Acta Numerica (2017), pp. 95–135
doi:10.1017/S0962492917000058

Randomized algorithms in numerical linear algebra

Ravindran Kannan
Microsoft Research Labs, Bangalore,
Karnataka 560001, India
E-mail: kannan@microsoft.com

Santosh Vempala
Georgia Institute of Technology,
North Avenue NW, Atlanta, GA 30332, USA
E-mail: vempala@gatech.edu

This survey provides an introduction to the use of randomization in the design of fast algorithms for numerical linear algebra. These algorithms typically examine only a subset of the input to solve basic problems approximately, including matrix multiplication, regression and low-rank approximation. The survey describes the key ideas and gives complete proofs of the main results in the field. A central unifying idea is sampling the columns (or rows) of a matrix according to their squared lengths.

CONTENTS

1. Introduction

Algorithms for matrix multiplication, low-rank approximations, singular value decomposition, dimensionality reduction and other compressed representations of matrices, linear regression, *etc.*, are widely used. For modern data sets, these computations take too much time and space to perform on the entire input matrix. Instead one can pick a random subset of columns (or rows) of the input matrix. If s (for sample size) is the number of columns we are willing to work with, we execute s statistically independent identical trials, each selecting a column of the matrix. Sampling uniformly at random (u.a.r.) is not always good, for example when only a few columns are significant.

Using sampling probabilities proportional to squared lengths of columns (henceforth called 'length-squared sampling') leads to many provable error bounds. If the input matrix A has n columns, we define[1]

$$p_j = \frac{|A(:,j)|^2}{\|A\|_F^2}, \quad \text{for } j = 1, 2, \ldots, n,$$

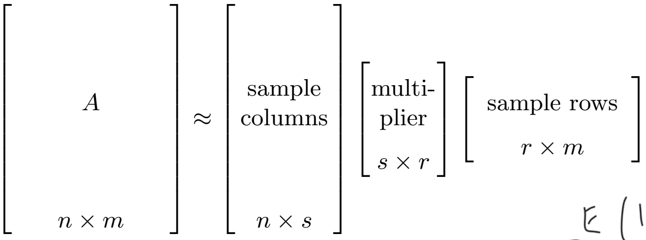
and in each trial, pick a random $X \in \{1, 2, \ldots, n\}$, with $\Pr(X = j) = p_j$.

We will prove error bounds of the form $\varepsilon\|A\|_F^2$, provided s grows as a function of $1/\varepsilon$ (s is independent of the size of the matrix) for *all* matrices. So, the guarantees are *worst-case* bounds rather than average-case bounds. They are most useful when $\|A\|_2^2/\|A\|_F^2$ is not too small, as is indeed the case for the important topic of principal component analysis (PCA). The algorithms are randomized (*i.e.*, they use a random number generator) and hence errors are random variables. We bound the expectations or tail probabilities of the errors. In this paper, we strike a compromise between readability and comprehensive coverage by presenting full proofs of conceptually central theorems and stating stronger results without proofs.

1.1. Overview of the paper

The first problem we consider is computing the matrix product AA^T. Given as input an $m \times n$ matrix A, we select a (random) subset of s columns of A (in s independent identical trials). We then scale the selected columns and form an $m \times s$ matrix C. We wish to satisfy two conditions: (i) (each entry of) CC^T is an unbiased estimator of (the corresponding entry of) AA^T, and (ii) the sum of the variances of all entries of CC^T (which we refer to as

[1] All matrices in this paper have real entries. We use the usual norms for matrices: the Frobenius norm ($\|\cdot\|_F$) and the spectral norm ($\|\cdot\|_2$). We also use standard MATLAB colon notation, *i.e.*, $A(:,j)$ is the jth column of A; see Golub and Van Loan (1996, Section 1.1.8).



$E\left(\|AAT\cdot CC^T\|_F\right)^2 \leq$

Figure 1.1. Approximating A by a sample of s columns and r rows.

$\varphi(E[x])$
$\leq E[\varphi(x)]$
for convex
φ

the ‘variance’) is at most $\varepsilon^2\|A\|_F^4$. Formally,

$$E(CC^T) = AA^T, \quad E(\|AA^T - CC^T\|_F^2) \leq \varepsilon^2\|A\|_F^4. \tag{1.1}$$

Note that (1.1) implies $E(\|AA^T - CC^T\|_F) \leq \varepsilon\|A\|_F^2$ by Jensen’s inequality.

(you are weighting $\varphi(x_i)$'s v.s. weighting x_i's)

The starting point of sampling-based matrix algorithms was the discovery of length-squared sampling by Frieze, Kannan and Vempala (1998, 2004), motivated by low-rank approximation. We start with two properties of length-squared sampling, which will be proved in Theorem 2.1.

- Length-squared sampling minimizes variance among all unbiased estimators.
- A sample of size $s = 1/\varepsilon^2$ suffices to ensure (1.1). In particular, s is independent of m and n.

Length-squared sampling achieves similar error bounds for general matrix multiplication, matrix sketches, low-rank approximation, *etc.* The main result in matrix sketching is as follows. For any $m \times n$ input matrix A, form an $m \times s$ sub-matrix C of the columns of A and an $r \times n$ sub-matrix R of the rows of A, both picked using length-squared sampling. We can then compute an $s \times r$ matrix U so that

$$E(\|A - CUR\|_2^2) \leq \varepsilon\|A\|_F^2$$

provided $r \geq c/\varepsilon^2, s \geq c/\varepsilon^3$. This is proved in Theorem 2.5. A schematic diagram of C, U and R is shown in Figure 1.1.

Section 2.3 shows a result based purely on matrix perturbation theory (no probabilities involved) which in words states: If C and A are any two matrices with $CC^T \approx AA^T$ (C, A may be of different dimensions), then the restriction of A to the space spanned by the top k singular vectors of C is a good approximation to A. Used in conjunction with (1.1), this reduces the problem of low-rank approximation of A to computing the singular value decomposition (SVD) of a submatrix C.

PCA on LS sample of A

In Section 3 we apply length-squared sampling to tensors (higher-dimensional arrays) to obtain good low-rank tensor approximation. Unlike matrices, finding the best low-rank approximation for tensors is NP-hard.

Low-rank tensor approximation has several applications, and we show that there is a natural class of tensors (which includes dense hypergraphs and generalizations of metrics) for which, after scaling, $\|A\|_2 = \Omega(\|A\|_F)$. Low-rank approximation based on length-squared sampling yields good results.

In Sections 4, 5 and 6 we turn to more refined error bounds for matrices. Can we improve the right-hand side of (1.1) if we want to bound only the spectral norm instead of the Frobenius norm? Using deep results from functional analysis, Rudelson and Vershynin (2007) showed that if we use length-squared sampling,

$$\|CC^T - AA^T\|_2 \leq \varepsilon\|A\|_2\|A\|_F \text{ holds with high probability}$$
$$\text{provided } s \geq \frac{c\ln(1/\varepsilon)}{\varepsilon^2}. \tag{1.2}$$

While the change from Frobenius to spectral norm was first considered for its mathematical appeal, it is also suitable for the Hoeffding–Chernoff-type inequality for matrix-valued random variables, proved by Ahlswede and Winter (2002). We present this theorem and proof (Theorem 4.1), since it is of independent interest. In Section 4 we prove the result of Rudelson and Vershynin.

A central question in probability is to determine how many i.i.d. samples from a probability distribution suffice to make the empirical estimate of the variance close to the true variance in every direction. Mathematically, there is an $n \times m$ matrix A (with $m > n$, m possibly infinite) such that AA^T is the covariance matrix, and for any unit vector $\mathbf{x} \in \mathbb{R}^n$, the variance of the distribution in direction $\mathbf{x}$ is $\mathbf{x}^T AA^T \mathbf{x} = |\mathbf{x}^T A|^2$. The problem is to find a (small) sample of s columns of A to form an $n \times s$ matrix C so that the variance is (approximately) preserved, to relative error ε, in every direction $\mathbf{x}$. That is, we wish to satisfy, for all $\mathbf{x}$,

$$\|\mathbf{x}^T C\|^2 \in \big[(1-\varepsilon)\|\mathbf{x}^T A\|^2,\ (1+\varepsilon)\|\mathbf{x}^T A\|^2\big], \text{ denoted } \|\mathbf{x}^T C\|^2 \cong_\varepsilon \|\mathbf{x}^T A\|^2. \tag{1.3}$$

It is important that (1.3) holds simultaneously for all $\mathbf{x}$. It turns out that, in general, if we sample columns of A with probability proportional to the squared length not of its own columns but of the columns of A^+A, where A^+ is the pseudo-inverse of A, then (1.3) follows from (1.2). We call this 'preconditioned' length-squared sampling (since multiplying A by A^+ can be thought of as preconditioning).

There is another, seemingly unrelated context, where exactly the same question (1.3) arises, namely, graph sparsification, considered by Spielman and Srivastava (2011). Here, A is the node–edge signed incidence matrix of a graph, and the goal is to find a subset of edges satisfying (1.3). Spielman and Srivastava (2011) showed in this case that the squared length of the columns of A^+A are proportional to the (weighted) electrical resistances and can also be computed in linear time (in the number of edges of the graph).

In theoretical computer science, preconditioned length-squared sampling (also called leverage score sampling) arose from a different motivation: Can the additive error guarantee (1.1) be improved to a relative error guarantee, perhaps with more sophisticated sampling methods? Several answers have been given to this question. The first is simply to iterate length-squared sampling on the residual matrix in the space orthogonal to the span of the sample chosen so far (Deshpande, Rademacher, Vempala and Wang 2006). Another is to use preconditioned length-squared sampling, where the preconditioning effectively makes the sampling probabilities proportional to the *leverage scores* (Drineas, Mahoney and Muthukrishnan 2008). A third is *volume sampling*, which picks a subset of k columns with probabilities proportional to the square of the volume of the k-simplex they span, together with the origin (Deshpande and Vempala 2006, Deshpande and Rademacher 2010, Anari, Gharan and Rezaei 2016). We discuss preconditioned length-squared sampling in Section 5.

In Section 6 we consider an approach pioneered by Clarkson and Woodruff (2009, 2013) for obtaining similar relative-error approximations but in input sparsity time, *i.e.*, time that is asymptotically linear in the number of non-zeros of the input matrix. This is possible via a method known as subspace embedding, which can be performed in linear time using the sparsity of the matrix. We first discuss an inefficient method using random projection, then an efficient method due to Clarkson and Woodruff (2013) based on a sparse projection.

As the title of this article indicates, we focus here on randomized algorithms with linear algebra as the important application area. For treatments from a linear algebra perspective (with randomized algorithms as one of the tools), the reader might consult Halko, Martinsson and Tropp (2011), Woodruff (2014) and references therein.

2. Basic algorithms

2.1. Matrix multiplication using sampling

Suppose A is an $m \times n$ matrix and B is an $n \times p$ matrix, and the product AB is desired. We can use sampling to get an approximate product faster than the traditional multiplication. Let $A(:,k)$ denote the kth column of A. $A(:,k)$ is an $m \times 1$ matrix. Let $B(k,:)$ be the kth row of B. $B(k,:)$ is a $1 \times n$ matrix. We have

$$AB = \sum_{k=1}^{n} A(:,k)B(k,:).$$

$$\begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} \begin{bmatrix} b_1 & \cdots & b_n \end{bmatrix}$$

$$(ab)_{ij} = \sum_k a_{ik} b_{kj}$$

Can we estimate the sum over all k by summing over a sub-sample? Let $p_1, p_2, \ldots, p_n$ be non-negative reals summing to 1 (to be determined later). Let z be a random variable that takes values in $\{1, 2, \ldots, n\}$ with $\Pr(z = j) = p_j$. Define an associated matrix random variable X such that

$$\Pr\left(X = \frac{1}{p_k} A(:,k)B(k,:)\right) = p_k. \tag{2.1}$$

Let $E(X)$ denote the entry-wise expectation, that is,

$$E(X) = \sum_{k=1}^{n} \Pr(z = k)\frac{1}{p_k} A(:,k)B(k,:) = \sum_{k=1}^{n} A(:,k)B(k,:) = AB.$$

The scaling by $1/p_k$ makes X an unbiased estimator of AB. We will be interested in $E(\|AB - X\|_F^2)$, which is just the sum of the variances of all entries of X,[2] since

$$\begin{aligned} E(\|AB - X\|_F^2) &= \sum_{i=1}^{m}\sum_{j=1}^{p} \mathrm{Var}(x_{ij}) = \sum_{ij} E(x_{ij}^2) - E(x_{ij})^2 \\ &= \left(\sum_{ij}\sum_{k} p_k \frac{1}{p_k^2} a_{ik}^2 b_{kj}^2\right) - \|AB\|_F^2. \end{aligned}$$

We can ignore the $\|AB\|_F^2$ term since it does not depend on the p_k. Now

$$\begin{aligned} \sum_{ij}\sum_{k} p_k \frac{1}{p_k^2} a_{ik}^2 b_{kj}^2 &= \sum_k \frac{1}{p_k}\left(\sum_i a_{ik}^2\right)\left(\sum_j b_{kj}^2\right) \\ &= \sum_k \frac{1}{p_k} |A(:,k)|^2 |B(k,:)|^2. \end{aligned}$$

It can be seen by calculus[3] that the minimizing p_k must be proportional to $|A(:,k)||B(k,:)|$. In the important special case when $B = A^T$, this means picking columns of A with probabilities proportional to the squared length of the columns. In fact, even in the general case when $B \neq A^T$, doing so simplifies the bounds, so we will use it. If p_k is proportional to $|A(:,k)|^2$, that is,

$$p_k = \frac{|A(:,k)|^2}{\|A\|_F^2},$$

then

$$E(\|AB - X\|_F^2) = \mathrm{Var}(X) \leq \|A\|_F^2 \sum_k |B(k,:)|^2 = \|A\|_F^2 \|B\|_F^2.$$

[2] We use a_{ij} to denote entries of matrix A.

[3] For any non-negative c_k, minimizing $\sum_k c_k p_k^{-1}$ subject to $\sum_k p_k = 1$ via Lagrange multipliers implies that p_k is proportional to $\sqrt{c_k}$.

To reduce the variance, we can take the average of s independent trials. Each trial i, $i = 1, 2, \ldots, s$ yields a matrix X_i as in (2.1). We take

$$\frac{1}{s}\sum_{i=1}^{s} X_i$$

as our estimate of AB. Since the variance of the sum of independent random variables is the sum of the variances, we find that

$$\operatorname{Var}\left(\frac{1}{s}\sum_{i=1}^{s} X_i\right) = \frac{1}{s}\operatorname{Var}(X) \leq \frac{1}{s}\|A\|_F^2\|B\|_F^2.$$

Let $k_1, \ldots, k_s$ be the k chosen in each trial. Expanding this, we obtain

$$\frac{1}{s}\sum_{i=1}^{s} X_i = \frac{1}{s}\left(\frac{A(:,k_1)B(k_1,:)}{p_{k_1}} + \frac{A(:,k_2)B(k_2,:)}{p_{k_2}} + \cdots + \frac{A(:,k_s)B(k_s,:)}{p_{k_s}}\right). \tag{2.2}$$

We write this as the product of an $m \times s$ matrix with an $s \times p$ matrix as follows. Let C be the $m \times s$ matrix consisting of the following columns, which are scaled versions of the chosen columns of A:

$$\frac{A(:,k_1)}{\sqrt{sp_{k_1}}}, \quad \frac{A(:,k_2)}{\sqrt{sp_{k_2}}}, \ldots, \frac{A(:,k_s)}{\sqrt{sp_{k_s}}}.$$

Note that this scaling has a nice property, which we leave to the reader to verify:

$$E(CC^T) = AA^T. \tag{2.3}$$

Define R to be the $s \times p$ matrix with the corresponding rows of B similarly scaled, namely, R has rows

$$\frac{B(k_1,:)}{\sqrt{sp_{k_1}}}, \quad \frac{B(k_2,:)}{\sqrt{sp_{k_2}}}, \ldots, \frac{B(k_s,:)}{\sqrt{sp_{k_s}}}.$$

The reader may also verify that

$$E(R^TR) = A^TA. \tag{2.4}$$

From (2.2), we see that

$$\frac{1}{s}\sum_{i=1}^{s} X_i = CR.$$

This is represented in Figure 2.1. We summarize our discussion in Theorem 2.1.

Theorem 2.1. Suppose A is an $m\times n$ matrix and B is an $n\times p$ matrix. The product AB can be estimated by CR, where C is an $m \times s$ matrix consisting of s columns of A picked according to length-squared distribution and scaled

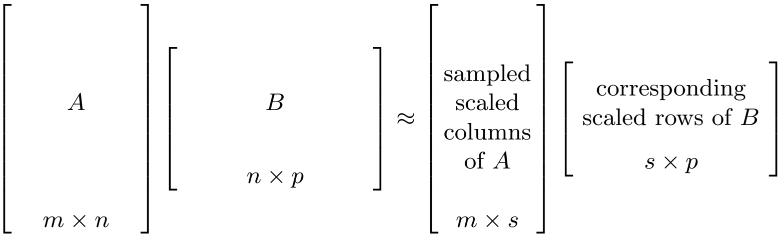


Figure 2.1. Approximate matrix multiplication using sampling.

to satisfy (2.3), and R is the $s \times p$ matrix consisting of the corresponding rows of B scaled to satisfy (2.4). The error is bounded by

$$E(\|AB - CR\|_F^2) \leq \frac{\|A\|_F^2 \, \|B\|_F^2}{s}.$$

Thus, to ensure that

$$E(\|AB - CR\|_F^2) \leq \varepsilon^2 \|A\|_F^2 \|B\|_F^2,$$

it suffices to choose $s \geq 1/\varepsilon^2$. If now $\varepsilon = \Omega(1)$ (and so $s = O(1)$), then the multiplication CR can be carried out in time $O(mp)$.

When is this the error bound useful? Let us focus on the case $B = A^T$ so that we have just one matrix to consider. If A is the identity matrix, then the guarantee is *not* very good. In this case, $\|AA^T\|_F^2 = n$, but the right-hand side of the inequality is n^2/s. So we would need $s > n$ for the bound to be any better than approximating the product with the zero matrix.

More generally, the trivial estimate of the zero matrix for AA^T makes an error in the Frobenius norm of $\|AA^T\|_F$. If $\sigma_1, \sigma_2, \ldots$ are the singular values of A, then the singular values of AA^T are $\sigma_1^2, \sigma_2^2, \ldots$, and we have

$$\|A\|_F^2 = \sum_t \sigma_t^2 \quad \text{and} \quad \|AA^T\|_F^2 = \sum_t \sigma_t^4.$$

So from the theorem we can assert

$$E(\|AA^T - CR\|_F^2) \leq \|AA^T\|_F^2$$

provided that

$$s \geq \frac{(\sigma_1^2 + \sigma_2^2 + \cdots)^2}{\sigma_1^4 + \sigma_2^4 + \cdots}.$$

If $\operatorname{rank}(A) = r$, then there are r non-zero σ_t and the best general upper bound on the ratio $(\sigma_1^2 + \sigma_2^2 + \cdots)^2/(\sigma_1^4 + \sigma_2^4 + \cdots)$ is r, so in general, s needs to be at least r. If A is of full rank, this means that sampling will not give us any gain over taking the whole matrix!

However, if there is a constant c and a small integer p such that

$$\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_p^2 \geq c(\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_r^2), \tag{2.5}$$

then

$$\frac{(\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_r^2)^2}{\sigma_1^4 + \sigma_2^4 + \cdots + \sigma_r^4} \leq \frac{1}{c^2} \frac{(\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_p^2)^2}{\sigma_1^4 + \sigma_2^4 + \cdots + \sigma_p^2} \leq \frac{p}{c^2},$$

and so $s \geq p/c^2$ gives us a better estimate than the zero matrix. Further increasing s by a factor decreases the error by the same factor. The condition (2.5) (in words, the top p singular values make up a constant fraction of the spectrum) is indeed the hypothesis of the subject of principal component analysis, and there are many situations when the data matrix does satisfy the condition and so sampling algorithms are useful.

2.1.1. Implementing length-squared sampling in two passes

Traditional matrix algorithms often assume that the input matrix is in random access memory (RAM) and so any particular entry of the matrix can be accessed in unit time. For massive matrices, RAM may be too small to hold the entire matrix, but may be able to hold and compute with the sampled columns/rows.

Let us consider a high-level model where the input matrix or matrices have to be read from 'external memory' using a 'pass'. In one pass, we can read sequentially all entries of the matrix in some order. We may do some 'sampling on the fly' as the pass is going on.

It is easy to see that two passes suffice to draw a sample of columns of A according to length-squared probabilities, even if the matrix is not in row order or column order and entries are presented as a linked list (as in sparse representations). In the first pass, we just compute the squared length of each column and store this information in RAM. The squared lengths can be computed as running sums. Then, we use a random number generator in RAM to figure out the columns to be sampled (according to length-squared probabilities). Then, we make a second pass in which we pick out the columns to be sampled.

What if the matrix is already presented in external memory in column order? In this case, one pass will do, based on a primitive using rejection sampling.

The primitive is as follows. We are given a stream (*i.e.* a read-once only input sequence) of positive real numbers $a_1, a_2, \ldots, a_n$. We seek to have a random $i \in \{1, 2, \ldots, n\}$ at the end, with the property that the probability of choosing i is exactly equal to $a_i / \sum_{j=1}^n a_j$, for all i. This is solved as follows. After having read $a_1, a_2, \ldots, a_i$, suppose we have (i) $\sum_{j=1}^i a_j$ and (ii) a sample $a_j, j \leq i$, picked with probability $a_j / \sum_{k=1}^i a_k$. On reading a_{i+1}, we update the sum and, with the correct probability, reject the earlier sample and replace it with a_{i+1}. If we need s independent identical samples, we just run s such processes in parallel.

2.2. Sketch of a large matrix

The main result of this section is that for any matrix, a sample of columns and rows, each picked according to length-squared distribution, provides a good sketch of the matrix in a formal sense that will be described briefly. Let A be an $m \times n$ matrix. Pick s columns of A according to length-squared distribution. Let C be the $m \times s$ matrix containing the picked columns scaled so as to satisfy (2.3), that is, if $A(:, k)$ is picked, it is scaled by $1/\sqrt{sp_k}$. Similarly, pick r rows of A according to length-squared distribution on the rows of A. Let R be the $r \times n$ matrix of the picked rows, scaled as follows. If row k of A is picked, it is scaled by $1/\sqrt{rp_k}$. We then have $E(R^TR) = A^TA$. From C and R, we can find a matrix U so that $A \approx CUR$.

One may recall that the top k singular vectors give a similar picture. But the SVD takes more time to compute, requires all of A to be stored in RAM, and does not have the property that the singular vectors, the basis of the reduced space, are directly from A. The last property – that the approximation involves actual rows/columns of the matrix rather than linear combinations – is called an *interpolative approximation*, and is useful in many contexts. Some structural results of such approximations are found in the work of Stewart (Stewart 1999, Stewart 2004, Berry, Pulatova and Stewart 2004) and Goreinov, Tyrtyshnikov and Zamarashkin (Goreinov, Tyrtyshnikov and Zamarashkin 1997, Goreinov and Tyrtyshnikov 2001).

We briefly mention two motivations for such a sketch. Suppose A is the document-term matrix of a large collection of documents. We are to 'read' the collection at the outset and store a sketch so that later, when a query represented by a vector with one entry per term arrives, we can find its similarity to each document in the collection. Similarity is defined by the dot product. In Figure 1.1 it is clear that the matrix–vector product of a query with the right-hand side can be done in time $O(ns + sr + rm)$, which would be linear in n and m if s and r are $O(1)$. To bound errors for this process, we need to show that the difference between A and the sketch of A has small 2-norm. The fact that the sketch is an interpolative approximation means that our approximation essentially consists a subset of documents and a subset of terms, which may be thought of as a representative set of documents and terms. Moreover, if A is *sparse* in its rows and columns – each document contains only a small fraction of the terms and each term is in only a small fraction of the documents – then this property will be preserved in C and R, unlike with the SVD.

A second motivation comes from recommendation systems. Here A would be a customer–product matrix whose (i, j)th entry is the preference of customer i for product j. The objective is to collect a few sample entries of A and, based on these, get an approximation to A so that we can make future recommendations. A few sampled rows of A (all preferences of a few customers) and a few sampled columns (all customer preferences for a few products) give a good approximation to A provided that the samples are drawn according to the length-squared distribution.

It now remains to describe how to find U from C and R. There is an $n \times n$ matrix P of the form $P = QR$ which acts as the identity on the space spanned by the rows of R and zeros out all vectors orthogonal to this space. The matrix Q is just the pseudo-inverse of R.

Lemma 2.2. If $\operatorname{rank}(R) = r'$ and $R = \sum_{t=1}^{r'} \sigma_t \mathbf{u}_t \mathbf{v}_t^T$ is the SVD of R, then the matrix $P = \bigl(\sum_{t=1}^{r'} {\sigma_t}^{-1} \mathbf{v}_t \mathbf{u}_t^T\bigr) R$ satisfies:

(i) $P\mathbf{x} = \mathbf{x}$ for every vector $\mathbf{x}$ of the form $\mathbf{x} = R^T\mathbf{y}$,

(ii) if $\mathbf{x}$ is orthogonal to the row space of R, then $P\mathbf{x} = \mathbf{0}$,

(iii) $P = R^T \bigl(\sum_{t=1}^{r'} {\sigma_t}^{-2} \mathbf{u}_t \mathbf{u}_t^T\bigr) R$.

We begin with some intuition. In particular, we first present a simpler idea that does not work but will then motivate the idea that *does* work. Write A as AI, where I is the $n \times n$ identity matrix. Now, let us approximate the product AI using the algorithm of Theorem 2.1 from the previous section, that is, by sampling s columns of A according to their squared lengths. Then, as in the last section, write $AI \approx CW$, where W consists of a scaled version of the s rows of I corresponding to the s columns of A that were picked. Theorem 2.1 bounds the error $\|A - CW\|_F^2$ by

$$\|A\|_F^2 \|I\|_F^2 / s = \|A\|_F^2 \frac{n}{s}.$$

But we would like the error to be a small fraction of $\|A\|_F^2$ which would require $s \geq n$, which clearly is of no use since this would pick at least as many columns as the whole of A.

Instead, let us use the identity-like matrix P instead of I in the above discussion. Using the fact that R is picked according to the squared length, we will show the following proposition later.

Proposition 2.3. $A \approx AP$ with error $E(\|A - AP\|_2^2) \leq \|A\|_F^2 / \sqrt{r}$.

We then use Theorem 2.1 to argue that instead of doing the multiplication AP, we can use the sampled columns of A and the corresponding rows of P. The sampled s columns of A form C. We have to take the corresponding s rows of the matrix

$$P = R^T \left(\sum_{t=1}^{r'} \frac{1}{\sigma_t^2} \mathbf{u}_t \mathbf{u}_t^T \right) R.$$

This is the same as taking the corresponding s rows of R^T and multiplying this by $\left(\sum_{t=1}^{r'} \sigma_t{}^{-2}\mathbf{u}_t\mathbf{u}_t^T\right)R$. It is easy to check that this leads to an expression of the form CUR. Moreover, by Theorem 2.1, the error is bounded by

$$E(\|AP - CUR\|_2^2) \le E(\|AP - CUR\|_F^2) \le \frac{\|A\|_F^2\|P\|_F^2}{s} \le \frac{r}{s}\|A\|_F^2, \quad (2.6)$$

since we will deduce the following inequality.

Proposition 2.4. $\|P\|_F^2 \le r$.

Combining (2.6) and Proposition 2.3, and using the fact that by the triangle inequality we have

$$\|A - CUR\|_2 \le \|A - AP\|_2 + \|AP - CUR\|_2,$$

which in turn implies that

$$\|A - CUR\|_2^2 \le 2\|A - AP\|_2^2 + 2\|AP - CUR\|_2^2,$$

we obtain the main result.

Theorem 2.5. Suppose A is any $m \times n$ matrix and r and s are positive integers. Suppose C is an $m \times s$ matrix of s columns of A picked according to length-squared sampling, and similarly R is a matrix of r rows of A picked according to length-squared sampling. Then we can find from C, R an $s \times r$ matrix U such that

$$E(\|A - CUR\|_2^2) \le \|A\|_F^2\left(\frac{2}{\sqrt{r}} + \frac{2r}{s}\right).$$

We see that if s is fixed, the error is minimized when $r = (s/2)^{2/3}$. Choosing $s = O(r/\varepsilon)$ and $r = 1/\varepsilon^2$, the bound becomes $O(\varepsilon)\|A\|_F^2$.

Now we prove Proposition 2.3. First,

$$\|A - AP\|_2^2 = \max_{\{\mathbf{x}:|\mathbf{x}|=1\}} |(A - AP)\mathbf{x}|^2.$$

Let us first suppose that $\mathbf{x}$ is in the row space V of R. We have $P\mathbf{x} = \mathbf{x}$, so for $\mathbf{x} \in V$ we have $(A - AP)\mathbf{x} = \mathbf{0}$. Now, since every vector can be written as the sum of a vector in V plus a vector orthogonal to V, this implies that the maximum must therefore occur at some $\mathbf{x} \in V^\perp$. For such $\mathbf{x}$, we have $(A - AP)\mathbf{x} = A\mathbf{x}$. Thus, the question now becomes: For unit-length $\mathbf{x} \in V^\perp$, how large can $\|A\mathbf{x}\|^2$ be? To analyse this, we can write

$$\begin{aligned}\|A\mathbf{x}\|^2 &= \mathbf{x}^T A^T A\mathbf{x} = \mathbf{x}^T(A^T A - R^T R)\mathbf{x}\\ &\le \|A^T A - R^T R\|_2|\mathbf{x}|^2 \le \|A^T A - R^T R\|_2.\end{aligned}$$

This implies that we get $\|A - AP\|_2^2 \le \|A^T A - R^T R\|_2$. So, it suffices to prove that

$$E(\|A^T A - R^T R\|_2^2) \le \|A\|_F^4/r,$$

which follows directly from Theorem 2.1, since we can think of $R^T R$ as a way of estimating $A^T A$ by picking (according to length-squared distribution) columns of A^T, that is, rows of A. By Jensen's inequality, this implies that

$$E(\|A^T A - R^T R\|_2) \leq \frac{\|A\|_F^2}{\sqrt{r}}$$

and proves Proposition 2.3.

Proposition 2.4 is easy to see. Since, by Lemma 2.2, P is the identity on the space V spanned by the rows of R, and $P\mathbf{x} = 0$ for $\mathbf{x}$ perpendicular to the rows of R, we have that $\|P\|_F^2$ is the sum of its singular values squared, which is at most r as claimed.

Finally, to bound the time needed to compute U, the only step involved in computing U is to find the SVD of R. But note that RR^T is an $r \times r$ matrix, and since r is much smaller than n, m, this is fast.

2.3. Low-rank approximations and the SVD

Singular value decomposition yields the best approximation of given rank to a matrix in both spectral norm and Frobenius norm. Here, we show that a near-optimal low-rank approximation (LRA) of a matrix A can be found by computing the SVD of a smaller matrix R. The main result is that for *any* matrix R with

$$A^T A \approx R^T R,$$

the restriction of A to the space spanned by the top few right singular vectors of R is a good approximation to A. This is purely based on matrix perturbation theory: there is no probability involved.

If we can find an R with smaller dimensions than A, with $R^T R \approx A^T A$, then computing the SVD for R would be faster. Indeed, we have already seen one way of getting such an R: pick a set of r rows of A in r i.i.d. trials according to the length-squared distribution and scale them so that $E(R^T R) = A^T A$. We saw in Theorem 2.1 that $R^T R \approx A^T A$.

The Hoffman–Wielandt inequality (stated below) implies that if $\|R^T R - A^T A\|_F$ is small, then the singular *values* of A and R are close. If the top singular vectors of R and A were close, the desired result would follow easily. However, as is well known, multiple singular values are points of discontinuity of singular vectors. But it is not difficult to see intuitively that LRA is not discontinuous: using a singular vector of a slightly worse singular value does not induce gross errors in LRA, and this intuition underlies the result proved formally here.

2.3.1. Restriction to the SVD subspace

If $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k$ is a basis for the vector space V, the restriction of A to V is

$$\bar{A} = A \sum_{t=1}^{k} \mathbf{w}_t \mathbf{w}_t^T \text{ satisfies } \bar{A}\mathbf{x} = \begin{cases} A\mathbf{x} & \text{if } x \in V, \\ 0 & \text{if } \mathbf{x} \perp V. \end{cases} \tag{2.7}$$

Theorem 2.6. For *any* $r \times n$ matrix R and *any* $m \times n$ matrix A, if $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$ are the top k right singular vectors of R and $A(k)$ is the best rank-k approximation to A, then

$$\left\| A - A \sum_{t=1}^{k} \mathbf{v}_t \mathbf{v}_t^T \right\|_F^2 \leq \|A - A(k)\|_F^2 + 2\sqrt{k}\|R^T R - A^T A\|_F, \tag{2.8}$$

$$\left\| A - A \sum_{t=1}^{k} \mathbf{v}_t \mathbf{v}_t^T \right\|_2^2 \leq \|A - A(k)\|_2^2 + 2\|R^T R - A^T A\|_2. \tag{2.9}$$

In (2.8), the first term $\|A - A(k)\|_F^2$ is the best possible error we can make with exact SVD of A. We cannot avoid that. The second term is the penalty we pay for computing with R instead of A, and similarly for (2.9). The theorem is for any r including the cases $r = 0$ and $r = n$. Central to the proof is the Hoffman–Wielandt inequality, stated next without proof.

Lemma 2.7. If P, Q are two real symmetric $n \times n$ matrices and $\lambda_1, \lambda_2, \dots$ denote eigenvalues in non-increasing order, then

$$\sum_{t=1}^{n} (\lambda_t(P) - \lambda_t(Q))^2 \leq \|P - Q\|_F^2.$$

We can now prove the low-rank approximation guarantee.

Proof of Theorem 2.6. Complete $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$ to a basis $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ of $\mathbb{R}^n$. Then

$$\begin{aligned}
&\left\| A - A \sum_{t=1}^{k} \mathbf{v}_t \mathbf{v}_t^T \right\|_F^2 - \|A - A(k)\|_F^2 \\
&\quad = \|A\|_F^2 - \left\| A \sum_{t=1}^{k} \mathbf{v}_t \mathbf{v}_t^T \right\|_F^2 - (\|A\|_F^2 - \|A(k)\|_F^2) \\
&\quad = \|A(k)\|_F^2 - \left\| A \sum_{t=1}^{k} \mathbf{v}_t \mathbf{v}_t^T \right\|_F^2 \\
&\quad = \sum_{t=1}^{k} \sigma_t^2(A) - \sum_{t=1}^{k} \|A\mathbf{v}_t\|^2
\end{aligned}$$

$$= \sum_{t=1}^{k} \sigma_t^2(A) - \sum_{t=1}^{k} \mathbf{v}_t^T (A^T A - R^T R)\mathbf{v}_t - \sum_{t=1}^{k} \mathbf{v}_t^T R^T R \mathbf{v}_t$$
$$= \sum_{t=1}^{k} (\sigma_t^2(A) - \sigma_t^2(R)) - \sum_{t=1}^{k} \mathbf{v}_t^T (A^T A - R^T R)\mathbf{v}_t.$$

We can now deduce that

$$\left| \sum_{t=1}^{k} \mathbf{v}_t^T (A^T A - R^T R)\mathbf{v}_t \right| \leq k \|A^T A - R^T R\|_2,$$

but we want $\sqrt{k}$ instead of k. For this, we use the Cauchy–Schwarz inequality to assert

$$\left| \sum_{t=1}^{k} \mathbf{v}_t^T (A^T A - R^T R)\mathbf{v}_t \right| \leq \sqrt{k} \left(\sum_{t=1}^{k} (\mathbf{v}_t^T (A^T A - R^T R)\mathbf{v}_t)^2 \right)^{1/2}$$
$$\leq \sqrt{k} \|A^T A - R^T R\|_F,$$

since the Frobenius norm is invariant under change of basis and, in a basis containing $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$,

$$\sum_{t=1}^{k} (\mathbf{v}_t^T (A^T A - R^T R)\mathbf{v}_t)^2$$

is the sum of squares of the first k diagonal entries of $A^T A - R^T R$. We still have to bound

$$\sum_{t=1}^{k} (\sigma_t^2(A) - \sigma_t^2(R)),$$

for which we use the Hoffman–Wielandt inequality, after another use of Cauchy–Schwarz:

$$\sum_{t=1}^{k} (\sigma_t^2(A) - \sigma_t^2(R)) \leq \sqrt{k} \left(\sum_{t=1}^{k} (\sigma_t^2(A) - \sigma_t^2(R))^2 \right)^{1/2}.$$

Now $\sigma_t^2(A) = \lambda_t(A^T A)$ and $\sigma_t^2(R) = \lambda_t(R^T R)$, and so

$$\sum_{t=1}^{k} (\sigma_t^2(A) - \sigma_t^2(R))^2 \leq \|A^T A - R^T R\|_F^2.$$

Plugging these in, we obtain (2.8).

For (2.9), first note that the top singular vector $\mathbf{u}$ of $A - A\sum_{t=1}^{k} \mathbf{v}_t\mathbf{v}_t^T$ must be orthogonal to the span of $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$. Hence

$$\left(A - A\sum_{t=1}^{k} \mathbf{v}_t\mathbf{v}_t^T\right)\mathbf{u} = A\mathbf{u}$$

by (2.7). Thus

$$\begin{aligned}
&\left\|A - A\sum_{t=1}^{k} \mathbf{v}_t\mathbf{v}_t^T\right\|_2^2 \\
&\quad = \|A\mathbf{u}\|^2 \\
&\quad = \mathbf{u}^T A^T A\mathbf{u} \\
&\quad = \mathbf{u}^T(A^TA - R^TR)\mathbf{u} + \mathbf{u}^T R^T R\mathbf{u} \leq \|A^TA - R^TR\|_2 + \sigma_{k+1}^2(R) \\
&\quad = \|A^TA - R^TR\|_2 + (\sigma_{k+1}^2(R) - \sigma_{k+1}^2(A)) + \sigma_{k+1}^2(A) \\
&\quad \leq 2\|A^TA - R^TR\|_2 + \sigma_{k+1}^2(A),
\end{aligned}$$

by Weyl's inequality (Horn and Johnson 2012, Section 4.3), which states that

$$\lambda_{k+1}(R^TR) \in [\lambda_{k+1}(A^TA) - \|A^TA - R^TR\|_2,\ \lambda_{k+1}(A^TA) + \|A^TA - R^TR\|_2].$$

□

To summarize, here is the algorithm for LRA of A after one final note: we need the right singular vectors of R. For this, we can compute the SVD of RR^T (an $r \times r$ matrix) to find the left singular vectors of R, from which we can get the right singular vectors.

(1) Pick r rows of A by length-squared sampling, and scale them so that for the resulting $r \times n$ matrix R, $E(R^TR) = A^TA$.

(2) Find RR^T. Find the left singular vectors of R by computing the SVD of the $r \times r$ matrix RR^T. Premultiply R by the left singular vectors to get the right singular vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ of R.

(3) Return $A\sum_{t=1} \mathbf{v}_t\mathbf{v}_t^T$ as the implicit LRA (or if required, multiply out and return the matrix).

Let p be the maximum number of non-zero entries in any row of A ($p \leq n$). The first step can be done in two passes from external memory. For the second step, RR^T can be found in $O(r^2p)$ time. The spectral decomposition of RR^T can be done in time $O(r^3)$ (or better). Finally, multiplying the k left singular values by R can be done in time $O(krp)$.

The running time has been improved by Drineas, Kannan and Mahoney (2006) and further by Clarkson and Woodruff (see Woodruff 2014).

3. Tensor approximation via length-squared sampling

An r-tensor A is an r-dimensional array with real entries $A_{i_1,i_2,\ldots,i_r}$, for $i_1, i_2, \ldots, i_r \in \{1, 2, \ldots, n\}$. Tensors encode mutual information about subsets of size three or higher (matrix entries are pairwise information) and arise naturally in a number of applications.

In analogy with matrices, we may define a rank-1 tensor to be the *outer product* of r vectors $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(r)}$ denoted $\mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)} \otimes \cdots \otimes \mathbf{x}^{(r)}$, with $(i_1, i_2, \ldots, i_r)$th entry $x^{(1)}_{i_1} x^{(2)}_{i_2} \cdots x^{(r)}_{i_r}$. We will show the following.

(1) For any r-tensor A, there exists a good approximation by the sum of a small number of rank-1 tensors (Lemma 3.2).

(2) We can algorithmically find such an approximation (Theorem 3.3).

In the case of matrices, traditional linear algebra algorithms find optimal approximations in polynomial time. Unfortunately, there is no such theory (or algorithm) for r-dimensional arrays when $r > 2$. Indeed, there are computational hardness results (Hillar and Lim 2013). But our focus here is what we can do, not hardness results. We assume throughout that r is a fixed number, whereas n grows to infinity. We will develop polynomial-time algorithms for finding low-rank approximations. The algorithms make crucial use of length-squared sampling.

Definition 3.1. Corresponding to an r-tensor A, there is an r-linear form defined as follows: for vectors $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(r)}$,

$$A(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(r)}) = \sum_{i_1,i_2,\ldots,i_r} A_{i_1,i_2,\ldots,i_r} \, x^{(1)}_{i_1} x^{(2)}_{i_2} \cdots x^{(r)}_{i_r}. \tag{3.1}$$

We use the following two norms of r-dimensional arrays corresponding to the Frobenius and spectral norms for matrices:

$$\|A\|_F = \Bigl(\sum A^2_{i_1,i_2,\ldots,i_r}\Bigr)^{1/2}, \tag{3.2}$$

$$\|A\|_2 = \max_{\mathbf{x}^{(1)},\mathbf{x}^{(2)},\ldots,\mathbf{x}^{(r)}} \frac{A(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(r)})}{\|\mathbf{x}^{(1)}\|\|\mathbf{x}^{(2)}\| \cdots \|\mathbf{x}^{(r)}\|}. \tag{3.3}$$

Lemma 3.2. For any tensor A and any $\epsilon > 0$, there exist $k \le 1/\epsilon^2$ rank-1 tensors $B_1, B_2, \ldots, B_k$ such that

$$\|A - (B_1 + B_2 + \cdots + B_k)\|_2 \le \epsilon \|A\|_F.$$

Theorem 3.3. For any tensor A and any $\epsilon > 0$, we can find $k \le 4/\epsilon^2$ rank-1 tensors $B_1, B_2, \ldots, B_k$, using a randomized algorithm in time $(n/\epsilon)^{O(1/\epsilon^4)}$, such that with high probability (over coin tosses of the algorithm), we have

$$\|A - (B_1 + B_2 + \cdots + B_k)\|_2 \le \epsilon \|A\|_F.$$

Proof of Lemma 3.2. If $\|A\|_2 \leq \epsilon\|A\|_F$, then we are done. If not, there are unit vectors $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(r)}$ such that $A(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(r)}) \geq \epsilon\|A\|_F$. Now consider the r-dimensional array

$$B = A - (A(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(r)}))\mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)} \otimes \cdots \otimes \mathbf{x}^{(r)}.$$

It is easy to see that $\|B\|_F^2 \leq \|A\|_F^2(1-\epsilon^2)$. We may repeat for B: either $\|B\|_2 \leq \epsilon\|A\|_F$ and we may stop, or there exist $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(r)}$ with $B(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(r)}) \geq \epsilon\|A\|_F$. Let

$$C = B - B(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(r)})\ (\mathbf{y}^{(1)} \otimes \mathbf{y}^{(2)} \otimes \cdots \otimes \mathbf{y}^{(r)}).$$

Each time the Frobenius norm squared falls by $\epsilon^2\|A\|_F^2$, so this process will only continue for at most $1/\epsilon^2$ steps. □

The algorithm that proves Theorem 3.3 will take up the rest of this section. First, from the proof of Lemma 3.2, it suffices to find $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(r)}$ all of length 1, maximizing $A(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(r)})$ to within *additive error* $\epsilon\|A\|_F/2$. We will give an algorithm to solve this problem. We need a bit more notation. For any $r-1$ vectors $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(r-1)}$, we define a vector with ith component

$$A(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(r-1)}, \cdot)_i = \sum_{i_1, i_2, \ldots, i_{r-1}} A_{i_1, i_2, \ldots, i_{r-1}, i} z^{(1)}_{i_1} z^{(2)}_{i_2} \cdots z^{(r-1)}_{i_{r-1}}. \quad (3.4)$$

Here is the idea behind the algorithm. Suppose $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(r)}$ are the unknown unit vectors that maximize $A(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots)$. Since

$$A(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(r-1)}, \mathbf{z}^{(r)}) = \mathbf{z}^{(r)^T} A(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(r-1)}, \cdot),$$

we have

$$\mathbf{z}^{(r)} = \frac{A(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(r-1)}, \cdot)}{|A(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(r-1)}, \cdot)|}.$$

Thus, if we knew $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(r-1)}$, then we could compute $\mathbf{z}^{(r)}$. In fact, each component of $\mathbf{z}^{(r)}$ is the sum of n^{r-1} terms as in (3.4). We can hope to estimate the components of $\mathbf{z}^{(r)}$ with a sample of fewer than n^{r-1} terms. In fact, the main point is that we will show that if we pick a sample I of $s = O(1/\epsilon^2)$ elements $(i_1, i_2, \ldots, i_{r-1})$, with probabilities proportional to $\sum_i A^2_{i_1, i_2, \ldots, i_{r-1}, i}$, which is the squared length of the 'line' or 'column' $(i_1, i_2, \ldots, i_{r-1})$, the sums are well estimated. In more detail, let $f((i_1, i_2, \ldots, i_{r-1})) = z^{(1)}_{i_1} z^{(2)}_{i_2} \cdots z^{(r-1)}_{i_{r-1}}$. We will show that

$$\sum_{(i_1, i_2, \ldots, i_{r-1}) \in I} A_{i_1, i_2, \ldots, i_{r-1}, i} f((i_1, i_2, \ldots, i_{r-1})) \approx c z^{(r)}_i,$$

where c is a scalar (independent of i).

Now we still need to compute the $s(r-1)$ real numbers $f((i_1, i_2, \dots, i_{r-1}))$, for all $(i_1, i_2, \dots, i_{r-1}) \in I$. We just enumerate all possibilities for the $s(r-1)$-tuple of values in steps of a certain size η, and for each possibility, we get a candidate $\mathbf{z}^{(r)}$. One of these candidates will be (close to) the optimal $\mathbf{z}^{(r)}$, but we do not know which one. To solve this puzzle, we just turn the problem on its head: for each candidate $\mathbf{z}^{(r)}$, we can define an $r-1$ tensor $A(\cdot, \dots, \cdot, \mathbf{z}^{(r)})$ similar to (3.4), and recursively find the $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(r-1)}$ (approximately) maximizing $A(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(r-1)}, \mathbf{z}^{(r)})$. The best of these will satisfy the theorem. One subtle point: each enumerated $s(r-1)$-tuple need not really come from $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots, \mathbf{z}^{(r-1)}$, since all we need for this argument is that one of the enumerated $s(r-1)$-tuples of values gets close to the true $\{z_{i_1}^{(1)} z_{i_2}^{(2)} \cdots z_{i_{r-1}}^{(r-1)} : (i_1, i_2, \dots, i_{r-1}) \in I\}$.

Algorithm 3.4 (tensor decomposition). Set

$$\eta = \frac{\epsilon^2}{100r\sqrt{n}} \quad \text{and} \quad s = \frac{10^5 r}{\epsilon^2}.$$

(1) Pick s random $(r-1)$-tuples $(i_1, i_2, \dots, i_{r-1})$ with probabilities proportional to the sum of squared entries on the corresponding line, that is,

$$p(i_1, i_2, \dots, i_{r-1}) = \frac{\sum_i A^2_{i_1, i_2, \dots, i_{r-1}, i}}{\|A\|_F^2}.$$

Let I be the set of s $(r-1)$tuples so chosen.

(2) Enumerate all functions

$$f : I \to \{-1, -1+\eta, -1+2\eta, \dots, 0, \dots, 1-\eta, 1\}.$$

(a) For each of the $((2/\eta)+1)^{s(r-1)}$ functions so enumerated, find a vector $\mathbf{y}$ defined by

$$y_i = \sum_{(i_1, i_2, \dots, i_{r-1}) \in I} \frac{A_{i_1, i_2, \dots, i_{r-1}, i} f((i_1, i_2, \dots, i_{r-1}))}{p(i_1, i_2, \dots, i_{r-1})}.$$

Replace $\mathbf{y}$ by $\mathbf{y}/\|\mathbf{y}\|$.

(b) Consider the $(r-1)$-dimensional array $A(\mathbf{y})$ defined by

$$(A(\mathbf{y}))_{i_1, i_2, \dots, i_{r-1}} = \sum_i A_{i_1, i_2, i_3 \dots, i_{r-1}, i}\ y_i$$

and apply the algorithm recursively to find the approximate maximum

$$A(\mathbf{y})(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(r-1)}) \text{ subject to } \|\mathbf{x}^{(1)}\| = \cdots = \|\mathbf{x}^{(r-1)}\| = 1,$$

to within additive error $\epsilon\|A(\mathbf{y})\|_F/2$. Note that $\|A(\mathbf{y})\|_F \leq \|A\|_F$ by Cauchy–Schwarz.

(3) Output the set of vectors that gives the maximum among all these candidates.

We now analyse Algorithm 3.4 and prove Theorem 3.3. We begin by showing that the discretization does not cause any significant loss.

Lemma 3.5. Let $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(r-1)}$ be the optimal unit vectors. Suppose $\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(r-1)}$ are obtained from the $\mathbf{z}^{(t)}$ by rounding each coordinate down to the nearest integer multiple of η. Then

$$\|A(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(r-1)}, \cdot) - A(\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(r-1)}, \cdot)\| \leq \frac{\epsilon^2}{100}\|A\|_F.$$

Proof. We write

$$\begin{aligned}
&|A(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(r-1)}, \cdot) - A(\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(r-1)}, \cdot)| \\
&\quad \leq |A(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(r-1)}, \cdot) - A(\mathbf{w}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(r-1)}, \cdot)| \\
&\qquad + |A(\mathbf{w}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(r-1)}, \cdot) - A(\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \mathbf{z}^{(3)}, \ldots, \mathbf{z}^{(r-1)}, \cdot)| + \cdots.
\end{aligned}$$

A typical term above is

$$\begin{aligned}
&\big|A(\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(t)}, \mathbf{z}^{(t+1)}, \ldots, \mathbf{z}^{(r-1)}, \cdot) \\
&\quad - A(\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(t)}, \mathbf{w}^{(t+1)}, \mathbf{z}^{(t+2)}, \ldots, \mathbf{z}^{(r-1)}, \cdot)\big|.
\end{aligned}$$

Define B to be the matrix with components

$$B_{ij} = \sum_{j_1, j_2, \ldots, j_t, j_{t+2}, \ldots, j_{r-1}} A_{j_1, j_2, \ldots, j_t, i, j_{t+2}, \ldots, j_{r-1}, j} w_{j_1}^{(1)} \cdots w_{j_t}^{(t)} z_{j_{t+2}}^{(t+2)} \cdots z_{j_{r-1}}^{(r-1)}$$

Then the term above is bounded by

$$|B(\mathbf{z}^{(t+1)} - \mathbf{w}^{(t+1)})| \leq \|B\|_2 |\mathbf{z}^{(t+1)} - \mathbf{w}^{(t+1)}| \leq \|B\|_F \eta \sqrt{n} \leq \|A\|_F \eta \sqrt{n}.$$

The claim follows from our choice of η. □

Next, we analyse the error incurred by sampling. Consider the $(r-1)$-tuple $(i_1, i_2, \ldots, i_{r-1}) \in I$, and define the random variables X_i by

$$X_i = \frac{A_{i_1, i_2, \ldots, i_{r-1}, i} w_{i_1}^{(1)} w_{i_2}^{(2)} \cdots w_{i_{r-1}}^{(r-1)}}{p(i_1, i_2, \ldots, i_{r-1})}.$$

It follows that

$$E(X_i) = A(\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(r-1)}, \cdot)_i.$$

We estimate the variance using a calculation similar to the matrix case:

$$\begin{aligned}\sum_i \operatorname{Var}(X_i) &\le \sum_i \sum_{i_1,i_2,\dots,i_{r-1}} \frac{A^2_{i_1,i_2,\dots,i_{r-1},i}(w^{(1)}_{i_1}\cdots w^{(r-1)}_{i_{r-1}})^2}{p(i_1,i_2,\dots,i_{r-1})}\\ &= \sum_{i_1,i_2,\dots,i_{r-1}} \frac{(w^{(1)}_{i_1}w^{(2)}_{i_2}\cdots w^{(r-1)}_{i_{r-1}})^2}{p(i_1,i_2,\dots,i_{r-1})}\sum_i A^2_{i_1,i_2,\dots,i_{r-1},i}\\ &= \|A\|_F^2 \sum_{i_1,i_2,\dots,i_{r-1}} (w^{(1)}_{i_1}w^{(2)}_{i_2}\cdots w^{(r-1)}_{i_{r-1}})^2 = \|A\|_F^2.\end{aligned}$$

Consider the y_i computed by the algorithm when all $\hat{z}^{(t)}_{i_t}$ are set to $w^{(t)}_{i_t}$. This will clearly happen at some point during the enumeration. This $\mathbf{y}_i$ is just the sum of s i.i.d. copies of X_i, one for each element of I. Thus we have

$$E(\mathbf{y}) = sA(\mathbf{w}^{(1)},\mathbf{w}^{(2)},\dots,\mathbf{w}^{(r-1)},\cdot), \quad \operatorname{Var}(\mathbf{y}) = E(\|\mathbf{y}-E(\mathbf{y})\|^2) \le s\|A\|_F^2.$$

Let

$$\zeta = sA(\mathbf{z}^{(1)},\mathbf{z}^{(2)},\dots,\mathbf{z}^{(r-1)}).$$

Since we only want to find the maximum to within an additive error of $\epsilon s\|A\|_F/2$, without loss of generality, we may assume $\|\zeta\| \ge \epsilon\|A\|_F/2$. By Chebyshev's inequality, it follows that with high probability $\|\mathbf{y}-\zeta\| \le c\sqrt{s}\|A\|_F$. One can now show that

$$\left\|\frac{\mathbf{y}}{\|\mathbf{y}\|} - \frac{\zeta}{\|\zeta\|}\right\| \le c\epsilon.$$

From this we obtain

$$\left\|A\left(\frac{\mathbf{y}}{\|\mathbf{y}\|}\right) - A\left(\frac{\zeta}{\|\zeta\|}\right)\right\|_F \le \frac{\epsilon}{10}\|A\|_F.$$

Thus, for any $r-1$ unit vectors $\mathbf{a}^{(1)},\mathbf{a}^{(2)},\dots,\mathbf{a}^{(r-1)}$, we have

$$\left\|A\left(\mathbf{a}^{(1)},\mathbf{a}^{(2)},\dots,\mathbf{a}^{(r-1)},\frac{\mathbf{y}}{\|\mathbf{y}\|}\right) - A\left(\mathbf{a}^{(1)},\mathbf{a}^{(2)},\dots,\mathbf{a}^{(r-1)},\frac{\zeta}{\|\zeta\|}\right)\right\| \le \frac{\epsilon}{10}\|A\|_F.$$

This implies that the optimal set of vectors for $A(\mathbf{y}/\|\mathbf{y}\|)$ are nearly optimal for $A(\zeta/\|\zeta\|)$. Since $\mathbf{z}^{(r)} = \zeta/\|\zeta\|$, the optimal vectors for the latter problem are $\mathbf{z}^{(1)},\dots,\mathbf{z}^{(r-1)}$. The error bound follows using Lemma 3.5.

Finally, the running time of Algorithm 3.4 is dominated by the number of candidates we enumerate, and is given by

$$\operatorname{poly}(n)\left(\frac{1}{\eta}\right)^{s^2 r} = \left(\frac{n}{\epsilon}\right)^{O(1/\epsilon^4)}.$$

3.1. Symmetric non-negative tensors with good low-rank approximations

As we mentioned earlier, a small set of samples cannot hope to deal with every matrix. But we saw that sampling yields good results for numerically low-rank matrices (matrices where, say, $\sigma_1^2(A)+\sigma_2^2(A)+\cdots+\sigma_p^2(A) \geq c\|A\|_F^2$ for a small p). Here, we will discuss instances of tensors with a similar property.

Property X. The spectral norm of the tensor is a constant fraction of the Frobenius norm after we scale the tensor, once for each dimension.

For a matrix, the permitted scaling is to scale the rows, scale the columns or both. In this section we only deal with tensors with non-negative real entries which are also symmetric, that is, for any permutation τ of $\{1,2,\ldots,r\}$, we have

$$A_{i_1,i_2,\ldots,i_r} = A_{i_{\tau(1)},i_{\tau(2)},\ldots,i_{\tau(r)}}.$$

We will show that two natural classes of tensors have Property X. We begin by discussing these classes for matrices. The first class of matrices are adjacency matrices of dense graphs, that is, $n\times n$ matrices with entries $\{0,1\}$ with $\Omega(n)$ non-zeros in each row. These arise in many applications. They are easily seen to satisfy Property X with no scaling, since

$$\|A\|_F = O(n), \quad \|A\|_2 \geq \frac{1}{n}\sum_{i,j} A_{ij} \in \Omega(n).$$

Another important class of matrices represent metrics, where A_{ij} is the distance between points i and j in a metric space, that is, distances satisfy the triangle inequality. Let $D_i = \sum_{j=1}^n A_{ij}$ be the total distance of point i to other points. Then, one can prove the following result.

Lemma 3.6 (local density of metrics). A_{ij} is at most $(D_i + D_j)/n$.

Proof. By the triangle inequality, $d(x_i,x_j) \leq d(x_i,x_k)+d(x_k,x_j)$, for all k. Summing, for $k=1,\ldots,n$, we obtain

$$n\, d(x_i,x_j) \leq \sum_{k=1}^n d(x_i,x_k) + \sum_{k=1}^n d(x_k,x_j) = D_i + D_j,$$

whence $d(x_i,x_j) \leq (D_i+D_j)/n$. □

We call the lemma the 'local density property' since, for the previous example of dense graphs, each entry of the adjacency matrix is indeed at most a constant times the row and column average. Metrics also have this property for each entry.

Using this, one can show that a row and column scaled version of A satisfies Property X. Let

$$\bar{D} = \frac{1}{n}\sum_{i=1}^{n} D_i.$$

The scaled matrix B is defined by

$$B_{ij} = \frac{A_{ij}}{\sqrt{(D_i+\bar{D})(D_j+\bar{D})}}.$$

We will not prove here that B has Property X, but this will follow from the general tensor case below. In fact, we will see that for tensors, a global averaged version of the density lemma holds, and implies Property X.

Let A be any symmetric non-negative tensor. Let

$$D_i = \sum_{i_2,i_3,\ldots,i_r} A_{i,i_2,i_3,\ldots,i_r}, \quad \bar{D} = \frac{1}{n}\sum_i D_i.$$

Definition 3.7. The *density* of a tensor A is defined by

$$\gamma(A) = \left(\sum_{i=1}^{n} D_i\right)^{r-2} \sum_{i_1,i_2,\ldots,i_r} \frac{A^2_{i_1,i_2,\ldots,i_r}}{\prod_{t=1}^{r}(D_{i_t}+\bar{D})}.$$

Lemma 3.8. Let A be an r-dimensional tensor satisfying the following local density condition:

$$A_{i_1,\ldots,i_r} \leq \frac{c}{rn^{r-1}}\sum_{j=1}^{r} D_{i_j}, \quad \text{for all } i_1,\ldots,i_r \in V,$$

where c is a constant. Then A has local density at most c.

Remark. Examples of graphs that satisfy the local density condition above include graphs with total number of edges at least cn^2, metrics and quasi-metrics where there is some $\gamma > 0$ with $A_{ij} = (\text{distance between } i \text{ and } j)^{\gamma}$.

Proof. We need to bound the density of A. To this end,

$$\begin{aligned}
&\sum_{i_1,i_2,\ldots,i_r\in V} \frac{A^2_{i_1,\ldots,i_r}}{\prod_{j=1}^{r}(D_{i_j}+\bar{D})} \\
&\quad \leq \frac{c}{rn^{r-1}} \sum_{i_1,i_2,\ldots,i_r\in V} \frac{A_{i_1,\ldots,i_r}\sum_{j=1}^{r} D_{i_j}}{\prod_{j=1}^{r}(D_{i_j}+\bar{D})} \\
&\quad \leq \frac{c}{rn^{r-1}} \sum_{i_1,i_2,\ldots,i_r\in V} A_{i_1,\ldots,i_r} \sum_{j=1}^{r} \frac{1}{\prod_{k\in\{1,\ldots,r\}\setminus j}(D_{i_k}+\bar{D})}
\end{aligned}$$

$$\begin{aligned}&\le \frac{c}{rn^{r-1}}\Biggl(\sum_{i_1,i_2,\ldots,i_r\in E} A_{i_1,\ldots,i_r}\Biggr)\frac{r}{\bar{D}^{r-1}}\\ &= \frac{c}{(\sum_{i=1}^n D_i)^{r-2}}.\end{aligned}$$

Thus, the density is at most

$$\Biggl(\sum_{i=1}^n D_i\Biggr)^{r-2}\sum_{i_1,i_2,\ldots,i_r\in E}\frac{A^2_{i_1,\ldots,i_r}}{\prod_{j=1}^r(D_{i_j}+\bar{D})}\le c. \qquad \square$$

Let B be a scaled version of A:

$$B_{i_1,i_2,\ldots,i_r}=\frac{A_{i_1,i_2,\ldots,i_r}}{\prod_{t=1}^r\alpha_{i_t}},\quad \text{where } \alpha_i=\sqrt{D_i+\bar{D}}.$$

Lemma 3.9. $\|B\|_F\le c\|B\|_2$.

Proof. We have

$$\|B\|_F^2=\frac{\gamma(A)}{(n\bar{D})^{r-2}},$$

while, if $\alpha=(\alpha_1,\ldots,\alpha_r)^T$, then

$$\begin{aligned}\|B\|_2&\ge\frac{B(\alpha,\alpha,\ldots,\alpha)}{\|\alpha\|^r}\\ &=\frac{1}{\|\alpha\|^r}\sum_{i_1,i_2,\ldots,i_r}B_{i_1,i_2,\ldots,i_r}\alpha_{i_1}\alpha_{i_2}\cdots\alpha_{i_r}\\ &=\frac{1}{\|\alpha\|^r}\sum_{i_1,i_2,\ldots,i_r}A_{i_1,i_2,\ldots,i_r}\\ &=\frac{n\bar{D}}{(2n\bar{D})^{r/2}}.\end{aligned}$$

Hence

$$\|B\|_F^2\le\gamma(A)2^r\|B\|_2^2. \qquad \square$$

4. Spectral norm error for matrices

In this section we prove the result of Rudelson and Vershynin (2007) summarized earlier in (1.2). The proof involves the Hoeffding–Chernoff inequality for matrix-valued random variables, and we prove this first. The Hoeffding–Chernoff inequality for real-valued independent identically distributed random variables $X_1,X_2,\ldots,X_s$ can be stated as follows. For any positive real numbers a, t, we have

$$\Pr\left(\sum_{i=1}^{s} X_i > a\right) \le \mathrm{e}^{-ta}\ (E(\mathrm{e}^{tX_1}))^s.$$

Also, recall that the matrix exponential

$$\mathrm{e}^U = \sum_{t=0}^{\infty} \frac{U^t}{t!}$$

exists for any square matrix U. For a real symmetric matrix U with eigenvalues $\lambda_i(U)$, e^U has eigenvalues $\mathrm{e}^{\lambda_i(U)}$ and the same eigenvectors as U. Therefore e^U is always positive semidefinite.

4.1. Hoeffding–Chernoff inequality for matrix-valued random variables

Theorem 4.1. Let X be a random variable taking values which are real symmetric $d \times d$ matrices. Suppose $X_1, X_2, \dots, X_s$ are i.i.d. draws of X. For any positive real numbers a, t, we have

$$\Pr\left(\lambda_{\max}\left(\sum_{i=1}^{s} X_i\right) \ge a\right) \le d\,\mathrm{e}^{-ta} \|E\,\mathrm{e}^{tX}\|_2^s, \tag{4.1}$$

$$\Pr\left(\left\|\sum_{i=1}^{s} X_i\right\|_2 \ge a\right) \le d\,\mathrm{e}^{-ta}\big(\|E\,\mathrm{e}^{tX}\|_2^s + \|E\,\mathrm{e}^{-tX}\|_2^s\big). \tag{4.2}$$

Remark. $\lambda_{\max}$ is the largest eigenvalue. Note that having the expectation inside the norm is better than having it outside on the right-hand side, since, by the convexity of the norm function and Jensen's inequality, it follows that for a matrix-valued random variable B, we have $\|E(B)\| \le E(\|B\|)$, and it can be much less. Also, it is easy to see that applying the real-valued Hoeffding–Chernoff inequality to each entry of $\sum_{i=1}^{s} X_i$ would not yield the theorem.

Proof. Inequality (4.2) follows from (4.1), since

$$\left\|\sum_i X_i\right\|_2 = \max\left(\lambda_{\max}\left(\sum_i X_i\right),\ \lambda_{\max}\left(\sum_i(-X_i)\right)\right),$$

and we can apply the first inequality twice: once with X_i and once with $-X_i$. So we prove only the first inequality. Let $S = X_1 + X_2 + \cdots + X_s$. Then

$$\lambda_{\max}(S) \ge a \iff \lambda_{\max}(tS) \ge ta \iff \lambda_{\max}(\mathrm{e}^{tS}) \ge \mathrm{e}^{ta} \implies \mathrm{Tr}[\mathrm{e}^{tS}] \ge \mathrm{e}^{ta}.$$

Now $\mathrm{Tr}[\mathrm{e}^{tS}]$ is a non-negative real-valued random variable, so by Markov's inequality, we get that

$$\Pr(\mathrm{Tr}[\mathrm{e}^{tS}] \ge \mathrm{e}^{ta}) \le \mathrm{e}^{-ta} E(\mathrm{Tr}[\mathrm{e}^{tS}]).$$

We will upper-bound $E(\mathrm{Tr}[\mathrm{e}^{tS}])$. To this end, we first use the Golden–Thomson inequality (Bhatia 1996), which asserts that for Hermitian matrices U, V,

$$\mathrm{Tr}[\mathrm{e}^{U+V}] \leq \mathrm{Tr}[\mathrm{e}^{U}\,\mathrm{e}^{V}].$$

(The proof is not given here. Note that this is false for three matrices.) We will use this with $U = t(X_1 + X_2 + \cdots + X_{s-1})$ and $V = tX_s$. Also note that Tr and E commute. Then

$$\begin{aligned} E(\mathrm{Tr}[\mathrm{e}^{tS}]) &\leq E(\mathrm{Tr}[\mathrm{e}^{U}\,\mathrm{e}^{V}]) \\ &= \mathrm{Tr}[E(\mathrm{e}^{U}\,\mathrm{e}^{V})] \\ &= \mathrm{Tr}\big[E_{X_1,X_2,\dots,X_{s-1}}(\mathrm{e}^{U}\,E_{X_s}(\mathrm{e}^{V}))\big] \qquad\qquad \text{(independence)} \\ &\leq \|E(\mathrm{e}^{tX})\|_2\ \mathrm{Tr}\big[E_{X_1,X_2,\dots,X_{s-1}}(\mathrm{e}^{U})\big] \\ &\qquad\quad (\mathrm{Tr}[BC] \leq \mathrm{Tr}[B]\ \|C\|_2 \text{ for positive semidefinite } B, C) \\ &\leq \|E(\mathrm{e}^{tX})\|_2^{s-1}\mathrm{Tr}\big[E(\mathrm{e}^{tX})\big] \\ &\leq d\|E(\mathrm{e}^{tX})\|_2^{s}, \end{aligned}$$

where, in the penultimate inequality, we have peeled off an X_i $s-2$ times. The factor of d arises because $\mathrm{Tr}[\mathrm{e}^{tX}] \leq d\lambda_{\max}(\mathrm{e}^{tX})$. No direct way of bounding $\lambda_{\max}$ without going to the trace is known. □

Notation. For two $n \times n$ real symmetric matrices B, C, we write $B \preceq C$ when $C - B$ is positive semidefinite.

Lemma 4.2. If B is a real symmetric matrix for which $\|B\|_2 \leq 1$, then $\mathrm{e}^{B} \preceq I + B + B^2$.

Proof. The inequality $\mathrm{e}^{\lambda_i} \leq 1 + \lambda_i + \lambda_i^2$, for $|\lambda_i| \leq 1$, implies

$$\mathrm{e}^{\lambda_i}\mathbf{v}_i\mathbf{v}_i^T \preceq (1 + \lambda_i + \lambda_i^2)\mathbf{v}_i\mathbf{v}_i^T,$$

whence

$$\mathrm{e}^{B} = \sum_{i=1}^{d} \mathrm{e}^{\lambda_i}\mathbf{v}_i\mathbf{v}_i^T \preceq \sum_{i=1}^{d}(1 + \lambda_i + \lambda_i^2)\mathbf{v}_i\mathbf{v}_i^T = I + B + B^2. \qquad \square$$

4.2. Applying Hoeffding–Chernoff to length-squared sampling

Suppose we use length-squared sampling on a matrix A to draw s columns in s i.i.d. trials. Let

$$p_j = \frac{|A(:,j)|^2}{\|A\|_F^2}$$

be the probability of picking column j. Define the real symmetric matrix-valued random variable Y satisfying

$$\Pr\biggl(Y = \frac{1}{s\,p_j}\;A(:,j)A(:,j)^T\biggr) = p_j. \tag{4.3}$$

We saw that $EY = AA^T/s$. Then the random variable

$$X = Y - EY \quad \text{satisfies } EX = 0. \tag{4.4}$$

Also, we have $E(X^2) \preceq E(Y^2)$; this is proved along the same lines as for real random variables, where the variance of a random variable is at most the second moment. Therefore

$$E(X^2) \preceq E(Y^2) = \sum_j p_j \frac{A(:,j)A(:,j)^T A(:,j)A(:,j)^T}{s^2\,p_j^2} = AA^T\|A\|_F^2 \frac{1}{s^2}, \tag{4.5}$$

which implies

$$\|E(X^2)\|_2 \leq \frac{1}{s^2}\|A\|_2^2\|A\|_F^2. \tag{4.6}$$

We will also need an absolute upper bound on $\|X\|_2$. For this, note that

$$\begin{aligned}\|X\|_2 &\leq \frac{1}{s}\max(\|A(:,j)A(:,j)^T/p_j\|_2, \|AA^T\|_2) \qquad (4.7)\\ &\leq \max\biggl(\frac{\|A(:,j)A(:,j)^T\|_2}{s\|A(:,j)\|^2}\|A\|_F^2, \frac{\|A\|_F^2}{s}\biggr) = \frac{1}{s}\|A\|_F^2.\end{aligned}$$

Proposition 4.3. If t is a positive real number such that $\|tX\|_2 \leq 1$ for all possible values of X, then

$$\|E\,\mathrm{e}^{\pm tX}\|_2 \leq 1 + \frac{t^2}{s^2}\|A\|_2^2\|A\|_F^2 \leq \mathrm{e}^{t^2\|A\|_2^2\|A\|_F^2/s^2}$$

Proof. We have $E(\mathrm{e}^{tX}) \preceq E(I + tX + t^2X^2) = I + t^2E(X^2)$, since $EX = 0$. Thus we have the proposition using (4.5). □

Note. It was important to argue that $\mathrm{e}^B \preceq I + B + B^2$ in the lemma. A weaker inequality such as $\|\mathrm{e}^B\|_2 \leq \|I + B + B^2\|_2$ does not suffice. We are ready to prove (1.2).

Theorem 4.4. Let A be any $m \times n$ matrix and let C be an $m \times s$ matrix obtained by length-squared sampling and scaling to have $E(CC^T) = AA^T$. (C consists of columns $Y_1, Y_2, \dots, Y_s$, which are i.i.d. copies of Y defined in (4.3).) Then, for all $\varepsilon \in [0, \|A\|_2/\|A\|_F]$,[4] we have

$$\Pr\bigl(\|CC^T - AA^T\|_2 \geq \varepsilon\|A\|_2\|A\|_F\bigr) \leq 2n\,\mathrm{e}^{-\varepsilon^2 s/4}.$$

[4] If $\varepsilon \geq \|A\|_2/\|A\|_F$, then the zero matrix is a good enough approximation to AA^T.

Hence, for $s \geq (c \ln n)/\varepsilon^2$, with high probability we have

$$\|CC^T - AA^T\|_2 \leq \varepsilon \|A\|_2 \|A\|_F. \tag{4.8}$$

Proof. It is easy to see that

$$CC^T - AA^T = \sum_{i=1}^{s} X_i,$$

where the X_i are i.i.d. copies of X defined in (4.4). Then

$$\begin{aligned}
\Pr\Biggl(\Biggl\|\sum_{i=1}^{s} X_i\Biggr\|_2 \geq \varepsilon \|A\|_2 \|A\|_F\Biggr) &\\
&\leq n\, \mathrm{e}^{-t\varepsilon \|A\|_2 \|A\|_F} \bigl(\|E(\mathrm{e}^{tX_1})\|_2^s + \|E(\mathrm{e}^{-tX_1})\|_2^s\bigr)\\
&\qquad\qquad\qquad\text{(for any } t > 0\text{, by Theorem 4.1)}\\
&\leq 2n\, \mathrm{e}^{-t\varepsilon \|A\|_2 \|A\|_F} \mathrm{e}^{t^2 \|A\|_2^2 \|A\|_F^2 / s}\\
&\qquad\text{(provided } t \leq s/\|A\|_F^2\text{, by Proposition 4.3 and (4.7))}\\
&\leq 2n\, \mathrm{e}^{-\varepsilon^2 s/4},
\end{aligned}$$

setting

$$t = \frac{\varepsilon s}{2\|A\|_F \|A\|_2} \leq \frac{s}{\|A\|_F^2}.$$

So we see that for $s \geq (c \ln n)/\varepsilon^2$ and a large enough constant c, with high probability,

$$\|CC^T - AA^T\|_2 \leq \varepsilon \|A\|_2 \|A\|_F. \qquad \square$$

5. Preconditioned length-squared sampling

In this section we show how to solve problem (1.3). We first discuss a principal motivation.

5.1. Relative error in the residual

We have seen that if we sample a set of columns of A and scale them to form an $m \times s$ matrix C, the restriction of A to the column space of the top k right singular vectors of C is a good approximation to A, in the sense of Theorem 2.1. But the error is in terms of $\|A\|_F$. In this section we see how to get errors of the form $O(\|A - A_k\|_F)$, where A_k denotes the best rank-k approximation to A. Of course, we can get such errors by using the SVD of A. The point here is that we will see that a random sample corresponding to preconditioned length-squared sampling on A_k (instead of A) gives us a subset of columns of A in whose span we find an $O(\|A - A_k\|_F)$-type error.

There are at least three approaches to getting such a relative error bound. The first is to use length-squared sampling *iteratively* (Deshpande *et al.* 2006). In this approach, after performing length-squared sampling on the initial matrix A to obtain a sample S_1, the columns of A are sampled again, but this time with probability proportional to their squared lengths in the residual, that is, after orthogonal projection onto the span of S_1. This process is repeated, and results in a geometrically decaying error factor with each round. Such an approach was also used by Li, Miller and Peng (2013). More recent papers analyse a similar approach with uniform sampling (Cohen *et al.* 2015, Cohen, Musco and Musco 2017).

The second approach is *volume sampling*, which extends length-squared sampling by picking subsets of k columns jointly (Deshpande and Vempala 2006). The probability of picking a k-subset is proportional to the squared volume of the k-dimensional simplex induced by the columns along with the origin, a generalization of length-squared sampling. This single sample approach has expected squared error $(k+1)\|A-A_k\|_F^2$, the best possible bound using k columns of the matrix. This can be improved by one round to length-squared sampling in the residual to $(1+\epsilon)$ (relative error) using $O(k/\epsilon)$ samples.

The third approach (Drineas *et al.* 2008), which we present here, uses only one round of length-squared sampling, but the lengths are according to a different matrix, which can be viewed as a preconditioned version of the original matrix, and corresponds to sampling columns according to their *leverage scores*. Before we go to the main proof of the preconditioned length-squared sampling method, we discuss its application to low-rank approximation in more detail.

Let S be the $n\times s$ column selector (no scaling of columns) matrix, so that $C=AS$. Let V_k be the $n\times k$ matrix with top k right singular vectors of A as columns. The probability of picking column j of A to include in C is according to the squared length of column j of $A_k^+A_k=V_kV_k^T$, where A_k^+ is the pseudo-inverse of A. Since V_k has orthonormal columns, this is the same as the squared length of column j of V_k^T. Then the approximation to A in the column space of C is

$$X=C(V_k^TS)^+V_k^T.$$

The error is bounded as follows:

$$\begin{aligned}A-X&=A-AS(V_k^TS)^+V_k^T\\&=A-(AV_kV_k^T+(A-A_k))S(V_k^TS)^+V_k^T\\&=A-AV_kV_k^T-(A-A_k)S(V_k^TS)^+V_k^T\\&=\underbrace{(A-A_k)}_{X_1}-\underbrace{(A-A_k)S(V_k^TS)^+V_k^T}_{X_2}.\end{aligned}$$

Then

$$\|X_1\|_F = \|A - A_k\|_F,$$
$$\|X_2\|_F \le \|(A - A_k)S\|_F \underbrace{\|(V_k^T S)^+ V_k^T\|_2}_{X_3}.$$

Hence $E(\|(A - A_k)S\|_F) \le \|A - A_k\|_F$. Now, $X_3 = 1/\sigma_{\min}(V_k^T S)$ (the last V_k^T can be omitted since it has orthonormal rows). We can bound this as follows:

$$X_3 = \frac{1}{\sqrt{\lambda_{\min}(V_k^T S S^T V_k)}}.$$

Noting that $V_k V_k^T = I_k$, we can view this as selecting columns (as in S) according to length-squared sampling on V_k to multiply V_k^T by V_k. By the guarantee for approximate matrix multiplication using length-squared sampling, the error is at most $(2 + \varepsilon)\|A - A_k\|_F$. This can be improved to $(1 + \epsilon)\|A - A_k\|_F$.

5.2. Leverage score sampling

Here we present the main guarantee for sampling according to the leverage scores, that is, the preconditioned squared lengths.

Theorem 5.1. Suppose A is any $d \times n$ matrix with rank r and SVD $A = \sum_{t=1}^r \sigma_t \mathbf{u}_t \mathbf{v}_t^T$, and W is an $n \times d$ symmetric matrix $\sum_{t=1}^r {\sigma_t}^{-1} \mathbf{v}_t \mathbf{u}_t^T$.[5] Let $p_j, j = 1, 2, \dots, n$ be defined by[6]

$$p_j = \frac{|(WA)(:, j)|^2}{\|WA\|_F^2}.$$

If $\varepsilon \in [0, 1]$ and $s \ge (cr \ln n)/\varepsilon^2$, and we draw s columns of A in i.i.d. trials, with probabilities $\{p_j\}$, scale the chosen column j of A by $1/\sqrt{sp_j}$, and form a $d \times s$ matrix C with these scaled columns, then with high probability we have

$$\|\mathbf{x}^T C\|^2 \in \bigl[(1 - \varepsilon)\|\mathbf{x}^T A\|^2,\ (1 + \varepsilon)\|\mathbf{x}^T A\|^2\bigr], \quad \text{for all } \mathbf{x}.$$

Remark. The p_j are proportional to the squared lengths of columns of WA, but the actual sampling picks corresponding columns of A.

Proof. Let V be the space spanned by columns of A. Write vector $\mathbf{x}$ as $\mathbf{x} = \mathbf{z} + \mathbf{y}$, with $\mathbf{z} \in V$ and $\mathbf{y}$ orthogonal to V. Clearly, $\|\mathbf{x}^T A\| = \|\mathbf{z}^T A\|$ and $\|\mathbf{x}^T C\| = \|\mathbf{z}^T C\|$. So it suffices to prove the theorem assuming $\mathbf{x} \in V$.

[5] W is the pseudo-inverse of A.

[6] We call p_j the 'preconditioned length-squared' probabilities.

For any $\mathbf{x} \in V$, there is a $\mathbf{y} \in \operatorname{span}(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r)$ such that $\mathbf{x}^T = \mathbf{y}^T W$. Note that

$$WA = \sum_{t=1}^{r} \mathbf{v}_t \mathbf{v}_t^T,$$

so $\|WA\|_2^2 = 1$ and $\|WA\|_F^2 = r$. Apply Theorem 4.4 to WA with the ε of that theorem set to the ε here divided by $\sqrt{r}$. Since $s \geq r \ln n/\varepsilon^2$, the hypothesis of Theorem 4.4 is satisfied. Thus

$$\|WC(WC)^T - WA(WA)^T\|_2 \leq \varepsilon,$$

which implies

$$|\|\mathbf{y}^T WC\|^2 - \|\mathbf{y}^T WA\|^2| \leq \varepsilon \|\mathbf{y}\|^2 = \varepsilon \|\mathbf{y}^T (WA)\|^2, \quad \text{for all } \mathbf{y},$$

since, for $\mathbf{y} \in \operatorname{span}(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r)$,

$$\|\mathbf{y}^T WA\| = \|\mathbf{y}\|.$$

This implies the theorem. □

6. Subspace embeddings and applications

The methods of the previous section allow us to get relative errors in the residual, but can be more expensive computationally. They can be implemented in time dominated by the number of non-zeros of the input matrix times k/ϵ for a rank-k approximation. In this section we will see methods to improve this to only the order of the number of non-zeros in A (*i.e.* input sparsity time) plus lower-order terms that depend only on one of the dimensions of A. The guarantee is slightly weaker, but suffices for applications.

6.1. Very sparse subspace embeddings

Suppose A is an $n \times d$ given data matrix. For this section we assume that n is much larger than d, so A is a tall skinny matrix. For positive reals a, b, we say $a \cong_\varepsilon b$ if $a \in ((1-\varepsilon)b, (1+\varepsilon)b)$.

Definition 6.1. A $t \times n$ matrix S is a subspace embedding (for A) with error ε if

$$\|SA\mathbf{x}\| \cong_\varepsilon \|A\mathbf{x}\|, \quad \text{for all } \mathbf{x} \in \mathbb{R}^d.$$

We aim to bound t by a function of only d and ε. The point of subspace embedding is as follows: we can compute with SA instead of A, and if $t \ll n$, the matrix SA, which is $t \times d$, is much smaller than A. For example, if we want to find the top singular value (vector) of A within relative error ε, we can work on SA instead of A. The next theorem is from Clarkson and Woodruff (2013).

Theorem 6.2 (Clarkson and Woodruff). For a matrix A of rank r, the following matrix S is a subspace embedding with high probability provided $t \geq \mathrm{poly}(r/\varepsilon)$. For each $j \in \{1, 2, \ldots, n\}$, pick an $i \in \{1, 2, \ldots, d\}$ uniformly at random, and set $S_{ij} = \pm 1$ with probability 1/2. Then, with high probability,

$$\|SA\mathbf{x}\| \cong_{\varepsilon} \|A\mathbf{x}\|, \quad \text{for all } \mathbf{x} \in \mathbb{R}^d.$$

Remarks.

(i) Note that the theorem states that, with high probability, $\cong_{\varepsilon}$ *for all* x. It is a much stronger statement than saying 'for each x, $\cong_{\varepsilon}$ holds, with high probability'.

(ii) Note that the construction of S is 'oblivious', *i.e.*, independent of A.

(iii) Since S has only one non-zero entry per column, SA can be computed in time equal to a constant times the number of non-zero entries of A (often denoted $\mathrm{nnz}(A)$), using a sparse representation of the input matrix.

Subspace embeddings as presented here are due to Clarkson and Woodruff (2013) using the work of Dasgupta, Kumar and Sarlós (2010). For further improvements and applications, see Woodruff (2014).

6.2. *Elementary subspace embedding via Johnson–Lindenstrauss*

First, we prove a simpler theorem using the Johnson–Lindenstrauss random projection theorem. It says that we can project high-dimensional vectors to a lower-dimensional space and still preserve lengths approximately.

The projection $f : \mathbb{R}^n \to \mathbb{R}^k$ that we will examine (in fact, many related projections are known to work as well) is as follows. Pick k vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$ in $\mathbb{R}^n$, by choosing all the nk coordinates independently, each from the Gaussian distribution

$$\frac{1}{(2\pi)^{n/2}} \exp(-\|\mathbf{x}\|^2/2).$$

Then, for any vector $\mathbf{v}$ in $\mathbb{R}^n$, we define the projection $f(\mathbf{v})$ by

$$f(\mathbf{v}) = (\mathbf{u}_1 \cdot \mathbf{v}, \mathbf{u}_2 \cdot \mathbf{v}, \ldots, \mathbf{u}_k \cdot \mathbf{v}). \tag{6.1}$$

We will show that with high probability $\|f(\mathbf{v})\| \approx \sqrt{k}\|\mathbf{v}\|$, so if we have to find, say, $\|\mathbf{v}\|$, it suffices to find $\|f(\mathbf{v})\|$ (since the factor of $\sqrt{k}$ is known). The original proof was to project $\mathbf{v}$ onto a random k-dimensional subspace of $\mathbb{R}^n$. The proof is more complicated for that situation, since projecting onto a random subspace is not equivalent to picking k vectors independently at random and taking dot products. The bound below for a Gaussian random projection is simpler to prove, and still has the same set of applications

(Indyk and Motwani 1998, Arriaga and Vempala 1999, Dasgupta and Gupta 2003, Arriaga and Vempala 2006, Vempala 2004).

Theorem 6.3 (Johnson–Lindenstrauss). Let $\mathbf{v}$ be a fixed vector in $\mathbb{R}^n$ and let f be defined by (6.1). Then, for $\varepsilon \in (0,1)$,

$$\Pr\bigl(\bigl|\|f(\mathbf{v})\| - \sqrt{k}\|\mathbf{v}\|\bigr| \geq \varepsilon\sqrt{k}\|\mathbf{v}\|\bigr) \leq 2\,\mathrm{e}^{-(\epsilon^2-\epsilon^3)k/4},$$

where the probability is taken over the random draws of vectors $\mathbf{u}_i$ used to construct f.

Proof. See Dasgupta and Gupta (2003, Theorem 2.1). □

The theorem deals with one vector. However, subspace embeddings have to work for infinitely many vectors. We cannot do a simple union bound because the number of vectors is infinite. Instead, we use what is called an ε-net.

Definition 6.4. A set N of unit vectors is an ε*-net* for the unit sphere $S = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| = 1\}$ if, for any $\mathbf{x} \in S$, there exists $\mathbf{y} \in N$ such that $\|\mathbf{x} - \mathbf{y}\| \leq \varepsilon$.

The next lemma bounds the size of the ϵ-net for a sphere.

Lemma 6.5. There is an ε-net N with $|N| \leq (1 + 2/\varepsilon)^d$.

Proof. The proof is by taking a maximal set N of points that are pairwise ϵ-separated and using a volume argument (see *e.g.* Lemma 4.10 of Pisier 1989), that is, balls of radii $\epsilon/2$ centred at each point of the net are disjoint and the union of all these balls is contained in a ball of radius $1 + (\epsilon/2)$. Hence

$$|N| \leq \left(\frac{1+\epsilon/2}{\epsilon/2}\right)^d = \left(1 + \frac{2}{\varepsilon}\right)^d. \qquad \square$$

Now we may apply Theorem 6.3 as follows. Suppose V is a d-dimensional subspace of $\mathbb{R}^n$, for example, $V = \{A\mathbf{x}\}$ for an $n \times d$ matrix A. The set of unit vectors in V has an ε-net N of size at most $\mathrm{e}^{cd\ln(1/\varepsilon)}$. Choose $k \geq cd\ln(1/\varepsilon)/\varepsilon^2$ so that, for a single $\mathbf{x} \in V$, the probability that $\|M\mathbf{x}\| \not\cong_\varepsilon \sqrt{k}\|\mathbf{x}\|$ is at most $1/|N|^2$. Then, just by union bounds, we get

$$\Pr\bigl(\text{for all } \mathbf{x} \in N,\ \|M\mathbf{x}\| \cong_\varepsilon \sqrt{k}\|\mathbf{x}\|\bigr) \geq 1 - \frac{1}{|N|}.$$

Now $f(\mathbf{v})$ can be written as a matrix product, $f(\mathbf{v}) = M\mathbf{v}$, where M is a $k \times n$ matrix with $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$ as its rows. We claim that the above suffices to prove that $\|M\mathbf{x}\| \cong_{3\varepsilon} \sqrt{k}\|\mathbf{x}\|$ for *every* $x \in V$.

Here is a first attempt to prove the claim. For any $\mathbf{x} \in V$, there is some $\mathbf{y} \in N$ with $\|\mathbf{x} - \mathbf{y}\| \leq \varepsilon$. So, $\|M\mathbf{x}\| = \|M\mathbf{y}\| + \delta$, where $|\delta| \leq \|M(\mathbf{x} - \mathbf{y})\|$. But bounding $\|M(\mathbf{x} - \mathbf{y})\|$ needs a good bound on $\|M\|_2$ which we do not yet have. Instead, we will see that by repeated use of the ε-net, we can write $\mathbf{x}$ as a convergent series of linear combinations of elements in N.

First, there exists a $\mathbf{y}^{(1)} \in N$ with $\|\mathbf{x} - \mathbf{y}^{(1)}\| = \theta_1 \leq \varepsilon$. Now there exists a $\mathbf{y}^{(2)} \in N$ with

$$\left\| \mathbf{y}^{(2)} - \frac{\mathbf{x} - \mathbf{y}^{(1)}}{\theta_1} \right\| = \theta_2 \leq \varepsilon.$$

Continuing, we get that $\mathbf{x} = \mathbf{y}^{(1)} + \theta_1 \mathbf{y}^{(2)} + \theta_1\theta_2\mathbf{y}^{(3)} + \cdots$, where $\theta_i \in [0, \varepsilon]$. Therefore,

$$\begin{aligned}
\frac{1}{\sqrt{k}}\|M\mathbf{x}\| &= \frac{1}{\sqrt{k}}\big\|M(\mathbf{y}^{(1)} + \theta_1\mathbf{y}^{(2)} + \cdots)\big\| \\
&\geq \frac{1}{\sqrt{k}}\big(\|M\mathbf{y}^{(1)}\| - \theta_1\|M\mathbf{y}^{(2)}\| - \theta_1\theta_2\|M\mathbf{y}^{(3)}\| - \cdots\big) \\
&\geq (1-\varepsilon) - (1+\varepsilon)(\theta_1 + \theta_1\theta_2 + \theta_1\theta_2\theta_3 + \cdots) \\
&\geq 1 - 3\varepsilon.
\end{aligned}$$

A similar argument also proves an upper bound of $1 + 3\varepsilon$ on $\|M\mathbf{x}\|$.

Now we return to the situation where A is an $n \times d$ matrix of rank r. The set $\{A\mathbf{x} : \mathbf{x} \in \mathbb{R}^d\}$ is an r-dimensional subspace of $\mathbb{R}^n$. Hence, by the above, as long as $k \geq \text{poly}(r/\varepsilon)$, we deduce that M is a subspace embedding for A. But M is dense, and computing MA is costly (and in general, superlinear).

6.3. Sparse embeddings

Now consider the sparse matrix S defined in Theorem 6.2. Let V be the subspace spanned by the columns of A. Unfortunately, we cannot carry out the same sort of argument as above, because it is not true that for every (individual) $\mathbf{x} \in V$ we have

$$\Pr(\|S\mathbf{x}\| \not\cong_\varepsilon \|\mathbf{x}\|) \leq \exp(-cd).$$

Counter-examples are 'lumpy' $\mathbf{x}$ with some 'big' coordinates: for example, if $x_1 = x_2 = 1/\sqrt{2}$, with probability poly$(1/t)$, we could have $S_{11}, S_{12} \neq 0$, whence $\|S\mathbf{x}\| \not\cong_\varepsilon |\mathbf{x}|$. On the other hand, 'non-lumpy' $\mathbf{x}$ – with, say, all $\|x_i\| \leq c\|\mathbf{x}\|/\sqrt{n}$ – can be shown to satisfy $\Pr(\|S\mathbf{x}\| \not\cong_\varepsilon \|\mathbf{x}\|) \leq \exp(-cd)$. We do not do this since it is not required for our proof.

Here is a simple and crucial point.

Lemma 6.6. Suppose $\alpha \in [0, 1]$. There is a subset H of $\{1, 2, \ldots, n\}$ with $|H| \leq d/\alpha$ such that

$$|(A\mathbf{x})_i| \leq \sqrt{\alpha}\|A\mathbf{x}\|, \quad \text{for all } \mathbf{x} \text{ and all } i \notin H.$$

The lemma asserts that only coordinates in H (for heavy) can be bigger than $\sqrt{\alpha}\|A\mathbf{x}\|$ in absolute value. Note that H is independent of $\mathbf{x}$.

Proof. Let $r = \operatorname{rank}(A)$. Form an $n \times r$ matrix U with the left singular vectors of A as its columns. Since $\|U\|_F^2 = r$, we have that[7]

$$H = \{i : \|U(i,:)\| \geq \sqrt{\alpha}\} \text{ satisfies } |H| \leq r/\alpha \leq d/\alpha. \tag{6.2}$$

We assert that this H satisfies the conclusion of the lemma. For any $\mathbf{x}$, we have that $A\mathbf{x}$ is a linear combination of the columns of U, and so $A\mathbf{x} = U\mathbf{z}$ for some $\mathbf{z}$. Since the columns of U are orthonormal, we have $\|\mathbf{z}\| = \|A\mathbf{x}\|$. Also $(A\mathbf{x})_i = U(i,:)^T\mathbf{z}$, and so for $i \notin H$,

$$|(A\mathbf{x})_i| \leq \|U(i,:)\|\|\mathbf{z}\| \leq \sqrt{\alpha}\|A\mathbf{x}\|. \qquad \square$$

Let $[n] \setminus H = L$ (L for light). We can write any $\mathbf{y}$ in the column space of A as the sum of two vectors $\mathbf{y}^H, \mathbf{y}^L$, where $y_i^H = y_i$ for $i \in H$ and $y_i^H = 0$ for $i \in L$, and similarly $\mathbf{y}^L$ is $\mathbf{y}$ restricted to L. Then

$$\|S\mathbf{y}\|^2 = \|S\mathbf{y}^H\|^2 + \|S\mathbf{y}^L\|^2 + 2(S\mathbf{y}^H)^T(S\mathbf{y}^L). \tag{6.3}$$

We deal with each of these three terms. For each $j \in \{1, 2, \ldots, n\}$, we picked an $i \in \{1, 2, \ldots, t\}$ at random and set $S_{i,j} = \pm 1$. It will be convenient to call the i we picked the 'hash of j'. Since $n \gg t$, $j \in \{1, 2, \ldots, n\}$ and $i \in \{1, 2, \ldots, t\}$, it resembles a random hash function.

Lemma 6.7. With probability at least $1 - (|H|^2/t)$, we have $\|S\mathbf{y}^H\| = \|\mathbf{y}^H\|$ when $\mathbf{y} = A\mathbf{x}$.

Proof. The probability that the hashes of two different $j \in H$ are the same is at most $|H|^2/t$, and we will choose $t > c|H|^2$, so the probability of this hash collision is small. (Recall the 'birthday paradox'.) Note that this event has nothing to do with a particular $\mathbf{y}$. If there is no hash collision, then the submatrix of S in the columns corresponding to H is just a permutation matrix (with signs), and the lemma follows. □

For bounding the other two terms of (6.3), the following theorem due to Dasgupta *et al.* (2010) (which we do not prove here) is useful. Call a vector $\mathbf{y} = A\mathbf{x}$ non-lumpy if no coordinate of $\mathbf{y}$ is too big in absolute value. The theorem asserts that with high probability, for all non-lumpy vectors simultaneously, we have bounds on the difference $\|S\mathbf{y}\| - \|\mathbf{y}\|$. Since we have already taken care of the heavy coordinates, this will help us take care of the rest.

Theorem 6.8 (Theorem 2 of Dasgupta *et al.* 2010). Suppose $\varepsilon, \delta \in (0, 1)$ and $t \geq 12\log(1/\delta)/\varepsilon^2$, and let S be a $t \times n$ matrix chosen as in Theorem 6.2. Suppose $\mathbf{y}$ is any vector in $\mathbb{R}^n$ with $|y_i| \leq \sqrt{\alpha}$, where $1/\alpha = 16\log(1/\delta)\log^2(t/\delta)/\varepsilon$. Then

$$\Pr\bigl(\bigl|\|S\mathbf{y}\| - \|\mathbf{y}\|\bigr| \geq \varepsilon\bigr) \leq 3\delta.$$

[7] $U(i,:)$ is the ith row of U.

Remark. The theorem is for one vector $\mathbf{y}$. It is a sparse version of Johnson–Lindenstrauss, and finds other uses as well.

To prove that the probability result holds simultaneously for all $\mathbf{y}$, we apply Theorem 6.8 as follows.

(i) We examine the construction of an ε-net to see that if we want to cover only non-lumpy $\mathbf{y}$ (each $|y_i| \le \sqrt{\alpha}$) in a d-dimensional subspace of $\mathbb{R}^n$ (like $\{A\mathbf{x}\}$), then we can get the same size net with all coordinates of vectors in the net having absolute value at most $2\sqrt{\alpha}$.

(ii) We choose $\delta \le \exp(-cd\ln d/\varepsilon^2)$ and apply Theorem 6.8 to show that, for each non-lumpy $\mathbf{y}$ *in the net*, the probability of failure is at most δ. Then use the union bound over all elements of the net.

(iii) Finally, we use the argument of Section 6.2 to write any non-lumpy $\mathbf{y}$ as a convergent series of linear combinations of net elements, and use this to conclude that the following holds.

Theorem 6.9. Suppose $\varepsilon \in (0,1)$ and $t = \mathrm{poly}(d/\varepsilon)$ and $\alpha = \mathrm{poly}(\varepsilon/d)$ satisfying[8]

$$t \ge \frac{d}{\alpha\varepsilon^2}, \quad \frac{1}{\alpha} \ge c(\log^2 t)(d^3(\ln d)^3/\varepsilon^6). \tag{6.4}$$

Let S be a $t \times n$ matrix chosen as in Theorem 6.2. Then,

$$\Pr\bigl(\text{for all } \mathbf{y},\ \text{with } |y_i| \le \sqrt{\alpha},\ \|S\mathbf{y}\| \cong_\varepsilon \|\mathbf{y}\|\bigr) \ge 1-\varepsilon.$$

Theorem 6.9 can be used to imply immediately that

$$|\|S\mathbf{y}^L\| - \|\mathbf{y}^L\|| \le c\varepsilon, \quad \text{for all } \mathbf{y} \in \{Ax\}.$$

This takes care of the second term on the right-hand side of (6.3).

The third term requires a further argument. Let

$$L' = \{j \in L : \text{the hash of } j = \text{the hash of some } j' \in H\}.$$

Note that L' depends only upon the choices made for S, that is, it does not depend on any particular y. We make three assertions, all holding with high probability:

$$|(S\mathbf{y}^H)^T(S\mathbf{y}^L)| = |(S\mathbf{y}^H)^T(S\mathbf{y}^{L'})| \le \|S\mathbf{y}^H\|\,\|S\mathbf{y}^{L'}\| \le \|S\mathbf{y}^{L'}\|, \tag{6.5}$$

$$\|S\mathbf{y}^{L'}\| \le O(\varepsilon) + \|\mathbf{y}^{L'}\|, \tag{6.6}$$

$$\|\mathbf{y}^{L'}\| \le \varepsilon. \tag{6.7}$$

Statement (6.5) is easy to check. For (6.6), note that for a $j \in L'$ conditioned on its hash being one of the $|H|$ hashes of $j' \in H$, it is uniform random. So, S restricted to columns in L' is an $|H| \times |L'|$ matrix constructed the

[8] It is easy to see that t, α satisfying these inequalities exist.

same way as the whole of S: pick a random row for each column and set the entry in that row and column to ± 1 with probability $1/2$ each. So we may apply Theorem 6.9 to this situation to get (6.6). If the actual $|H|$ is much smaller than the upper bound in (6.2), we can just augment it to get to size $cd\ln d/\varepsilon^2$. Statement (6.7) will follow using a version of Bernstein's inequality (often called Freedman's inequality) from probability theory, as follows.

Lemma 6.10. Suppose $Y_1, Y_2, \dots, Y_n$ are independent real-valued random variables with $|Y_i| \le M$. Let $b = \mathrm{Var}(\sum_{i=1}^n Y_i)$, and let a be any positive real number. Then

$$\Pr\biggl(\biggl|\sum_{i=1}^n Y_i\biggr| \ge a\biggr) \le 2\exp\biggl(-\frac{a^2}{2(aM+b)}\biggr).$$

Proof. See Freedman (1975). □

To prove (6.7), consider a single $\mathbf{y}$ first. Since for $j \in L$, $y_j^{L'} = y_j \mathbf{1}(j \in L')$ (where $\mathbf{1}$ is an indicator random variable), we obtain

$$\mathrm{Var}((y_j^{L'})^2) \le y_j^4 \frac{|H|}{t} \le y_j^4 \frac{d}{t\alpha} \le \varepsilon^2 y_j^4,$$

using (6.2) and (6.4). Thus

$$\mathrm{Var}\biggl(\sum_{j\in L} (y_j^{L'})^2\biggr) \le \varepsilon^2 \sum_j y_j^4 \le \varepsilon^2 |\mathbf{y}|^2 \alpha \le \alpha\varepsilon^2.$$

Hence Lemma 6.10 with $a = \varepsilon^2$ implies that

$$\Pr(\|\mathbf{y}^{L'}\|^2 \ge \varepsilon^2) \le 2\exp(-c\varepsilon^4/(\varepsilon^2\alpha + \varepsilon^2\alpha)) \le 2\exp(-cd\ln d/\varepsilon^2).$$

This probability is small enough that we can take a union bound over an ε-net and then extend to all vectors again by expressing the vector as a convergent series of net vectors.

7. Conclusion

The goal of this survey is to convey some core ideas of using random sampling to obtain fast algorithms for matrix problems in numerical linear algebra. These algorithms work for arbitrary input matrices (and tensors where appropriate). They are randomized approximation algorithms with a probability of failure that can be controlled by using a larger sample.

The problems and their randomized algorithms have many applications, including machine learning, combinatorial optimization, solving linear systems and graph decompositions. For a more extensive discussion of these applications we refer the reader to Kannan and Vempala (2009), Mahoney (2011), Vishnoi (2013) and Woodruff (2014).

Randomized numerical linear algebra is now a large and active field. The basic theorems presented here have also been extended in many ways. We mention some of them briefly.

Achlioptas and McSherry (2007) gave a different method for additive error low-rank approximation based on sparsifying the original matrix by sampling each entry independently. The CUR approximation has been improved by Boutsidis and Woodruff (2014). For multiplicative error low-rank approximation, the methods based on iterative length-squared sampling, volume sampling and preconditioned length-squared sampling all require two or more passes over the data. Sarlós (2006) gave the first algorithm that uses only one pass over the data and gives a multiplicative error. Deshpande and Rademacher (2010) showed that volume sampling can be implemented efficiently with random projections, while Anari *et al.* (2016) gave a Markov chain method to sample efficiently from the volume sampling distribution. Drineas, Magdon-Ismail, Mahoney and Woodruff (2012) improved the complexity of preconditioned length-squared sampling (leverage score sampling). While the original method of Spielman and Srivastava for graph sparsification was a polynomial-time algorithm establishing an asymptotically optimal bound on the size of the sparse graph, Lee and Sun (2015) have improved the running time to almost linear. For sparse embeddings, the work of Clarkson and Woodruff described here has been improved by Meng and Mahoney (2013), Nelson and Nguyên (2013) and further by Cohen (2016). In related work, Tropp (2011) and Kane and Nelson (2014) have improved the sparsity of random projection matrices for Euclidean length preservation.

REFERENCES

D. Achlioptas and F. McSherry (2007), 'Fast computation of low-rank matrix approximations', *J. Assoc. Comput. Mach.* **54**, 9.

R. Ahlswede and A. Winter (2002), 'Strong converse for identification via quantum channels', *IEEE Trans. Inform. Theory* **48**, 568–579.

N. Anari, S. Gharan and A. Rezaei (2016), Monte Carlo Markov chain algorithms for sampling strongly Rayleigh distributions and determinantal point processes. In *COLT 2016: 29th Conference on Learning Theory*, pp. 103–115.

R. Arriaga and S. Vempala (1999), An algorithmic theory of learning: Robust concepts and random projection. In *FOCS 1999: 40th Annual Symposium on Foundations of Computer Science*, pp. 616–623.

R. Arriaga and S. Vempala (2006), 'An algorithmic theory of learning: Robust concepts and random projection', *Machine Learning* **63**, 161–182.

M. Berry, S. Pulatova and G. Stewart (2004), Computing sparse reduced-rank approximations to sparse matrices. Technical report, UMIACS, University of Maryland.

R. Bhatia (1996), *Matrix Analysis*, Vol. 169 of Graduate Texts in Mathematics, Springer.

C. Boutsidis and D. Woodruff (2014), Optimal CUR matrix decompositions. In *STOC 2014: Symposium on Theory of Computing*, pp. 353–362.

K. Clarkson and D. Woodruff (2009), Numerical linear algebra in the streaming model. In *STOC 2009: 41st Annual ACM Symposium on Theory of Computing*, pp. 205–214.

K. Clarkson and D. Woodruff (2013), Low rank approximation and regression in input sparsity time. In *STOC 2013: Symposium on Theory of Computing Conference*, pp. 81–90.

M. Cohen (2016), Nearly tight oblivious subspace embeddings by trace inequalities. In *SODA 2016: 27th Annual ACM–SIAM Symposium on Discrete Algorithms*, pp. 278–287.

M. Cohen, Y. Lee, C. Musco, C. Musco, R. Peng and A. Sidford (2015), Uniform sampling for matrix approximation. In *ITCS 2015: Conference on Innovations in Theoretical Computer Science*, pp. 181–190.

M. Cohen, C. Musco and C. Musco (2017), Ridge leverage scores for low-rank approximation. In *SODA 2017: 27th Annual ACM–SIAM Symposium on Discrete Algorithms*, pp. 1758–1777.

A. Dasgupta, R. Kumar and T. Sarlós (2010), A sparse Johnson–Lindenstrauss transform. In *STOC 2010: 42nd ACM Symposium on Theory of Computing*, pp. 341–350.

S. Dasgupta and A. Gupta (2003), ‘An elementary proof of a theorem of Johnson and Lindenstrauss’, *Random Struct. Alg.* **22**, 60–65.

A. Deshpande and L. Rademacher (2010), Efficient volume sampling for row/column subset selection. In *FOCS 2010: 51th Annual IEEE Symposium on Foundations of Computer Science*, pp. 329–338.

A. Deshpande and S. Vempala (2006), Adaptive sampling and fast low-rank matrix approximation. In *APPROX–RANDOM 2006*, Vol. 4110 of Lecture Notes in Computer Science, Springer, pp. 292–303.

A. Deshpande, L. Rademacher, S. Vempala and G. Wang (2006), ‘Matrix approximation and projective clustering via volume sampling’, *Theory of Computing* **2**, 225–247.

P. Drineas, R. Kannan and M. Mahoney (2006), ‘Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix’, *SIAM J. Comput.* **36**, 158–183.

P. Drineas, M. Magdon-Ismail, M. Mahoney and D. Woodruff (2012), ‘Fast approximation of matrix coherence and statistical leverage’, *J. Mach. Learn. Res.* **13**, 3475–3506.

P. Drineas, M. Mahoney and S. Muthukrishnan (2008), ‘Relative-error CUR matrix decompositions’, *SIAM J. Matrix Anal. Appl.* **30**, 844–881.

D. Freedman (1975), ‘On tail probabilities for martingales’, *Ann. Probab.* **3**, 100–118.

A. Frieze, R. Kannan and S. Vempala (1998), Fast Monte-Carlo algorithms for finding low-rank approximations. In *FOCS 1998: 39th Annual Symposium on Foundations of Computer Science*, pp. 370–378.

A. Frieze, R. Kannan and S. Vempala (2004), ‘Fast Monte-Carlo algorithms for finding low-rank approximations’, *J. Assoc. Comput. Mach.* **51**, 1025–1041.

G. Golub and C. Van Loan (1996), *Matrix Computations*, third edition, Johns Hopkins University Press.

S. Goreinov and E. Tyrtyshnikov (2001), 'The maximum-volume concept in approximation by low-rank matrices', *Contemp. Math.* **280**, 47–51.

S. Goreinov, E. Tyrtyshnikov and N. Zamarashkin (1997), 'A theory of pseudoskeleton approximations', *Linear Algebra Appl.* **261**, 1–21.

N. Halko, P. Martinsson and J. Tropp (2011), 'Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions', *SIAM Review* **53**, 217–288.

C. Hillar and L. Lim (2013), 'Most tensor problems are NP-hard', *J. Assoc. Comput. Mach.* **60**, 45.

R. Horn and C. Johnson (2012), *Matrix Analysis*, second edition, Cambridge University Press.

P. Indyk and R. Motwani (1998), Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC 1998: 30th Annual ACM Symposium on Theory of Computing*, pp. 604–613.

D. Kane and J. Nelson (2014), 'Sparser Johnson–Lindenstrauss transforms', *J. Assoc. Comput. Mach.* **61**, 4.

R. Kannan and S. Vempala (2009), 'Spectral algorithms', *Found. Trends Theoret. Comput. Sci.* **4**, 157–288.

Y. Lee and H. Sun (2015), Constructing linear-sized spectral sparsification in almost-linear time. In *FOCS 2015: IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 250–269.

M. Li, G. Miller and R. Peng (2013), Iterative row sampling. In *FOCS 2013: 54th Annual IEEE Symposium on Foundations of Computer Science*, pp. 127–136.

M. Mahoney (2011), 'Randomized algorithms for matrices and data', *Found. Trends Mach. Learning* **3**, 123–224.

X. Meng and M. Mahoney (2013), Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *STOC 2013: Symposium on Theory of Computing Conference*, pp. 91–100.

J. Nelson and H. Nguyên (2013), OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *FOCS 2013: IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 117–126.

G. Pisier (1989), *The Volume of Convex Bodies and Banach Space Geometry*, Cambridge University Press.

M. Rudelson and R. Vershynin (2007), 'Sampling from large matrices: An approach through geometric functional analysis', *J. Assoc. Comput. Mach.* **54**, 21.

T. Sarlós (2006), Improved approximation algorithms for large matrices via random projections. In *FOCS 2006: 47th Annual IEEE Symposium on Foundations of Computer Science*, pp. 143–152.

D. Spielman and N. Srivastava (2011), 'Graph sparsification by effective resistances', *SIAM J. Comput.* **40**, 1913–1926.

G. Stewart (1999), 'Four algorithms for the efficient computation of truncated QR approximations to a sparse matrix', *Numer. Math.* **83**, 313–323.

G. Stewart (2004), Error analysis of the quasi-Gram–Schmidt algorithm. Technical report, UMIACS, University of Maryland.

J. Tropp (2011), 'Improved analysis of the subsampled randomized Hadamard transform', *Adv. Adapt. Data Anal.* **3**, 115–126.

S. Vempala (2004), *The Random Projection Method*, Vol. 65 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, DIMACS/AMS.

N. Vishnoi (2013), 'Lx = b', *Found. Trends Theoret. Comput. Sci.* **8**, 1–141.

D. Woodruff (2014), 'Sketching as a tool for numerical linear algebra', *Found. Trends Theoret. Comput. Sci.* **10**, 1–157.