Fast Monte Carlo Algorithms for Matrices I: Approximating Matrix Multiplication [*]

Petros Drineas [†] Ravi Kannan [‡§] Michael W. Mahoney [¶]

Technical Report, YALEU/DCS/TR-1269, February 2004.

Abstract

Motivated by applications in which the data may be formulated as a matrix, we consider algorithms for several common Linear Algebra problems. These algorithms make more efficient use of computational resources, such as the computation time, Random Access Memory (RAM), and the number of passes over the data, than do previously known algorithms for these problems; in addition, they achieve their greater efficiency at the cost of some error. In this paper, we devise two algorithms for the Matrix Multiplication Problem. Suppose A and B (which are $m \times n$ and $n \times p$ respectively) are the two input matrices. In our main algorithm, we perform $c = O(1)$ independent trials, where in each trial we randomly sample an element of $\{1, 2, \ldots n\}$ with an appropriate probability distribution $\mathcal{P}$ on $\{1, 2, \ldots n\}$. We form a $m \times c$ matrix C consisting of the sampled columns of A, each scaled appropriately, and we form a $c \times n$ matrix R using the same rows of B, again scaled appropriately. The choice of $\mathcal{P}$ and the column and row scaling are crucial features of the algorithm. When these are chosen judiciously, we show that CR is a good approximation to AB; more precisely, we show that, with high probability,

$$\|AB - CR\|_F \in O(\|A\|_F \, \|B\|_F / \sqrt{c}),$$

where $\|\cdot\|_F$ denotes the Frobenius norm, i.e., $\|A\|_F^2 = \sum_{i,j} A_{ij}^2$. This algorithm can be implemented without storing the matrices A and B in RAM, provided it can make two passes over the matrices stored in external memory and use $O(m + p)$ additional RAM memory to construct C and R. We then present a second matrix multiplication algorithm which is similar in spirit to our main algorithm. In addition, we present a model (the Pass-Efficient model) in which the efficiency of these and other approximate matrix algorithms may be studied and which we argue is well-suited to many applications involving massive data sets. In this model, the scarce computational resources are the number of passes over the data and the additional space and time required by the algorithm. The input matrices may be presented in any order of the entries (and not just row or column order), as is the case in many applications where, e.g., the data has been written in by multiple agents. In addition, the input matrices may be presented in a sparse representation, where only the non-zero entries are written.

[*] A preliminary version of this paper appeared as "Fast Monte-Carlo algorithms for approximate matrix multiplication" by P. Drineas and R. Kannan in the *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science*, 2001, pp. 452–459.

[†] Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York 12180, drinep@cs.rpi.edu

[‡] Department of Computer Science, Yale University, New Haven, Connecticut, USA 06520, kannan@cs.yale.edu

[§] Supported in part by a grant from the NSF.

[¶] Department of Mathematics, Yale University, New Haven, Connecticut, USA 06520, mahoney@cs.yale.edu

1 Introduction

We are interested in developing and analyzing fast Monte Carlo algorithms for performing useful computations on large matrices. Examples of such computations include matrix multiplication, the computation of the Singular Value Decomposition of a matrix, and the computation of compressed approximate decompositions of a matrix. In this paper, we present a computational model for computing on massive data sets (the Pass-Efficient model) and in which our algorithms may naturally be formulated; we also present two algorithms for the approximation of the product of two matrices. In a second paper we present two algorithms for the computation of low-rank approximations to a matrix [12]. Finally, in a third paper we present two algorithms to compute a compressed approximate decomposition to a matrix that has several appealing properties [13]. We expect our algorithms to be useful in many applications where data sets are modeled by matrices and are extremely large. For example, in Information Retrieval and Data Mining (two rapidly growing areas of research in computer science and scientific computation that build on techniques and theories from fields such as statistics, linear algebra, database theory, pattern recognition and learning theory) a large collection of n objects, e.g., documents, genomes, images, or web pages, is implicitly presented as a set of points in an m-dimensional Euclidean space, where m is the number of features that describe the object; thus, this collection may be represented by an $m \times n$ matrix A, the columns of which are the object vectors and the rows of which are the feature vectors.

Recent interest in computing with massive data sets has led to the development of computational models in which the usual notions of time-efficiency and space-efficiency have been modified [23, 19, 3, 14, 11, 5]. In the applications that motivate these data-streaming models [19, 5], e.g., the observational sciences and the monitoring and operation of large networked systems, the data sets are much too large to fit into main memory. Thus, they are either not stored or are stored in a secondary storage device which may be read sequentially as a data stream but for which random access is very expensive. Typically, algorithms that compute on a data stream examine the data stream, keep a small "sketch" of the data, and perform computations on the sketch. Thus, these algorithms are usually randomized and approximate, and their performance is evaluated by considering resources such as the time to process an item in the data stream, the number of passes over the data, the additional workspace and additional time required, and the quality of the approximations returned. (Note that in some cases the term "data-streaming model" refers to a model in which only a single pass over the data is allowed [19, 5].)

The motivation for our particular "pass-efficient" approach is that in modern computers the amount of disk storage (external memory) has increased enormously, while RAM and computing speeds have increased, yet at a substantially slower pace. Thus, we have the ability to store large amounts of data, but not in RAM, and we do not have the computational ability to process these data with algorithms that require superlinear time. In order to provide a framework in which to view the algorithms presented herein, we first introduce and describe the Pass-Efficient model of data-streaming computation [11]. In the Pass-Efficient model the computational resources are the number of passes over the data and the additional RAM space and the additional time required. Thus, our algorithms are quite different from traditional numerical analysis approaches and generally fit within the following framework. Our algorithms will be allowed to read the matrices from external storage a few, e.g., one or two or three, times and keep a small randomly-chosen and rapidly-computable "sketch" of the matrices in RAM. Our algorithms will also be permitted additional space and time that is linear or sublinear in the number of data elements in order to perform computations on the "sketch". The results of these computations will be returned as approximations to the solution of the original problem.

In all of our algorithms, an important implementational issue will be how to form the random sample. An obvious choice is to use uniform sampling, where each data object is equally likely to be picked. Uniform sampling can be performed blindly, in which case the sample to be chosen can be decided before seeing the data. Even when the number N of data elements is not known in advance an element can be selected uniformly at random in one pass over the data; see Lemma 1. Uniform sampling fits within our framework and is useful for certain (restricted) classes of problems. To obtain much more generality, we will sample according to a judiciously chosen (and data-dependent) set of nonuniform sampling probabilities. This nonuniform sampling, in which in the first pass through the data we compute sampling probabilities (e.g., we may keep rows or columns of a data matrix with probability proportional to the square of their lengths) and in the second pass we draw the sample, offers substantial gains. For example, it allows us to approximately solve problems in sparse matrices as well as dense matrices.

The idea of sampling rows or columns of matrices in order to approximate various operations is not new; indeed, a motivation for our main matrix multiplication algorithm came from [15]. In this paper and accompanying work [12, 13], we extend those ideas and develop algorithms with provable error bounds for a variety of matrix operations. One of the main contributions of our work is to demonstrate that a "sketch" consisting of a small judiciously chosen random sample of rows and/or columns of the input matrix or matrices is adequate for provably rapid and efficient approximation of several common matrix operations. We believe that the underlying principle of using nonuniform sampling to create "sketches" of the data in a small number of passes (and "pass-efficient" approaches more generally) constitutes an appealing and fruitful direction for algorithmic research in order to address the size and nature of modern data sets.

In the present paper, we present two simple and intuitive algorithms which, when given an $m \times n$ matrix A and an $n \times p$ matrix B, compute an approximation to the product AB. In the first algorithm, the BasicMatrixMultiplication algorithm of Section 4, we perform $c = O(1)$ independent trials, where in each trial we randomly sample an element of $\{1, 2, \dots n\}$ with an appropriate probability distribution $\mathcal{P}$ on $\{1, 2, \dots n\}$. We form a $m \times c$ matrix C consisting of the sampled columns of A, each scaled appropriately, and we form a $c \times n$ matrix R using the same rows of B, again scaled appropriately. The choice of $\mathcal{P}$ and the column and row scaling are crucial features of the algorithm. When these are chosen judiciously, we show that CR is a good approximation to AB; more precisely, we show that

$$\|AB - CR\|_F \in O(\, \|A\|_F \, \|B\|_F \, / \sqrt{c}),$$

where $\|\cdot\|_F$ denotes the Frobenius norm, i.e., $\|A\|_F^2 = \sum_{i,j} A_{ij}^2$, holds in expectation and with high probability. Thus, in particular, when $B = A^T$ we have that if $c = \Omega(1/\epsilon^2)$ then $\left\|AA^T - CC^T\right\|_F \leq \epsilon \|A\|_F^2$ holds with high probability. This algorithm can be implemented without storing the matrices A and B in RAM, provided it can make two passes over the matrices stored in external memory and use $O(m + p)$ additional RAM memory; thus it will be efficient within the Pass-Efficient model. In the second algorithm, the ElementwiseMatrixMultiplication algorithm of Section 5, which is an extension of ideas from [2, 1], elements of A and B, rather than columns and rows, are randomly either zeroed out or kept and rescaled, thereby constructing matrices $\tilde{A}$ and $\tilde{B}$. Although this algorithm lacks a useful bound on $\left\|AB - \tilde{A}\tilde{B}\right\|_F$, under appropriate assumptions a bound on the spectral norm of the form

$$\left\|AB - \tilde{A}\tilde{B}\right\|_2 \in O(\, \|A\|_F \, \|B\|_F \, / \sqrt{c})$$

holds with high probability.

After this introduction, we provide in Section 2 a review of the relevant linear algebra and in Section 3 we introduce the Pass-Efficient model of data-streaming computation and discuss several technical sampling lemmas. In Section 4 we introduce and analyze in detail the BasicMatrixMultiplication algorithm to approximate the product of two matrices. Then, in Section 5 we describe and analyze the ElementwiseMatrixMultiplication algorithm which is based on the ideas of [2, 1]. Finally, in Section 6 we provide a discussion and conclusion. In Appendix A, we provide further analysis of the BasicMatrixMultiplication algorithm.

2 Review of Linear Algebra

This section contains a review of some linear algebra that will be useful throughout the paper. For more detail, see [18, 20, 25, 6] and references therein.

For a vector $x \in \mathbb{R}^n$ we let $|x| = \left(\sum_{i=1}^n |x_i|^2\right)^{1/2}$ denote its Euclidean length. For a matrix $A \in \mathbb{R}^{m\times n}$ we let $A^{(j)}$, $j = 1, \ldots, n$, denote the j-th column of A as a column vector and $A_{(i)}$, $i = 1, \ldots, m$, denote the i-th row of A as a row vector. We denote matrix norms by $\|A\|_\xi$, using subscripts to distinguish between various norms. Of particular interest will be the Frobenius norm which is defined by

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2}, \tag{1}$$

and the spectral norm which is defined by

$$\|A\|_2 = \sup_{x\in\mathbb{R}^n,\ x\neq 0} \frac{|Ax|}{|x|}. \tag{2}$$

These norms are related to each other as: $\|A\|_2 \leq \|A\|_F \leq \sqrt{n}\,\|A\|_2$. Both of these norms provide a measure of the "size" of the matrix A.

3 The Pass-Efficient Model and Sampling Lemmas

In this section, we informally define a computational model in which the computational resources are the number of passes over the data and the additional space and additional time required. In addition, we present several technical sampling lemmas.

3.1 The Pass-Efficient Model

The Pass-Efficient model of data-streaming computation is a model that is motivated by the observation that in modern computers the amount of disk storage, i.e., sequential access memory, has increased very rapidly while random access memory (RAM) and computing speeds have increased at a substantially slower pace [11]. Thus, one has the ability to store very large amounts of data but not have random access to the data. Additionally, processing the data with algorithms that take low polynomial time or linear time with large constants is prohibitive.

To model this phenomenon, we consider the Pass-Efficient model, in which the three computational resources of interest are number of passes over the data and the additional space and time required [11]. The data are assumed to be stored in an external disk space, to consist of elements whose size is bounded by a constant, and to be presented to an algorithm on a read-only tape. The only access an algorithm has to the data is via a pass, where a *pass* over the data is a sequential read of the entire input from disk where only a constant amount of processing time is permitted per bit read. Note that this is a more restrictive notion of a pass over the data than in other data-streaming models [23, 19, 14]; in particular, in the Pass-Efficient model

Select Algorithm

Input: $\{a_1, \ldots, a_n\}$, $a_i \geq 0$, read in one pass, i.e., one sequential read, over the data.

Output: i^*, a_{i^*}.

- $D = 0$.
- For $i = 1$ to n,
  - $D = D + a_i$.
  - With probability a_i/D, let $i^* = i$ and $a_{i^*} = a_i$.
- Return i^*, a_{i^*}.

Figure 1: The Select Algorithm

only a constant rather than a logarithmic (in the data input length) amount of computation is permitted per bit read. In addition to the external disk space to store the data and to a small number of passes over the data, an algorithm in the Pass-Efficient model is permitted to use *additional RAM space* and *additional computation time.* An algorithm operating in this model is considered *pass-efficient* if it requires a small constant number of passes and additional space and time which are sublinear in the length of the data stream in order to compute a "description" of the solution, which is then returned by the algorithm. A *description* of the solution is either an explicit solution (if that is possible within the specified additional space and time) or an implicit representation of the solution that can be computed in the allotted additional space and time, and that can be expanded into an explicit solution with the additional expense of one pass over the data and linear (in the data input length) additional space and time. Note that depending on the application, this last step may or may not be necessary. Note also that if the data are represented by a $m \times n$ matrix, e.g., n vectors $a_i \in \mathbb{R}^m$, $i = 1, \ldots, n$, then the data stream has length $O(mn)$ and an algorithm which uses additional space and time that is linear in the number of data points or in the dimensionality of the data points, i.e., that is $O(m)$ or $O(n)$, is sublinear in the length of the data stream and thus is pass-efficient. We will be primarily interested in models that require additional space and time that is either $O(m+n)$ or constant with respect to m and n.

The *sparse-unordered representation* of data is a form of data representation in which each element of the data stream consists of a pair $((i,j), A_{ij})$ where the elements in the data stream may be unordered with respect to the indices (i,j) and only the nonzero elements of the matrix A need to be presented. This very general form is suited to applications where, e.g., multiple agents may write parts of a matrix to a central database and where one cannot make assumptions about the rules for write-conflict resolution. The data stream read by algorithms in the Pass-Efficient model is assumed to be presented in the *sparse-unordered representation.* Other related methods of data representation have been studied within the data-streaming context; see, e.g., [17] for applications to the problem of dynamic histogram maintenance.

3.2 Sampling Lemmas

In this section we present two sampling lemmas that will be used by our algorithms. Consider the Select algorithm presented in Figure 1. The following lemma establishes that in one pass over the data one can sample an element according to certain probability distributions.

Lemma 1 *Suppose that* $\{a_1, \ldots, a_n\}$, $a_i \geq 0$, *are read in one pass, i.e., one sequential read over the data, by the* SELECT *algorithm. Then the* SELECT *algorithm requires* $O(1)$ *additional storage space and returns* i^* *such that* $\mathbf{Pr}[i^* = i] = a_i / \sum_{i'=1}^{n} a_{i'}$.

Proof: First, note that retaining the selected value and the running sum requires $O(1)$ additional space. The remainder of the proof is by induction. After reading the first element a_1, $i^* = 1$ with probability $a_1/a_1 = 1$. Let $D_\ell = \sum_{i'=1}^{\ell} a_{i'}$ and suppose that the algorithm has read $a_1, \ldots, a_\ell$ thus far and has retained the running sum D_ℓ and a sample i^* such that $\mathbf{Pr}[i^* = i] = a_i/D_\ell$. Upon reading $a_{\ell+1}$ the algorithm lets $i^* = \ell + 1$ with probability $a_{\ell+1}/D_{\ell+1}$ and retains i^* at its previous value otherwise. At that point, clearly $\mathbf{Pr}[i^* = \ell + 1] = a_{\ell+1}/D_{\ell+1}$; furthermore for $i = 1, \ldots, \ell$, $\mathbf{Pr}[i^* = i] = \frac{a_i}{D_\ell}\left(1 - \frac{a_{\ell+1}}{D_{\ell+1}}\right) = \frac{a_i}{D_{\ell+1}}$. By induction this results holds when $\ell + 1 = n$ and the lemma follows.

◇

Clearly, in a single pass over the data this algorithm can be run in parallel with $O(s)$ total memory units to return s independent samples $i_1^*, \ldots, i_s^*$ such that for each i_t^*, $t = 1, \ldots, s$, we have $\mathbf{Pr}[i_t^* = i] = a_i / \sum_{i'=1}^{n} a_{i'}$.

The next lemma is a modification of the previous lemma to deal with the case where a matrix is read in the sparse-unordered representation and one wants to choose a row label with a certain probability. This can also be implemented in $O(1)$ additional space and time. Note that a trivial modification would permit choosing a column label.

Lemma 2 *Suppose that* $A \in \mathbb{R}^{m \times n}$, *is presented in the sparse-unordered representation and is read in one pass, i.e., one sequential read over the data, by the* SELECT *algorithm. Then the algorithm requires* $O(1)$ *additional storage space and returns* i^*, j^* *such that* $\mathbf{Pr}[i^* = i \wedge j^* = j] = A_{i^*j^*}^2 / \|A\|_F^2$ *and thus* $\mathbf{Pr}[i^* = i] = \left|A_{(i^*)}\right|^2 / \|A\|_F^2$.

Proof: Since $A_{i^*j^*}^2 > 0$ the first claim follows from Lemma 1; the second follows since

$$\mathbf{Pr}[i^* = i] = \sum_{j=1}^{n} \mathbf{Pr}[i^* = i \wedge j^* = j] = \sum_{j=1}^{n} \frac{A_{i^*j^*}^2}{\|A\|_F^2} = \frac{\left|A_{(i^*)}\right|^2}{\|A\|_F^2}.$$

◇

4 The Basic Matrix Multiplication Approximation Algorithm

In this section, which describes the main result of the paper, the BASICMATRIXMULTIPLICATION algorithm to approximate the product of two matrices is presented; it is analyzed in this section and in Appendix A. After describing the algorithm in Section 4.1 we describe its implementation and running time issues in Section 4.2. In Section 4.3 we analyze the algorithm and provide error bounds for arbitrary probability distributions; in Section 4.4 error bounds are derived for probability distributions which are nearly optimal in a well defined sense. We provide further discussion of the algorithm in Section 6 and in Appendix A we provide further analysis of the BASICMATRIXMULTIPLICATION algorithm.

BasicMatrixMultiplication Algorithm

Input: $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$, $c \in \mathbb{Z}^+$ s.t. $1 \le c \le n$, and $\{p_i\}_{i=1}^n$ s.t. $p_i \ge 0$ and $\sum_{i=1}^n p_i = 1$.

Output: $C \in \mathbb{R}^{m \times c}$ and $R \in \mathbb{R}^{c \times p}$.

- For $t = 1$ to c,
  - Pick $i_t \in \{1, \ldots, n\}$ with $\mathbf{Pr}[i_t = k] = p_k$, $k = 1, \ldots, n$, independently and with replacement.
  - Set $C^{(t)} = A^{(i_t)} / \sqrt{c p_{i_t}}$ and $R_{(t)} = B_{(i_t)} / \sqrt{c p_{i_t}}$.
- Return C, R.

Figure 2: The BasicMatrixMultiplication Algorithm

4.1 The Algorithm

Recall that for $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times p}$, the product AB may be written as the sum of n rank one matrices

$$AB = \sum_{t=1}^{n} A^{(t)} B_{(t)}. \tag{3}$$

When matrix multiplication is formulated in this manner, a simple randomized algorithm to approximate the product matrix AB suggests itself: randomly sample with replacement from the terms in the summation c times according to a probability distribution $\{p_i\}_{i=1}^n$, scale each term in an appropriate manner, and output the sum of the scaled terms. If $m = p = 1$ then $A^{(t)}, B_{(t)} \in \mathbb{R}$ and it is straightforward to show that this sampling procedure produces an unbiased estimator for the sum. When the terms in the sum are rank one matrices, as in (3), we show that similar results hold.

Consider the BasicMatrixMultiplication algorithm described in Figure 2. When this algorithm is given as input two matrices A and B, a probability distribution $\{p_i\}_{i=1}^n$, and a number c of column-row pairs to choose, it returns as output matrices C and R such that the product CR is an approximation to AB. Observe that since

$$CR = \sum_{t=1}^{c} C^{(t)} R_{(t)} = \sum_{t=1}^{c} \frac{1}{c p_{i_t}} A^{(i_t)} B_{(i_t)}$$

the procedure for sampling and scaling column and row pairs that is used in the BasicMatrix-Multiplication algorithm corresponds to sampling terms in (3) and rescaling by dividing by cp_t if the t-th term is sampled. Alternatively, one could define the sampling matrix $S \in \mathbb{R}^{n \times c}$ to be the zero-one matrix where $S_{ij} = 1$ if the i-th column of A (and thus also the i-th row of B) is chosen in the j-th independent random trial and $S_{ij} = 0$ otherwise. If the rescaling matrix $D \in \mathbb{R}^{c \times c}$ is the diagonal matrix with $D_{tt} = 1/\sqrt{c p_{i_t}}$ then

$$C = ASD \text{ and } R = (SD)^T B$$

so that $CR = ASD(SD)^T B \approx AB$. Figure 3 presents a diagram illustrating the action of the BasicMatrixMultiplication algorithm. The product AB is shown as B and then A operating between the high-dimensional $\mathbb{R}^p$ and $\mathbb{R}^m$ via the high-dimensional $\mathbb{R}^n$; this is approximated by

$$
\begin{tikzcd}
\mathbb{R}^p \arrow[r, "B"] \arrow[dr, "R"] & \mathbb{R}^n \arrow[r, "A"] & \mathbb{R}^m \\
& \mathbb{R}^c \arrow[u, "S"'] \arrow[ur, "C"'] \arrow[loop below, "D"] &
\end{tikzcd}
$$

Figure 3: Diagram for the BasicMatrixMultiplication Algorithm

CR, which is shown as R and then C operating between $\mathbb{R}^p$ and $\mathbb{R}^m$ via the low-dimensional subspace $\mathbb{R}^c$. Also shown are the sampling matrix S and the diagonal rescaling matrix D.

An important issue is the choice of the probabilities $\{p_i\}_{i=1}^n$ and the scaling. It is easily seen that the scaling of $1/\sqrt{cp_{i_t}}$ used in the BasicMatrixMultiplication algorithm makes CR an unbiased estimator of AB; see Lemma 3. Lemma 3 also computes $\mathbf{Var}\left[(CR)_{ij}\right]$ under general probabilities $\{p_i\}_{i=1}^n$. We then compute $\mathbf{E}\left[\|AB - CR\|_F^2\right]$ and see that probabilities of the form $p_k = \left|A^{(k)}\right| \left|B_{(k)}\right| / N, k = 1, \ldots, n$, where N is a normalization, are optimal in that they minimize this quantity; see Lemma 4.

This approach for approximating matrix multiplication has several advantages. First, it is conceptually simple and can be generalized to approximate the product of more than two matrices; see Section A.1 for more on the latter point. Second, since the heart of the algorithm involves matrix multiplication of smaller matrices, it can use any algorithms that exist in the literature for performing the desired matrix multiplication [18, 26, 8]. Third, this approach does not tamper with the sparsity of the matrices, unlike an algorithm that would project both A and B to the same random c dimensional subspace and take the product of the projections. Finally, the algorithm can be easily implemented; see Sections 4.2 and 6 for more discussion. This algorithm is of independent interest and has applications beyond those of the present paper; see [10] for its analysis and [12, 13, 2, 1, 9] for examples of application areas.

4.2 Implementation of the Sampling and Running Time

To implement the BasicMatrixMultiplication algorithm, it must be decided which elements of the input to sample and those elements must then be sampled. In the case of uniform sampling it may be decided before the input is seen which column-row pairs to sample. In this case, a single pass over the matrices is sufficient to sample the columns and rows of interest and to construct C and R; this requires $O(m+p)$ additional time and space. We will see below that it is useful to sample according to a nonuniform probability distribution that depends on column and row lengths, e.g., see (5) and (7). In order to decide which column-row pairs to sample in such a case, one pass through the matrices and $O(n)$ additional time and space is sufficient; in the additional space running totals of $\left|A^{(k)}\right|^2$ and $\left|B_{(k)}\right|^2$ are kept, so that after the first pass $\left|A^{(k)}\right|$, $\left|B_{(k)}\right|$, $k = 1, \ldots, n$, and thus the probabilities, can be calculated in $O(n)$ additional time. Then in a second pass the columns and rows of interest can be sampled and C and R can be constructed and stored; this requires $O(m+p)$ additional space and time. Thus, in addition to either one or two passes over the data, for both uniform and nonuniform sampling $O(m+n+p)$ additional space and time is sufficient to sample from the matrices A and B of the input and to construct the matrices C and R.

If $B = A^T$, then the additional space and time requirements for uniform sampling are similar to the general case; for nonuniform sampling, however, the situation is different (assuming probabilities of the form (5) or (7)) since the sampling probabilities are then of the form $p_k = \left|A^{(k)}\right|^2 / \|A\|_F^2$. Due to Lemma 2 we can select which columns of A to choose using constant additional space and time during the first pass. Then, during the second pass, these columns may be extracted and the matrices C and $R = C^T$ may be constructed using $O(m+p)$ additional space and time; this will be used in the LINEARTIMESVD algorithm of [12]. Note that if only a constant-sized part of the columns of C are needed, as for example in the CONSTANTTIMESVD algorithm of [12], then extracting and storing this constant sized subset of the samples desired may be performed using $O(1)$ additional space and time.

4.3 Analysis of the Algorithm for Arbitrary Probabilities

In this section we prove upper bounds for $\|AB - CR\|_F^2$, where C and R are returned from the BASICMATRIXMULTIPLICATION algorithm. Recall that by Jensen's inequality bounding $\|AB - CR\|_F^2$ (in expectation) implies a bound for $\|AB - CR\|_F$. Recall also that a bound on $\|AB - CR\|_F$ immediately provides a bound on $\|AB - CR\|_2$ since $\|AB - CR\|_2 \leq \|AB - CR\|_F$.

Our first lemma proves that the expectation of the (i,j)-th element of the approximation is equal to the (i,j)-th element of the exact product; it also describes the variance of the approximation of the (i,j)-th element.

Lemma 3 *Suppose $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$, $c \in \mathbb{Z}^+$ such that $1 \leq c \leq n$, and $\{p_i\}_{i=1}^n$ are such that $p_i \geq 0$ and $\sum_{i=1}^n p_i = 1$. Construct C and R with the* BASICMATRIXMULTIPLICATION *algorithm, and let CR be an approximation to AB. Then,*

$$\mathbf{E}\left[(CR)_{ij}\right] = (AB)_{ij}$$

and

$$\mathbf{Var}\left[(CR)_{ij}\right] = \frac{1}{c}\sum_{k=1}^n \frac{A_{ik}^2 B_{kj}^2}{p_k} - \frac{1}{c}(AB)_{ij}^2.$$

Proof: Fix i,j. For $t = 1, \ldots, c$, define $X_t = \left(\frac{A^{(i_t)} B_{(i_t)}}{c p_{i_t}}\right)_{ij} = \frac{A_{i i_t} B_{i_t j}}{c p_{i_t}}$. Thus,

$$\mathbf{E}\left[X_t\right] = \sum_{k=1}^n p_k \frac{A_{ik} B_{kj}}{c p_k} = \frac{1}{c}(AB)_{ij} \quad \text{and} \quad \mathbf{E}\left[X_t^2\right] = \sum_{k=1}^n \frac{A_{ik}^2 B_{kj}^2}{c^2 p_k}.$$

Since by construction $(CR)_{ij} = \sum_{t=1}^c X_t$, we have $\mathbf{E}\left[(CR)_{ij}\right] = \sum_{t=1}^c \mathbf{E}\left[X_t\right] = (AB)_{ij}$. Since $(CR)_{ij}$ is the sum of c independent random variables, $\mathbf{Var}\left[(CR)_{ij}\right] = \sum_{t=1}^c \mathbf{Var}\left[X_t\right]$. Since $\mathbf{Var}\left[X_t\right] = \mathbf{E}\left[X_t^2\right] - \mathbf{E}\left[X_t\right]^2$ we see that

$$\mathbf{Var}\left[X_t\right] = \sum_{k=1}^n \frac{A_{ik}^2 B_{kj}^2}{c^2 p_k} - \frac{1}{c^2}(AB)_{ij}^2$$

and the lemma follows.

$\diamond$

Using the previous lemma, we bound in the next lemma $\mathbf{E}\left[\|AB - CR\|_F^2\right]$ and note how this measure of the error depends on the p_i's.

Lemma 4 *Suppose $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$, $c \in \mathbb{Z}^+$ such that $1 \leq c \leq n$, and $\{p_i\}_{i=1}^n$ are such that $p_i \geq 0$ and $\sum_{i=1}^n p_i = 1$. Construct C and R with the* BasicMatrixMultiplication *algorithm, and let CR be an approximation to AB. Then,*

$$\mathbf{E}\left[\|AB - CR\|_F^2 \right] = \sum_{k=1}^n \frac{\left|A^{(k)}\right|^2 \left|B_{(k)}\right|^2}{cp_k} - \frac{1}{c} \|AB\|_F^2 . \tag{4}$$

Furthermore, if

$$p_k = \frac{\left|A^{(k)}\right| \left|B_{(k)}\right|}{\sum_{k'=1}^n \left|A^{(k')}\right| \left|B_{(k')}\right|}, \tag{5}$$

then

$$\mathbf{E}\left[\|AB - CR\|_F^2 \right] = \frac{1}{c} \left(\sum_{k=1}^n \left|A^{(k)}\right| \left|B_{(k)}\right| \right)^2 - \frac{1}{c} \|AB\|_F^2 . \tag{6}$$

This choice for p_k minimizes $\mathbf{E}\left[\|AB - CR\|_F^2 \right]$.

Proof: First, note that

$$\mathbf{E}\left[\|AB - CR\|_F^2 \right] = \sum_{i=1}^m \sum_{j=1}^p \mathbf{E}\left[(AB - CR)_{ij}^2 \right] = \sum_{i=1}^m \sum_{j=1}^p \mathbf{Var}\left[(CR)_{ij} \right].$$

Thus, from Lemma 3 it follows that

$$\begin{aligned} \mathbf{E}\left[\|AB - CR\|_F^2 \right] &= \frac{1}{c} \sum_{k=1}^n \frac{1}{p_k} \left(\sum_i A_{ik}^2 \right) \left(\sum_j B_{kj}^2 \right) - \frac{1}{c} \|AB\|_F^2 \\ &= \frac{1}{c} \sum_{k=1}^n \frac{1}{p_k} \left|A^{(k)}\right|^2 \left|B_{(k)}\right|^2 - \frac{1}{c} \|AB\|_F^2 . \end{aligned}$$

If the value $p_k = \frac{\left|A^{(k)}\right| \left|B_{(k)}\right|}{\sum_{k'=1}^n \left|A^{(k')}\right| \left|B_{(k')}\right|}$ is used in this expression, then

$$\mathbf{E}\left[\|AB - CR\|_F^2 \right] = \frac{1}{c} \left(\sum_{k=1}^n \left|A^{(k)}\right| \left|B_{(k)}\right| \right)^2 - \frac{1}{c} \|AB\|_F^2 .$$

Finally, to prove that this choice for the p_k's minimizes $\mathbf{E}\left[\|AB - CR\|_F^2 \right]$ define the function

$$f(p_1, \ldots p_n) = \sum_{k=1}^n \frac{1}{p_k} \left|A^{(k)}\right|^2 \left|B_{(k)}\right|^2,$$

which characterizes the dependence of $\mathbf{E}\left[\|AB - CR\|_F^2 \right]$ on the p_k's. To minimize f subject to $\sum_{k=1}^n p_k = 1$, introduce the Lagrange multiplier λ and define the function

$$g(p_1, \ldots p_n) = f(p_1, \ldots p_n) + \lambda \left(\sum_{k=1}^n p_k - 1 \right).$$

We then have at the minimum that

$$0 = \frac{\partial g}{\partial p_i} = \frac{-1}{p_i^2} \left|A^{(i)}\right|^2 \left|B_{(i)}\right|^2 + \lambda.$$

Thus,

$$p_i = \frac{\left|A^{(i)}\right| \left|B_{(i)}\right|}{\sqrt{\lambda}} = \frac{\left|A^{(i)}\right| \left|B_{(i)}\right|}{\sum_{i'=1}^{n} \left|A^{(i')}\right| \left|B_{(i')}\right|},$$

where the second equality comes from solving for $\sqrt{\lambda}$ in $\sum_{k=1}^{n-1} p_k = 1$. That these probabilities are a minimum follows since $\frac{\partial^2 g}{\partial {p_i}^2} > 0 \; \forall i$ s.t. $\left|A^{(i)}\right|^2 \left|B_{(i)}\right|^2 > 0$.

◇

4.4 Analysis of the Algorithm for Nearly Optimal Probabilities

With Lemma 4 and using Jensen's inequality upper bounds on quantities such as $\mathbf{E}\left[\, \|AB - CR\|_F^2 \right]$ and $\mathbf{E}\left[\, \|AB - CR\|_F \right]$ may be obtained for various sampling probabilities $\{p_i\}_{i=1}^n$. In many cases, by using a Martingale argument to show that the error is tightly concentrated around its mean, the expectations in these bounds may be removed and the corresponding results can be shown to hold with high probability.

Rather than presenting these results in their full generality, we restrict attention to two particular sets of probabilities. We will say that the sampling probabilities $p_k = \frac{\left|A^{(k)}\right|\left|B_{(k)}\right|}{\sum_{k'=1}^{n}\left|A^{(k')}\right|\left|B_{(k')}\right|}$ are the *optimal probabilities* since they minimize $\mathbf{E}\left[\, \|AB - CR\|_F^2 \right]$, which as Lemma 4 shows is one natural measure of the error. We will say that a set of sampling probabilities $\{p_i\}_{i=1}^n$ are *nearly optimal probabilities* if $p_k \geq \frac{\beta\left|A^{(k)}\right|\left|B_{(k)}\right|}{\sum_{k'=1}^{n}\left|A^{(k')}\right|\left|B_{(k')}\right|}$ for some positive constant $\beta \leq 1$.

We now prove, for nearly optimal sampling probabilities, results analogous to those of Lemma 4, and also that the corresponding results with the expectations removed hold with high probability. Notice that if $\beta \neq 1$ then we suffer a small β-dependent loss in accuracy.

Theorem 1 *Suppose $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$, $c \in \mathbb{Z}^+$ such that $1 \leq c \leq n$, and $\{p_i\}_{i=1}^n$ are such that $\sum_{i=1}^n p_i = 1$ and such that for some positive constant $\beta \leq 1$*

$$p_k \geq \frac{\beta\left|A^{(k)}\right| \left|B_{(k)}\right|}{\sum_{k'=1}^{n} \left|A^{(k')}\right| \left|B_{(k')}\right|}. \tag{7}$$

Construct C and R with the BasicMatrixMultiplication *algorithm, and let CR be an approximation to AB. Then,*

$$\mathbf{E}\left[\, \|AB - CR\|_F^2 \right] \leq \frac{1}{\beta c} \|A\|_F^2 \, \|B\|_F^2 \,. \tag{8}$$

Furthermore, let $\delta \in (0,1)$ and $\eta = 1 + \sqrt{(8/\beta)\log(1/\delta)}$. Then, with probability at least $1-\delta$,

$$\|AB - CR\|_F^2 \leq \frac{\eta^2}{\beta c} \|A\|_F^2 \, \|B\|_F^2 \,. \tag{9}$$

Proof: Following reasoning similar to that of lemma 4 and using the probabilities of (7), we see that

$$\begin{aligned}
\mathbf{E}\left[\, \|AB - CR\|_F^2 \right] &\leq \frac{1}{c}\sum_{k=1}^{n} \frac{1}{p_k} \left|A^{(k)}\right|^2 \left|B_{(k)}\right|^2 \\
&\leq \frac{1}{\beta c}\left(\sum_{k=1}^{n} \left|A^{(k)}\right| \left|B_{(k)}\right|\right)^2 \\
&\leq \frac{1}{\beta c} \|A\|_F^2 \, \|B\|_F^2 \,,
\end{aligned}$$

where the last inequality follows due to the Cauchy-Schwartz inequality. Next, define the event $\mathcal{E}_2$ to be

$$\|AB - CR\|_F \leq \frac{\eta}{\sqrt{\beta c}} \|A\|_F \|B\|_F \tag{10}$$

and note that to prove the remainder of the theorem it suffices to prove that $\mathbf{Pr}[\mathcal{E}_2] \geq 1 - \delta$. To that end, note that C and R and thus $CR = \sum_{t=1}^{c} \frac{1}{cp_{i_t}} A^{i_t} B_{i_t}$ are formed by randomly selecting c elements from $\{1, \ldots, n\}$, independently and with replacement. Let the sequence of elements chosen be $\{i_t\}_{t=1}^{c}$. Consider the function

$$F(i_1, \ldots, i_c) = \|AB - CR\|_F. \tag{11}$$

We will show that changing one i_t at a time does not change F too much; this will enable us to apply a martingale inequality. To this end, consider changing one of the i_t to i'_t while keeping the other i_t's the same. Then, construct the corresponding C' and R'. Note that C' differs from C in only a single column and that R' differs from R in only a single row. Thus,

$$\begin{aligned}
\left\|CR - C'R'\right\|_F &= \left\| \frac{A^{(i_t)} B_{(i_t)}}{cp_{i_t}} - \frac{A^{(i'_t)} B_{(i'_t)}}{cp_{i'_t}} \right\|_F & (12)\\
&\leq \frac{1}{cp_{i_t}} \left\| A^{(i_t)} B_{(i_t)} \right\|_F + \frac{1}{cp_{i'_t}} \left\| A^{(i'_t)} B_{(i'_t)} \right\|_F & (13)\\
&= \frac{1}{cp_{i_t}} \left| A^{(i_t)} \right| \left| B_{(i_t)} \right| + \frac{1}{cp_{i'_t}} \left| A^{(i'_t)} \right| \left| B_{(i'_t)} \right| & (14)\\
&\leq \frac{2}{c} \max_{\alpha} \frac{\left| A^{(\alpha)} \right| \left| B_{(\alpha)} \right|}{p_\alpha}. & (15)
\end{aligned}$$

(12) follows by construction and (14) follows since $\left\|xy^T\right\|_F = |x|\,|y|$ for $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$. Thus, using the probabilities (7) and employing the Cauchy-Schwartz inequality we see that

$$\begin{aligned}
\left\|CR - C'R'\right\|_F &\leq \frac{2}{\beta c} \sum_{k=1}^{n} \left| A^{(k)} \right| \left| B_{(k)} \right| & (16)\\
&\leq \frac{2}{\beta c} \|A\|_F \|B\|_F. & (17)
\end{aligned}$$

Therefore, using the triangle inequality we see that

$$\begin{aligned}
\|AB - CR\|_F &\leq \left\|AB - C'R'\right\|_F + \left\|C'R' - CR\right\|_F \\
&\leq \left\|AB - C'R'\right\|_F + \frac{2}{\beta c} \|A\|_F \|B\|_F. & (18)
\end{aligned}$$

By similar reasoning, we can derive

$$\left\|AB - C'R'\right\|_F \leq \|AB - CR\|_F + \frac{2}{\beta c} \|A\|_F \|B\|_F. \tag{19}$$

Define $\Delta = \frac{2}{\beta c} \|A\|_F \|B\|_F$; thus,

$$\left| F(i_1, \ldots, i_k, \ldots, i_c) - F\left(i_1, \ldots, i'_k, \ldots, i_c\right) \right| \leq \Delta. \tag{20}$$

Let $\gamma = \sqrt{2c \log(1/\delta)}\,\Delta$ and consider the associated Doob martingale. By the Hoeffding-Azuma inequality [22],

$$\mathbf{Pr}\left[\|AB - CR\|_F \geq \frac{1}{\sqrt{\beta c}} \|A\|_F \|B\|_F + \gamma \right] \leq \exp\left(-\gamma^2 / 2c\Delta^2\right) = \delta \tag{21}$$

and theorem follows.

$\diamond$

An immediate consequence of Theorem 1 is that by choosing enough column-row pairs, the error in the approximation of the matrix product can be made arbitrarily small. In particular, if $c \geq 1/\beta\epsilon^2$ then by using Jensen's inequality it follows that

$$\mathbf{E}\left[\, \|AB - CR\|_F \right] \leq \epsilon \, \|A\|_F \, \|B\|_F \tag{22}$$

and if, in addition, $c \geq \eta^2/\beta\epsilon^2$ then with probability at least $1-\delta$

$$\|AB - CR\|_F \leq \epsilon \, \|A\|_F \, \|B\|_F \, . \tag{23}$$

In certain applications, e.g. [12, 13], one is interested in an application of Theorem 1 to the case that $B = A^T$, i.e., one is interested in approximating $\left\|AA^T - CC^T\right\|_F^2$. In this case, sampling column-row pairs corresponds to sampling columns of A, and nearly optimal probabilities will be those such that $p_k \geq \frac{\beta\left|A^{(k)}\right|}{\|A\|_F}$ for some positive $\beta \leq 1$. By taking $B = A^T$ and applying Jensen's inequality, we have the following theorem as a corollary of Theorem 1.

Theorem 2 *Suppose* $A \in \mathbb{R}^{m \times n}$, $c \in \mathbb{Z}^+$, $1 \leq c \leq n$, *and* $\{p_i\}_{i=1}^n$ *are such that* $\sum_{i=1}^n p_i = 1$ *and such that* $p_k \geq \frac{\beta\left|A^{(k)}\right|^2}{\|A\|_F^2}$ *for some positive constant* $\beta \leq 1$. *Furthermore, let* $\delta \in (0,1)$ *and* $\eta = 1 + \sqrt{(8/\beta)\log(1/\delta)}$. *Construct* C *(and* $R = C^T$*) with the* BasicMatrixMultiplication *algorithm, and let* CC^T *be an approximation to* AA^T. *Then,*

$$\mathbf{E}\left[\, \left\|AA^T - CC^T\right\|_F \right] \leq \frac{1}{\sqrt{\beta c}} \, \|A\|_F^2 \tag{24}$$

and with probability at least $1-\delta$,

$$\left\|AA^T - CC^T\right\|_F \leq \frac{\eta}{\sqrt{\beta c}} \, \|A\|_F^2 \, . \tag{25}$$

5 A Second Matrix Multiplication Algorithm

In this section we describe the ElementwiseMatrixMultiplication algorithm to approximate the product of two matrices. First, in Section 5.1, we describe the algorithm, its implementation, and running time issues; then in Section 5.2 we analyze the algorithm and bound its error with respect to both the Frobenius and spectral norms. We will see that the algorithm returns good approximations with respect to the spectral norm but not with respect to the Frobenius norm.

5.1 The Algorithm and Its Implementation

The method to approximate the product of two matrices that is presented in this section differs from the previous algorithm and is inspired by [2] and [1]. In [2] the singular value decomposition of a matrix is approximated using element-wise uniform sampling; in [1] this approach is extended to include nonuniform sampling probabilities of a certain natural form. Since neither of these papers apply these methods to approximate matrix multiplication, we do so here for comparison with the BasicMatrixMultiplication algorithm.

Consider the ElementwiseMatrixMultiplication algorithm which is presented in Figure 4. When this algorithm is given as input two matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times p}$ it creates two matrices $S \in \mathbb{R}^{m \times n}$ and $R \in \mathbb{R}^{n \times p}$ by keeping a few elements of A and a few elements of B, ElementwiseMatrixMultiplication Algorithm

Input: $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$, $\{p_{ij}\}_{i,j=1}^{m,n}$ such that $0 \leq p_{ij} \leq 1$, and $\{q_{ij}\}_{i,j=1}^{n,p}$ such that $0 \leq q_{ij} \leq 1$.

Output: $S \in \mathbb{R}^{m \times n}$ and $R \in \mathbb{R}^{n \times p}$.

Algorithm:

- For $i = 1$ to m and $j = 1$ to n,
  - Set
  $$S_{ij} = \begin{cases} A_{ij}/p_{ij} & \text{with probability } p_{ij} \\ 0 & \text{otherwise.} \end{cases}$$
- For $i = 1$ to n and $j = 1$ to p,
  - Set
  $$R_{ij} = \begin{cases} B_{ij}/q_{ij} & \text{with probability } q_{ij} \\ 0 & \text{otherwise.} \end{cases}$$
- Return S, R.

Figure 4: The ElementwiseMatrixMultiplication Algorithm

respectively, scaling in an appropriate manner those elements that are kept, and zeroing out the remaining elements. The algorithm then returns matrices S and R such that the product SR is an approximation to AB. Note that since S and R are formed independently of each other the algorithm does not keep "corresponding" elements; doing so would introduce dependence that would complicate the analysis.

The ElementwiseMatrixMultiplication algorithm can be implemented with the nonuniform probabilities used in this section with two passes over the data; we leave it as an open problem whether a single pass suffices when working within the pass efficient framework. This algorithm differs from the BasicMatrixMultiplication algorithm in that we get an expected number of elements so we have an expected additional space required for storage and an expected additional time required for the associated sparse matrix multiplication. We do not provide a detailed analysis of these random variables.

5.2 Analysis of the Algorithm

In this section we present error bounds for both $\|AB - SR\|_F$ and $\|AB - SR\|_2$. While the Frobenius norm error bound for this algorithm is rather easy to derive using very intuitive probability distributions, the spectral norm bound is more complicated and requires some additional technicalities.

Since whether or not (for a given i,j) $S_{ij} = 0$ or $S_{ij} = A_{ij}/p_{ij}$ we have that $A_{ij} - S_{ij}$ is large (and similarly for the matrix R and thus the matrix SR) it is plausible that the ElementwiseMatrixMultiplication algorithm does not have a good bound for $\mathbf{E}\left[\|AB - SR\|_F^2\right]$. This intuition is formalized in the following lemma. Note that ℓ and ℓ' are chosen such that not more than ℓ and ℓ' of the elements of the matrices A and B are retained in expectation, respectively.

Lemma 5 *Suppose $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times p}$, let $\ell, \ell' \in \mathbb{Z}^+$, and let $p_{ij} = \min\{1, \ell A_{ij}^2 / \|A\|_F^2\}$ and $q_{ij} = \min\{1, \ell' B_{ij}^2 / \|B\|_F^2\}$. Construct S and R with the* ElementwiseMatrixMultipli-

CATION *algorithm, and let SR be an approximation to AB. Then, for all i, j,*

$$\mathbf{E}\left[(SR)_{ij}\right] = (AB)_{ij},$$

$$\mathbf{Var}\left[(SR)_{ij}\right] = \sum_{k=1}^{n} \frac{A_{ik}^2}{p_{ik}} \frac{B_{kj}^2}{q_{kj}} - \sum_{k=1}^{n} A_{ik}^2 B_{kj}^2$$

$$\mathbf{E}\left[\left\|AB - SR\right\|_F^2\right] \geq \frac{mpn}{\ell\ell'} \left\|A\right\|_F^2 \left\|B\right\|_F^2 - \sum_{k=1}^{n} \left|A^{(k)}\right|^2 \left|B_{(k)}\right|^2. \tag{26}$$

Proof: Let us first fix i, j. Then, since for every k we have that $S_{ik} = A_{ik}/p_{ik}$ with probability p_{ik} and $S_{ik} = 0$ with probability $1 - p_{ik}$ we have that $\mathbf{E}\left[S_{ik}\right] = A_{ik}$; similarly for R_{kj} we have that $\mathbf{E}\left[R_{kj}\right] = B_{kj}$. Thus, since S and R have been constructed independently, we have that

$$\mathbf{E}\left[(SR)_{ij}\right] = \mathbf{E}\left[\sum_{k=1}^{n} S_{ik} R_{kj}\right] = \sum_{k=1}^{n} \mathbf{E}\left[S_{ik}\right] \mathbf{E}\left[R_{kj}\right] = (AB)_{ij}.$$

Since $\mathbf{Var}\left[(SR)_{ij}\right] = \mathbf{E}\left[(SR)_{ij}^2\right] - \mathbf{E}\left[(SR)_{ij}\right]^2$ and since $(SR)_{ij} = \sum_{k=1}^{n} S_{ik} R_{kj}$ we get that

$$\begin{aligned}
\mathbf{Var}\left[(SR)_{ij}\right] &= \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} \mathbf{E}\left[S_{ik_1} R_{k_1 j} S_{ik_2} R_{k_2 j}\right] - \mathbf{E}\left[(SR)_{ij}\right]^2 \\
&= \sum_{k=1}^{n} \mathbf{E}\left[S_{ik}^2\right] \mathbf{E}\left[R_{kj}^2\right] + \sum_{k_1=1}^{n} \sum_{k_2 \neq k_1} \mathbf{E}\left[S_{ik_1}\right] \mathbf{E}\left[R_{k_1 j}\right] \mathbf{E}\left[S_{ik_2}\right] \mathbf{E}\left[R_{k_2 j}\right] - (AB)_{ij}^2 \\
&= \sum_{k=1}^{n} \frac{A_{ik}^2}{p_{ik}} \frac{B_{kj}^2}{q_{kj}} + \sum_{k_1=1}^{n} \sum_{k_2 \neq k_1} A_{ik_1} B_{k_1 j} A_{ik_2} B_{k_2 j} - (AB)_{ij}^2 \\
&= \sum_{k=1}^{n} \frac{A_{ik}^2}{p_{ik}} \frac{B_{kj}^2}{q_{kj}} - \sum_{k=1}^{n} A_{ik}^2 B_{kj}^2,
\end{aligned}$$

where the last line follows by adding and subtracting $\sum_{k_1=1}^{n} \sum_{k_2=k_1} A_{ik_1} B_{k_1 j} A_{ik_2} B_{k_2 j}$ from the second to last line.

Thus, since $\mathbf{E}\left[\left\|AB - SR\right\|_F^2\right] = \sum_{i=1}^{m} \sum_{j=1}^{p} \mathbf{Var}\left[(SR)_{ij}\right]$ and since the probabilities p_{ij} and q_{ij} are such that $1/p_{ik} \geq \left\|A\right\|_F^2 / \ell A_{ik}^2$ and $1/q_{kj} \geq \left\|B\right\|_F^2 / \ell' B_{kj}^2$ we get that

$$\begin{aligned}
\mathbf{E}\left[\left\|AB - SR\right\|_F^2\right] &= \sum_{i=1}^{m} \sum_{j=1}^{p} \sum_{k=1}^{n} \frac{A_{ik}^2}{p_{ik}} \frac{B_{kj}^2}{q_{kj}} - \sum_{i=1}^{m} \sum_{j=1}^{p} \sum_{k=1}^{n} A_{ik}^2 B_{kj}^2 \\
&\geq \sum_{i,j=1}^{m,p} \sum_{k=1}^{n} \frac{\left\|A\right\|_F^2 \left\|B\right\|_F^2}{\ell\ell'} - \sum_{k=1}^{n} \left|A^{(k)}\right|^2 \left|B_{(k)}\right|^2.
\end{aligned}$$

The lemma then follows.

◇

Next we show that although the ElementwiseMatrixMultiplication algorithm does not yield a nice error bound for the Frobenius norm, it does for the spectral norm. In order to prove Theorem 4, which provides our bound on $\left\|AB - SR\right\|_2$, we will use the following theorem which follows immediately from a result that was proved in [1] and which shows that with high probability the spectrum of a random matrix is close to its expectation. The theorem is proved by using a generalization of a result of Füredi and Komlós [16], combined with a more recent concentration result of Alon, Krivelevich, and Vu based on Talagrand's inequality [21].

Theorem 3 *Given an $n \times n$ matrix A, let $\hat{A}$ be any random matrix whose entries are independent random variables such that for all i, j:* $\mathbf{E}\left[\hat{A}_{ij}\right] = A_{ij}$, $\mathbf{Var}\left[\hat{A}_{ij}\right] \leq \sigma^2$, *and*

$$\left|\hat{A}_{ij} - A_{ij}\right| \leq \frac{\sigma\sqrt{2n}}{\log^3(2n)}. \tag{27}$$

For any $n \geq 10$, with probability at least $1 - 1/(2n)$,

$$\left\|A - \hat{A}\right\|_2 < 7\sigma\sqrt{2n}. \tag{28}$$

Prior to stating the main result of this section, we must address a technical issue that arises in our effort to apply the above theorem in order to bound $\|AB - SR\|_2$. Note that the construction of the matrices S and R by the ElementwiseMatrixMultiplication algorithm may be viewed as adding carefully constructed random matrices E and D such that $S = A + E$ and $R = B + D$; see [2] and [1] for a discussion. As we will see below, if we can bound $\|E\|_2$ and $\|D\|_2$, then a bound for $\|AB - SR\|_2$ follows easily. Since we will apply Theorem 3 in order to obtain such bounds, we need to satisfy the range constraint (27). Sampling with respect to the nonuniform probability distribution of Lemma 5 might violate this constraint since, in the unlikely event that a small element is kept, the resulting entry $S_{ij} = A_{ij}/p_{ij}$ will be very large (and similarly for R). Thus, following [1], we modify our sampling probabilities so that small elements are kept with a slightly larger probability which is proportional to $|A_{ij}|$ instead of A_{ij}^2:

$$p_{ij} = \begin{cases} \min\{1, \ell A_{ij}^2 / \|A\|_F^2\} & \text{if } |A_{ij}| > \frac{\|A\|_F \log^3(2n)}{\sqrt{2n\ell}} \\ \min\{1, \frac{\sqrt{\ell}|A_{ij}|\log^3(2n)}{\sqrt{2n}\|A\|_F}\} & \text{otherwise} \end{cases} \tag{29}$$

$$q_{ij} = \begin{cases} \min\{1, \ell' B_{ij}^2 / \|B\|_F^2\} & \text{if } |B_{ij}| > \frac{\|B\|_F \log^3(2n)}{\sqrt{2n\ell'}} \\ \min\{1, \frac{\sqrt{\ell'}|B_{ij}|\log^3(2n)}{\sqrt{2n}\|B\|_F}\} & \text{otherwise.} \end{cases} \tag{30}$$

We now state and prove our main theorem of this section. In the interests of clarity we make several simplifying assumptions in the statement of the theorem.

Theorem 4 *Suppose $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$, and let p_{ij} and q_{ij} be as specified in (29) and (30) with $\ell = \ell' \geq 1$. Assume that $\ell \leq \|A\|_F^2 / \max_{i,j} A_{ij}^2$ and that $\ell \leq \|B\|_F^2 / \max_{i,j} B_{ij}^2$; assume also that $m = n = p$ and that n is large enough so that $2n \geq \log^6(2n)$. Construct S and R with the* ElementwiseMatrixMultiplication *algorithm, and let SR be an approximation to AB. Then, with probability at least $1 - 1/n$,*

$$\|AB - SR\|_2 \leq \left(20\sqrt{\frac{n}{\ell}} + \frac{100n}{\ell}\right) \|A\|_F \|B\|_F. \tag{31}$$

Proof: By the assumptions on n and ℓ, neither p_{ij} nor q_{ij} exceed 1 for any i, j. Letting $E = S - A$ and $D = R - B$, we have

$$SR = (A + E)(B + D) = AB + AD + EB + ED. \tag{32}$$

Thus, by the triangle inequality and submultiplicitivity, we have that

$$\|AB - SR\|_2 \leq \|A\|_2 \|D\|_2 + \|E\|_2 \|B\|_2 + \|E\|_2 \|D\|_2. \tag{33}$$

In order to apply Theorem 3 to $\|E\|_2$ and $\|D\|_2$ we first verify that the assumptions of the theorem are satisfied. From the proof of Lemma 5, we have that $\mathbf{E}[S_{ij}] = A_{ij}$. In addition,

$$\mathbf{Var}[S_{ij}] \le \mathbf{E}\left[S_{ij}^2\right] = \frac{A_{ij}^2}{p_{ij}} \le \frac{\|A\|_F^2}{\ell}$$

holds regardless of whether $|A_{ij}|$ is larger or smaller than the threshold. Similarly, we get that $\mathbf{E}[R_{ij}] = B_{ij}$ and that $\mathbf{Var}[D_{ij}] \le \frac{\|B\|_F^2}{\ell}$. It is straightforward to show that regardless of whether or not $|A_{ij}|$ is above or below the threshold and regardless of whether or not $S_{ij} = 0$ or $S_{ij} = A_{ij}/p_{ij}$ we have that

$$|A_{ij} - S_{ij}| \le \frac{\|A\|_F \sqrt{2n}}{\sqrt{\ell}\log^3(2n)}. \tag{34}$$

Similarly, one can show that

$$|B_{ij} - R_{ij}| \le \frac{\|B\|_F \sqrt{2n}}{\sqrt{\ell}\log^3(2n)}. \tag{35}$$

Thus, the conditions of Theorem 3 are satisfied and with probability at least $1 - 1/2n$ each of the following holds:

$$\|E\|_2 \le 7\|A\|_F \sqrt{2n}/\sqrt{\ell} \tag{36}$$

$$\|D\|_2 \le 7\|B\|_F \sqrt{2n}/\sqrt{\ell}. \tag{37}$$

Thus, with probability at least $1 - 1/n$ both of these inequalities hold. Combining the bounds (36) and (37) with (33), and since $\|\cdot\|_2 \le \|\cdot\|_F$, we have

$$\begin{aligned}
\|AB - SR\|_2 &\le \|A\|_2 \|D\|_2 + \|E\|_2 \|B\|_2 + \|E\|_2 \|D\|_2 \\
&\le \frac{7\sqrt{2n}\,\|A\|_F \|B\|_F}{\sqrt{\ell}} + \frac{7\sqrt{2n}\,\|A\|_F \|B\|_F}{\sqrt{\ell}} + \frac{98n\,\|A\|_F \|B\|_F}{\ell} \\
&\le \left(20\sqrt{n/\ell} + 100n/\ell\right)\|A\|_F \|B\|_F .
\end{aligned}$$

$\diamond$

Notice that if we let $\ell = cn$ in Theorem 4 then the error bound (31) becomes

$$\|AB - SR\|_2 \le \left(\frac{20}{\sqrt{c}} + \frac{100}{c}\right)\|A\|_F \|B\|_F = O(1/\sqrt{c})\,\|A\|_F \|B\|_F .$$

Comparison with (9) of Theorem 1 reveals that (since $\|\cdot\|_2 \le \|\cdot\|_F$) both of our matrix multiplication algorithms have, asymptotically, a similar bound with respect to the spectral norm.

6 Discussion and Conclusion

To the best of our knowledge, the only previous randomized algorithm that approximates the product of two matrices is that of Cohen and Lewis [7]. This algorithm is based on random walks in a graph representation of the input matrices and attempts to identify all high-valued entries in nonnegative matrix products in order to improve estimates (relative to exact sparse multiplication) by spending less time on small valued entries. Their algorithm is more complicated than ours, it requires different graph representations of the input matrices if the matrices are allowed to contain negative entries, it needs to store the complete input matrices, and it is especially useful when the matrices are not sparse.

It is worth emphasizing how the BasicMatrixMultiplication algorithm behaves when A and B are well approximated by low-rank matrices. Since a low-rank matrix or a matrix that is well approximated by a low-rank matrix is a matrix whose rows and columns contain much redundant information in terms of the subspaces they span, it is plausible that if the range of B overlaps appropriately with the domain of A, then we can get a good approximation to AB by carefully sampling a small number c of appropriately rescaled rank one approximations to AB. Theorem 1 shows that if the $\{p_i\}_{i=1}^n$ are chosen judiciously then this is the case and Figure 3 illustrates this.

We emphasize that in the case of sampling with nonuniform probabilities our sampling can be viewed as a two-pass algorithm; in the first pass the algorithm reads the matrix, it then decides which columns and rows to keep, and then in the second pass it extracts these columns and rows. In certain applications, two passes through the matrix are not possible and only one pass is allowed [14]. In these cases, we can still perform uniform sampling; in this case, if column-row pairs are all approximately the same size, i.e., $\left|A^{(k)}\right| \left|B_{(k)}\right|$ is close to its mean value (more precisely, if there exists some positive constant $\beta \leq 1$ such that $\forall k \ \left|A^{(k)}\right| \left|B_{(k)}\right| \leq \frac{1}{\beta n} \sum_{k'=1}^{n} \left|A^{(k')}\right| \left|B_{(k')}\right|$) then the uniform probabilities are nearly optimal and we can sample uniformly with a small β-dependent loss in accuracy.

Given the information about the elements of A and B that can be obtained, e.g., after one pass through the data, we have shown that certain nonuniform sampling probabilities (in which the probability that larger column-row pairs are drawn is higher – see (7)) are better in a well defined sense. Note that although larger columns and rows get picked more often, the scaling is such that their weight is deemphasized in the estimator sum. One could imagine a situation when detailed information about the elements of, e.g., A may be obtained after a single pass but no information or no information except general bounds on the size of the elements may be possible for B. In this case, a set of sampling probabilities other than those discussed in Section 4 may be appropriate. See Figure 5 for a summary of the results for different probability distributions; these results are proven in Appendix A.3

The ElementwiseMatrixMultiplication algorithm has been presented for completeness and since in some applications its use may be more appropriate than the use of the BasicMatrixMultiplication algorithm. It is worth emphasizing that the ElementwiseMatrixMultiplication algorithm achieves its spectral norm bound since its sampling procedure may be viewed as adding a carefully constructed random perturbation to every element of the original matrix; see [2, 1] for a nice discussion of these ideas.

Recent work has focused on establishing lower bounds on the number of queries a sampling algorithm is required to perform in order to approximate a given function accurately with low probability of error; see, e.g., [4]. See also [24, 27] for recent related work.

Acknowledgments We would like to thank Dimitris Achlioptas for bringing to our attention the results of [21] and the National Science Foundation for partial support of this work.

References

[1] D. Achlioptas and F. McSherry. Fast computation of low rank matrix approximations. *submitted.*

	$\mathbf{E}\left[\|AB-CR\|_F\right] \leq$	w.h.p. $\|AB-CR\|_F \leq$	comments and restrictions
$p_k \geq \frac{\beta\left\|A^{(k)}\right\|\left\|B_{(k)}\right\|}{\sum_{k'}\left\|A^{(k')}\right\|\left\|B_{(k')}\right\|}$	$\frac{1}{\sqrt{\beta c}}\|A\|_F\,\|B\|_F$	$\frac{\eta}{\sqrt{\beta c}}\|A\|_F\,\|B\|_F$	$\eta = 1+\sqrt{\frac{8}{\beta}\log\left(\frac{1}{\delta}\right)}$
$p_k \geq \frac{\beta\left\|A^{(k)}\right\|^2}{\|A\|_F^2}$	$\frac{1}{\sqrt{\beta c}}\|A\|_F\,\|B\|_F$	$\frac{\eta}{\sqrt{\beta c}}\|A\|_F\,\|B\|_F$	$\eta = 1+\frac{\|A\|_F}{\|B\|_F}\mathcal{M}\sqrt{\frac{8}{\beta}\log\left(\frac{1}{\delta}\right)}$ $\mathcal{M} = \max_\alpha \frac{\left\|B_{(\alpha)}\right\|}{\left\|A^{(\alpha)}\right\|}$
$p_k \geq \frac{\beta\left\|B_{(k)}\right\|^2}{\|B\|_F^2}$	$\frac{1}{\sqrt{\beta c}}\|A\|_F\,\|B\|_F$	$\frac{\eta}{\sqrt{\beta c}}\|A\|_F\,\|B\|_F$	$\eta = 1+\frac{\|B\|_F}{\|A\|_F}\mathcal{M}\sqrt{\frac{8}{\beta}\log\left(\frac{1}{\delta}\right)}$ $\mathcal{M} = \max_\alpha \frac{\left\|A^{(\alpha)}\right\|}{\left\|B_{(\alpha)}\right\|}$
$p_k \geq \frac{\beta\left\|A^{(k)}\right\|}{\sum_{k'=1}^n\left\|A^{(k')}\right\|}$	$\frac{1}{\sqrt{\beta c}}\|A\|_F\,\sqrt{n}\mathcal{M}$	$\frac{\eta}{\sqrt{\beta c}}\|A\|_F\,\sqrt{n}\mathcal{M}$	$\eta = 1+\sqrt{\frac{8}{\beta}\log\left(\frac{1}{\delta}\right)}$ $\mathcal{M} = \max_\alpha \left\|B_{(\alpha)}\right\|$
$p_k \geq \frac{\beta\left\|B_{(k)}\right\|}{\sum_{k'=1}^n\left\|B_{(k')}\right\|}$	$\frac{1}{\sqrt{\beta c}}\sqrt{n}\mathcal{M}\,\|B\|_F$	$\frac{\eta}{\sqrt{\beta c}}\sqrt{n}\mathcal{M}\,\|B\|_F$	$\eta = 1+\sqrt{\frac{8}{\beta}\log\left(\frac{1}{\delta}\right)}$ $\mathcal{M} = \max_\alpha \left\|A^{(\alpha)}\right\|$
$p_k \geq \frac{\beta\left\|A^{(k)}\right\|\left\|B_{(k)}\right\|}{\|A\|_F\|B\|_F}$	$\frac{1}{\sqrt{\beta c}}\|A\|_F\,\|B\|_F$	$\frac{\eta}{\sqrt{\beta c}}\|A\|_F\,\|B\|_F$	$\eta = 1+\sqrt{\frac{8}{\beta}\log\left(\frac{1}{\delta}\right)}$
$p_k = \frac{1}{n}$	See Lemma 11.	See Lemma 11.	See Lemma 11.

Figure 5: Summary of results for different probability distributions

[2] D. Achlioptas and F. McSherry. Fast computation of low rank matrix approximations. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 611–618, 2001.

[3] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 20–29, 1996.

[4] Z. Bar-Yossef. Sampling lower bounds via information theory. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 335–344, 2003.

[5] D. Barbara, C. Faloutsos, J. Hellerstein, Y. Ioannidis, H.V. Jagadish, T. Johnson, R. Ng, V. Poosala, K. Ross, and K.C. Sevcik. The New Jersey data reduction report. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 1997.

[6] R. Bhatia. *Matrix Analysis.* Springer-Verlag, New York, 1997.

[7] E. Cohen and D.D. Lewis. Approximating matrix multiplication for pattern recognition tasks. *Journal of Algorithms*, 30(2):211–252, 1999.

[8] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251–280, 1990.

[9] P. Drineas. *Randomized Algorithms for Matrix Operations.* PhD thesis, Yale University, 2002.

[10] P. Drineas and R. Kannan. Fast Monte-Carlo algorithms for approximate matrix multiplication. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 452–459, 2001.

[11] P. Drineas and R. Kannan. Pass efficient algorithms for approximating large matrices. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 223–232, 2003.

[12] P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. Technical Report YALEU/DCS/TR-1270, Yale University Department of Computer Science, New Haven, CT, February 2004.

[13] P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. Technical Report YALEU/DCS/TR-1271, Yale University Department of Computer Science, New Haven, CT, February 2004.

[14] J. Feigenbaum, S. Kannan, M. Strauss, and M. Viswanathan. An approximate L^1-difference algorithm for massive data sets. In *Proceedings of the 40th Annual IEEE Symposium on the Foundations of Computer Science*, pages 501–511, 1999.

[15] A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 370–378, 1998.

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

[17] A.C. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and M. Strauss. Fast, small-space algorithms for approximate histogram maintenance. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 389–398, 2002.

[18] G.H. Golub and C.F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 1989.

[19] M.R. Henzinger, P. Raghavan, and S. Rajagopalan. Computing on data streams. Technical Report 1998-011, Digital Systems Research Center, Palo Alto, CA, May 1998.

[20] R.A. Horn and C.R. Johnson. *Matrix Analysis.* Cambridge University Press, New York, 1985.

[21] M. Krivelevich and V.H. Vu. On the concentration of eigenvalues of random symmetric matrices. Technical Report MSR-TR-2000-60, Microsoft Research, Redmond,WA, 2000.

[22] C. McDiarmid. On the method of bounded differences. In J. Siemons, editor, *Surveys in Combinatorics, 1989*, London Mathematical Society Lecture Notes Series, pages 148–188. Cambridge University Press, 1989.

[23] J.I. Munro and M.S. Paterson. Selection and sorting with limited storage. In *Proceedings of the 19th Annual IEEE Symposium on Foundations of Computer Science*, pages 253–258, 1978.

[24] M. Rudelson and R. Vershynin. Approximation of matrices. *manuscript.*

[25] G.W. Stewart and J.G. Sun. *Matrix Perturbation Theory.* Academic Press, New York, 1990.

[26] V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 14(3):354–356, 1969.

[27] R. Vershynin. Coordinate restrictions of linear operators in l_2^n. *manuscript.*

$$
\begin{tikzcd}
\mathbb{R}^q \arrow[r, "C"] \arrow[dr, "\tilde{C}"'] & \mathbb{R}^p \arrow[r, "B"] & \mathbb{R}^n \arrow[r, "A"] & \mathbb{R}^m \\
& \mathbb{R}^{c_2} \arrow[u, "S^{(C,c_2)}"'] \arrow[r, "\tilde{B}"] \arrow[loop below, "D^{\{q_k\}}"] & \mathbb{R}^{c_1} \arrow[u, "S^{(A,c_1)}"] \arrow[ur, "\tilde{A}"] \arrow[loop below, "D^{\{p_k\}}"] &
\end{tikzcd}
$$

Figure 6: Diagram for the algorithm to approximately multiply three matrices

A Further Analysis of the Basic Matrix Multiplication Algorithm

In this section we provide further analysis of the BasicMatrixMultiplication algorithm. In Section A.1 we consider approximating the product of more than two matrices by a similar sampling process. Then, in Section A.2 we examine element-wise error bounds for the algorithm and in Section A.3 we consider error bounds for probability distributions which are not nearly optimal in the sense of Section 4.4.

A.1 Approximating the Product of More than Two Matrices

In this section we consider the task of approximating the product of three or more matrices using the ideas of the BasicMatrixMultiplication algorithm of Section 4. For simplicity our exposition will be restricted to the case of approximating the product ABC of three matrices. Recall that given matrices $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{p\times q}$, the product ABC may be written as

$$ABC = \sum_{s=1}^{n}\sum_{t=1}^{p} A^{(s)} B_{st} C_{(t)}. \tag{38}$$

One possible way of extending the ideas of Section 4.1 is the following. Randomly choose $i_s \in \{1,\ldots,n\}$ independently and with replacement c_1 times according to a probability distribution $\{p_i\}_{i=1}^{n}$ and randomly choose $j_t \in \{1,\ldots,p\}$ independently and with replacement c_2 times according to a probability distribution $\{q_j\}_{j=1}^{p}$. Then form the matrix $\tilde{A} \in \mathbb{R}^{m\times c_1}$ with columns $\tilde{A}^{(s)} = A^{(i_s)}/\sqrt{c_1 p_{i_s}}$, the matrix $\tilde{B} \in \mathbb{R}^{c_1\times c_2}$ with elements $\tilde{B}_{st} = B_{i_s j_t}/\sqrt{c_1 c_2 p_{i_s} q_{j_t}}$, and the matrix $\tilde{C} \in \mathbb{R}^{c_2\times q}$ with rows $\tilde{C}_{(t)} = C_{(j_t)}/\sqrt{c_2 q_{j_t}}$ so that

$$\tilde{A}\tilde{B}\tilde{C} = \sum_{s=1}^{c_1}\sum_{t=1}^{c_2} \frac{A^{(i_s)} B_{i_s j_t} C_{(j_t)}}{c_1 c_2 p_{i_s} q_{j_t}}.$$

Figure 6 presents a diagram illustrating the action of the algorithm just described to approximate the product of three matrices. One could then define sampling matrices $S^{(A,c_1)}$ and $S^{(C,c_2)}$ and diagonal rescaling matrices $D^{\{p_k\}}$ and $D^{\{q_k\}}$ in a manner analogous to that of Section 4.1 and as indicated in Figure 6. Then $\tilde{A}\tilde{B}\tilde{C} = AS^{(A,c_1)}{D^{\{p_k\}}}^2{S^{(A,c_1)}}^T BS^{(C,c_2)}{D^{\{q_k\}}}^2{S^{(C,c_2)}}^T C \approx ABC$. An intuitively appealing aspect of this algorithm is that the product ABC is shown as C, B, and then A operating between the high-dimensional $\mathbb{R}^q$ and $\mathbb{R}^m$ via the high-dimensional $\mathbb{R}^p$ and $\mathbb{R}^n$; this is approximated by $\tilde{A}\tilde{B}\tilde{C}$, which acts between $\mathbb{R}^q$ and $\mathbb{R}^m$ via the low-dimensional subspaces $\mathbb{R}^{c_2}$ and $\mathbb{R}^{c_1}$. A difficulty with this algorithm is that its analysis is quite complicated due to the correlation in the non-independent sampling of the elements of the matrix B.

A second way of extending the ideas of Section 4.1 is the following. Randomly choose $(i_s, j_t) \in \{1, \ldots, n\} \times \{1, \ldots, p\}$ independently and with replacement c times according to a probability distribution $\{p_{kl}\}_{(k,l)=1}^{n,p}$. This corresponds to sampling c terms from the sum (38). Then define

$$P = \sum_{u \equiv (s,t)=1}^{c} \frac{1}{cp_{k_s l_t}} A^{(k_s)} B_{k_s l_t} C_{(l_t)},$$

where the summation is a single sum over the c pairs $(k_s, l_t) \in \{1, \ldots, n\} \times \{1, \ldots, p\}$ chosen by the algorithm. In this second algorithm the subspace interpretation of the first algorithm is lost but the analysis simplifies considerably. Using ideas similar to those in Section 4 we can prove the following lemma about this algorithm.

Lemma 6 *Given matrices $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$, $C \in \mathbb{R}^{p \times q}$, construct an approximation P to the product ABC by sampling as described in the second algorithm above with probabilities $\{p_{k,l}\}_{(k,l)=1}^{np}$. Then, for every i, j we have that $\mathbf{E}\left[(P)_{ij}\right] = (ABC)_{ij}$ and that*

$$\mathbf{Var}\left[(P)_{ij}\right] = \frac{1}{c} \sum_{k=1}^{n} \sum_{l=1}^{p} \frac{1}{p_{kl}} A_{ik}^2 B_{kl}^2 C_{lj}^2 - \frac{1}{c} (ABC)_{ij}^2 .$$

In addition,

$$\mathbf{E}\left[\|ABC - P\|_F^2\right] = \frac{1}{c} \sum_{k=1}^{n} \sum_{l=1}^{p} \frac{1}{p_{kl}} \left|A^{(k)}\right|^2 B_{kl}^2 \left|C_{(l)}\right|^2 - \frac{1}{c} \|ABC\|_F^2$$

and the probabilities

$$p_{kl} = \frac{\left|A^{(k)}\right| |B_{kl}| \left|C_{(l)}\right|}{\sum_{k'} \sum_{l'} \left|A^{(k')}\right| |B_{k'l'}| \left|C_{(l')}\right|}$$

minimize $\mathbf{E}\left[\|ABC - P\|_F^2\right]$

Proof: Similar to the proof of Lemma 3 and Lemma 4.

$\diamond$

As in Section 4.4 we will define probabilities $\{p_{kl}\}$ to be nearly optimal if

$$p_{kl} \geq \beta \frac{\left|A^{(k)}\right| |B_{kl}| \left|C_{(l)}\right|}{\sum_{k'} \sum_{l'} \left|A^{(k')}\right| |B_{k'l'}| \left|C_{(l')}\right|}$$

for some $\beta \leq 1$. If sampling is performed with these probabilities, one can show that

$$\mathbf{E}\left[\|ABC - P\|_F^2\right] \leq \frac{1}{c\beta} \sum_{k} \sum_{l} \left|A^{(k)}\right| |B_{kl}| \left|C_{(l)}\right|$$

and a similar result can be shown to hold with high probability.

Unfortunately, computing the optimal probabilities in the general case is not pass-efficient since it would require $O(np)$ additional space and time. This situation would be relatively worse if one wanted to compute the product of more than three matrices, rendering this method uncompetitive with the exact algorithm. On the other hand, if the matrices are known to have a special structure or if the data are presented in a more specialized format then this algorithm may be useful. For example, if it is known that none of the elements of B are too big, i.e., that the elements of B are such that there exists a ξ_B such that for all i, j we have that $B_{ij} \leq \xi_B \|B\|_F^2 / np$ then there will exist a set of probabilities that are nearly optimal that do not depend on B and that can be computed efficiently.

A.2 Element-wise Error Bounds

In this section we provide element-wise error bounds on $|(AB)_{ij} - (CR)_{ij}|$ for the BASICMATRIXMULTIPLICATION algorithm for two different probability distributions. We have the following lemma.

Lemma 7 *Suppose $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$, $c \in \mathbb{Z}^+$ such that $1 \le c \le n$, and $\{p_i\}_{i=1}^n$ are such that $p_i \ge 0$ and $\sum_{i=1}^n p_i = 1$. Let M be such that $|A_{ij}| \le M$ and $|B_{ij}| \le M$ for every appropriate i, j. Construct C and R with the* BASICMATRIXMULTIPLICATION *algorithm, and let CR be an approximation to AB. If $p_k = 1/n$ for every k, then for every $\delta > 0$ with probability at least $1-\delta$*

$$|(AB)_{ij} - (CR)_{ij}| < \frac{nM^2}{\sqrt{c}}\sqrt{8\ln(2mp/\delta)} \quad \forall i,j. \tag{39}$$

If $p_k \ge \frac{\beta|A^{(k)}||B_{(k)}|}{\sum_{k'=1}^n |A^{(k')}||B_{(k')}|}$ for some positive constant $\beta \le 1$, then for every $\delta > 0$ with probability at least $1-\delta$

$$|(AB)_{ij} - (CR)_{ij}| < \frac{n\sqrt{mp}M^2}{\sqrt{\beta c}}\sqrt{(8/\beta)\ln(2mp/\delta)} \quad \forall i,j. \tag{40}$$

Proof: Let us first consider the case of uniform sampling probabilities, i.e., when $p_k = 1/n$. First, fix attention on one particular $(i,j) \in (\{1,\ldots,m\},\{1,\ldots,p\})$. Define $X_t^{(ij)} = \left(\frac{A^{(i_t)}B_{(i_t)}}{cp_{i_t}}\right)_{ij} = \frac{A_{ii_t}B_{i_tj}}{cp_{i_t}}$. From lemma 3 we see that $\mathbf{E}\left[X_t^{(ij)}\right] = \frac{1}{c}(AB)_{ij}$. Define $Y_t^{(ij)} = X_t^{(ij)} - \frac{1}{c}(AB)_{ij}$, $t = 1,\ldots,c$, and note that the Y_t's are independent random variables with $\mathbf{E}\left[Y_t^{(ij)}\right] = 0$ for every $t = 1,\ldots,c$. In addition,

$$\begin{aligned} \left|Y_t^{(ij)}\right| &\le \left|\frac{A_{ii_t}B_{i_tj}}{cp_{i_t}}\right| + \left|\frac{1}{c}(AB)_{ij}\right| \\ &\le \left|\frac{A_{ii_t}B_{i_tj}}{cp_{i_t}}\right| + \frac{nM^2}{c} &(41)\\ &\le \frac{2nM^2}{c} &(42)\end{aligned}$$

(42) follows since for the uniform probabilities $\left|\frac{A_{ii_t}B_{i_tj}}{cp_{i_t}}\right| \le \frac{nM^2}{c}$. By combining the upper and lower bounds provided by (42) with Hoeffding's inequality, we have that for any $t > 0$

$$\mathbf{Pr}\left[\left|\sum_{t=1}^c Y_t^{(ij)}\right| \ge ct\right] \le 2\exp\left(-\frac{2c^2t^2}{\sum_{i=1}^c\left(4nM^2/c\right)^2}\right) = 2\exp\left(-\frac{c^3t^2}{8n^2M^4}\right). \tag{43}$$

Define the event $\mathcal{E}_{ij}$ to be $\left|\sum_{t=1}^c Y_t^{(ij)}\right| \ge ct$ and the event $\mathcal{E} = \bigcup_{i=1}^m\bigcup_{j=1}^p \mathcal{E}_{ij}$. If we then let $t = nM^2\frac{2\sqrt{2}}{c^{3/2}}\sqrt{\ln(2mp/\delta)}$ then by (43) we have that $\mathbf{Pr}[\mathcal{E}_{ij}] \le \frac{\delta}{mp}$. Thus, (39) then follows since

$$\mathbf{Pr}[\mathcal{E}] \le \sum_{i=1}^m\sum_{j=1}^p \mathbf{Pr}[\mathcal{E}_{ij}] \le \sum_{ij}\delta/mp = \delta.$$

When applied to the nonuniform probabilities $p_k \ge \frac{\beta|A^{(k)}||B_{(k)}|}{\sum_{k'=1}^n|A^{(k')}||B_{(k')}|}$ a similar line of reasoning establishes (40). The key step is to note that when using these probabilities we have that

$$\left|\frac{A_{ii_t}B_{i_tj}}{cp_{i_t}}\right| \le \left|\frac{A_{ii_t}B_{i_tj}}{c\beta\left|A^{(k)}\right|\left|B_{(k)}\right|}\sum_{k'=1}^n\left|A^{(k')}\right|\left|B_{(k')}\right|\right| \le \left|\frac{n\sqrt{mp}}{c\beta}M^2\right|. \tag{44}$$

Since $nM^2/c \leq n\sqrt{mp}M^2/(c\beta)$ this, when combined with (41), implies that

$$\left|Y_t^{(ij)}\right| \leq \frac{2n\sqrt{mp}M^2}{c\beta} \tag{45}$$

which provides the upper and lower bounds on the random variable required to apply Hoeffding's inequality.

$\diamond$

When the uniform probabilities are used

$$\|AB - CR\|_F^2 = \sum_{ij} |(AB)_{ij} - (CR)_{ij}|^2 < \frac{mn^2pM^4}{c} 8\log(2mp/\delta)$$

holds with probability greater than $1-\delta$. The difference between this result and the result of Theorem 1 or its variants such as Lemma 11 is that Lemma 7 guarantees that every element of the approximation will have small additive error, while Theorem 1 provides a tighter Frobenius norm bound but not element-wise guarantees.

It may seem counterintuitive that by sampling with respect to the optimal probabilities of Section 4 the bound of (40) is worse than that of (39) by a factor of $\sqrt{mp/\beta}$. (Relatedly, when the nonuniform probabilities of Lemma 7 are used, we have that

$$\|AB - CR\|_F^2 < \frac{m^2n^2p^2M^4}{\beta^2 c} 8\log(2mp/\delta)$$

with probability greater than $1-\delta$.) The reason for this is that the optimal probabilities are optimal with respect to minimizing $\mathbf{E}\left[\|AB - CR\|_F^2\right]$, in which case elements corresponding to smaller columns and rows contribute relatively little. On the other hand, the two statements of Lemma 7 are required to hold for every i and j. Thus, (whether or not the uniform probabilities are nearly optimal) because the optimal sampling probabilities bias toward elements corresponding to larger columns and rows an extra factor of $\sqrt{mp}$ is needed.

A.3 Analysis of the Algorithm for Non-nearly Optimal Probabilities

Note that the nearly optimal probabilities (7) use information from both matrices A and B in a particular form. In some cases, such detailed information about both matrices may not be available. Thus, we present results for the BasicMatrixMultiplication algorithm for several other sets of probabilities. See Figure 5 in Section 6 for a summary of these results.

In the first case, to estimate the product AB one could use the probabilities (46) which use information from the matrix A only. In this case $\|AB - CR\|_F$ can still be shown to be small in expectation and under an additional assumption the expectation can be removed and the corresponding result can be shown to hold with high probability.

Lemma 8 *Suppose $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$, $c \in \mathbb{Z}^+$ such that $1 \leq c \leq n$, and $\{p_i\}_{i=1}^n$ are such that $\sum_{i=1}^n p_i = 1$ and such that*

$$p_k \geq \frac{\beta\left|A^{(k)}\right|^2}{\|A\|_F^2} \tag{46}$$

for some positive constant $\beta \leq 1$. Construct C and R with the BasicMatrixMultiplication *algorithm, and let CR be an approximation to AB. Then:*

$$\mathbf{E}\left[\|AB - CR\|_F^2\right] \leq \frac{1}{\beta c}\|A\|_F^2 \|B\|_F^2 . \tag{47}$$

Furthermore, let $\mathcal{M} = \max_\alpha \frac{|B_{(\alpha)}|}{|A^{(\alpha)}|}$, *let* $\delta \in (0,1)$ *and let* $\eta = 1 + \frac{\|A\|_F}{\|B\|_F}\mathcal{M}\sqrt{(8/\beta)\log(1/\delta)}$. *Then with probability at least* $1-\delta$*:*

$$\|AB - CR\|_F^2 \leq \frac{\eta^2}{\beta c} \|A\|_F^2 \, \|B\|_F^2 . \tag{48}$$

Proof: The proof is similar to that of Theorem 1 except that the indicated probabilities are used. $\diamond$

Alternatively, to estimate the product AB one could use the probabilities (49) which also use information from the matrix A only, but in a different form than the probabilities (46). In this case, under an additional assumption $\|AB - CR\|_F$ can still be shown to be small both in expectation and with high probability.

Lemma 9 *Suppose* $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$, $c \in \mathbb{Z}^+$ *such that* $1 \leq c \leq n$, *and* $\{p_i\}_{i=1}^n$ *are such that* $\sum_{i=1}^n p_i = 1$ *and such that*

$$p_k \geq \frac{\beta \left|A^{(k)}\right|}{\sum_{k'=1}^n \left|A^{(k')}\right|} \tag{49}$$

for some positive constant $\beta \leq 1$. *Let* $\mathcal{M} = \max_\alpha \left|B_{(\alpha)}\right|$. *Construct* C *and* R *with the* BasicMatrixMultiplication *algorithm, and let* CR *be an approximation to* AB. *Then:*

$$\mathbf{E}\left[\|AB - CR\|_F^2\right] \leq \frac{1}{\beta c} \|A\|_F^2 \, n\mathcal{M}^2. \tag{50}$$

Furthermore, let $\delta \in (0,1)$ *and* $\eta = 1 + \sqrt{(8/\beta)\log(1/\delta)}$. *Then with probability at least* $1-\delta$*:*

$$\|AB - CR\|_F^2 \leq \frac{\eta^2}{\beta c} \|A\|_F^2 \, n\mathcal{M}^2. \tag{51}$$

Proof: The proof is similar to that of Theorem 1 except that the indicated probabilities are used. $\diamond$

The probabilities (46) and (49) depend on only the lengths of the columns of A. Results similar to those of the previous two lemmas hold if the probabilities depend on the rows of B rather than the columns of A; see Figure 5.

Alternatively, to estimate the product of AB one could use the probabilities (52); interestingly, although the probabilities differ from those of (7) we are able to derive the same bounds as those of Theorem 1 without additional assumptions.

Lemma 10 *Suppose* $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$, $c \in \mathbb{Z}^+$ *such that* $1 \leq c \leq n$, *and* $\{p_i\}_{i=1}^n$ *are such that* $\sum_{i=1}^n p_i = 1$ *and such that*

$$p_k \geq \frac{\beta \left|A^{(k)}\right| \left|B_{(k)}\right|}{\|A\|_F \, \|B\|_F} \tag{52}$$

for some positive constant $\beta \leq 1$. *Construct* C *and* R *with the* BasicMatrixMultiplication *algorithm, and let* CR *be an approximation to* AB. *Then:*

$$\mathbf{E}\left[\|AB - CR\|_F^2\right] \leq \frac{1}{\beta c} \|A\|_F^2 \, \|B\|_F^2 . \tag{53}$$

Furthermore, let $\delta \in (0,1)$ *and* $\eta = 1 + \sqrt{(8/\beta)\log(1/\delta)}$. *Then with probability at least* $1-\delta$*:*

$$\|AB - CR\|_F^2 \leq \frac{\eta^2}{\beta c} \|A\|_F^2 \, \|B\|_F^2 . \tag{54}$$

Proof: The proof is similar to that of Theorem 1 except that the indicated probabilities are used. ◇

Of course one could estimate the product AB using the uniform probabilities (55). In this case for simplicity we consider bounding $\|AB - CR\|_F$ directly.

Lemma 11 *Suppose $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$, $c \in \mathbb{Z}^+$ such that $1 \le c \le n$, and $\{p_i\}_{i=1}^n$ are such that*

$$p_k = \frac{1}{n}. \tag{55}$$

Construct C and R with the BasicMatrixMultiplication *algorithm, and let CR be an approximation to AB. Then:*

$$\mathbf{E}\left[\|AB - CR\|_F\right] \le \sqrt{\frac{n}{c}} \left(\sum_{k=1}^{n} \left|A^{(k)}\right|^2 \left|B_{(k)}\right|^2\right)^{1/2}. \tag{56}$$

Furthermore, let $\delta \in (0,1)$ and $\gamma = \frac{n}{\sqrt{c}}\sqrt{8\log(1/\delta)}\max_\alpha \left|A^{(\alpha)}\right| \left|B_{(\alpha)}\right|$; then with probability at least $1-\delta$:

$$\|AB - CR\|_F \le \sqrt{\frac{n}{c}} \left(\sum_{k=1}^{n} \left|A^{(k)}\right|^2 \left|B_{(k)}\right|^2\right)^{1/2} + \gamma. \tag{57}$$

Proof: The proof is similar to that of Theorem 1 except that the indicated probabilities are used. ◇