Fast Operator Norm Low Rank Approximation Inside a Subspace

Chirag Gupta [1] **Praneeth Kacham** [1] **David P. Woodruff** [1]

Abstract

Given a matrix $A \in \mathbb{R}^{n \times d_A}$ and a matrix $B \in \mathbb{R}^{n \times d_B}$, we consider solving the low rank approximation problem inside a subspace with operator norm error:

$$\text{OPT} := \min_{\text{rank}(X) \leqslant k} \|AX - B\|_2.$$

In the Fröbenius norm case, $X = (\Sigma V^T)^{-1}[U^T B]_k$ is optimal for the above problem, where $A = U\Sigma V^T$, but no closed form solution is known for the operator norm. The problem for operator norm has important applications in control theory and constrained low rank approximation.

We work in the high-dimensional and big-data setting for B: $d_A \ll d_B, n$ and $\text{sr}(B) \ll d_B$, where $\text{sr}(B)$ denotes the stable rank of B. The work of Sou and Rantzer (2012) shows how to obtain a $(1+\epsilon)$-accurate solution in time

$$O(nd_B^2 + d_B^3 + d_A d_B^2(\log(\|B\|_2/\text{OPT})/\epsilon).$$

Using dimensionality reduction techniques, we show how to obtain an additive approximation error of $\epsilon\|B\|_2$ for this problem with a substantially improved running time of

$$\begin{aligned} &O(\text{nnz}(B) + nd_A^2 + r^2 d_B + r^3) \\ &+ O(d_A^3 d_B + (r d_A d_B \log d_B)/\epsilon) \\ &+ \tilde{O}(k^2 d_B) \cdot \text{poly}\,(1/\epsilon) \end{aligned}$$

where $k = \max(d_A, \text{sr}(B))$ and $r = O(k/\epsilon^4)$. We also empirically confirm our improved algorithm.

1. Introduction

There are a number of constructions of so-called coresets, which are succinct representations which preserve properties of one's underlying data. In the context of matrices, a common technique, introduced in (Cohen et al., 2015), is that of a projection cost-preserving sketch (PCP). Here the idea is to first replace a matrix A with AR, where R has a small number r of columns. One can view the rows of A as points and the columns of AR as the important features or directions in one's data set. For example, R could correspond to a column subset selection matrix (Boutsidis et al., 2009; Guruswami and Sinop, 2012; Boutsidis and Malioutov, 2013; Boutsidis et al., 2014; Boutsidis and Woodruff, 2017; Chierichetti et al., 2017; Song et al., 2019a;b), and AR would be a small subset of important columns. Given a new set B of points, one may want to map it to the feature space, namely, project its columns onto the column span of AR, while at the same time replacing B with a low rank approximation for efficiency purposes. Letting k be the rank of B and ϵ an accuracy parameter for spectral norm low rank approximation, this leads us to exactly the situation studied in this paper, where we would like to solve for

$$\min_{\text{rank}-k\ X} \|ARX - B\|_2.$$

As another applications, consider the problem of approximating a matrix $B \in \mathbb{R}^{n \times d_B}$ with a rank$-k$ matrix $B_k \in \mathbb{R}^{n \times d_B}$. If n and d_B are very large, the optimal rank$-k$ projection (in any Schatten $p-$norm) is inefficient to compute. One common solution is to select a random subset of columns of B with size $d_A > r$ to obtain a matrix $A \in \mathbb{R}^{n \times d_A}$ and solve the problem

$$\min_{\text{rank}-k\ X} \|AX - B\|_*, \tag{1}$$

for an appropriate norm. In this case we would like to identify efficient algorithms to compute:

(a) a good column subset matrix A such that $\min_{\text{rank}-k\ X} \|AX - B\|_*$ is not much worse than $\|B_k - B\|_*$;

(b) exact or approximate solutions to (1).

While (a) has received considerable attention for the Fröbenius and operator norm settings (see Boutsidis et al. (2014) and references therein), (b) remains open for operator norm (see discussion in Section 1.3 of Boutsidis et al. (2014)). In this paper, we discuss efficient algorithms to obtain solutions $\hat{X}$ of rank k that satisfy:

$$\|A\hat{X} - B\|_2 \leqslant \text{OPT} + \epsilon\|B\|_2,$$

where

$$\text{OPT} := \min_{\text{rank}-k\ X} \|AX - B\|_2. \tag{2}$$

The Fröbenius norm version of (1) has the solution

$$(\Sigma V^T)^{-1}[U^T B]_k \tag{3}$$

where $[M]_k$ denotes the best rank$-k$ approximation of matrix M in Fröbenius norm. Indeed, for the Fröbenius norm, it is equivalent to minimizing, over rank-k matrices X, the problem $\|ARX - B\|_F^2$. But now, by the Pythagorean theorem, we can write this cost as $\|ARX - P_{AR}B\|_F^2 + \|P_{AR}B - B\|_F^2$. Importantly, the latter term does not depend on X, and is the fixed cost any solution pays to move to the column span of AR. Now $P_{AR}B$ is a matrix in the column span of AR already, so its best rank-k approximation is given by its singular value decomposition (or SVD), and is $[P_{AR}B]_k$, and setting $X = (AR)^-[P_{AR}B]_k$ then gives us the desired X.

It is natural to try to use the Fröbenius norm solution X_F as a solution for the operator norm. Indeed, Boutsidis (2011) showed that this is a $\sqrt{2}$-approximation, namely, that $\|AX_F - B\|_2 \leqslant \sqrt{2}$ OPT. An open question, posed in Remark 2 of Boutsidis (2011), was to design a fast algorithm with a better approximation factor. Unfortunately, there are examples where the Fröbenius norm solution really does give at best a 2-approximation. Suppose, for example

$$B = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1+\gamma \end{bmatrix}, A = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Then for the problem $\min_{\text{rank}-1\ X} \|AX - B\|_F$, the solution is

$$X = \begin{bmatrix} 0 & 0 \\ 0 & 1+\gamma \end{bmatrix}, \text{with } AX - B = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}$$

and thus $\|AX - B\|_2^2 = 2$. On the other hand, if

$$X = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \text{ then } AX - B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1+\gamma \end{bmatrix},$$

and so $\|AX - B\|_2^2 = (1+\gamma)^2$. As $\gamma \to 0$, the approximation factor becomes arbitrarily close to 2.

While it is possible to solve the problem for operator norm in polynomial time, the only known solution, discussed in the following section, takes the same amount of time to compute the SVD of A and B, which is prohibitive.

2. Related work

Let X^* be the minimizer of (2). Let $U\Sigma V^T$ be the SVD of A. The first algorithm to solve (2) was an iterative method by Sou and Rantzer (2012) (Algorithm 1). They considered the following auxiliary problem for some $s > 0$:

Algorithm 1 (Sou and Rantzer, 2012)

Require: $A \in \mathbb{R}^{n \times d_A}, B \in \mathbb{R}^{n \times d_B}, k \in \mathbb{Z}$
procedure SOURANTZER(A, B, k, ϵ)
  $[U, \Sigma, V^T] \leftarrow \text{SVD}(A)$
  $N \leftarrow$ Basis for orthogonal space to colspan(U)
  $s \leftarrow \|B\|_2$
  while $\sigma_{k+1}(U^T B(s^2 I - \Delta)^{-1/2}) < 1$ **do**
    $s \leftarrow s/(1+\epsilon)$
  end while
  $s \leftarrow s(1+\epsilon)$
  $D \leftarrow s^2 I - \Delta$
  $Y \leftarrow (\Sigma V^T)^{-1}(U^T B D^{-1/2})_k (D)^{1/2}$
  return Y
end procedure

$$\begin{aligned} &\text{minimize rank}(X) \\ &\text{such that } \|AX - B\|_2 < s. \end{aligned} \tag{4}$$

It is easy to see that the solution of (2) is the infimum over s such that the solution for objective (4) is at most k. The following theorem characterizes the solution for (4). Before stating it we introduce some notation (to be used in the rest of the paper). We use N to denote any orthonormal basis for the space orthogonal to colspan(A) and define $\Delta := B^T N N^T B$.

Theorem 2.1 (Sou and Rantzer (2012)). *Given a full rank matrix $A \in \mathbb{R}^{n \times d_A}$, a matrix $B \in \mathbb{R}^{n \times d_B}$ and $s > 0$, there exists a rank k matrix $X \in \mathbb{R}^{d_A \times d_B}$ with $\|AX - B\|_2 < s$ if and only if*

$$\sigma_{k+1}(U^T B(s^2 I - \Delta)^{-1/2}) < 1$$

where U is a basis for the column span of A and N is a basis for the space orthogonal to . A feasible rank-k X is given by

$$(\Sigma V^T)^{-1}[U^T B(s^2 I - \Delta)^{-1/2}]_k \Delta(S)^{1/2}$$

where $U\Sigma V^T$ is the thin SVD of A.

The theorem also shows that $\sigma_{k+1}(U^T B(s^2 I - \Delta)^{-1/2})$ is a decreasing function of s. Therefore, using a binary search like procedure, we identify s such that $s \leqslant (1+\epsilon)$OPT in $\log(\|B\|_2/\text{OPT})/\epsilon$ iterations. Then the final solution is computed as:

$$\widehat{X} = (\Sigma V^T)^{-1}[U^T B(s^2 I - \Delta)^{-1/2}]_k (s^2 I - \Delta)^{1/2}.$$

An efficient implementation of Algorithm 1 exhibits the run time:

$$O(nd_B^2 + nd_A^2 + d_B^3 + d_A d_B^2 (\log(\|B\|_2/\text{OPT})/\epsilon), \tag{5}$$

which is very inefficient if n and d_B are very large.

In Section 3, we use oblivious subspace embeddings and then run Algorithm 1 on the sketched matrices and show that we can obtain a $(1+\epsilon)$ approximation to OPT in time

$$O(\text{nnz}(A) + \text{nnz}(B) + d_B^3 \text{poly}\,(1/\epsilon))$$

However, even this algorithm is slow in the case of very large d_B. In Section 4, we use fast sketching techniques that guarantee Approximate-Matrix-Matrix multiplication (AMM) and show that we can obtain an algorithm with running time

$$\begin{aligned} &O(\text{nnz}(B) + nd_A^2 + r^2 d_B + r^3) \\ &+ O(d_A^3 d_B + r d_A d_B \log(d_B)/\epsilon) \\ &+ \tilde{O}(k^2 d_B) \cdot \text{poly}\,(1/\epsilon) \end{aligned}$$

where $k = \max(d_A, \text{sr}(B))$ and $r = O(k/\epsilon^4)$. Here $\text{sr}(B)$ refers to the stable rank of B defined as

$$\text{sr}(B) := \frac{\|B\|_F^2}{\|B\|_2^2}.$$

It is easy to see that $\text{sr}(B) \leqslant \text{rank}(B)$ but it can be much smaller in more structured cases. For example, Gaussian kernel matrices have small stable rank even though their rank is very high (Wang et al., 2019). In the experiments section (Section 5.1), we discuss results on some very high dimensional datasets which have stable ranks at most 22. Our main result is Theorem 4.3.

3. Oblivious Subspace Embedding

Let $d = d_A + d_B$. There exists a distribution $\mathcal{D}$ over sketching matrices $R \in \mathbb{R}^{r \times n}$ where $r = O((d + \log(1/\delta))/\epsilon^2)$ such that with probability $1-\delta$, R is a $1 \pm \epsilon$ subspace embedding for the column span of A, B: that is for every u in the column span of $[A\ B]$,

$$\|Ru\|_2 \in (1 \pm \epsilon)\, \|u\|_2 .$$

Such a distribution can be obtained following (Cohen et al., 2016, Remark 3). For these matrices, $\widehat{A} := RA$ and $\widehat{B} := RB$ can be computed in time $O(\text{nnz}(A) + \text{nnz}(B) + d^3 \text{poly}\,(1/\epsilon))$. We then solve the following:

$$\min_{\text{rank}-k\ X} \|\widehat{A}X - \widehat{B}\|_2.$$

Let the minimum be attained for some rank$-k$ matrix $\widehat{X}$. It is easy to see that $\widehat{X}$ satisfies

$$\|\widehat{A}\widehat{X} - \widehat{B}\|_2 \leqslant (1 + O(\epsilon))\text{OPT}.$$

The overall computation time of $\widehat{X}$ is thus

$$O(\text{nnz}(A) + \text{nnz}(B) + d^3 \text{poly}\,(1/\epsilon))$$

which is faster than (5) if $n \ll d_B$. However this is still prohibitively slow if $d_B = \Omega(n)$ and n is very large.

Observe that although using $O(d/\epsilon^2)$ rows in the sketch ensures a full subspace embedding for vectors in the entire column span of $[A\ B]$, we really only want X^* to be preserved. Notice that Algorithm 1 only uses the following matrix-matrix products: $U \times B$ and $B \times B$. As shown by Cohen et al. (2016), matrix products between these matrices can be preserved using only $O(\max(\text{sr}(U), \text{sr}(B))/\epsilon^2)$ many rows. In the next section, we show that such an Approximate-Matrix-Matrix product (AMM) guarantee is sufficient to ensure $\epsilon\|B\|_2$ additive approximate solutions.

4. Approximate Matrix-Matrix Product

From Theorem 2.1, we have that problem (2) can be solved by using a binary search over s, where we search for the smallest value of s such that

$$\sigma_{k+1}(U^T B(s^2 I - \Delta)^{-1/2}) < 1.$$

This is equivalently the smallest s such that

$$\sigma_{k+1}(U^T B(s^2 I - \Delta)^{-1} B^T U) < 1.$$

Naively computing these singular values has an initial cost of $O(nd_B^2 + nd_A d_B + d_A d_B^2 + d_B^3)$ and $O(d_A d_B^2)$ for each value of s. Instead, we show we can compute approximate singular values efficiently using Approximate-Matrix-Matrix products. The procedure is summarized as Algorithm 2. In the rest of the section we prove theoretical guarantees on its approximation error.

To perform the binary search, we first need an upper bound on the value of OPT. Using Algorithm 3 we obtain such an upper bound between OPT and $O(\sqrt{d_B})\cdot$OPT. This is done by utilizing affine embeddings (Clarkson and Woodruff, 2017) to obtain a solution for the sketched version of the problem (in Fröbenius norm). The following lemma shows that the solution for the sketched problem gives an upper-bound approximation for OPT at the desired level.

Lemma 4.1 (Obtaining an $O(\sqrt{d_B})$ upper bound for OPT)**.** *Given matrices $A \in \mathbb{R}^{n \times d_A}$ and $B \in \mathbb{R}^{n \times d_B}$, with constant probability Algorithm 3 obtains s such that* $\text{OPT} \leqslant s \leqslant \sqrt{3d_B}\ \text{OPT}$ *in time $O(\text{nnz}(A) + \text{nnz}(B) + d_A^3 d_B + d_A^4)$ time.*

Proof. Consider any $1 \pm 1/2$ affine embedding S of A. A Countsketch matrix S with $O(d_A^2)$ rows works. We have for all X

$$\|SAX - SB\|_F^2 = \left(1 \pm \frac{1}{2}\right) \|AX - B\|_F^2$$

Let $U\Sigma V^T$ be the basis for colspan(A) and $U'\Sigma' V'^T$ be the SVD of colspan(SA). We have that

$$\|U[U^T B]_k - B\|_2 \leqslant \sqrt{2}\text{OPT}$$

Algorithm 2 Fast Low Rank Approximation

Require: $A \in \mathbb{R}^{n\times d_A}$, $B \in \mathbb{R}^{n\times d_B}$, $S \in \mathbb{R}^{r\times n}$

procedure FASTLRA(A, B, k, ϵ, S)
  $[U, \Sigma, V^T] \leftarrow$ SVD(A)
  Compute SA, SB
  Compute $U^T S^T SB$
  $M_1 \leftarrow (U^T S^T SB)(U^T S^T SB)^T$
  $[Q, D] \leftarrow$ FASTEIGEN(SU, SB)
  ▷ Computes Eigen Decomposition of $B^T S^T SB - B^T S^T SUU^T S^T SB$
  $M_2 \leftarrow U^T S^T SBQ$
  diag($\lambda_1, \ldots, \lambda_{O(r)}$) $\leftarrow D$
  $s \leftarrow$ APPROX(A, B)
  while true **do**
    Eig $\leftarrow$ diag($\frac{\lambda_1}{s^2(s^2-\lambda_1)}, \ldots, \frac{\lambda_{O(r)}}{s^2(s^2-\lambda_{O(r)})}$)
    $M \leftarrow (1/s^2)M_1 + M_2 \cdot$ Eig $\cdot M_2^T$
    if $\sigma_{k+1}(M) < 1$ **then**
      $s \leftarrow s/(1+\epsilon)$
    else
      $s \leftarrow \sqrt{s^2(1+\epsilon)^2 + O(\epsilon^2)\|B\|_2^2}$
      $D_{\text{half}} \leftarrow sI + Q((s^2 I - D)^{1/2} - s)Q^T$
      $D_{\text{neg-half}} \leftarrow \frac{I}{s} + Q((s^2 I - D)^{-1/2} - s^{-1/2})Q^T$
      $\widehat{X} \leftarrow (\Sigma V^T)^{-1}[U^T S^T SBD_{\text{neg-half}}]_k D_{\text{half}}$
      return $\widehat{X}$
    end if
  end while
end procedure

$$\implies \|A(\Sigma V^T)^{-1}[U^T B]_k - B\|_2 \leqslant \sqrt{2}\,\text{OPT}$$

and thus

$$\|U'[U'^T(SB)]_k - SB\|_F = \min_{\text{rank-}k\ X} \|SAX - SB\|_F.$$

This implies that

$$\begin{aligned}&\|SA(\Sigma' V'^T)^{-1}[U'^T(SB)]_k - SB\|_F \\ &\quad = \min_{\text{rank-}k\ X} \|SAX - SB\|_F. \end{aligned} \tag{6}$$

Finally, we have

$$\begin{aligned}&\|A(\Sigma' V'^T)^{-1}[U'^T(SB)]_k - B\|_F^2 \\ &\leqslant 2\|SA(\Sigma' V'^T)^{-1}[U'^T(SB)]_k - SB\|_F^2 \\ &\quad \text{(Since } S \text{ is Affine Embedding)} \\ &\leqslant 2\|SA(\Sigma V^T)^{-1}[U^T B]_k - SB\|_F^2 \\ &\quad \text{(Since } (\Sigma' V'^T)^{-1}[U'^T(SB)]_k \text{ is optimal for (6))} \\ &\leqslant 2 \cdot \frac{3}{2} \cdot \|A(\Sigma V^T)^{-1}[U^T B]_k - B\|_F^2 \\ &\quad \text{(Affine Embedding)} \\ &\leqslant 3d_B\|A(\Sigma V^T)^{-1}[U^T B]_k - B\|_2^2 \\ &\leqslant 6d_B\text{OPT}^2. \end{aligned}$$

Algorithm 3 $O(\sqrt{d_B})$ approximation to OPT

Require: $A \in \mathbb{R}^{n\times d_A}, B \in \mathbb{R}^{n\times d_B}$

▷ This procedure returns s such that OPT $\leqslant s \leqslant \sqrt{3d_B}$ OPT

procedure APPROX(A,B)
  $S \leftarrow (1 \pm 1/2)$-affine embedding for A
  ▷ Count-Sketch with $O(d_A^2)$ rows
  Compute SA, SB ▷ nnz(A) + nnz(B)
  $U'\Sigma' V'^T \leftarrow$ SVD(SA)
  $s \leftarrow \|U'[U'^T(SB)]_k - SB\|_F$
  return s.
end procedure

Hence, we obtain that $\|SA(\Sigma' V'^T)^{-1}[U'^T(SB)]_k - SB\|_F = \|U'[U'^T(SB)]_k - SB\|_F \leqslant \sqrt{3d_B}$OPT.

Here SA, SB can be computed in nnz(A) + nnz(B) time, U' can be computed in $O(d_A^4)$ time, $U'^T(SB)$ can be computed in $O(d_A^3 d_B)$ time and its SVD can also be computed in $O(d_A^2 d_B)$ time and the matrix $U'[U'^T(SB)]_k - SB$ and its Fröbenius norm can be computed in $O(d_A^3 d_B + d_A^2 d_B)$ time. Thus, we can obtain an s such that OPT $\leqslant s \leqslant \sqrt{3d_B}$ OPT in $O(\text{nnz}(A)+\text{nnz}(B)+d_A^3 d_B + d_A^4)$ time. □

Having obtained an upper bound for OPT, we can now perform a binary search. Notice that to perform steps 5,9 of Algorithm 1 efficiently, all we need to do is to approximate Δ and $U^T B$. We do so using sketching matrices that give us an AMM guarantee. The following lemma relates the AMM guarantee to the error obtained by Algorithm 2.

Lemma 4.2. *Given matrices $A \in \mathbb{R}^{n\times d_A}$ with U being an orthonormal basis for colspan(A), $B \in \mathbb{R}^{n\times d_B}$ and $S \in \mathbb{R}^{r\times n}$ satisfying*

$$\begin{aligned}\|U^T B - U^T S^T SB\|_2 &\leqslant \epsilon\|B\|_2 \\ \|B^T S^T SB - B^T B\|_2 &\leqslant \epsilon\|B\|_2^2,\end{aligned}$$

Algorithm 2 computes an s such that

$$s \leqslant \sqrt{\text{OPT}^2 + O(\epsilon)\|B\|_2^2}$$

and returns a rank-k matrix Y such that

$$\|AY - B\|_2 \leqslant \sqrt{\text{OPT}^2 + O(\epsilon)\|B\|_2^2}.$$

Lemma 4.2 is proved at the end of this section.

The next theorem gives a distribution of sketching matrices which satisfy the AMM property and gives our main result.

Theorem 4.3. *Given a full rank matrix $A \in \mathbb{R}^{n\times d_A}$, matrix $B \in \mathbb{R}^{n\times d_B}$, $1 \leqslant k \leqslant d_A$ and $0 \leqslant \epsilon \leqslant 1$, there exists a distribution of matrices $\mathcal{D}$ such that for $S \in$*

Algorithm 4 Fast Eigen Decomposition of $\Delta(S)$

Require: $SU \in \mathbb{R}^{r\times d_A}$, $SB \in \mathbb{R}^{r\times d_B}$
Ensure: Returns $[V'Q, D]$ such that $(V'Q)D(V'Q)^T$ is Eigen Decomposition of $\Delta(S)$
procedure FASTEIGEN(SU, SB)
  $R \leftarrow O(1)$, an Oblivious Subspace Embedding for r dimensional space ▷ $R \in \mathbb{R}^{O(r)\times d_B}$
  $M \leftarrow R(SB)^T(SB) - R(U^TS^TSB)^T(U^TS^TSB)$ ▷ $M \in \mathbb{R}^{O(r)\times d_B}$. Takes $O(r^2d_B + rd_Ad_B)$
  $[U', \Sigma', (V')^T] \leftarrow$ ThinSVD(M) ▷ $V'^T \in \mathbb{R}^{O(r)\times d_B}$. Takes $\tilde{O}(r^2d_B)$
  $M' \leftarrow (V')^T(SB)^T(SB)V' - (V')^T(U^TS^TSB)^T(U^TS^TSB)V'$ ▷ $O(r^2d_B + r^3)$
  $QDQ^T \leftarrow$ EigenDecomposition(M') ▷ $O(r^3)$
  Return $[V'Q, D]$
end procedure

$\mathbb{R}^{r\times n}$ *drawn from* $\mathcal{D}$ *where* $r = O(\max(\text{sr}(B), d_A)/\epsilon^4)$, FASTLRA$(A, B, k, \epsilon, S)$ *returns a solution* $\widehat{X}$ *satisfying*

$$\|A\widehat{X} - B\|_2 \leqslant \text{OPT} + \epsilon\|B\|_2. \tag{7}$$

with time complexity

$$\begin{aligned} &O(nnz(B) + nd_A^2 + r^2d_B + r^3) \\ &+ O(d_A^3d_B + rd_Ad_B\log(d_B)/\epsilon) \\ &+ \tilde{O}(k^2d_B)\cdot\text{poly}\,(1/\epsilon)\,. \end{aligned}$$

where $k = \max(d_A, \text{sr}(B))$ *and* $r = O(k/\epsilon^4)$ *with* $O(1)$ *probability.*

Proof. Let $U\Sigma V^T$ be the Singular Value Decomposition of A. For AMM guarantees, we use the construction from Remark 3 of Cohen et al. (2016) where S is given by

$$S = \Pi_1 \cdot \Pi_2 \cdot \Pi_3.$$

Here, Π_3 is a countsketch matrix, Π_2 is a Subsampled Randomized Hadamard Transform and Π_1 is a subgaussian matrtix with $O(\max(\text{sr}(B), d_A)/\epsilon^4)$ rows and S satisfies the AMM property i.e.,

$$\begin{aligned} \|U^TS^TSB - U^TB\|_2 &\leqslant O(\epsilon^2)\|B\|_2 \\ \|B^TS^TSB - B^TB\|_2 &\leqslant O(\epsilon^2)\|B\|_2^2 \end{aligned}$$

The matrix products SU, SB can be computed in $O(\text{nnz}(B) + nd_A) + \tilde{O}((k^3 + k^2(d_A + d_B))\cdot\text{poly}\,(1/\epsilon))$. Now, from Lemma 4.2, we obtain that FASTLRA (Algorithm 2) returns a rank-k matrix Y such that

$$\begin{aligned} \|AY - B\|_2 &\leqslant \sqrt{\text{OPT}^2 + \epsilon^2\|B\|_2^2} \\ &\leqslant \text{OPT} + \epsilon\|B\|_2 \end{aligned}$$

Also the total running time is

$$\begin{aligned} &O(\text{nnz}(B) + nd_A^2 + r^2d_B + r^3) \\ &+ O(d_A^3d_B + rd_Ad_B\log(d_B)/\epsilon) \\ &+ \tilde{O}(k^2d_B)\cdot\text{poly}\,(1/\epsilon) \end{aligned}$$

where $k = \max(d_A, \text{sr}(B))$ and $r = O(k/\epsilon^4)$. □

We prove the following Lemmas which bound norms of various matrices before going on to the proof of Lemma 4.2.

We will use the following matrix in the proof to follow:

$$\Delta(S) := B^TS^TSB - B^TS^TSUU^TS^TSB.$$

Lemma 4.4. *Given full rank matrix* $A \in \mathbb{R}^{n\times d_A}$*, and matrix* $B \in \mathbb{R}^{n\times d_B}$*, let* $U\Sigma V^T$ *be the SVD of* A*. If* $S \in \mathbb{R}^{r\times n}$ *satisfies*

$$\|U^TB - U^TS^TSB\|_2 \leqslant \epsilon\|B\|_2 \tag{8}$$

$$\|B^TS^TSB - B^TB\|_2 \leqslant \epsilon\|B\|_2^2, \tag{9}$$

then

$$\begin{aligned} \|B^TS^TSUU^TS^TSB - B^TUU^TB\|_2 &\leqslant 3\epsilon\|B\|_2^2, \\ \|\Delta(S) - \Delta\|_2 &\leqslant 4\epsilon\|B\|_2^2. \end{aligned}$$

Proof. Given that $\|U^TB - U^TS^TSB\|_2 \leqslant \epsilon\|B\|_2$, we have that

$$B^TS^TSB = U^TB + \Delta_1$$

with $\|\Delta_1\|_2 \leqslant \epsilon\|B\|_2$. Therefore,

$$\begin{aligned} &B^TS^TSUU^TS^TSB \\ &= (U^TS^TSB)^T(U^TS^TSB) \\ &= (U^TB + \Delta_1)^T(U^TB + \Delta_1) \\ &= B^TUU^TB + \Delta_1^TU^TB + U^TB\Delta_1 + \Delta_1^T\Delta_1 \end{aligned}$$

which implies that

$$\begin{aligned} &\|B^TS^TSUU^TS^TSB - B^TUU^TB\|_2 \\ &= \|\Delta_1^TU^TB + U^TB\Delta_1 + \Delta_1^T\Delta_1\|_2 \\ &\leqslant \|\Delta_1^TU^TB\|_2 + \|U^TB\Delta_1\|_2 + \|\Delta_1^T\Delta_1\|_2 \\ &\leqslant 3\epsilon\|B\|_2^2. \end{aligned} \tag{10}$$

Now,

$$\|\Delta(S) - \Delta\|_2$$

$$
\begin{aligned}
&= \|(B^TS^TSB - B^TS^TSUU^TS^TSB) \\
&\quad - (B^TB - B^TUU^TB)\|_2 \\
&= \|(B^TS^TSB - B^TB) - \\
&\quad (B^TS^TSUU^TS^TSB - B^TUU^TB)\|_2 \\
&\leqslant \|B^TS^TSB - B^TB\|_2 + \\
&\quad \|B^TS^TSUU^TS^TSB - B^TUU^TB\|_2 \\
&\leqslant 4\epsilon\|B\|_2^2 \qquad \text{(From (9) and (10))}. \qquad \square
\end{aligned}
$$

Lemma 4.5 (Upper bound on $\|X_U^*\|_2$)**.** *For X_U^* defined as the solution of*

$$\min_{\text{rank}-k\ X} \|UX - B\|_2$$

where U is an orthonormal matrix,

$$\|X_U^*\|_2 \leqslant 2\|B^*\|_2$$

Proof. It is important to note that the optimum value for the above problem and the optimum for (2) are equal given that U is an orthonormal basis for colspan(A). We have OPT $= \|UX_U^* - B\|_2 \leqslant \|U \cdot 0 - B\|_2 = \|B\|_2$. Therefore,

$$
\begin{aligned}
\|UX_U^*\|_2 &\leqslant \|UX_U^* - B\|_2 + \|B\|_2 \\
&= \text{OPT} + \|B\|_2 \\
&\leqslant 2\|B\|_2
\end{aligned}
$$

As U has orthonormal columns,

$$\|X_U^*\|_2 = \|UX_U^*\|_2 \leqslant 2\|B\|_2 \qquad \square$$

Proof of Lemma 4.2. Let $\Delta(S) = B^TS^TSB - B^TS^TSUU^TS^TSB$. We have $\sigma'_{k+1}(s) = \sigma_{k+1}([U^TS^TSB(s^2I - \Delta(S))^{-1/2}]) < 1$ iff there exists a rank-k matrix X' such that

$$\|X' - U^TS^TSB(s^2I - \Delta(S))^{-1/2}\|_2 < 1$$

Letting $X' = X(s^2I - \Delta(S))^{-1/2}$ for some rank-k X, we have

$$
\begin{aligned}
&\sigma_{k+1}([U^TS^TSB(s^2I - \Delta(S))^{-1/2}]) \leqslant 1 \\
&\iff \|(X - U^TS^TSB)(s^2I - \Delta(S))^{-1/2}\|_2 \leqslant 1 \\
&\iff (s^2I - \Delta(S))^{-1/2}(X - U^TS^TSB)^T \cdot \\
&\quad (X - U^TS^TSB)(s^2I - \Delta(S))^{-1/2} \preceq I \\
&\iff (X - U^TS^TSB)^T(X - U^TS^TSB) \preceq s^2I - \Delta(S) \\
&\iff (X - U^TB)^T(X - U^TB) + (B^TU - B^TS^TSU)X + \\
&\quad (B^TS^TSUU^TS^TSB - B^TUU^TB) + \\
&\quad X^T(U^TB - U^TS^TSB) \preceq (s^2I - \Delta(S)) \\
&\iff (X - U^TB)^TU^TU(X - U^TB) + \\
&\quad (B^TU - B^TS^TSU)X + X^T(U^TB - U^TS^TSB) + \\
&\quad (B^TS^TSUU^TS^TSB - B^TUU^TB) \preceq (s^2I - \Delta(S)) \\
&\iff (UX - B)^T(UX - B) - B^TNN^TB + \\
&\quad (B^TS^TSUU^TS^TSB - B^TUU^TB) + \\
&\quad (B^TU - B^TS^TSU)X + X^T(U^TB - U^TS^TSB) \\
&\quad \preceq s^2I - \Delta(S) \\
&\iff (UX - B)^T(UX - B) \preceq s^2I - \\
&\quad (\Delta(S) - \Delta) - (B^TU - B^TS^TSU)X \\
&\quad - X^T(U^TB - U^TS^TSB) \\
&\quad - (B^TS^TSUU^TS^TSB - B^TUU^TB).
\end{aligned}
$$

Also, there exists a rank-k matrix X_U^* such that

$$\|UX_U^* - B\|_2 = \text{OPT}$$

Therefore, if

$$
\begin{aligned}
&\lambda_{\min}(s^2I - (\Delta(S) - \Delta)) - (B^TU - B^TS^TSU)X_U^* \\
&- X_U^{*T}(U^TB - U^TS^TSB) \\
&- (B^TS^TSUU^TS^TSB - B^TUU^TB) \geqslant \text{OPT}^2
\end{aligned}
$$

then $\sigma'_{k+1}(s)$ computed is $\leqslant 1$.

We have the following from Lemmas 4.5, 4.4:

$$
\begin{aligned}
&\|\Delta(S) - \Delta\|_2 \leqslant 4\epsilon\|B\|_2^2 \\
&\|(B^TU - B^TS^TSU)X_U^*\|_2 \\
&\leqslant \|(B^TU - B^TS^TSU)\|_2\|X_U^*\|_2 \\
&\leqslant 2\epsilon\|B\|_2^2 \\
&\|X^{*T}_U(U^TB - U^TS^TSB)\|_2 \\
&\leqslant \|X_U^*\|_2\|(U^TB - U^TS^TSB)\|_2 \\
&\leqslant 2\epsilon\|B\|_2^2 \\
&\|B^TS^TSUU^TS^TSB - B^TUU^TB\|_2 \leqslant 3\epsilon\|B\|_2^2
\end{aligned}
$$

which implies that

$$
\begin{aligned}
&\lambda_{\min}(s^2I - (\Delta(S) - \Delta) \\
&- (B^TU - B^TS^TSU)X_U^* - X_U^{*T}(U^TB - U^TS^TSB) \\
&- (B^TS^TSUU^TS^TSB - B^TUU^TB)) \\
&\geqslant s^2I - O(\epsilon)\|B\|_2^2
\end{aligned}
$$

and therefore, for all

$$s \geqslant \sqrt{\text{OPT}^2 + O(\epsilon)\|B\|_2^2},$$

$$\sigma'_{k+1}(s) < 1.$$

Similarly, we have

$$\lambda_{\max}(s^2I - (\Delta(S) - \Delta) - (B^TU - B^TS^TSU)X$$

$$- X^T(U^TB - U^TS^TSB)$$
$$- (B^TS^TSUU^TS^TSB - B^TUU^TB)) \leqslant s^2I + O(\epsilon)\|B\|_2^2$$

and therefore, for all

$$s \leqslant \sqrt{\text{OPT}^2 - O(\epsilon)\|B\|_2^2}$$

$\sigma'_{k+1}(s) > 1$ Thus, the algorithm computes an s which lies in

$$\left[\sqrt{\text{OPT}^2 - O(\epsilon)\|B\|_2^2}, \sqrt{\text{OPT}^2 + O(\epsilon)\|B\|_2^2}\right].$$

Let t be such that $t^2 = s^2 + O(\epsilon)\|B\|_2^2$ and

$$t \in [\text{OPT}, \sqrt{\text{OPT}^2 + O(\epsilon^2)}]$$

As $t > s$, $\sigma'_{k+1}(t) \leqslant 1$. This implies from the above that there is a rank-k matrix X such that

$$\|UX - B\|_2^2 \leqslant t^2 + O(\epsilon)\|B\|_2^2$$

and as $t^2 \leqslant \text{OPT} + O(\epsilon)\|B\|_2^2$, we have

$$\|UX - B\|_2^2 \leqslant \text{OPT}^2 + O(\epsilon)\|B\|_2^2$$

and this matrix X is given by $[U^TS^TSB(t^2I - \Delta(S))^{-1/2}]_k(t^2I - \Delta(S))^{1/2}$ and the matrix $Y = (\Sigma V^T)^{-1}X$ satisfies

$$\begin{aligned}\|AY - B\|_2^2 &= \|U(\Sigma V^T)Y - B\|_2^2 \\ &= \|U(\Sigma V^T)(\Sigma V^T)^{-1}X - B\|_2^2 \\ &= \|UX - B\|_2^2 \\ &\leqslant \text{OPT}^2 + O(\epsilon)\|B\|_2^2\end{aligned}$$

□

5. Experiments

We now validate Algorithm 2 experimentally. First, we consider a simple synthetic example with small stable rank to perform a sanity check. Then, we run it on large-scale real-world datasets.

5.1. Synthetic matrix with small stable-rank

Consider a Gaussian kernel matrix $K \in \mathbb{R}^{n\times n}$ where $K_{ij} = \exp(-h\|x_i - x_j\|_2^2)$. In our experiments we set $h = 0.1$ and draw $x_i \overset{iid}{\sim} N(0, I_{10})$. Let $n = d_B = 100000$ and set $B = K$. A is set to a random sample of some $d_A = 20$ columns of B, and we set $k = 10$. In Figure 5.1 we plot the performance of Algorithm 2 on this dataset versus the running time for increasing sketching dimension. We also plot a horizontal line for the naive solution (Equation (3)) and the optimal solution computed using Algorithm 2 which corresponds to the optimal rank$-k$ solution.

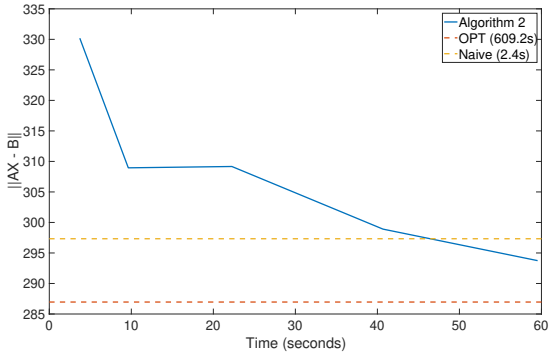


Figure 1: $\|AX - B\|_2$ vs. computation time where B is a Gaussian Kernel Matrix and A is a random subset of its columns. The time for OPT and naive are in the legend.

5.2. Real-world datasets

We consider four real-world datasets[1] to empirically validate Algorithm 2. We observe that Algorithm 2 gives fast running times with a few number of rows while giving a good approximation to OPT, while Algorithm 1 is very slow or impossible to run. Parameters of the datasets we consider are summarized in the following table:

Dataset	n	d_B	sr(B)	nnz(B)
ENRON	39861	28102	4.44	6.4×10^6
NIPS	1500	12419	5.25	1.9×10^6
KOS	3430	6906	7.81	4.6×10^5
NYTimes	3×10^5	1×10^5	21.75	1×10^8

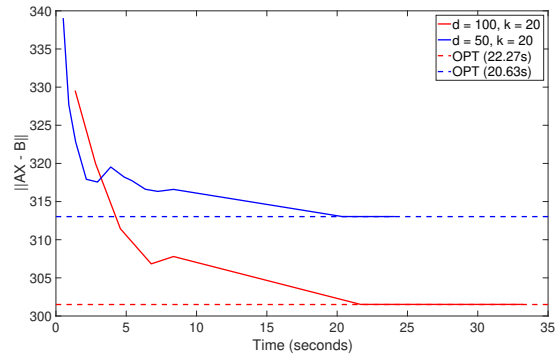
Table 1: Large scale UCI datasets

These datasets correspond to bag-of-words matrices where each data-point is a document and each feature represents the frequency of a certain corpus word in the document. These bag-of-words matrices form the large matrix B. To form A, we randomly select d_A columns of B for $d_A = 50, 100$. In our experiments, $k = 20$ and the number r of rows in the sketch are varied between a suitable range determined empirically. For every value of r, $\widehat{X}$ is computed using Algorithm 2, and the value of the objective $\|A\widehat{X} - B\|_2$ and the time taken for the computation are recorded. All experiments were run in MATLAB on a machine with 28 cores. Results are shown in Figure 2.

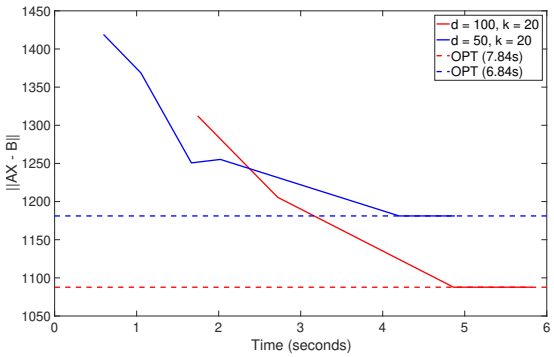
We make two key observations:

- For large datasets (ENRON and NYTIMES), Algorithm 2 obtains very good approximations to OPT with a very few number of rows in running times or-

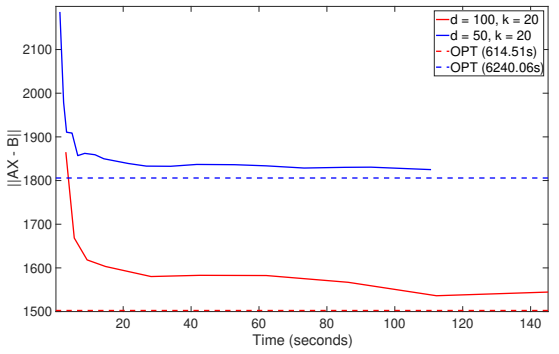
[1] obtained from https://archive.ics.uci.edu/ml/datasets/Bag+of+Words



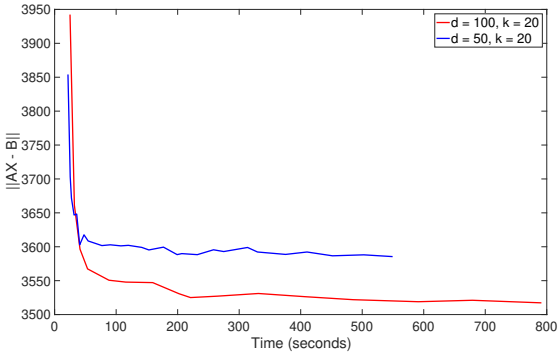
(a) KOS blog entries



(b) NIPS full papers



(c) ENRON emails



(d) NYTIMES news articles

Figure 2: $\|A\widehat{X} - B\|_2$ vs. computation time for four UCI bag-of-words datasets with $d_A = 50, 100$ and $k = 20$. Each observation of computation time corresponds to a different number of rows in the sketching matrix. The dashed horizontal lines refer to OPT and the computation time for OPT is given in the legend. We could not compute OPT for NYTIMES because it is too large.

ders of magnitude faster than the running time of Algorithm 1. In fact we could not compute OPT for NYTIMES in a reasonable amount of time.

- For small datasets (KOS and NIPS), where we expect OPT to be quite fast, Algorithm 2 still gives good approximations to OPT faster than the running time of OPT. This speedup could be useful even in these settings. For example, in column subset based low-rank reconstruction one may need to resample many column subsets to identify the best one.

References

Christos Boutsidis. *Topics in Matrix Sampling Algorithms*. PhD thesis, USA, 2011.

Christos Boutsidis and Dmitry Malioutov. Equity factor analysis via column subset selection. In *IEEE Global Conference on Signal and Information Processing, GlobalSIP 2013, Austin, TX, USA, December 3-5, 2013*, page 1131, 2013.

Christos Boutsidis and David P Woodruff. Optimal cur matrix decompositions. *SIAM Journal on Computing*, 46 (2):543–589, 2017.

Christos Boutsidis, Michael W. Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, page 968–977, USA, 2009. Society for Industrial and Applied Mathematics.

Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-optimal column-based matrix reconstruc-

tion. *SIAM Journal on Computing*, 43(2):687–717, 2014.

Flavio Chierichetti, Sreenivas Gollapudi, Ravi Kumar, Silvio Lattanzi, Rina Panigrahy, and David P. Woodruff. Algorithms for ℓ_p low-rank approximation. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 806–814, 2017.

Kenneth L. Clarkson and David P. Woodruff. Low-rank approximation and regression in input sparsity time. *J. ACM*, 63(6):54:1–54:45, January 2017. ISSN 0004-5411. doi: 10.1145/3019134. URL http://doi.acm.org/10.1145/3019134.

Michael B. Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, page 163–172, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450335362. doi: 10.1145/2746539.2746569. URL https://doi.org/10.1145/2746539.2746569.

Michael B. Cohen, Jelani Nelson, and David P. Woodruff. Optimal approximate matrix product in terms of stable rank. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 11:1–11:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi: 10.4230/LIPIcs.ICALP.2016.11. URL https://doi.org/10.4230/LIPIcs.ICALP.2016.11.

Venkatesan Guruswami and Ali Kemal Sinop. Optimal column-based low-rank matrix reconstruction. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, page 1207–1214, USA, 2012. Society for Industrial and Applied Mathematics.

Zhao Song, David P. Woodruff, and Peilin Zhong. Towards a zero-one law for column subset selection. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 6120–6131, 2019a.

Zhao Song, David P. Woodruff, and Peilin Zhong. Average case column subset selection for entrywise 120001_1-norm loss. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 10111–10121, 2019b.

Kin Cheong Sou and Anders Rantzer. On generalized matrix approximation problem in the spectral norm. *Linear Algebra and its Applications*, 436(7):2331–2341, 2012.

Ruoxi. Wang, Yingzhou. Li, Michael W. Mahoney, and Eric. Darve. Block basis factorization for scalable kernel evaluation. *SIAM Journal on Matrix Analysis and Applications*, 40(4):1497–1526, 2019. doi: 10.1137/18M1212586. URL https://doi.org/10.1137/18M1212586.