepsilon^2), s = 1$. The construction in [11] with similar m and larger s for $1-\delta$ success probability followed a similar but more complicated analysis; that construction hashed $[n]$ into buckets then used the sparse JL matrices of [22] in each bucket. Meanwhile, our analyses use the matrices of [22] directly without the extra hashing step.

We remark that in our analyses, the properties we need from an OSE are the following.

- For each Π in the support of the distribution, we can write $\Pi_{i,j} = \delta_{i,j}\sigma_{i,j}/\sqrt{s}$, where the σ are i.i.d. ± 1 random variables, and $\delta_{i,j}$ is an indicator random variable for the event $\Pi_{i,j} \neq 0$.
- $\forall\ j \in [n]$, $\sum_{i=1}^m \delta_{i,j} = s$ with probability 1, i.e. every column has *exactly* s non-zero entries.
- For any $S \subseteq [m] \times [n]$, $\mathbb{E} \prod_{(i,j)\in S} \delta_{i,j} \leq (s/m)^{|S|}$.
- The columns of Π are i.i.d.

We call any Π drawn from an OSE with the above properties an *oblivious sparse norm-approximating projection* (OSNAP). In our analyses, the last condition and the independence of the $\sigma_{i,j}$ can be weakened to only be (2ℓ)-wise independent, since our analyses use ℓth moment bounds.

We now sketch a brief technical overview of our proofs. When $\ell = 2$, we have $\text{tr}((S-I)^2) = \|S-I\|_F^2$, and our analysis becomes a short computation (Theorem 2). For larger ℓ, we expand $\text{tr}((S-I)^\ell)$ and compute its expectation. This expression is a sum of exponentially many monomials, each involving a product of ℓ terms. Without delving into all technical details, each such monomial can be thought of as being in correspondence with some undirected multigraph (see the dot product multigraphs in the proof of Theorem 5). We group monomials with isomorphic graphs, bound the contribution from each graph separately, then sum over all graphs. Multigraphs whose edges all have even multiplicity turn out to be easier to handle (Lemma 6). However most graphs G do not have this property. Informally speaking, the contribution of a graph turns out to be related to the product over its edges of the contribution of that edge. Let us informally call this "contribution" $F(G)$. Thus if $E' \subset E$ is a subset of the edges of G, we can write $F(G) \leq F((G|_{E'})^2)/2 + F((G|_{E\backslash E'})^2)/2$ by AM-GM, where squaring a multigraph means duplicating every edge, and $G|_{E'}$ is G with all edges in $E\backslash E'$ removed. This reduces back to the case of even edge multiplicities, but unfortunately the bound we desire on $F(G)$ depends exponentially on the number of connected components. Thus this step is bad, since if G is connected, then one of $G|_{E'}, G|_{E\backslash E'}$ can have *many* connected components for any choice of E'. For example if G is a cycle on N vertices, then any partition of the edges into two sets $E', E\backslash E'$ will have that either $G_{E'}$ or $G_{E\backslash E'}$ has at least $N/2$ components. We overcome this by showing that any $F(G)$ is bounded by some $F(G')$ with the property that every component of G' has two edge-disjoint spanning trees. We then put one such spanning tree into E' for each component, so that $G|_{E\backslash E'}$ and $G|_{E'}$ both have the same number of components as G.

C. Other Related Work

Simultaneously and independently of this work, Mahoney and Meng [28] showed that one can set $m = O(d^4/\varepsilon^4), s = 1$. Their argument was somewhat similar, although rather than using $\|S-I\|^2 \leq \text{tr}((S-I)^2)$ as in Eq. (1), [28] used the Gershgorin circle theorem. After receiving our manuscript as well as an independent tip from Petros Drineas, the authors produced a second version of their manuscript with a proof and result that match our Theorem 2. Their work also gives an alternate algorithm for $(1+\varepsilon)$ approximate ℓ_p regression in $O(\text{nnz}(A)\log n + \text{poly}(d/\varepsilon))$ time, without using a known black-box reduction from ℓ_p regression to OSE's in [9]. Their ℓ_p regression algorithm has the advantage over this work and over [11] of requiring only $\text{poly}(d)$ space, but has the disadvantage of only working for $1 \leq p \leq 2$, whereas both this work and [11] handle all $p \in [1, \infty)$. We remark that after our work, Woodruff and Zhang used our OSE's in a black box way to give even further improved ℓ_p regression algorithms for all $p \in [1, \infty)$, also using $\text{poly}(d)$ space [40].

Another simultaneous and independent related work is that of Li, Miller and Peng [26]. They provide a subspace embedding with $m = (d^{1+\gamma} + \text{nnz}(A)/d^3)/\varepsilon^2$, $s = 1$. Their embedding is non-oblivious, meaning the construction of Π requires looking at the matrix A. Their work has the advantage of smaller s by a factor $O_\gamma(1/\varepsilon)$ (although for all problems considered here except approximating leverage scores, one only needs OSE's with $\varepsilon = \Theta(1)$), and has the disadvantage of m depending on $\text{nnz}(A) \geq n$, and being non-oblivious, so that they cannot provide a $\text{poly}(d)$-space algorithm in one-pass streaming applications. Furthermore their embeddings do not have a property required for known applications of OSE's to the low rank approximation problem, and it is thus not known how to use their embeddings for this problem.

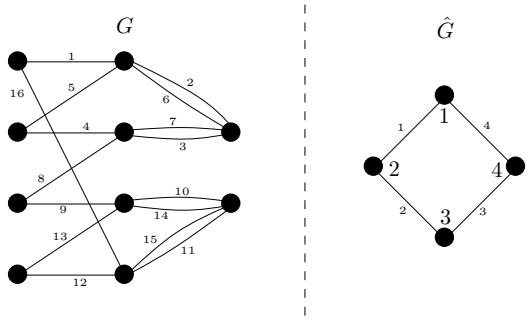

Figure 2. Example of $G \in \mathcal{G}$ and corresponding $\hat{G}$, $\ell = 4$. Here $y = 4$ (middle) and $z = 2$ (right), and $i_1 = i_2, i_3 = i_4, j_1 = j_2, j_3 = j_4, r_1 = r_2, r_3 = r_4$, where i_1, j_1, i_3, j_3 are all distinct, and $r_1 \neq r_3$. Also $w = 2$ (the top half of $MR(G)$ is disconnected from the bottom half), and $b = 4$.

II. Analysis

In this section the orthonormal columns of $U \in \mathbb{R}^{n\times d}$ are denoted $u^1, \ldots, u^d$. We implement Eq. (1) and show $\mathbb{E}\mathrm{tr}((S-I)^\ell) < t^\ell \cdot \delta$ for $t = 2\varepsilon - \varepsilon^2$ and $\delta \in (0,1)$ a failure probability parameter. Before proceeding, we assert straightforward calculations show that for all $k, k' \in [d]$

$$(S-I)_{k,k'} = \frac{1}{s}\sum_{r=1}^{m}\sum_{i\neq j\in[n]} \delta_{r,i}\delta_{r,j}\sigma_{r,i}\sigma_{r,j}u_i^k u_j^{k'}. \quad (2)$$

A. Analysis for $\ell = 2$

We first show that one can set $m = O(d^2/\varepsilon^2), s = 1$ by performing a 2nd moment computation.

Theorem 2. *For Π an OSNAP with $s = 1$ and $\varepsilon \in (0,1)$, with probability at least $1-\delta$ all singular values of ΠU are $1 \pm \varepsilon$ as long as $m \geq \delta^{-1}(d^2+d)/(2\varepsilon-\varepsilon^2)^2$.*

Proof: We need only show $\mathbb{E}\mathrm{tr}((S-I)^2) \leq (2\varepsilon-\varepsilon^2)^2 \cdot \delta$. Since $\mathrm{tr}((S-I)^2) = \|S-I\|_F^2$, we bound the expectation of this latter quantity. We first deal with the diagonal terms of $S-I$. By Eq. (2),

$$\mathbb{E}(S-I)_{k,k}^2 = \sum_{r=1}^{m}\sum_{i\neq j}\frac{2}{m^2}(u_i^k)^2(u_j^k)^2 \leq \frac{2}{m}\cdot\|u^k\|^4 = \frac{2}{m}.$$

Thus summing diagonal terms contributes at most $2d/m$.

We now focus on the off-diagonal terms. By Eq. (2),

$$\mathbb{E}(S-I)_{k,k'}^2 = \frac{1}{m^2}\sum_{r=1}^{m}\sum_{i\neq j}\left((u_i^k)^2(u_j^{k'})^2 + u_i^k u_i^{k'} u_j^k u_j^{k'}\right)$$
$$= \frac{1}{m}\sum_{i\neq j}\left((u_i^k)^2(u_j^{k'})^2 + u_i^k u_i^{k'} u_j^k u_j^{k'}\right).$$

Noting $0 = \langle u^k, u^{k'}\rangle^2 = \sum_{k=1}^{n}(u_i^k)^2(u_i^{k'})^2 + \sum_{i\neq j} u_i^k u_i^{k'} u_j^k u_j^{k'}$ we have $\sum_{i\neq j} u_i^k u_i^{k'} u_j^k u_j^{k'} \leq 0$, so

$$\mathbb{E}(S-I)_{k,k'}^2 \leq \frac{1}{m}\sum_{i\neq j}(u_i^k)^2(u_j^{k'})^2 \leq \frac{1}{m}\|u^k\|^2\cdot\|u^{k'}\|^2,$$

which equals $1/m$. Summing over $k \neq k'$, the total contribution from off-diagonal terms to $\mathbb{E}\|S-I\|_F^2$ is at most $d(d-1)/m$. Thus $\mathbb{E}\|S-I\|_F^2 \leq d(d+1)/m$, so it suffices to set $m \geq \delta^{-1}d(d+1)/(2\varepsilon-\varepsilon^2)^2$. ∎

B. Analysis for $\ell = \Theta(\log d)$

We now show that one can set $m \approx d/\varepsilon^2$ for slightly larger s by performing a $\Theta(\log d)$th moment computation. Before proceeding, we state two standard facts. Also recall u^i denotes the ith column of U, and we use u_i to denote the ith row. Full proofs can be found in the full version.

Fact 3. *Let G be a multigraph formed by removing at most k edges from a multigraph G' that has edge-connectivity at least $2k$. Then G must have at least k edge-disjoint spanning trees.*

Fact 4. *For any matrix $B \in \mathbb{C}^{d\times d}$, $\|B\| = \sup_{\|x\|,\|y\|=1} x^*By$.*

Theorem 5. *For Π an OSNAP with $s = \Theta(\log^3(d/\delta)/\varepsilon)$ and $\varepsilon \in (0,1)$, with probability at least $1-\delta$, all singular values of ΠU are $1\pm\varepsilon$ as long as $m = \Omega(d\log^6(d/\delta)/\varepsilon^2)$.*

Proof (Sketch). We show $\mathbb{E}\mathrm{tr}((S-I)^\ell) \leq (2\varepsilon-\varepsilon^2)^\ell\cdot\delta$ for $\ell = \Theta(\log d)$ an even integer then apply Eq. (1). Induction yields that $\mathbb{E}\mathrm{tr}((S-I)^\ell)$ equals

$$\frac{1}{s^\ell}\cdot\mathbb{E}\sum_{\substack{k_1,k_2,\ldots,k_{\ell+1}\\ k_1=k_{\ell+1}\\ i_1\neq j_1,\ldots,i_\ell\neq j_\ell\\ r_1,\ldots,r_\ell}}\prod_{t=1}^{\ell}\delta_{r_t,i_t}\delta_{r_t,j_t}\sigma_{r_t,i_t}\sigma_{r_t,j_t}u_{i_t}^{k_t}u_{j_t}^{k_{t+1}}. \quad (3)$$

For each monomial ψ appearing on the right hand side of Eq. (3) we associate a three-layered undirected multigraph G_ψ with labeled edges and unlabeled vertices (see Figure 2). We call these three layers the *left*, *middle*, and *right* layers, and we refer to vertices in the left layer as *left vertices*, and similarly for vertices in the other layers. Define $y = |\{i_1,\ldots,i_\ell,j_1,\ldots,j_\ell\}|$ and $z = |\{r_1,\ldots,r_\ell\}|$. The graph G_ψ has ℓ left vertices, y middle vertices corresponding to the distinct i_t, j_t in ψ, and z right vertices corresponding

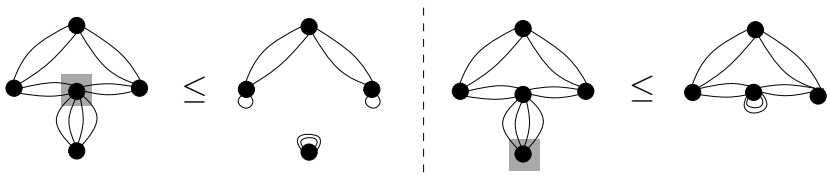

Figure 3. Choosing a good summation order π. The boxed vertex is the next vertex we sum over. The left side chose a bad vertex, since we lost connectivity, but the choice on the right is good.

to the distinct r_t. For the sake of brevity, often we refer to the vertex corresponding to i_t (resp. j_t, r_t) as simply i_t (resp. j_t, r_t). Thus note that when we refer to for example some vertex i_t, it may happen that some other $i_{t'}$ or $j_{t'}$ is also the same vertex. We now describe the edges of G_ψ. For $\psi = \prod_{t=1}^{\ell} \delta_{r_t,i_t}\delta_{r_t,j_t}\sigma_{r_t,i_t}\sigma_{r_t,j_t}u_{i_t}^{k_t}u_{j_t}^{k_{t+1}}$ we draw 4ℓ labeled edges in G_ψ with distinct labels in $[4\ell]$. For each $t \in [\ell]$ we draw an edge from the tth left vertex to i_t with label $4(t-1)+1$, from i_t to r_t with label $4(t-1)+2$, from r_t to j_t with label $4(t-1)+3$, and from j_t to the $(t+1)$st left vertex with label $4(t-1)+4$. Many different monomials ψ will map to the same graph G_ψ; in particular the graph maintains no information concerning equalities amongst the k_t, and the y middle vertices may map to any y distinct values in $[n]$, and the right vertices to any z distinct values in $[m]$. We handle the right hand side of Eq. (3) by grouping monomials ψ mapping to the same G, bounding the total contribution of G in terms of its graph structure when summing all ψ with $G_\psi = G$, then summing contributions over all G.

Before continuing further we introduce some more notation then make a few observations. For a graph G as above, recall G has 4ℓ edges, and we refer to the *distinct* edges (ignoring labels) as *bonds*. We let $E(G)$ denote the edge multiset of a multigraph G and $B(G)$ denote the bond set. We refer to the number of bonds a vertex is incident upon as its *bond-degree*, and the number of edges as its *edge-degree*. We do not count self-loops for calculating bond-degree, and we count them twice for edge-degree. We let $LM(G)$ be the induced multigraph on the left and middle vertices of G, and $MR(G)$ be the induced multigraph on the middle and right vertices. We let $w = w(G)$ be the number of connected components in $MR(G)$. We let $b = b(G)$ denote the number of bonds in $MR(G)$ (note $MR(G)$ has 2ℓ edges, but it may happen that $b < 2\ell$ since G is a multigraph). Given G we define the undirected *dot product multigraph* $\widehat{G}$ with vertex set $[y]$. Note every left vertex of G has edge-degree 2. For each $t \in [\ell]$ an edge (i, j) is drawn in $\widehat{G}$ between the two middle vertices that the tth left vertex is adjacent to (we draw a self-loop on i if $i = j$). We label the edges of $\widehat{G}$ according to the natural tour on G (by following edges in increasing label order), and the vertices with distinct labels in $[y]$ in increasing order of when each vertex was first visited by the same tour. We name $\widehat{G}$ the dot product multigraph since if some left vertex t has its two edges connected to vertices $i, j \in [n]$, then summing over $k_t \in [d]$ produces the dot product $\langle u_i, u_j\rangle$.

Now we make some observations. Due to the random signs $\sigma_{r,i}$, a monomial ψ has expectation zero unless every bond in $MR(G)$ has even multiplicity, in which case the product of random signs is 1. Also, note the expected product of the $\delta_{r,i}$ is at most $(s/m)^b$ by OSNAP properties. Thus letting $\mathcal{G}$ be the set of all such graphs G with even bond multiplicity in $MR(G)$ that arise from some monomial ψ appearing in Eq. (3), some manipulations yield (see full version) that $\mathbb{E}\mathrm{tr}((S-I)^\ell)$ is upper bounded by

$$\frac{1}{s^\ell} \cdot \sum_{G\in\mathcal{G}} \left(\frac{s}{m}\right)^b \cdot m^z \cdot \left| \sum_{\substack{a_1,\ldots,a_y\in[n] \\ \forall i\neq j\ a_i\neq a_j}} \prod_{\substack{e\in E(\widehat{G}) \\ e=(i,j)}} \langle u_{a_i}, u_{a_j}\rangle \right|. \quad (4)$$

It will also be convenient to introduce a notion we will use in our analysis called a *generalized dot product multigraph*. Such a graph $\widehat{G}$ is just as in the case of a dot product multigraph, except that each edge $e = (i, j)$ is associated with some matrix M_e. We call M_e the *edge-matrix* of e. Furthermore, for an edge $e = (i, j)$ with edge-matrix M_e, we also occasionally view e as the edge (j, i), in which case we say its associated edge-matrix is M_e^*. We associate with $\widehat{G}$ the product over all $e = (i, j) \in \widehat{G}$ of $\langle u_{a_i}, M_e u_{a_j}\rangle$. Note that a dot product multigraph is simply a generalized dot product multigraph in which $M_e = I$ for all e. Also, in such a generalized dot product multigraph, we treat multiedges as representing the same bond iff the associated edge-matrices are equal (multiedges may have different edge-matrices).

Lemma 6. *Let H be a connected generalized dot product multigraph on vertex set $[N]$ with $E(H) \neq \emptyset$ and where every bond has even multiplicity. Also suppose that for all $e \in E(H)$, $\|M_e\| \leq 1$. Then*

$$\sum_{a_2=1}^{n} \cdots \sum_{a_N=1}^{n} \prod_{\substack{e\in E(H) \\ e=(i,j)}} \langle v_{a_i}, M_e v_{a_j}\rangle \leq \|c\|^2, \quad (5)$$

where $v_{a_i} = u_{a_i}$ for $i \neq 1$, and v_{a_1} equals some fixed vector c with $\|c\| \leq 1$.

Proof (Sketch). Let π be some permutation of $\{2, \ldots, N\}$. For a bond $q = (i, j) \in B(H)$, let $2\alpha_q$ denote the

multiplicity of q in H. Then by ordering the assignments of the a_t in the summation on the left hand side of Eq. (5) according to π, we obtain the exactly equal expression

$$\sum_{a_{\pi(N)}=1}^{n} \prod_{\substack{q\in B(H)\\ q=(\pi(N),j)\\ N\le\pi^{-1}(j)}} \langle v_{a_{\pi(N)}}, M_q v_{a_j}\rangle^{2\alpha_q} \cdots \sum_{a_{\pi(2)}=1}^{n} \prod_{\substack{q\in B(H)\\ q=(\pi(1),j)\\ 2\le\pi^{-1}(j)}} \langle v_{a_{\pi(2)}}, M_q v_{a_j}\rangle^{2\alpha_q}. \quad (6)$$

Here we took the product over $t \le \pi^{-1}(j)$ as opposed to $t < \pi^{-1}(j)$ since there may be self-loops.

What we show in the full proof, using that $\sum_{i=1}^{n} u_i u_i^* = I$ and $\forall i\ \|u_i\| \le 1$, is the inequality in Figure 3. That is, if the boxed vertex is the next vertex we sum over according to π (going from right to left in Eq. (6)), then we show Eq. (6) is at most the summation over $\{a_i\}$ for a similar sum but over a new graph H'. To form H', eliminate the boxed vertex and add a self-loop to each of its neighbors (one self-loop to each pair of edges to that neighbor). Each new self-loop has edge-matrix I. By iteratively choosing the next vertex to sum over, one specifies π. Note we do not sum over vertex 1. If π is chosen so that at each step the new graph H' is connected, then the final graph is vertex 1 with some positive number of self-loops, so Eq. (6) is upper bounded by

$$\prod_{j=1}^{t} \|M_{e_j} c\|^2 \le \prod_{j=1}^{t} \|M_{e_j}\|^2 \|c\|^2 \le \|c\|^2$$

for some $t \ge 1$, where $\forall j\ \|M_{e_j}\| \le 1$. Such a π exists: take a spanning tree of H rooted at vertex 1 then sum over vertices in reverse breadth first search order (i.e. from the leaves upward). ■

Lemma 7. *Let $\widehat{G}$ be any dot product graph as in Eq.* (4)*. Then*

$$\left| \sum_{\substack{a_1,\ldots,a_y\in[n]\\ \forall i\neq j\ a_i\neq a_j}} \prod_{\substack{e\in\widehat{G}\\ e=(i,j)}} \langle u_{a_i}, u_{a_j}\rangle \right| \le y! \cdot d^{y-w+1}. \quad (7)$$

Proof (Sketch). In the full proof we first show how to eliminate the restrictions $a_i \neq a_j$ on the left hand side of Eq. (7) at the cost of a multiplicative $y!$. We then wish to show

$$F(\widehat{G}) \stackrel{\text{def}}{=} \left| \sum_{a_1,\ldots,a_y=1}^{n} \prod_{\substack{e\in E(\widehat{G})\\ e=(i,j)}} \langle u_{a_i}, M_e u_{a_j}\rangle \right| \le d^{y-w+1} \quad (8)$$

where $M_e = I$ for all $e \in \widehat{G}$. To upper bound $F(\widehat{G})$, let its connected components be $C_1, \ldots, C_{CC(\widehat{G})}$, where $CC(\cdot)$ counts connected components. We treat $\widehat{G}$ as a generalized dot product multigraph so that each edge e has an associated matrix M_e (though in fact $M_e = I$ for all e). Define an undirected multigraph to be *good* if all its connected components have two edge-disjoint spanning trees. We will show that $F(\widehat{G}) = F(G')$ for some good G' and such that $F(G') \le d^{y-w+1}$. If $\widehat{G}$ itself is good then we set $G' = \widehat{G}$. Otherwise, we will show $F(\widehat{G}) = F(H_0) = \ldots = F(H_\tau)$ for smaller and smaller generalized dot product multigraphs H_t (i.e. with successively fewer vertices) whilst maintaining the invariant that each H_t has Eulerian connected components (a simple argument shows that $\widehat{G}$ itself has Eulerian components) and has $\|M_e\| \le 1$ for all e. We stop when some H_τ is good and we can set $G' = H_\tau$. This iterative process terminates, since every successive H_t has at least one fewer vertex, and when the number of vertices of any connected component drops to 2 or lower then that connected component has two edge-disjoint spanning trees.

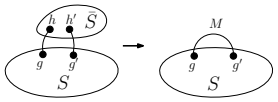

Figure 5. Forming H_t from H_{t-1}.

Let us now focus on constructing this sequence of H_t in the case that $\widehat{G}$ is not good. Let $H_0 = \widehat{G}$. Suppose we have constructed $H_0, \ldots, H_{t-1}$ for $i \ge 1$ none of which are good, and now we want to construct H_t. Since H_{t-1} is not good it cannot be 4-edge-connected by Fact 3, so there is some connected component C_{j^*} of H_{t-1} with some $S \subsetneq V(C_{j^*})$ with 2 edges crossing the cut $(S, \bar{S})$, where $\bar{S}$ represents the complement of S in C_{j^*}. This is because since C_{j^*} is Eulerian, any cut has an even number of edges crossing it. Choose such an $S \subset V(C_{j^*})$ with $|\bar{S}|$ minimum amongst all such cuts. Let the two edges crossing the cut be $(g,h),(g',h')$ with $g, g' \in S$ (note that it may be the case that $g = g'$ or $h = h'$, or both). Note that $F(H_{t-1})$ equals the magnitude of

$$\sum_{\substack{\{a_i\}\\ i\notin C_{j^*}}} \prod_{\substack{e\in H_{t-1}\\ e\notin C_{j^*}\\ e=(i,j)}} \langle u_{a_i}, M_e u_{a_j}\rangle \sum_{\substack{\{a_i\}\\ i\in S}} \left(\prod_{\substack{e\in H_{t-1}\\ e=(i,j)\\ i,j\in S}} \langle u_{a_i}, M_e u_{a_j}\rangle \right) u^*_{a_g} \underbrace{\left(\sum_{\substack{\{a_i\}\\ i\in\bar{S}}} M_{(g,h)} u_{a_h} \left(\prod_{\substack{e\in H_{t-1}\\ e=(i,j)\\ i,j\in\bar{S}}} \langle u_{a_i}, M_e u_{a_j}\rangle \right) u^*_{a_{h'}} M_{(h',g')} \right)}_{M} u_{a_{g'}}. \quad (9)$$

We define H_t to be H_{t-1} but where in the j^*th component we eliminate all the vertices and edges in $\bar{S}$ and add an additional edge from g to g' with edge-matrix M (see Figure 5). We thus have that $F(H_{t-1}) = F(H_t)$. Furthermore each component of H_t is still Eulerian since every vertex in

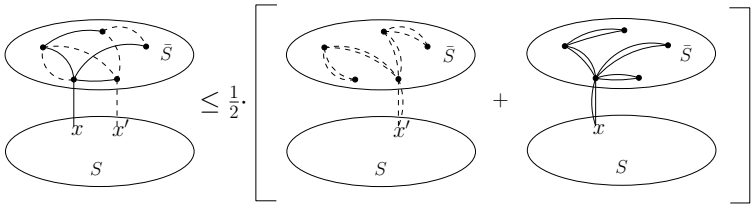

Figure 4. Showing that $\|M\| \leq 1$ by AM-GM on two edge-disjoint spanning subgraphs.

H_{t-1} has either been eliminated, or its edge-degree has been preserved and thus all edge-degrees are even. By iteratively eliminating bad cuts $(S, \bar{S})$ in this way, we eventually arrive at a generalized dot product multigraph H_τ with two edge-disjoint spanning trees in every component.

In the full proof we show that the graph induced on $\bar{S}$ has two edge-disjoint spanning trees, using the minimality of $|\bar{S}|$ and Fact 3. To show $\|M\| \leq 1$ as required by our invariant, by Fact 4 we have $\|M\| = \sup_{\|x\|,\|x'\|=1} x^* M x'$. The product $x^* M x'$ is displayed in Figure 4, where M is the summation over all assignments to vertices in $\bar{S}$ of the edges displayed in the figure (all edges within $\bar{S}$, plus the two edges between S and $\bar{S}$). Since $\bar{S}$ has two edge-disjoint spanning subgraphs, we can partition its edges into two sets (the dashed and solid edges in Figure 4) such that $\bar{S}$ is connected using either edge set. Then by applying the AM-GM inequality as in the figure, we arrive at two connected graphs that each have all even edge multiplicities. We apply Lemma 6 to each graph, where in the first graph $c = x$, and in the second graph $c = x'$. The right hand side of Figure 4 is thus bounded by $(\|x\|^2 + \|x'\|^2)/2 = 1$, showing $\|M\| \leq 1$.

It remains to show that for our final good G' we have $F(G') \leq d^{y-w+1}$. We first show $CC(G') \leq d^{y-w+1}$ then $F(G') \leq d^{CC(G')}$. For the first claim, note $CC(G') = CC(\widehat{G})$ since every H_t has the same number of connected components as $\widehat{G}$. Now, all middle vertices in G lie in one connected component (since G is connected) and $MR(G)$ has w connected components. Thus the at least $w-1$ edges connecting these components in G must come from $LM(G)$, implying that $LM(G)$ (and thus $\widehat{G}$) has at most $y - w + 1$ connected components, and thus $CC(G') = CC(\widehat{G}) \leq y - w + 1$.

It only remains to show $F(G') \leq d^{CC(G')}$. Let G' have connected components $C_1, \ldots, C_{CC(G')}$ with each C_j having 2 edge-disjoint spanning trees (see Figure 6). A simple observation is that $F(H) = \prod_i F(C_i)$, and thus we need only show $F(C_i) \leq d$, where we abuse notation to let C_i denote the generalized dot product multigraph induced on C_i. Label an arbitrary vertex in C_i as vertex 1. Then applying AM-GM after partitioning the edges into two edge-disjoint spanning subgraphs (Figure 6), Lemma 6 gives $F(C_i) \leq \sum_{a_1=1}^{n} \|u_{a_1}\|^2 = \|U\|_F^2 = d$.

■

Some basic estimates discussed in the full proof yield that for any $G \in \mathcal{G}$, we have $y + z \leq b + w$, $b \geq 2z$, $y \leq b \leq \ell$, and that the number of different G with a given b, y, z is at most $(b^3)^\ell / y!$. These estimates can be combined with Lemma 7 and Eq. (4) to show the inequality

$$\mathbb{E}\mathrm{tr}((S-I)^\ell) \leq d\ell^4 \cdot \max_{2 \leq b \leq \ell} \left(\frac{b^3}{s}\right)^{\ell-b} \left(b^3\sqrt{\frac{d}{m}}\right)^b.$$

Define $\epsilon = 2\varepsilon - \varepsilon^2$. For $\ell \geq \ln(d\ell^4/\delta) = O(\ln(d/\delta))$, $s \geq e\ell^3/\epsilon = O(\log(d/\delta)^3/\varepsilon)$, and $m \geq e^2 d\ell^6/\epsilon^2 = O(d\log(d/\delta)^6/\varepsilon^2)$, we then have that $\mathbb{E}tr((S-I)^\ell) \leq \delta\epsilon^\ell$. ■

A change of parameters also yields the following (see full version).

Theorem 8. *Let $\alpha, \gamma > 0$ be arbitrary constants. For Π an OSNAP with $s = \Theta(1/\varepsilon)$ and $\varepsilon \in (0,1)$, with probability at least $1 - 1/d^\alpha$, all singular values of ΠU are $1 \pm \varepsilon$ for $m = \Omega(d^{1+\gamma}/\varepsilon^2)$. The constants in the big-Θ and big-Ω depend on α, γ.*

III. Applications

Here we state the consequences of plugging our OSE's into known previous work for various numerical linear algebra problems [8], [9], [11], [12], [14], [15], [22]. Details can be found in the full version. In the statements of our bounds we implicitly assume $\mathrm{nnz}(A) \geq n$, since otherwise fully zero rows of A can be ignored without affecting the problem solution.

Leverage Scores (reduction to OSE's in [15]):

Theorem 9. *For any constant $\varepsilon > 0$, there is an algorithm that with probability at least $2/3$, approximates all leverage scores of a $n \times d$ matrix A up to $1 \pm \varepsilon$ in time $\tilde{O}(\mathrm{nnz}(A)/\varepsilon^2 + r^\omega \varepsilon^{-2\omega})$.*

Least Squares Regression (reduction to OSE's in [11]):

Theorem 10. *There is an algorithm for $(1+\varepsilon)$-approximate least squares regression running in time $O_\gamma(\mathrm{nnz}(A)/\varepsilon) + O(d^3 \log(d/\varepsilon)/\varepsilon^2)$ and succeeding with probability $2/3$.*

Theorem 11. *Let r be the rank of A. There is an algorithm for $(1+\varepsilon)$-approximate least squares regression running in*

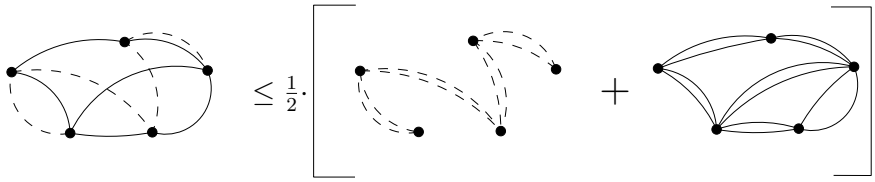

Figure 6. AM-GM on two edge-disjoint spanning subgraphs of one connected component of G'.

time $O(\mathrm{nnz}(A)((\log r)^{O(1)} + \log(n/\varepsilon)) + r^\omega(\log r)^{O(1)} + r^2\log(1/\varepsilon))$ *and succeeding with probability at least* $2/3$.

ℓ_p Regression, $p \in [1,\infty)$ (reduction to OSE's in [9]; see also [11]): Let $\hat{\alpha} = d^{1/p+1/2}, \hat{\beta}_m = O(\max\{1, d^{1/q-1/2}\} \cdot d(m^2d^3)^{|1/p-1/2|})$, where $1/p + 1/q = 1$.

Theorem 12 (follows from [9], [12])**.** *Suppose $A \in \mathbb{R}^{n\times d}$ has rank d. Given an OSE distribution over $\mathbb{R}^{m\times n}$ with column sparsity s, with $\varepsilon = 1/2$ and failure probability $\delta < 1/n$, one can find $\hat{x}' \in \mathbb{R}^d$ in time $O(\mathrm{nnz}(A)(s + \log n) + d^3\log n + \phi(O(2^p d^{1+|p/2-1|}(\hat{\alpha}\hat{\beta}_m)^p, d)) + \phi(O(\varepsilon^{-2}24^p d(\hat{\alpha}\hat{\beta}_m)^p \log(1/\varepsilon)), d)$ satisfying $\|A\hat{x}' - b\|_p \le (1+\varepsilon)\min_x \|Ax - b\|_p$ with probability $1/2$. Here $\phi(n,d)$ is the time to exactly solve an $n \times d$ ℓ_p regression problem, and $\hat{\alpha}, \hat{\beta}_m$ are as above.*

The work [11] plugged their OSE with $m = d^2 \cdot \mathrm{polylog}(n), s = \log n$ into Theorem 12 above (recall $\hat{\beta}_m$ depends on m^2). On the other hand, one obtains improved dependence on d by using our Theorem 5 with $m = d\,\mathrm{polylog}\, n, s = \mathrm{polylog}\, n$. If n, d are polynomially related one can also use Theorem 8 with $m = O(d^{1+\gamma}), s = O_\gamma(1)$ for any $\gamma > 0$. Also see [40] for an improved reduction from ℓ_p regression to OSE's.

Low Rank Approximation (reduction to OSE's in [11]):

Theorem 13. *Given $A \in \mathbb{R}^{n\times n}$, there are 2 algorithms that, with probability at least $3/5$, find 3 matrices U, Σ, V where U is of size $n \times k$, Σ is of size $k \times k$, V is of size $n \times k$, $U^TU = V^TV = I_k$, Σ is a diagonal matrix, and*

$$\|A - U\Sigma V^*\|_F \le (1+\varepsilon)\Delta_k$$

The first algorithm runs in time $O(\mathrm{nnz}(A)) + \tilde{O}(nk^2 + nk^{\omega-1}\varepsilon^{-1-\omega} + k^\omega\varepsilon^{-2-\omega})$. The second algorithm runs in time $O_\gamma(\mathrm{nnz}(A)) + \tilde{O}(nk^{\omega+\gamma-1}\varepsilon^{-1-\omega-\gamma} + k^{\omega+\gamma}\varepsilon^{-2-\omega-\gamma})$ for any constant $\gamma > 0$.

IV. Open Problem

As discussed previously, the work here can be seen as a natural extension of the sparse Johnson Lindenstrauss lemma [22]. Indeed, $\mathcal{D}$ being an OSE means that for all $U \in \mathbb{R}^{n\times d}$ with orthonormal columns,

$$\mathbb{P}_\Pi\left(\|(\Pi U)^*(\Pi U) - I\| > \varepsilon\right) < \delta \qquad (10)$$

The case $d = 1$ corresponds to the Johnson-Lindenstrauss lemma. We state the following conjecture.

Conjecture 14. *Let Π be an OSNAP with $m = \Omega((d + \log(1/\delta))/\varepsilon^2)$, $s = \Omega(\log(d/\delta)/\varepsilon)$. Then Eq.* (10) *holds. In fact, with these parameter settings, it holds that $\varepsilon^{-\ell} \cdot \mathbb{E}\mathrm{tr}(((\Pi U)^*(\Pi U) - I)^\ell) < \delta$ for $\ell = \Theta(\log(d/\delta))$ an even integer, so that the analysis can follow the moment method.*

Conjecture 14 is true for $d = 1$ [22], and for larger d holds up to $\log^{O(1)}(d/\delta)$ factors in m and s by Theorem 5. We remark that it is even open to resolve Conjecture 14 in the dense case of $s = m$ by using the moment method.

Acknowledgments

We thank Andrew Drucker for suggesting the SNAP acronym for our OSE's, to which we added the "oblivious" descriptor. This work was done while JN was a member at the Institute for Advanced Study, supported by NSF CCF-0832797 and NSF DMS-1128155. HN was supported by NSF CCF-0832797 and a Gordon Wu fellowship.

References

[1] D. Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *J. Comput. Syst. Sci.*, 66(4):671–687, 2003.

[2] N. Ailon and B. Chazelle. The Fast Johnson–Lindenstrauss transform and approximate nearest neighbors. *SIAM J. Comput.*, 39(1):302–322, 2009.

[3] N. Ailon and E. Liberty. Fast dimension reduction using Rademacher series on dual BCH codes. *Discrete Comput. Geom.*, 42(4):615–630, 2009.

[4] N. Ailon and E. Liberty. Almost optimal unrestricted fast Johnson-Lindenstrauss transform. In *SODA*, pages 185–191, 2011.

[5] Z. Bai and Y. Yin. Limit of the smallest eigenvalue of a large dimensional sample covariance matrix. *Ann. Probab.*, 21(3):1275–1294, 1993.

[6] V. Braverman, R. Ostrovsky, and Y. Rabani. Rademacher chaos, random Eulerian graphs and the sparse Johnson-Lindenstrauss transform. *CoRR*, abs/1011.2590, 2010.

[7] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[8] H. Y. Cheung, T. C. Kwok, and L. C. Lau. Fast matrix rank algorithms and applications. In *STOC*, pages 549–562, 2012.

[9] K. Clarkson, P. Drineas, M. Magdon-Ismail, M. Mahoney, X. Meng, and D. Woodruff. The fast Cauchy transform and faster robust linear regression. In *SODA*, pages 466–477, 2013.

[10] K. L. Clarkson and D. P. Woodruff. Numerical linear algebra in the streaming model. In *STOC*, pages 205–214, 2009.

[11] K. L. Clarkson and D. P. Woodruff. Low rank approximation and regression in input sparsity time. In *STOC*, pages 81–90, 2013.

[12] A. Dasgupta, P. Drineas, B. Harb, R. Kumar, and M. W. Mahoney. Sampling algorithms and coresets for ℓ_p regression. *SIAM J. Comput.*, 38(5):2060–2078, 2009.

[13] A. Dasgupta, R. Kumar, and T. Sarlós. A sparse Johnson-Lindenstrauss transform. In *STOC*, pages 341–350, 2010.

[14] J. Demmel, I. Dumitriu, and O. Holtz. Fast linear algebra is stable. *Numer. Math.*, 108(1):59–91, Oct. 2007.

[15] P. Drineas, M. Magdon-Ismail, M. Mahoney, and D. Woodruff. Fast approximation of matrix coherence and statistical leverage. In *ICML*, 2012.

[16] Z. Füredi and J. Komlós. The eigenvalues of random symmetric matrices. *Combinatorica*, 1(3):233–241, 1981.

[17] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev., Survey and Review section*, 53(2):217–288, 2011.

[18] D. L. Hanson and F. T. Wright. A bound on tail probabilities for quadratic forms in independent random variables. *Ann. Math. Statist.*, 42(3):1079–1083, 1971.

[19] A. Hinrichs and J. Vybíral. Johnson-Lindenstrauss lemma for circulant matrices. *Random Struct. Algorithms*, 39(3):391–398, 2011.

[20] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[21] D. M. Kane and J. Nelson. A derandomized sparse Johnson-Lindenstrauss transform. *CoRR*, abs/1006.3585, 2010.

[22] D. M. Kane and J. Nelson. Sparser Johnson-Lindenstrauss transforms. In *SODA*, pages 1195–1206, 2012.

[23] D. M. Kane, J. Nelson, E. Porat, and D. P. Woodruff. Fast moment estimation in data streams in optimal space. In *STOC*, pages 745–754, 2011.

[24] F. Krahmer, S. Mendelson, and H. Rauhut. Suprema of chaos processes and the restricted isometry property. *Comm. Pure Appl. Math.*, to appear.

[25] F. Krahmer and R. Ward. New and improved Johnson-Lindenstrauss embeddings via the Restricted Isometry Property. *SIAM J. Math. Anal.*, 43(3):1269–1281, 2011.

[26] M. Li, G. L. Miller, and R. Peng. Iterative row sampling. In *FOCS*, 2013.

[27] M. W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.

[28] X. Meng and M. W. Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *STOC*, pages 91–100, 2013.

[29] J. Nelson and H. L. Nguyễn. Lower bounds for oblivious subspace embeddings. *CoRR*, abs/1308.3280, 2013.

[30] J. Nelson and H. L. Nguyễn. Sparsity lower bounds for dimensionality-reducing maps. In *Proceedings of the 45th ACM Symposium on Theory of Computing (STOC)*, pages 101–110, 2013.

[31] N. H. Nguyen, T. T. Do, and T. D. Tran. A fast and efficient algorithm for low-rank approximation of a matrix. In *STOC*, pages 215–224, 2009.

[32] T. Sarlós. Improved approximation algorithms for large matrices via random projections. In *FOCS*, pages 143–152, 2006.

[33] T. Tao. *Topics in random matrix theory*, volume 132 of *Graduate Studies in Mathematics*. American Mathematical Society, 2012.

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

[35] J. A. Tropp. Improved analysis of the subsampled randomized Hadamard transform. *Adv. Adapt. Data Anal., Special Issue on Sparse Representation of Data and Images*, 3(1–2):115–126, 2011.

[36] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Y. Eldar and G. Kutyniok, editors, *Compressed Sensing, Theory and Applications*, chapter 5, pages 210–268. Cambridge University Press, 2012.

[37] J. Vybíral. A variant of the Johnson-Lindenstrauss lemma for circulant matrices. *J. Funct. Anal.*, 260(4):1096–1105, 2011.

[38] E. P. Wigner. Characteristic vectors of bordered matrices with infinite dimensions. *Ann. Math.*, 62:548–564, 1955.

[39] V. V. Williams. Multiplying matrices faster than Coppersmith-Winograd. In *STOC*, pages 887–898, 2012.

[40] D. P. Woodruff and Q. Zhang. Subspace embeddings and ℓ_p-regression using exponential random variables. In *COLT*, 2013.

[41] Y. Zhou, D. M. Wilkinson, R. Schreiber, and R. Pan. Large-scale parallel collaborative filtering for the Netflix prize. In *AAIM*, pages 337–348, 2008.