Provable Deterministic Leverage Score Sampling

Dimitris Papailiopoulos  
Electrical and Computer Engineering  
UT Texas  
dimitris@utexas.edu

Anastasios Kyrillidis  
School of Computer and Communication Sciences  
EPFL  
anastasios.kyrillidis@epfl.ch

Christos Boutsidis  
Yahoo! Labs  
New York, NY  
boutsidis@yahoo-inc.com

ABSTRACT

We explain theoretically a curious empirical phenomenon: “Approximating a matrix by deterministically selecting a subset of its columns with the corresponding largest leverage scores results in a good low-rank matrix surrogate”. In this work, we provide a novel theoretical analysis of deterministic leverage score sampling. We show that such sampling can be provably as accurate as its randomized counterparts, if the leverage scores follow a moderately steep power-law decay. We support this power-law assumption by providing empirical evidence that such decay laws are abundant in real-world data sets. We then demonstrate empirically the performance of deterministic leverage score sampling, which many times matches or outperforms the state-of-the-art techniques.

Categories and Subject Descriptors

G.1.3 [Mathematics of Computing]: Numerical Analysis
G.1.2 [Numerical Linear Algebra]: Miscellaneous
G.4 [Mathematics of Computing]: Mathematical Software
H.2.5 [Information Systems]: Online Information Services
I.2.8 [Artificial Intelligence]: Machine Learning
J.5.6 [Computing Methodologies]: Mathematics of Computing
K.6.1 [Computing Milieux]: Probability and Statistics

Keywords

Subset selection; low-rank matrix approximation; leverage scores; deterministic sampling; power law distributions

1. INTRODUCTION

Recently, there has been a lot of interest on selecting the “best” or “more representative” columns from a data matrix [13, 26]. Qualitatively, these columns reveal the most important information hidden in the underlying matrix structure. This is similar to what principal components carry, as extracted via Principal Components Analysis (PCA) [23]. In sharp contrast to PCA, using actual columns of the data matrix to form a low-rank surrogate offers interpretability, making it more attractive to practitioners and data analysts [33, 5, 34, 26].

To make the discussion precise and to rigorously characterize the “best” columns of a matrix, let us introduce the following Column Subset Selection Problem (CSSP).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

KDD’14, August 24–27, 2014, New York, NY, USA.
Copyright © 2014 by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2956-9/14/08 ...$15.00.
http://dx.doi.org/10.1145/2623330.2623698.

Column Subset Selection Problem. Let \( A \in \mathbb{R}^{m \times n} \) and let \( c < n \) be a sampling parameter. Find \( c \) columns of \( A \) — denoted as \( C \in \mathbb{R}^{m \times c} \) — that minimize

\[
\| A - CC^\dagger \|_F 	ext{ or } \| A - CC^\dagger \|_2,
\]

where \( C^\dagger \) denotes the Moore-Penrose pseudo-inverse.

State of the art algorithms for the CSSP utilize both deterministic and randomized techniques; we discuss related work in Section 5. Here, we describe two algorithms from prior literature that suffice to highlight our contributions.

A central part of our discussion will involve the leverage scores of a matrix \( A \), which we define below.

Definition 1. [Leverage scores] Let \( V_k \in \mathbb{R}^{m \times k} \) contain the top \( k \) right singular vectors of a \( m \times n \) matrix \( A \) with rank \( p = \text{rank}(A) \geq k \). Then, the (rank-\( k \)) leverage score of the \( i \)-th column of \( A \) is defined as

\[
\ell_i^{(k)} = \| [V_k]_{i, \cdot} \|_2^2, \quad i = 1, 2, \ldots, n.
\]

Here, \( [V_k]_{i, \cdot} \) denotes the \( i \)-th row of \( V_k \).

One of the first algorithms for column subset selection dates back to 1972: in [21], Jolliffe proposes a deterministic sampling of the columns of \( A \) that correspond to the largest leverage scores \( \ell_i^{(k)} \), for some \( k < \text{rank}(A) \). Although this simple approach has been extremely successful in practice [21, 22, 29, 8], to the best of our knowledge, there has been no theoretical explanation why the approximation errors \( \| A - CC^\dagger \|_F \) and \( \| A - CC^\dagger \|_2 \) should be small.

One way to circumvent the lack of a theoretical analysis for the above deterministic algorithm is by utilizing randomization. Drineas et al. [13] proposed the following approach: for a target rank \( k < \text{rank}(A) \), define a probability distribution over the columns of \( A \), i.e., the \( i \)-th column is associated with a probability \( p_i = \ell_i^{(k)}/k \); observe that \( \sum p_i = 1 \), since \( \sum \ell_i^{(k)} = \| [V_k] \|_F^2 = k \). Then, in a independent and identically distributed samples, sample with replacement \( c \) columns from \( A \), with probabilities given by \( p_i \). Drineas et al. [13], using results in [30], show that this random subset of columns \( C \in \mathbb{R}^{m \times c} \) approximates \( A \), with constant probability, within relative error:

\[
\| A - CC^\dagger \|_F \leq (1 + \varepsilon) \| A - A_k \|_F,
\]

where the number of sampled columns is \( c = O(k \log k / \varepsilon^2) \), for some \( 0 < \varepsilon < 1 \). Here, \( A_k \in \mathbb{R}^{m \times n} \) is the best rank-\( k \) matrix obtained via the SVD.

There are two important remarks that need to be made: (i) the randomized algorithm in [13] yields a matrix estimate that is “near optimal”, i.e., has error close to that of the best rank-\( k \) approximation; and (ii) the above random
sampling algorithm is a straightforward randomized version of the deterministic algorithm of Jolliffe [21].

From a practical perspective, the deterministic algorithm of Jolliffe [21] is extremely simple to implement, and is computationally efficient. Unfortunately, as of now, it did not admit provable performance guarantees. An important open question [13, 29, 8] is: Can one simply keep the columns having the largest leverage scores, as suggested in [21], and still have a provably tight approximation?

1.1 Contributions

In this work, we establish a new theoretical analysis for the deterministic leverage score sampling algorithm of Jolliffe [21]. We show that if the leverage scores \( \ell_{i}^{(k)} \) follow a sufficiently steep power-law decay, then this deterministic algorithm has provably similar or better performance to its randomized counterparts (see Theorems 2 and 3 in Section 2). This means that under the power-law decay assumption, deterministic leverage score sampling algorithm obtains near optimal low-rank approximations and it can be as accurate as the “best” algorithms in the literature [4, 18].

From an applications point of view, we support the power law decay assumption of our theoretical analysis by demonstrating that several real-world data-sets have leverage scores following such decays. We further run several experiments on synthetic and real data sets, and compare deterministic leverage score sampling with the state of the art algorithms for the CSSP. In most experiments, the deterministic algorithm obtains tight low-rank approximations, and is shown to perform similar, if not better, than the state of the art.

1.2 Notation

We use \( A, B, \ldots \) to denote matrices and \( a, b, \ldots \) to denote column vectors. \( I_{n} \) is the \( n \times n \) identity matrix; \( 0_{m \times n} \) is the \( m \times n \) matrix of zeros; \( e_{i} \) belongs to the standard basis (whose dimensionality will be clear from the context). Let \( C = [a_{1}, \ldots, a_{c}] \in \mathbb{R}^{m \times c} \) contain \( c \) columns of \( A \). We can equivalently write \( C = AS \), where the sampling matrix is \( S = [e_{1}, \ldots, e_{c}] \in \mathbb{R}^{n \times c} \). We define the Frobenius and the spectral norm of a matrix as \( \|A\|_{F}^{2} = \sum_{i,j} A_{i,j}^{2} \) and \( \|A\|_{2} = \max_{x : \|x\|_{2} = 1} \|Ax\|_{2} \), respectively.

2. DETERMINISTIC COLUMN SAMPLING

In this section, we describe the details of the deterministic leverage score sampling algorithm. In Section 3, we state our approximation guarantees. In the remaining of the text, given a matrix \( A \) of rank \( \rho \), we assume that the “target rank” is \( k < \rho \). This means that we wish to approximate \( A \) using a subset of \( c \geq k \) of its columns, such that the resulting matrix has an error close to that of the best rank-\( k \) approximation.

The deterministic leverage score sampling algorithm can be summarized in the following three steps:

Step 1: Obtain \( V_{k} \), the top-\( k \) right singular vectors of \( A \). This can be carried by simply computing the singular value decomposition (SVD) of \( A \) in \( O(\min\{m, n\})m \) time.

Step 2: Calculate the leverage scores \( \ell_{i}^{(k)} \). For simplicity, we assume that \( \ell_{i}^{(k)} \) are sorted in descending order, hence the columns of \( A \) have the same sorting as well.

Step 3: Output the \( c \) columns of \( A \) that correspond to the largest \( c \) leverage scores \( \ell_{i}^{(k)} \) such that their sum \( \sum_{i=1}^{c} \ell_{i}^{(k)} \) is more than \( \theta \). This ensures that the selected columns have accumulated “energy” at least \( \theta \). In this step, we have to carefully pick \( \theta \), our stopping threshold. This parameter essentially controls the quality of the approximation.

In Section 7, we provide some guidance on how the stopping parameter \( \theta \) should be chosen. Note that, if \( \theta \) is such that \( c < k \), we force \( c = k \). This is a necessary step that prevents the error in the approximation from “blowing up” (see Section 7). The exact steps are given in Algorithm 1.

Algorithm 1 LeverageScoresSampler(A, k, \( \theta \))

1. Compute \( V_{k} \in \mathbb{R}^{n \times k} \) (top \( k \) right singular vectors of \( A \))
2. for \( i = 1, 2, \ldots, n \):
   1. \( \ell_{i}^{(k)} = \|V_{k}i\|_{2}^{2} \)
end for

Without loss of generality, let \( \ell_{1}^{(k)} \leq \cdots \leq \ell_{k}^{(k)} \) be sorted:

\[ \ell_{1}^{(k)} \geq \cdots \geq \ell_{i}^{(k)} \geq \ell_{i+1}^{(k)} \geq \cdots \geq \ell_{n}^{(k)}. \]

3. Find index \( c \in \{1, \ldots, n\} \) such that:

\[ c = \arg\min_{c} \left( \sum_{i=1}^{c} \ell_{i}^{(k)} > \theta \right). \]

4. If \( c < k \), set \( c = k \).

Output: \( S \in \mathbb{R}^{n \times c} \) s.t. \( AS \) has the top \( c \) columns of \( A \).

Algorithm 1 requires \( O(\min\{m, n\})mn \) arithmetic operations. In the full version of this paper [28], we discuss modifications that improve the running time.

3. APPROXIMATION GUARANTEES

Our main technical innovation is a bound on the approximation error of Algorithm 1 in regard to the CSSP; the proof of the following theorem can be found in Section 6.

Theorem 2. Let \( \theta = k - \varepsilon \), for some \( \varepsilon \in (0, 1) \), and let \( S \in \mathbb{R}^{n \times c} \) be the output sampling matrix of Algorithm 1. Then, for \( C = AS \) and \( \xi = (2, F) \), we have \( \|A - CC^{\dagger}A\|_{F}^{2} < (1 - \varepsilon)^{-1} \cdot \|A - A_{k}\|_{F}^{2} \).

Choosing \( \varepsilon \in (0, 1/2) \) implies \( (1 - \varepsilon)^{-1} \leq 1 + 2\varepsilon \) and, hence, we have a relative-error approximation:

\[ \|A - CC^{\dagger}A\|_{F}^{2} < (1 + 2\varepsilon) \cdot \|A - A_{k}\|_{F}^{2}. \]

3.1 Bounding the number of sampled columns

Algorithm 1 extracts at least \( c \geq k \) columns of \( A \). However, an upper bound on the number of output columns \( c \) is not immediate. We study such upper bounds below.

From Theorem 2, it is clear that the stopping parameter \( \theta = k - \varepsilon \) directly controls the number of output columns \( c \). This number, extracted for a specific error requirement \( \varepsilon \), depends on the decay of the leverage scores. For example, if the leverage scores decay fast, then we intuitively expect \( \sum_{i=1}^{c} \ell_{i}^{(k)} = k - \varepsilon \) to be achieved for a “small” \( c \).
Let us for example consider a case where the leverage scores follow an extremely fast decay:

\[ \ell_1^{(k)} = k - 2k \cdot \varepsilon, \]
\[ \ell_2^{(k)} = \ldots = \ell_k^{(k)} = \varepsilon, \]
\[ \ell_{2k-1}^{(k)} = \ldots = \ell_n^{(k)} = \frac{\varepsilon}{n - 2k}. \]

Then, in this case, \( \sum_{i=1}^{2k} \ell_i^{(k)} = k - \varepsilon \), and Algorithm 1 outputs the \( c = 2k \) columns of \( A \) that correspond to the 2k largest leverage scores. Due to Theorem 2, this subset of columns \( C \in \mathbb{R}^{n \times 2k} \) comes with the following guarantee:

\[ \|A - CC^T A\|_F^2 < \frac{1}{1 - \varepsilon} \|A - A_k\|_F^2. \]

Hence, from the above example, we expect that, when the leverage scores decay fast, a small number of columns of \( A \) will offer a good approximation of the form \( CC^T A \).

However, in the worst case Algorithm 1 can output a number of columns \( c \) that can be as large as \( \Omega(n) \). To highlight this subtle point, consider the case where the leverage scores are uniform \( \ell_i^{(k)} = \frac{1}{n} \). Then, one can easily observe that if we want to achieve an error of \( \varepsilon \) according to Theorem 2, we have to set \( \theta = k - \varepsilon \). This directly implies that we need to sample \( c > (n/k) \theta \) columns. Hence, if \( \varepsilon = o(1) \), then

\[ c \geq (n/k) \theta = (1 - \varepsilon/k)n = \Omega(n). \]

Hence, for \( \varepsilon \to 0 \) we have \( c \to n \), which makes the result of Theorem 2 trivial.

We argued above that when the leverage scores decay is “fast” then a good approximation is to be expected with a “small” \( c \). We make this intuition precise below \(^2\). The next theorem considers the case where the leverage scores follow a power-law decay; the proof can be found in Section 6.

**Theorem 3.** Let the leverage scores follow a power-law decay with exponent \( c_h = 1 + \eta \), for \( \eta > 0 \):

\[ \ell_i^{(k)} = \frac{1}{i^{\eta}}. \]

Then, if we set the stopping parameter to \( \theta = k - \varepsilon \), for some \( \varepsilon \) with \( 0 < \varepsilon < 1 \), the number of sampled columns in \( C = AS \) that Algorithm 1 outputs is

\[ c = \max \left\{ \left( \frac{2k}{\varepsilon} \right)^{\frac{1}{\eta}}, 1, \left( \frac{2k}{\varepsilon \cdot \eta} \right)^{\frac{1}{\eta}}, 1, k \right\}, \]

and \( C \) achieves the following approximation error

\[ \|A - CC^T A\|_F^2 < \frac{1}{1 - \varepsilon} \|A - A_k\|_F^2, \text{ for } \xi \in \{2, F\}. \]

### 3.2 Theoretical comparison to state of the art

We compare the number of chosen columns \( c \) in Algorithm 1 to the number of columns chosen in the randomized leverage scores sampling case \([13]\). The algorithm of \([13]\) requires

\[ c = O(k \log k / \varepsilon^2) \]

\(^2\)We chose to analyze in detail the case where the leverage scores follow a power law decay; other models for the leverage scores, example, exponential decay, are also interesting, and will be the subject of the full version of this work.

columns for a relative-error bound with respect to the Frobenius error in the CSSP:

\[ \|A - CC^T A\|_F^2 \leq (1 + \varepsilon) \|A - A_k\|_F^2. \]

Assuming leverage scores follow a power-law decay, Algorithm 1 requires fewer columns for the same \( \varepsilon \) when:

\[ \max \left\{ \left( \frac{2k}{\varepsilon} \right)^{\frac{1}{\eta}}, \left( \frac{2k}{\varepsilon \cdot \eta} \right)^{\frac{1}{\eta}}, 1 \right\} < C \cdot \frac{k \log k}{\varepsilon^2}, \]

where \( C \) is an absolute constant. Hence, under the power law decay, Algorithm 1 offers provably a matrix approximation similar or better than \([13]\).

Let us now compare the performance of Algorithm 1 with \([4]\), which are the current state of the art for the CSSP. Theorem 1.5 in \([4]\) provides a randomized algorithm which selects

\[ c = \frac{2k}{\varepsilon} (1 + o(1)) \]

columns in \( C \) such that

\[ \|A - CC^T A\|_F^2 < (1 + \varepsilon) \cdot \|A - A_k\|_F^2 \]

holds in expectation. This result is in fact optimal, up to a constant 2, since there is a lower bound indicating that such a relative error approximation is not possible unless \( c = k / \varepsilon \) (see Section 9.2 in \([4]\)). The approximation bound of Algorithm 1 is indeed better than the upper/lower bounds in \([4]\) for any \( \eta > 1 \). We should note here that the lower bound in \([4]\) is for general matrices; however, the upper bound of Theorem 3 is applied to a specific class of matrices whose leverage scores follow a power law decay.

Next, we compare the spectral norm bound of Theorem 3 to the spectral norm bound of Theorem 1.1 in \([4]\), which indicates that there exists a deterministic algorithm selecting \( c > k \) columns with error

\[ \|A - CC^T A\|_2^2 < O(n/c) \cdot \|A - A_k\|_2^2. \]

This upper bound is also tight, up to constants, since \([4]\) provides a matching lower bound. Notice that a relative error upper bound requires \( c = \Omega(n/(1 + \varepsilon)) \) in the general case. However, under the power law assumption in Theorem 3, we provide such a relative error bound with asymptotically fewer columns. To our best knowledge, fixing \( \eta \) to a constant, this is the first relative-error bound for the spectral norm version of the CSSP with \( c = \text{poly}(k, 1/\varepsilon) \) columns.

### 4. EXPERIMENTS

In this section, we first provide evidence that power law decays are prevalent in real-world data sets. Then, we investigate the empirical performance of Algorithm 1 on real and synthetic data sets.

Our experiments are not meant to be exhaustive; however, they provide clear evidence that: (i) the leverage scores of real world matrices indeed follow “sharp” power law decays; and (ii) deterministic leverage score sampling in such matrices is particularly effective.

#### 4.1 Power-law decays in real data sets

We demonstrate the leverage score decay behavior of many real-world data sets. These range from social networks and product co-purchasing matrices to document-term bag-of-words data sets, citation networks, and medical imaging.
Table 1: Summary of datasets used in the experiments of Subsection 4.1

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$m \times n$</th>
<th>Description</th>
<th>Dataset</th>
<th>$m \times n$</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon</td>
<td>262111 $\times$ 262111</td>
<td>Purchase netw. [25]</td>
<td>Citeseer</td>
<td>723131 $\times$ 723131</td>
<td>Citation netw. [24]</td>
</tr>
<tr>
<td>Slashdot</td>
<td>875713 $\times$ 875713</td>
<td>Social netw. [24]</td>
<td>LJournal</td>
<td>4847571 $\times$ 4847571</td>
<td>Social netw. [25]</td>
</tr>
<tr>
<td>Cora</td>
<td>1696415 $\times$ 1696415</td>
<td>System netw. [24]</td>
<td>CT slices</td>
<td>386 $\times$ 53500</td>
<td>CT images [2]</td>
</tr>
<tr>
<td>Youtube</td>
<td>1134890 $\times$ 1134890</td>
<td>Citation netw. [24]</td>
<td>Writer</td>
<td>81067 $\times$ 42714</td>
<td>Writers/Works [24]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Video netw. [25]</td>
<td>YT groups</td>
<td>94238 $\times$ 30087</td>
<td>Users/Groups [24]</td>
</tr>
</tbody>
</table>

![Figure 1](image.png)

Figure 1: We plot the top 1,000 leverage scores for 16 different data sets, obtained through $V_k$ for $k = 10$. The plots are in logarithmic scale. For each data-set, we plot a fitting power-law curve $\beta \cdot x^{-\alpha_k}$. The exponent is listed on each figure as $\alpha_{10}$. The leverage scores are plotted with a red marker, and the fitted curves are denoted with a solid blue line. We observe that the power law fit offers a good approximation of the true leverage scores. We further observe that many data sets exhibit sharp decays ($\alpha_k > 1$), while only a few have leverage scores that decay slowly ($\alpha_k < 1$).

A fitting power-law curve of the form $\beta \cdot x^{-\alpha_k}$, where $\alpha_k$ is the exponent of interest. We can see from the plots that a power law indeed seems to closely match the behavior of the top leverage scores. What is more interesting is that for many of our data sets we observe a decay exponent of $\alpha_k > 1$: this is the regime where deterministic sampling is expected to perform well. It of the speed of decay. This is to be expected, since for the case of $k = \text{rank}(A)$ all leverage scores are equal.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$m \times n$</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gnutella</td>
<td>106218 $\times$ 106218</td>
<td>Social netw. [36]</td>
</tr>
<tr>
<td>Amazon</td>
<td>262111 $\times$ 262111</td>
<td>Purchase netw. [25]</td>
</tr>
<tr>
<td>Youtube</td>
<td>1134890 $\times$ 1134890</td>
<td>Video netw. [25]</td>
</tr>
</tbody>
</table>

samples. Their dimensions vary from thousands to millions of variables. The data-set description is given in Table 1.

In Figure 1, we plot the top 1,000 leverage scores extracted from the matrix of the right top-$k$ singular vectors $V_k$. In all cases we set $k = 10$.

We performed various experiments for larger $k$, e.g., $k = 30$ or $k = 100$ (not shown due to space limitations). We found that as we move towards higher $k$, we observe a “smoothing” effect, where the leverage scores are more uniformly distributed. This is to be expected, since for the case of $k = \text{rank}(A)$ all leverage scores are equal.
seems that these sharp decays are naturally present in many real-world data sets.

We would like to note that as we move to smaller scores (i.e., after the 10,000-th score), we empirically observe that the leverage scores tail usually decays much faster than a power law. This only helps the bound of Theorem 2.

4.2 Synthetic Experiments

In this subsection, we are interested in understanding the performance of Algorithm 1 on matrices with (i) uniform and (ii) power-law decaying leverage scores.

To generate matrices with a prescribed leverage score distribution, we use the implementation of [20]. Let $V_k \in \mathbb{R}^{n \times k}$ denote the matrix we want to construct, for some $k < n$. Then, [20] provides algorithms to generate tall-and-skinny orthonormal matrices with specified row norms (i.e., leverage scores). Given the $V_k$ that is the output of the matrix generation algorithm in [20], we use a basis completion algorithm to find the perpendicular matrix $V'_k \in \mathbb{R}^{n \times (n-k)}$ such that $V_k V'_k = 0_{k \times (n-k)}$. Then, we create an $n \times n$ matrix $V = [V_k V'_k]$ where the first $k$ columns of $V$ are the columns of $V_k$ and the rest $n - k$ columns are the columns of $V'_k$; hence, $V$ is a full orthonormal basis. Finally we generate $A \in \mathbb{R}^{m \times n}$ as $A = U \Sigma V^T$; where $U \in \mathbb{R}^{m \times n}$ is any orthonormal matrix, and $\Sigma \in \mathbb{R}^{n \times n}$ any diagonal matrix with $\min\{m, n\}$ positive entries along the main diagonal. Therefore, $A = U \Sigma V^T$ is the full SVD of $A$ with leverage scores equal to the squared $\ell_2$-norm of the rows of $V_k$. In our experiments, we pick $U$ as an orthonormal basis for an $m \times m$ matrix where each entry is chosen i.i.d. from the Gaussian distribution. Also, $\Sigma$ contains $\min\{m, n\}$ positive entries (sorted) along its main diagonal, where each entry was chosen i.i.d. from the Gaussian distribution.

4.2.1 Nearly-uniform scores

We set the number of rows to $m = 200$ and the number of columns to $n = 1000$ and construct $A = U \Sigma V^T \in \mathbb{R}^{m \times n}$ as described above. The row norms of $V_k$ are chosen as follows: First, all row norms are chosen equal to $k/n$, for some fixed $k$. Then, we introduce a small perturbation to avoid singularities: for every other pair of rows we add $\beta \in N(0, 1/100)$ to a row norm and subtract the same $\beta$ from the row norm - hence the sum of $\ell^2(k)$ equals to $k$.

We set $k$ to take the values $\{5, 10, 50, 100\}$ and for each $k$ we choose: $c = \{1, 2, \ldots, 1000\}$. We present our findings in Figure 2, where we plot the relative error achieved $\|A - C C^T A\|_2^2 / \|A - A_k\|_2^2$, where the $n \times m$ matrix $C$ contains the first $c$ columns of $A$ that correspond to the $k$ largest leverage scores of $V_k$, as sampled by Algorithm 1. Then, the leftmost vertical cyan line corresponds to the point where $k = c$. When $c < k$ the output error can be large; this justifies why we enforce the algorithm to output $c \geq k$ columns. The rightmost vertical magenta line indicates the point where the $c$ sampled columns achieve an error of $\|A - A_k\|_2^2$, where $A_k$ is the best rank-$k$ approximation.

In the plots of Figure 2, we see that as we move to larger values of $k$, if we wish to achieve an error of $\|A - C C^T A\|_2^2 \approx \|A - A_k\|_2^2$, then we need to keep in $C$, approximately almost half the columns of $A$. This agrees with the uniform scores example that we showed earlier in Subsection 3.1. However, we observe that Algorithm 1 can obtain a moderately small relative error, with significantly smaller $c$. See for example the case where $k = 100$: then, $c \approx 200$ sampled columns suffice for a relative error approximately equal to 2, i.e., $\|A - C C^T A\|_2^2 \approx 2 \cdot \|A - A_k\|_2^2$. This indicates that our analysis could be loose in the general case.

4.2.2 Power-law decay

In this case, our synthetic eigenvector matrices $V_k$ have leverage scores that follow a power law decay. We choose two power-law exponents: $\alpha_k = 0.5$ and $\alpha_k = 1.5$. Observe that the latter complies with Theorem 3, that predicts the near optimality of leverage score sampling under such decay.

In the first row of Figure 3, we plot the relative error vs. the number of output columns $c$ of Algorithm 1 for $\alpha_k = 0.5$. Then, in the second row of Figure 3, we plot the relative error vs. the number of output columns $c$ of Algorithm 1 for $\alpha_k = 1.5$. The blue line represents the relative error in terms of spectral norm. We can see that the performance of Algorithm 1 in the case of the fast decay is surprising: $c \approx 1.5-k$ suffices for an approximation as good as that of the best rank-$k$ approximation. This confirms the approximation performance in Theorem 3.

4.3 Comparison with other techniques

We will now compare the proposed algorithm to state of the art approaches for the CSSP, both for $\xi = 2$ and $\xi = F$. We report results for the errors $\|A - C C^T A\|_F^2 / \|A - A_k\|_F^2$. A comparison of the running time complexity of those algorithms is out of the scope of our experiments.

Table 2 contains a brief description of the datasets used in our experiments. We employ the datasets used in [15], which
sampling method with replacement, presented in [13].

Ac-formation vector such that $R$ where $\begin{bmatrix} c \end{bmatrix}^T$ built-in Matlab function, where one can select $c \leq k$ columns of $A$ in $C \in \mathbb{R}^{m \times c}$ with

$$\|A - CC^T A\|_F \leq \left(1 + \left(1 - \sqrt{k/c}\right)^{-1}\right) \|A - A_k\|_F.$$ 

Second, in [16], the authors present a deterministic pivoted QR algorithm such that:

$$\|A - CC^T A\|_2 \leq \left(1 + \sqrt{n - k} \cdot 2^k\right) \|A - A_k\|_2.$$ 

This bound was proved in [17]. In our tests, we use the $\text{qr}(\cdot)$ built-in Matlab function, where one can select $c \leq k$ columns of $A$ as:

$$Q, R, \pi = \text{qr}(A, 0); \quad C = A_{\cdot \pi, 1:c},$$

where $A = QR, Q \in \mathbb{R}^{m \times n}$ contains orthonormal columns, $R \in \mathbb{R}^{n \times n}$ is upper triangular, and $\pi$ is a permutation information vector such that $A_{\cdot \pi} = QR$.

Third, we also consider the randomized leverage-scores sampling method with replacement, presented in [13]. According to this work and given $A, k < \text{rank}(A)$, and $c = \Omega(k \log k)$, the bound provided by the algorithm is

$$\|A - CC^T A\|_F \leq \left(1 + O\left(\sqrt{k \log k/c}\right)\right) \|A - A_k\|_F,$$

which holds only with constant probability. In our experiments, we use the software tool developed in [20] for the randomized sampling step.

We use our own Matlab implementation for each of these approaches. For [13], we execute 10 repetitions and report the one that minimizes the approximation error.

4.3.2 Performance Results

Table 3 contains a subset of our results; a complete set is reserved for an extended version of this work. We observe that the performance of Algorithm 1 is particularly appealing: It is almost as good as randomized leverage scores sampling in almost all cases - when randomized sampling is better, the difference is on the first or second decimal digit.

Figure 4 shows the leverage scores for the three matrices used in our experiments. Although the decay for the first data sets does not fit a “sharp” power law as it is required in Theorem 3, the performance of the algorithm is still competitive in practice. Interestingly, we do observe good performance for the third data set (Enron). For this case, the power law decay fits the decay profile needed to establish the near optimality of Algorithm 1.

5. RELATED WORK

One of the first deterministic results regarding the CSSP goes back to the seminal work of Gene Golub on pivoted QR factorizations [16]. Similar algorithms have been developed in [16, 19, 10, 11, 17, 35, 32, 3, 27]; see also [6] for a recent survey. The best of these algorithms is the so-called Strong Rank-revealing QR (Strong RRQR) algorithm in [17]:

![Figure 3: Power-law decaying leverage scores case: We choose two power-law exponents: $\alpha_k = 0.5$ and $\alpha_k = 1.5$.
In the first row we plot the relative error of Algorithm 1 vs. $c$ for the first decay profile, and the second row is the error performance of Algorithm 1 for the second, sharpest decay profile. The vertical cyan line corresponds to the point where $k = c$, and the vertical magenta line indicates the point where the $c$ sampled columns offer a better approximation compared to the best rank-$k$ matrix $A_k$.](image)
Table 3: We present the performance of Algorithm 1 as compared to the state of the art in CSSP. We run experiments on 3 data sets described in the above table, for various values of \(k\) and \(c\). The performance of Algorithm 1, especially in terms of the Frobenius norm error, is very close to optimal, while at the same time similar, if not better, to the performance of the more sophisticated algorithms of the comparison.

Given \(A\), \(c = k\), and constant \(f \geq 1\), Strong RRQR requires \(O(mnk \log f n)\) arithmetic operations to find \(k\) columns of \(A\) in \(C \in \mathbb{R}^{m \times k}\) that satisfy

\[
\|A - CC^\top A\|_2 \leq \left(1 + f^2 \sqrt{k(n-k)+1}\right) \cdot \|A - A_k\|_2.
\]

As discussed in Section 1, \([21]\) suggests column sampling with the largest corresponding leverage scores. A related result in \([35]\) suggests column sampling through selection over \(V_k^\top\) with Strong RRQR. Notice that the leverage scores sampling approach is similar, but the column selection is based on the largest Euclidean norms of the columns of \(V_k^\top\).

From a probabilistic point of view, much work has followed the seminal work of \([14]\) for the CSSP. \([14]\) introduced the idea of randomly sampling columns based on specific probability distributions. \([14]\) use a simple probability distribution where each column of \(A\) is sampled with probability proportional to its Euclidean norm. The approximation bound achieved, which holds only in expectation, is

\[
\|A - CC^\top A\|_F^2 \leq \|A - A_k\|_F^2 + (k/c)\|A\|_F^2.
\]

![Image of Table 3](image-url)
Let $\mathbf{C} = \mathbf{A} \mathbf{S} \in \mathbb{R}^{m \times c}$. Then, for $\xi = 2, F$

$$
\|\mathbf{A} - \mathbf{C} \mathbf{C}^\dagger \mathbf{A}\|_F^2 \leq \|\mathbf{E}\|_F^2 \cdot \|\mathbf{S}(\mathbf{Z}^T \mathbf{S})^\dagger\|_F^2.
$$

We will also use the following novel lower bound on the smallest singular value of the matrix $\mathbf{V}_k$, after deterministic selection of its rows based on the largest leverage scores.

**Lemma 5.** Repeat the conditions of Theorem 2. Then,

$$
\sigma_k(\mathbf{V}_k^T \mathbf{S}) > 1 - \varepsilon.
$$

**Proof.** We use the following perturbation result on the sum of eigenvalues of symmetric matrices.

**Lemma 6.** (Theorem 2.8.1; part (i) in [9]) Let $\mathbf{X}$ and $\mathbf{Y}$ be symmetric matrices of order $k$ and let, $1 \leq i, j \leq n$ with $i + j \leq k + 1$. Then,

$$
\lambda_i(\mathbf{X}) \geq \lambda_{i+j-1}(\mathbf{X} + \mathbf{Y}) - \lambda_j(\mathbf{Y}).
$$

Let $\mathbf{S} \in \mathbb{R}^{n \times c}$ sample $c$ columns from $\mathbf{A}$ with $c \geq k$. Similarly, let $\hat{\mathbf{S}} \in \mathbb{R}^{n \times (n-c)}$ sample the rest $n - c$ columns from $\mathbf{A}$. Hence,

$$
\mathbf{I}_k = \mathbf{V}_k^T \mathbf{V}_k = \mathbf{V}_k^T \mathbf{S}^T \mathbf{V}_k + \mathbf{V}_{k}^T \hat{\mathbf{S}}^T \hat{\mathbf{V}}_k.
$$

Let $\mathbf{X} = \mathbf{V}_k^T \mathbf{S}^T \mathbf{V}_k, \mathbf{Y} = \mathbf{V}_{k}^T \hat{\mathbf{S}}^T \hat{\mathbf{V}}_k, i = k, \text{ and } j = 1$, in Lemma 6. Notice that $i + j \leq k + 1$, and $\lambda_k(\mathbf{X} + \mathbf{Y}) = 1$; hence:

$$
\lambda_k(\mathbf{V}_k^T \mathbf{S}^T \mathbf{V}_k) \geq 1 - \lambda_1(\mathbf{V}_k^T \hat{\mathbf{S}}^T \hat{\mathbf{V}}_k) = 1 - \|\mathbf{V}_k^T \hat{\mathbf{S}}\|_F^2 \geq 1 - \|\mathbf{V}_k^T \hat{\mathbf{S}}\|_F^2 > 1 - (k - \theta).
$$

Replacing $\theta = k - \varepsilon$ and $\lambda_k(\mathbf{V}_k^T \mathbf{S}^T \mathbf{V}_k) = \sigma_k^2(\mathbf{V}_k^T \mathbf{S})$ concludes the proof.

The proof of Theorem 2 is a straightforward combination of the Lemmas 4 and 5. First, by picking $\mathbf{Z} = \mathbf{V}_k$ in Lemma 4 we obtain:

$$
\|\mathbf{A} - \mathbf{C} \mathbf{C}^\dagger \mathbf{A}\|_F^2 \leq \|\mathbf{A} - \mathbf{A}_k\|_F^2 \cdot \|\mathbf{S}(\mathbf{V}_k^T \mathbf{S})^\dagger\|_F^2 \leq \|\mathbf{A} - \mathbf{A}_k\|_F^2 \cdot \|\mathbf{S}\|_F^2 \cdot \|\mathbf{V}_k^T \mathbf{S}\|_F^2 = \|\mathbf{A} - \mathbf{A}_k\|_F^2 \cdot \|\mathbf{V}_k\|_F^2 = \|\mathbf{A} - \mathbf{A}_k\|_F^2 \sigma_k^2(\mathbf{V}_k^T \mathbf{S}).
$$

In the above, we used the facts that

$$
\mathbf{E} = \mathbf{A} - \mathbf{A} \mathbf{V}_k \mathbf{V}_k^T = \mathbf{A} - \mathbf{A}_k,
$$

and the spectral norm of the sampling matrix $\mathbf{S}$ equals one. Also, we used that $\text{rank}(\mathbf{V}_k^T \mathbf{S}) = k$, which is implied from Lemma 5. Next, via the bound in Lemma 5:

$$
\|\mathbf{A} - \mathbf{C} \mathbf{C}^\dagger \mathbf{A}\|_F^2 \leq \|\mathbf{A} - \mathbf{A}_k\|_F^2 \|\mathbf{S}(\mathbf{V}_k^T \mathbf{S})^\dagger\|_F^2 = \|\mathbf{A} - \mathbf{A}_k\|_F^2 \|\mathbf{V}_k\|_F^2(1 + \varepsilon).
$$

**6.2 Proof of Theorem 3**

Let $\alpha_k = 1 + \eta$ for some $\eta > 0$. We assume that the leverage scores follow a power law decay such that: $\ell_i^{(k)} = \ell_i^{(k)} / i^{\eta}$. According to the proposed algorithm, we select $c$ columns such that $\sum_{i=1}^c \ell_i^{(k)} > \theta$. Here, we bound the number of columns $c$ required to achieve an $\varepsilon := k - \theta$ approximation in Theorem 2. To this end, we use the extreme case $\sum_{i=1}^c \ell_i^{(k)} = \theta$ which guarantees an $(1 + \varepsilon)$-approximation.

For our analysis, we use the following well-known result.
Proposition 7. [Integral test for convergence] Let \( f(\cdot) \geq 0 \) be a function defined over the set of positive reals. Furthermore, assume that \( f(\cdot) \) is monotone decreasing. Then,
\[
\int_j^{j+1} f(i)dx \leq \sum_{i=j}^j f(i) \leq f(j) + \int_j^j f(x)dx,
\]
over the interval \([j, \ldots, J]\) for \( j, J \) positive integers.

In our case, consider \( f(i) = \frac{1}{i^{1+\eta}} \). By definition of the leverage scores, we have:
\[
k = \sum_{i=1}^n \ell_i^{(k)} = \sum_{i=1}^n \frac{1}{i^{1+\eta}} \implies \ell_i^{(k)} = \frac{k}{\sum_{i=1}^n i^{1+\eta}}.
\]

By construction, we collect \( c \) leverage scores such that \( k - \theta = \varepsilon \). This leads to:
\[
k - \varepsilon = \ell_i^{(k)} = \sum_{i=1}^c \frac{1}{i^{1+\eta}} = \frac{k}{\sum_{i=1}^c i^{1+\eta}} \implies \\
k = \sum_{i=c+1}^n \frac{1}{i^{1+\eta}} = \frac{1}{c^{1+\eta}} + \int_{c+1}^n \frac{1}{i^{1+\eta}}dx = \frac{1}{c^{1+\eta}} + \left[-\frac{1}{i^{1+\eta}}\right]_{i=c+1}^n = \frac{1}{c^{1+\eta}} + \frac{1}{\eta} \left(\frac{1}{(c+1)^{1+\eta}} - \frac{1}{n^{1+\eta}}\right) \\
= \frac{1}{(c+1)^{1+\eta}} + \frac{1}{\eta} \left(\frac{1}{(c+1)^{1+\eta}} - \frac{1}{n^{1+\eta}}\right) = \frac{1}{(c+1)^{1+\eta}} + \frac{1}{\eta} \left(\frac{1}{(c+1)^{1+\eta}} - \frac{1}{n^{1+\eta}}\right) \\
= \frac{1}{(c+1)^{1+\eta}} + \frac{1}{\eta} \left(\frac{1}{(c+1)^{1+\eta}} - \frac{1}{n^{1+\eta}}\right)
\]
To bound the quantity on the right hand side, we observe
\[
\sum_{i=c+1}^n \frac{1}{i^{1+\eta}} < \int_{c+1}^n \frac{1}{i^{1+\eta}}dx = \frac{1}{(c+1)^{1+\eta}} + \int_{c+1}^n \frac{1}{i^{1+\eta}}dx = \frac{1}{(c+1)^{1+\eta}} + \left[-\frac{1}{i^{1+\eta}}\right]_{c+1}^n = \frac{1}{(c+1)^{1+\eta}} + \frac{1}{\eta} \left(\frac{1}{(c+1)^{1+\eta}} - \frac{1}{n^{1+\eta}}\right)
\]
where the first inequality is due to the right hand side of the integral test and the third inequality is due to \( 1 + \sum_{i=2}^n \frac{1}{i^{1+\eta}} > 1 \). Hence, we may conclude:
\[
\varepsilon < k \cdot \max \left\{ \frac{2}{(c+1)^{1+\eta}}, \frac{2}{\eta \cdot (c+1)^\eta} \right\}
\]
The above lead to the following two cases: if
\[
\varepsilon < \frac{2k}{(c+1)^{1+\eta}},
\]
we have:
\[
c < \left( \frac{2 \cdot k}{\varepsilon} \right)^{\frac{1}{\eta}} - 1,
\]
whereas in the case where
\[
\varepsilon < \frac{2 \cdot k}{\eta \cdot (c+1)^\eta},
\]
we get
\[
c < \left( \frac{2 \cdot k}{\eta \cdot \varepsilon} \right)^{\frac{1}{\eta}} - 1.
\]

7. THE KEY ROLE OF \( \theta \)

In the proof of Theorem 2, we require that
\[
\sigma_k^2(V_k^\top S) > 1 - (k - \theta) := 1 - \varepsilon.
\]
For this condition to hold, the sampling matrix \( S \) should preserve the rank of \( V_k \) in \( V_k^\top S \), i.e., choose \( \theta \) such that \( \operatorname{rank}(V_k^\top S) = k \).

Failing to preserve the rank of \( V_k \) has immediate implications for the CSSP. To highlight this, let \( A \in \mathbb{R}^{m \times n} \) of rank \( k < \min\{m, n\} \) with SVD \( A = U_k \Sigma_k V_k^\top \). Further, assume that the \( k \)th singular value of \( A \) is arbitrary large, i.e., \( \sigma_k(A) \to \infty \). Also, let rank\( V_k^\top S = \gamma < k \) and \( C = AS \).

Then,
\[
\|A - CC^\top A\|_\xi
\]
\[
= \|U_k \Sigma_k V_k^\top - U_k \Sigma_k V_k^\top S(U_k \Sigma_k V_k^\top)^\top U_k \Sigma_k V_k^\top\|_\xi
\]
\[
= \|\Sigma_k - \Sigma_k V_k^\top S(U_k \Sigma_k V_k^\top)^\top U_k \Sigma_k\|_\xi
\]
\[
= \|\Sigma_k - \Sigma_k V_k^\top S(U_k \Sigma_k V_k^\top)^\top U_k \Sigma_k\|_\xi
\]
\[
= \|\Sigma_k - \Sigma_k V_k^\top S(U_k \Sigma_k V_k^\top)^\top U_k \Sigma_k\|_\xi
\]
\[
\geq \sigma_k(A)
\]
The second equality is due to the fact that both spectral and Frobenius norms are invariant to unitary transformations. In the third equality, we used the fact that \( (WZ)^\top = Z^\top W^\top \) if \( W^\top W \) is the identity matrix. Then, set \( X = \Sigma_k V_k^\top S \in \mathbb{R}^{k \times \gamma} \) where \( \operatorname{rank}(X) = \gamma \). Using this notation, let \( U_X \in \mathbb{R}^{m \times \gamma} \) be any orthonormal basis for \( \text{span}(X) \). Observe \( U_X U_X^\top = XX^\top \). The last inequality is due to \( U_X U_X^\top \) being an \( m \times m \) diagonal matrix with \( \gamma \) ones along its main diagonal and the rest zeros. Thus, we may conclude that for this \( A \):
\[
\|A - CC^\top A\|_\xi \geq \sigma_k(A) \to \infty.
\]

8. CONCLUDING REMARKS

We provided a rigorous theoretical analysis of an old and popular deterministic feature selection algorithm from the statistics literature [21]. Although randomized algorithms are often easier to analyze, we believe that deterministic algorithms are simpler to implement and explain, hence more attractive to practitioners and data analysts.

One interesting path for future research is understanding the connection of this work with the so-called “spectral graph sparsification problem” [31]. In that case, edge selection in a graph is implemented via randomized leverage scores sampling from an appropriate matrix (see Theorem 1 in [31]). Note that in the context of graph sparsification, leverage scores correspond to the so-called “effective resistances” of the graph. Can deterministic effective resistances sampling be rigorously analyzed? What graphs have effective resistances following a power law distribution?
References


