On Sparse Representations of Linear Operators and the Approximation of Matrix Products

Mohamed-Ali Belabbas and Patrick J. Wolfe

Abstract—Thus far, sparse representations have been exploited largely in the context of robustly estimating functions in a noisy environment from a few measurements. In this context, the existence of a basis in which the signal class under consideration is sparse is used to decrease the number of necessary measurements while controlling the approximation error. In this paper, we instead focus on applications in numerical analysis, by way of sparse representations of linear operators with the objective of minimizing the number of operations needed to perform basic operations (here, multiplication) on these operators. We represent a linear operator by a sum of rank-one operators, and show how a sparse representation that guarantees a low approximation error for the product can be obtained from analyzing an induced quadratic form. This construction in turn yields new algorithms for computing approximate matrix products.

I. INTRODUCTION

OPERATIONS on large matrices are a cornerstone of computational linear algebra. With a few exceptions such as the approximate Lanczos and power methods, most algorithms used by practitioners aimed to optimize speed of computation under the constraint of obtaining an exact result. Recently, spurred by the seminal paper of Frieze et al. [1], there has been a greater interest in finding algorithms which sacrifice the precision of the result for a gain in the speed of execution.

Consider the low-rank approximation problem; i.e., finding a matrix $A_k$ of rank at most $k$ which approximates a given matrix $A$. The best matrix $A_k$, best in the sense that it minimizes $\|A - A_k\|$ for any unitarily invariant norm (e.g., spectral or Frobenius norms), can be obtained by computing the singular value decomposition (SVD) of $A$. Throughout this paper, we adopt the Frobenius norm; we use the notation $A_k$ to denote the best rank-$k$ approximation to $A$ and $\tilde{A}_k$ to denote an approximation to it—it will be easy to avoid confusion with $A_k$, which is used to denote the $i^{th}$ column of $A$.) But in some instances, evaluating the SVD, which scales as $O(n^3)$ where $n$ is the largest dimension of $A$, may be too costly. Frieze et al. in [1] showed that the approximation $A_k$ can be reasonably well approximated by computing the SVD of a subset of the columns of $A$ only, where the columns are sampled according to their relative powers—i.e., Pr(pick column $i$) $\propto \|A_i\|^2/\sum \|A_i\|^2$, with the expected error coming from using the approximation $A_k$ instead of $A_k$ being of the form $\mathbb{E}\|A - A_k\|^2 \leq \|A - A_k\|^2 + \varepsilon \|A\|^2$. In subsequent papers it has been argued that the additive error term in $\|A\|^2$ may be large, and thus other sampling techniques have been introduced to obtain relative approximation error bounds (see, e.g., [2], [3]).

In this paper, we address the sparse representation of linear operators for the approximation of matrix products. An important object that will appear in our study is the so-called Nyström method (see Section II) to find a low-rank approximation to a positive kernel. This method, familiar to numerical analysts, has nowadays found applications beyond its original field, most notably in machine learning. In previous work [4], we proposed an approach for low-rank approximation in such applications; here we show that the task of finding optimal sparse representation of linear operators, in order to evaluate their product, is related to the Nyström extension of a certain positive definite kernel. We will use this connection to derive and bound the error of two new algorithms, which our simulations indicate perform well in practice.

Related to our work is that of [5] for matrix products. In that paper, Drineas et al. showed that a randomized algorithm sampling columns $A$ and rows $B^j$ of $A$ and $B$ in proportion to their relative powers $\|A_i\|^2$ and $\|B_j\|^2$ yields an expected error of $\mathbb{E}\|AB - \tilde{AB}\|^2 \leq \text{const} \cdot k^{-1}\|A\|^2\|B\|^2$. Notice that this bound does not involve a low-rank approximation of $A$ or $B$. In contrast, we obtain a randomized algorithm bound in which the approximating rank $k$ of a kernel related to $A$ and $B$ appears explicitly.

The methods mentioned above are all adaptive, in the sense that they require some knowledge about $A$ and $B$. A very simple non-adaptive method is given by an application of the Johnson-Lindenstrauss Lemma: it is easy to show [6] that if $W$ is a $k \times n$ matrix with independent unit Normal entries and $x, y \in \mathbb{R}_n$, then for $0 < \varepsilon < 1$ we have that

$\Pr\left(\|\langle x, y \rangle - k^{-1}\langle Wx, Wy \rangle\| \leq \varepsilon \|x\|\|y\|\right) \geq 1 - 4e^{-\frac{n^2}{2} \left(\frac{\varepsilon^2}{4} - \frac{k^2}{2}\right)}.$

Letting $A^i$ denote the $i^{th}$ row of $A$ and $B^j$ the $j^{th}$ column of $B$, we observe the following element-wise relation:

$$(AB)_{ij} - k^{-1}(AW^TWB)_{ij} = \langle A^i, B^j \rangle - k^{-1}(WA^i, WB^j),$$

and thus we see that approximating $AB$ by $k^{-1}AW^TWB$ yields a good result with high probability. Later we compare this method to our algorithms described below.

The remainder of this paper is organized as follows. In Section II, we briefly review the Nyström method used to approximate positive definite matrices [7], [8]. In Section III, we introduce the problem of approximating a matrix product and highlight two key aspects: the issue of best subset selection and the issue of optimal rescaling. We then solve the optimal rescaling problem and analyze a randomized and a deterministic algorithm for subset selection; we conclude with simulations and a brief discussion of algorithmic complexity.
II. APPROXIMATION VIA THE NYSTRÖM METHOD

To provide context for our results, we first introduce the so-called Nyström method to approximate the eigenvectors of a symmetric positive semi-definite (SPSD) matrix.

A. The Nyström Method for Kernel Approximation

The Nyström method, familiar in the context of finite element methods, has found many applications in machine learning and computer vision in recent years (see, e.g., [8] and references therein). We give here a brief overview: Let $k : [0,1] \times [0,1] \to \mathbb{R}$ be a positive semi-definite kernel and $(\lambda_i, f_i)$, $i = 0, 1, \ldots, M$, denote pairs of eigenvalues and eigenvectors such that

$$
\int_{[0,1]} k(x,y)f_i(y)dy = \lambda_i f_i(x).
$$

The Nyström extension is a method to approximate the eigenvectors of $k(x,y)$ based on a discretization of the interval $[0,1]$. Define the $M+1$ points $x_i$ by $x_m = x_{m-1}+1/M$ with $x_0 = 0$, so that the $x_i$'s are evenly spaced along the interval $[0,1]$. Then form the Gram matrix $K_{mn} := k(x_m,x_n)$, which in turn is used to approximate (1) by a finite-dimensional spectral problem

$$
\frac{1}{M+1} \sum_n K_{mn} v_i(n) = \lambda_i v_i(m), \quad i = 0, 1, \ldots, M.
$$

The Nyström extension then uses these $v_i$ to give an estimate $\hat{f}_i$ of the $i$th eigenfunction as follows:

$$
\hat{f}_i(x) = \frac{1}{(M+1)\lambda_i} \sum_m k(x_m,x_m)v_i(m).
$$

This method can also be applied in the context of matrices. Let $Q$ be an $n \times n$ SPSD matrix, partitioned as

$$
Q = \begin{bmatrix} Q_J & Y \\ Y^T & Z \end{bmatrix},
$$

where $Q_J \in \mathbb{R}^{k \times k}$ and $k$ is typically much smaller than $n$. It is then possible to approximate $k$ eigenvectors and eigenvalues of $Q$ by using the eigendecomposition of $Q_J$ as follows. Define $Q = U\Lambda U^T$ and $Q_J = U_J\Lambda_J U_J^T$ with $U, U_J$ orthogonal and $\Lambda, \Lambda_J$ diagonal. The Nyström extension then tells us that an approximation for $k$ eigenvectors in $U$ is given by

$$
\tilde{U} = \begin{bmatrix} U_J \\ Y^T U_J \Lambda_J^{-1} \end{bmatrix}.
$$

These approximations $\tilde{U} \cong U$ and $\Lambda_J \cong \Lambda$ in turn yield an approximation $\tilde{Q}$ to $Q$ as follows:

$$
\tilde{Q} = \tilde{U} \Lambda_J \tilde{U}^T = \begin{bmatrix} Q_J & Y \\ Y^T & Y^T Q_J^{-1} Y \end{bmatrix}.
$$

The quality of this approximation can then be measured as the (e.g., Frobenius) norm of the Schur complement of $Q_J$ in $Q$:

$$
\|Q - \tilde{Q}\| = \|Z - Y^T Q_J^{-1} Y\|.
$$

III. APPROXIMATION OF MATRIX PRODUCTS

Let $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times p}$. We use the notation $A_i$ to denote the columns of $A$ and $B^i$ the rows of $B$. We can write the product $AB$ as the sum of rank-one matrices as follows:

$$
AB = \sum_{i=1}^n A_i B^i. \tag{2}
$$

Our approach to estimate the product $AB$, akin to model selection in statistics, will consist of keeping only a few terms in the sum of (2); this entails choosing a subset of columns of $A$ and of rows of $B$, and rescaling their outer products as appropriate. To gain some basic insight into this problem, we may consider the following two extreme cases with $A \in \mathbb{R}^{n \times 2}$ and $B = A^T$. First, suppose that the vectors $A_1$ and $A_2$ are collinear. Then $AB = A_1 A_1^T + \alpha A_1 A_2^T = (1+\alpha)A_1 A_1^T$, hence we can recover the product without error by only keeping one term of the sum of (2) and rescaling it appropriately provided that we know the correlation between $A_1$ and $A_2$. At the other extreme, if $A_1$ and $A_2$ are orthogonal, rescaling will not decrease the error no matter which term in (2) is kept.

Hence we see that there are two key aspects to the problem of sparse matrix product approximation as formulated above:

Optimal Model Selection

Which rows/columns should be retained?

Optimal Reweighting

How should these rows/columns be rescaled?

As we show below, the latter of these problems can be solved exactly for a relatively low complexity. For the former, which is combinatorial in nature and seemingly much harder to solve, we give an efficient approximation procedure.

IV. SOLVING THE OPTIMAL REWEIGHTING PROBLEM

We first consider the problem of optimal reweighting, conditioned upon a choice of subset. In particular, suppose that an oracle gives us the best subset $J \subset \{1,\ldots,n\}$ of cardinality $k$ to estimate the product $AB$. Without loss of generality, we assume that $J = \{1,\ldots,k\}$. We then have the following result characterizing how well one can estimate the product $AB$:

**Theorem 1.** Let the $n \times n$ SPSD matrix $Q$ be defined as $Q_{ij} = \langle A_i, A_j \rangle \langle B^i, B^j \rangle$ (i.e., $Q = (A^T A) \odot (B B^T)$, where $\odot$ is the Hadamard or entrywise product of matrices) and have the partition

$$
Q = \begin{bmatrix} Q_J & Y \\ Y^T & Z \end{bmatrix}, \tag{3}
$$

where $J = \{1,\ldots,k\}$ without loss of generality, and $Q_J$ is the corresponding principal submatrix.

Then the best approximation to the product $AB$ using the terms $\{A_i B^i\}_{i \in J}$ is given by

$$
AB \cong \tilde{A} \tilde{B} := \sum_{i \in J} w_i A_i B^i, \tag{4}
$$

where

$$
w := Q_{,\tilde{J}}^{-1} r, \quad r := \sum_{j=1}^n \langle A_i, A_j \rangle \langle B^i, B^j \rangle, \quad i \in J.
$$
Moreover, if $E$ is the $(n-k) \times (n-k)$ matrix with all entries equal to one, then the squared approximation error in Frobenius norm is given by
\[
\|AB - \tilde{AB}\|_F^2 = \text{tr}(S_C(Q_j)E),
\]
with $S_C(Q_j) := Z - Y^T Q_j^{-1} Y$ the Schur complement of $Q_j$ in $Q$.

This result tells us how well we can approximate the product $AB$ granted that we know only a few rows/columns of $A$ and $B$, and their correlations with the remaining rows and columns. It also allows us to characterize the best subset $J$ of size $k$; it is the subset that minimizes $\text{tr}(S_C(Q_j)E)$.

**Proof:** Given the subset $J$ of $\{1, \ldots, n\}$, we seek the best scaling factors $w_i$ to minimize the squared approximation error $\|AB - \tilde{AB}\|^2$. We can write the squared error as
\[
\|AB - \sum_{i=1}^k w_i A_i B_i^T\|^2 = \text{tr}\left((AB - \sum_{i=1}^k w_i A_i B_i^T)(AB - \sum_{i=1}^k w_i A_i B_i^T)^T\right).
\]

By distributing the product and using the linearity of the trace, we get
\[
\text{tr}\left((AB - \sum_{i=1}^k w_i A_i B_i^T)(AB - \sum_{i=1}^k w_i A_i B_i^T)^T\right) = \text{tr}\left(((AB)^T AB - 2 \sum_{i=1}^k w_i \text{tr}((AB)^T A_i B_i^T) + \sum_{i=1}^k \text{tr}(w_i A_i B_i^T)\sum_{i=1}^k w_i A_i B_i^T)\right),
\]

where we made use of the following equality:
\[
\text{tr}\left((AB)^T A_i B_i^T\right) = \text{tr}\left(((AB)^T A_i B_i^T)^T\right) = \text{tr}\left((A_i B_i^T)^T AB\right).
\]

We now work towards rewriting (5) in a more manageable form. First, using the fact that $\text{tr}(AB) = \text{tr}(BA)$, we see that
\[
\text{tr}(A_i B_i^T) = \langle B_i^T, A_i \rangle.
\]

By combining (2) and (6), we have
\[
\sum_{i=1}^k w_i \text{tr}\left((AB)^T A_i B_i^T\right) = \sum_{i=1}^k w_i \text{tr}\left(\sum_{j=1}^n (B_j)^T A_i B_i^T\right) = \sum_{i=1}^k w_i \sum_{j=1}^n \langle B_i^T, B_j^T \rangle \langle A_i, A_j \rangle = \langle w, r \rangle.
\]

Similarly, using (6), we get after an easy computation that
\[
\text{tr}\left(\sum_{i=1}^k w_i A_i B_i^T\sum_{i=1}^k w_i A_i B_i^T\right) = \sum_{i,j} w_i w_j \langle A_i, A_j \rangle \langle B_i^T, B_j^T \rangle = w^T Q_j w.
\]

We now rewrite (5) in a more manageable form:
\[
\|AB - \tilde{AB}\|^2 = \|AB\|^2 - 2\langle w, r \rangle + w^T Q_j w.
\] (7)

The optimal weight vector is now obtained by setting the gradient of (7) to zero. Hence we obtain
\[
w = Q_j^{-1} r,
\]
which proves the first part of the statement.

For the second part, first notice that if $[1]$ is the vector whose entries are all one, we have the following expression for $r$:
\[
r = [Q_j Y][1].
\]

Hence, at the optimum, the error is
\[
\|AB - \tilde{AB}\|^2 = \|AB\|^2 - \|[1]^T (Q_j Y) Y^{-1} [Q_j Y][1]\|
\]
\[
= \|AB\|^2 - \|[1]^T \tilde{Q}[1]\|
\]
where we see that $\tilde{Q}$ is the Nyström approximation of $Q$ as described in Section II. Using Lemma 1 below, we have
\[
\|AB - \tilde{AB}\|^2 = \|[1]^T (Q - \tilde{Q})[1]\|
\]
which finishes the proof of the Theorem.

The proof of the second part of Theorem 1 is based on the identity proven below:

**Lemma 1.** Let $A$ and $B$ be real matrices of dimensions $m \times n$ and $n \times p$, respectively, and let $E$ be the $n \times n$ matrix with all entries equal to one. The following identity holds:
\[
\|AB\|^2 = \text{tr}((A^T A) \odot (BB^T)E).
\] (8)

**Proof:** Recall that we can write the product $AB$ as a sum of $n$ rank-one terms as follows:
\[
AB = \sum_{i=1}^n A_i B_i^T.
\]

We thus have, by definition of the Frobenius norm, that
\[
\|AB\|^2 = \text{tr}((AB)^T AB) = \text{tr}\left(\sum_i (A_i B_i^T)^T \sum_j (A_j B_j^T)\right) = \text{tr}\left(\sum_{i,j} (B_i^T A_i^T A_j B_j^T)\right) = \sum_{i,j} \text{tr}(B_i^T A_i^T A_j B_j^T).
\]

Using the invariance of the trace with respect to cyclic permutations, the last equation yields
\[
\|AB\|^2 = \sum_{i,j} \langle A_i, A_j \rangle \langle B_i^T, B_j^T \rangle,
\]
and the relation (8) is proved.
A. Approximating the optimal Subset Selection Procedure

Having shown a solution to the optimal reweighting problem according to Theorem 1, we now turn our attention to the companion problem of optimal subset selection. In order to minimize the approximation error, we have to find the subset \( J \) whose associated Schur complement \( S_C(Q_J) \) has the lowest possible power along the one-dimensional subspace of \( \mathbb{R}^{n-k} \) spanned by the vector \([1]\). Determining the eigenvectors and eigenvalues of this Schur complement, and relating them to \( A \) and \( B \), is not an easy task. Here we present two approximations: one based on a random choice of subsets, and an alternative “greedy” approach which yields a worst-case error bound.

1) Random Choice of Subset: We first discuss a random oracle which outputs a subset \( J \) with probability \( p_{Q,k}(J) \) defined below. Recall our earlier definition of the matrix \( Q = A^T A \circ BB^T \) according to Theorem 1; this approach is motivated by the expression of the resultant squared error, conditioned upon having chosen a subset \( J \subset \{1, \ldots, n\} \), as \( \|AB - \tilde{AB}\|^2 = \text{tr}(S_C(Q_J))E \). Since \( S_C(Q_J) \) is positive definite, we have that \( \text{tr}(S_C(Q_J)) \) is larger than the largest eigenvalue of \( S_C(Q_J) \), and we can bound this error as follows:

\[
\|AB - \tilde{AB}\|^2 \leq (n-k) \text{tr}(S_C(Q_J)). \tag{9}
\]

Note that equality is obtained when \( S_C(Q_J) \propto E \), and hence this bound is tight. We have investigated in \([9]\) an algorithm to minimize \( \|S_C(Q_J)\| \), which has been shown to be effective in the context of low-rank covariance matrix approximation.

Returning to our random oracle, note that both \( A^T A \) and \( BB^T \) are positive definite, and thus by the Schur Theorem \([10]\), \( Q \) is also positive definite. From this, we conclude:

1) There exists a matrix \( X \in \mathbb{R}^{n \times n} \) such that \( Q = X^T X \);
2) All the principal minors \( \det(Q_J) \) of \( Q \) are positive.

Consequently, we assume here that an oracle returns a subset \( J \subset \{1, \ldots, n\} \), \(|J| = k\) with probability

\[
p_{Q,k}(J) := \frac{1}{K} \det(Q_J), \tag{10}
\]

where \( K = \sum_{J,|J|=k} \det(Q_J) \) is a normalizing constant, and the second fact above ensures that this probability distribution is well defined. We may then adapt the following result from the proof of Theorem 1 in \([4]\):

**Theorem 2.** Let \( Q \in \mathbb{R}^{n \times n} \) be a positive quadratic form with eigenvalues \( \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \). If \( J \subset \{1, \ldots, n\} \), \(|J| = k\) is chosen with probability \( p_{Q,k}(J) \propto \det(Q_J) \), then

\[
\mathbb{E} \text{tr}(S_C(Q_J)) \leq (k+1) \sum_{i=k+1}^n \lambda_i. \tag{11}
\]

Combining (9) with (11) leads, via Jensen’s inequality, to an upper bound on the average error of this approach to random subset selection:

\[
\mathbb{E} \|AB - \tilde{AB}\| \leq \sqrt{(n-k)(k+1)} \|X - X_k\|, \\
\text{where } X \text{ is defined via the relation } X^T X = A^T A \circ BB^T, \text{ and } X_k \text{ denotes the optimal rank-}k \text{ approximation to } X \text{ obtained by truncating its singular value decomposition.}
\]

Despite the appearance of the term \( \sqrt{n-k} \) in this bound, it serves to relate the resultant approximation quality to the ranks of \( A \) and \( B \), a feature reinforcing the well-foundedness of the accompanying algorithm we present below. In particular, if \( k > \text{rank}(A) \text{ rank}(B) \), then the approximation error is zero as expected. For practical reasons, we may also wish to relate this error to the eigenvalues of \( A \) and \( B \). To this end, let \( M \) and \( N \) be two \( n \times n \) matrices, \( P = M \odot N \), and let \( \sigma_i(M) \) (resp. \( \sigma_i(N) \)) be the singular values of \( M \) (resp. \( N \)) sorted in non-increasing order. We then have the following majorization relation \([11]\):

\[
\sum_{i=1}^m \sigma_i(P) \leq \sum_{i=1}^m \sigma_i(M) \sigma_i(N), \text{ for } m = 1, 2, \ldots, n.
\]

In particular, if \( M = A^T A, N = BB^T \), and \( Q = X^T X = M \odot N \), then the singular values of \( Q, M, \) and \( N \) are the squares of the singular values of \( X, A \) and \( B \) respectively:

\[
\sum_{i=1}^m \sigma_i^2(Q) \leq \sum_{i=1}^m \sigma_i^2(A) \sigma_i^2(B), \text{ for } m = 1, 2, \ldots, n. \tag{12}
\]

We may then conclude from (12) that

\[
\|X - X_k\|^2 \leq \min \left( \sigma_i^2(A) \|B\|^2, \sigma_i^2(B) \|A\|^2 \right) - \|X_k\|^2.
\]

Although the approach presented above relies on an oracle to sample in proportion to \( \det(Q_J) \), we will subsequently outline a realizable algorithm based on these results.

2) Deterministic Choice of Subset: Recall that Theorem 1 indicates we should ensure that the diagonal terms of \( Z \) are kept as small as possible. Hence, as a deterministic approximation to the optimal subset selection procedure, we may take \( J \) such that it contains the indices of the \( k \) largest terms \( \{A_i, A_j\} \langle B^i, B^j \rangle \). While yielding only a worst-case error bound, this approach has the advantage of being easily implementable (as it does not require sampling according to \( \det(Q_J) \)); it also appears to perform well in practice \([9]\). This greedy algorithm proceeds as follows:

**Algorithm 1 Greedy Approximate Matrix Multiplication**

Given matrices \( A \in \mathbb{R}^{n \times n} \) and \( B \in \mathbb{R}^{n \times p} \) and a positive integer \( k \leq n \):

1) Set \( T := \{\langle A_i, A_j \rangle \langle B^i, B^j \rangle \}, i = 1, \ldots, n \), and take \( J := \{i_1, \ldots, i_k\} \) to be the indices of the \( k \) largest elements of \( T \).
2) Set \( Q \in \mathbb{R}^{k \times k} \) as \( Q_{ij} := \langle A_i, A_j \rangle \langle B^i, B^j \rangle \), for \( i, j \in J \).
3) Set \( r \in \mathbb{R}^k \) as \( r_i := \sum_{j=1}^n \langle A_j, B_j \rangle \langle B^i, B^j \rangle \), for \( i \in J \).
4) Set \( w := Q^{-1} r \) and \( A_J := \{A_i\}, B_J := \{B^i\} \) for \( i \in J \).
5) Return \( \tilde{AB} := A_J \text{ diag}(w) B_J \) as an approximation to \( AB \).

Since the error term is the sum of all the terms in the Schur complement, we can look to bound its largest element. To this end, we have the following result:

**Lemma 2.** The largest entry in \( S_C(Q_J) \) is smaller than the largest diagonal element of \( Z \) in (3).
This lemma confirms that a good error-minimization strategy is to make sure that the diagonal terms of $Z$ are as small as possible, or equivalently to take $J$ such that it contains the indices of the $k$ largest $\langle A_i, A_j \rangle \langle B^i, B^j \rangle$ as per Algorithm 1.

The proof of Lemma 2 is based on the following set of simple results:

**Lemma 3.** If $Q$ is a positive definite matrix, then $\max_{i,j} Q_{ij}$ is positive and on the diagonal of $Q$.

*Proof:* Since $Q$ is positive definite, we know there exists a matrix $X \in \mathbb{R}^{n \times n}$ such that

$$Q_{ij} = \langle X_i, X_j \rangle.$$

By the Cauchy-Schwartz inequality, we have

$$\langle X_i, X_j \rangle^2 \leq \langle X_i, X_i \rangle \langle X_j, X_j \rangle,$$

from which we deduce that one of the following inequalities has to be satisfied:

$$\langle X_i, X_j \rangle \leq \langle X_i, X_i \rangle \quad \text{or} \quad \langle X_i, X_j \rangle \leq \langle X_j, X_j \rangle. \quad (13)$$

Now if we suppose that $\max_{i,j} Q_{ij}$ is not a diagonal element, the relations of (13) yield a contradiction—and hence the largest entry of $Q$ is on its main diagonal.

The entries of $S_C(Q_J)$, the Schur complement of $Q_J$ in $Q$, can be characterized explicitly according to the following formula:

**Lemma 4 (Crabtree-Haynsworth [12]).** Let $Q_J = Q_{1\ldots k, 1\ldots k}$ be a nonsingular leading principal submatrix of $Q$ obtained by keeping the rows and columns with indices $1, \ldots, k$. Then $S_C(Q_J)$, the Schur complement of $Q_J$ in $Q$, is given element-wise by

$$(S_C(Q_J))_{ij} = \frac{\det(Q_{1\ldots k, i\ldots k, 1\ldots k})}{\det(Q_J)}. \quad (14)$$

Furthermore, it is possible to bound the diagonal entries of $S_C(Q_J)$ as follows:

$$\det(Q_{J∪\{i\}}) < \det(Q_J)Q_{ii}.$$

We are now ready to give the proof of Lemma 2:

*Proof of Lemma 2:* The preceding two lemmas tell us that the diagonal entries of $S_C(Q_J)$ are bounded by $\max_{i,j} Q_{ii}$ (i.e., the largest diagonal element of $Z$, according to the partition of (3)). And using Lemma 3, we know that every entry of $S_C(Q_J)$ is bounded by these diagonal entries.

Lemma 2 can be further refined to give a worst-case error bound for deterministic matrix product approximation, conditioned on a choice of subset $J$ and the corresponding optimal reweighting procedure. Appealing to the inequality of arithmetic and geometric means to further bound the elements of $S_C(Q_J)$, the results of Theorem 1 and Lemmas 3–5 yield:

$$\|AB - \tilde{A}\tilde{B}\| \leq \sqrt{(n - k) \sum_{i \in J} (\|A_i\|^2 \|B^i\|^2)}.$$

V. SIMULATION STUDIES AND COMPLEXITY

A. Experimental Results

We now present preliminary experimental results and discuss the computational complexity of the algorithms under consideration. Three sets of experiments were performed, in which we compared the performance of four subset selection methods: a baseline uniform sampling on $k$-subsets; sampling according the row/column powers [5]; sampling in proportion to the $k$-principal minors of $Q$ according to (10); and selecting greedily according to Step 1 of Algorithm 1. We also compared the choice of reweighting following subset selection, in one case applying the optimal reweighting of Theorem 1 and in the other simply reweighting according to the row/column powers (see [1], [5]):

$$\tilde{A}\tilde{B} = \sum_{i \in J} \frac{1}{\sqrt{|J|\|A_i\|^2 \|B^i\|^2}} A_iB^i. \quad (15)$$

Fig. 1. Matrix product approximation error using the power rescaling of (15) applied to each of the four subset selection algorithms described in Sec. V-A.

Fig. 2. Matrix product approximation error using the optimal rescaling of Theorem 1 applied to the subset selection algorithms described in Sec. V-A.
To test these various experimental conditions, we drew 200 random matrices $A \in \mathbb{R}^{60 \times 15}$ and $B \in \mathbb{R}^{15 \times 90}$ in total, each having independent unit Normal entries. We then averaged the error of the randomized algorithm over 20 trials per matrix product, and report relative error in dB as $20 \log_{10} \left( \frac{\|AB - \bar{AB}\|}{\|A\|\|B\|} \right)$ for each test condition.

In the first set of experiments, shown in Figure 1, we compare the four different algorithms for subset selection described above, applied in conjunction with a reweighting according to row/column powers. The highest-error method in this case corresponds to choosing the subset $J$ uniformly at random, and thus should be understood as a baseline measure of performance as a function of approximant rank $k$. It can also be seen that sampling $J$ according to the relative powers of the row/columns of $A$ and $B$, and sampling via a Metropolis-Hastings algorithm (with independent proposal distributions taken in proportion to row/column powers), yield similar results, with both improving upon the baseline performance. The best results in this case are obtained by the greedy subset selection method indicated by Step 1 of Algorithm 1.

In a second set of experiments, we followed the same procedure as above to compare subset selection procedures, but applied the optimal reweighting of Theorem 1 rather than a rescaling according to row/column powers. Performance in this case is (as expected) seen to be better overall, but with the ordering of the methods unchanged. As our final experiment, we compare the method of Algorithm 1 (greedy subset selection followed by optimal rescaling) to two non-adaptive methods: choosing row/columns of $A$ and $B$ uniformly at random and rescaling according to $n/k$, and the simple Johnson-Lindenstrauss random projection approach outlined in Section I. These non-adaptive methods can be seen to yield significantly worse performance than Algorithm 1, suggesting its potential as a practical method of selecting sparse representations of linear operators that yield low approximation errors for the resultant matrix products.

We conclude with a brief discussion of the algorithmic complexity of Algorithm 1. First, assume without loss of generality that $m \geq n, p$, and recall that straightforward matrix multiplication requires $O(m^3)$ operations, though the best algorithm known so far (the Coppersmith-Winograd algorithm [13]) can perform this operation in $O(m^{2.38})$. Evaluating $T$ in Algorithm 1 requires the computation of $2n$ inner products of $m$-or $p$-dimensional vectors, and hence requires $O(2mn)$ operations. Extracting the $k$ largest elements of a set of size $m$, as is necessary to construct $J$, can be done efficiently using a variation on the Quicksort algorithm (see [14]) in $O(m \log k)$. The matrix $Q$ is symmetric and its diagonal is a restriction of $T$. Hence it requires the computation of an additional $2 \times k(k-1)/2$ inner products, and thus $O(mk(k-1))$ operations. Evaluating $r$ requires $O(2m(n-k))$ operations, taking into account the fact that $k$ terms of the sum appear in $Q$. Finally, evaluating $w$ can be done using Gaussian elimination in $O(k^3)$ operations. Hence the overall complexity is given by $O(m(k(k-1)+2(n-k)+\log k)+k^3) = O(m(n+k^2)+k^3)$.

**REFERENCES**

[1] A. M. Frieze, R. Kannan, and S. Vempala, “Fast Monte-Carlo algorithms for finding low-rank approximations,” in Proc. 39th IEEE Sympos. Found. Comp. Sci. (FOCS), 1998, pp. 370–378.

[2] A. Deshpande and S. Vempala, “Adaptive sampling and fast low-rank matrix approximation,” in Proc. 10th Internat. Worksh. Randomiz. Computat. (RANDOM’06), 2006.

[3] T. Sarlos, “Improved approximation algorithms for large matrices via random projection,” in Proc. 47th IEEE Sympos. Found. Comp. Sci. (FOCS), 2006, pp. 143–152.

[4] M.-A. Belabbas and P. J. Wolfe, “Spectral methods in machine learning: New strategies for very large datasets,” submitted to Proc. Natl. Acad. Sci. USA, 2007.

[5] P. Drineas, R. Kannan, and M. W. Mahoney, “Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication,” SIAM J. Comput., vol. 36, pp. 132–157, 2006.

[6] R. I. Arriaga and S. Vempala, “An algorithmic theory of learning: Robust concepts and random projection,” in Proc. 40th IEEE Sympos. Found. Comp. Sci. (FOCS), 1999, pp. 616–623.

[7] C. K. I. Williams and M. Seeger, “Using the Nyström method to speed up kernel machines,” in Neural Information Processing Systems, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., pp. 682–688, MIT Press, 2001.

[8] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, “Spectral grouping using the Nyström method,” IEEE Trans. Patt. Anal. Mach. Intell., vol. 2, pp. 214–225, 2004.

[9] M.-A. Belabbas and P. J. Wolfe, “Fast low-rank approximation for covariance matrices,” in Proc. 2nd IEEE Intl. Worksh. Computat. Adv. Multi-Sensor Adapt. Process., 2007, in press.

[10] R. A. Horn and C. R. Johnson, Matrix Analysis, Cambridge University Press, 1999.

[11] R. A. Horn and C. R. Johnson, Topics in Matrix Analysis, Cambridge University Press, 1994.

[12] D. E. Crabtree and E. V. Haynsworth, “An identity for the Schur complement of a matrix,” Proc. Amer. Math. Soc., vol. 22, pp. 364–366, 1969.

[13] D. Coppersmith and S. Winograd, “Matrix multiplication via arithmetic progressions,” J. Symbol. Computat., vol. 9, no. 3, pp. 251–280, 1990.

[14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, MIT Press, 2001.