An adaptive randomized pivoting strategy for low-rank approximation

Alice Cortinovis (Università di Pisa)

Joint work with Daniel Kressner (EPFL)

February 5, 2026

Randomized Numerical Linear Algebra Workshop @ ICERM

Low-rank approximation by column subset selection

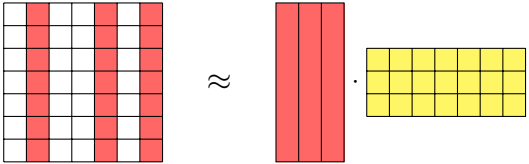
Notation:

- Matrix $A \in \mathbb{R}^{n \times n}$, target rank $k \ll n$
- Singular values are $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$
- Gold standard for low-rank approximation is truncated singular value decomposition (SVD). However, it is expensive to compute!

Error of best rank-k approximation A_k is, by Eckart-Young theorem,

a) σ_{k+1} for the spectral norm $\|\cdot\|_2$,
b) $\sqrt{\sigma_{k+1}^2 + \ldots + \sigma_n^2}$ for the Frobenius norm $\|\cdot\|_F$.

Column subset selection:



$C = A(:, J)$ for some $J \in \{1, \ldots, n\}^k$, $|J| = k$

Yellow matrix $= C^\dagger A$

Objective: Choose k columns such that $\|A - CC^\dagger A\|_F$ is small.

A good column subset

$$\min_{|J|=k} \|A - \Pi_{A(:,J)} A\|_F = ?$$

Theorem ([Deshpande/Rademacher/Vempala/Wang'2006])

For any matrix $A \in \mathbb{R}^{n\times n}$ and $k \leq n$, there exists an index set J of cardinality k such that

$$\|A - \Pi_{A(:,J)} A\|_F = \|A - A(:,J)A(:,J)^\dagger A\|_F \leq \sqrt{k+1}\|A - A_k\|_F.$$

Proof based on volume sampling.

Deterministic algorithm(s) to find a suitable J exist, but they are at least as expensive as SVD!

A. Deshpande, L. Rademacher, Efficient volume sampling for row/column subset selection. 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, IEEE, 2010.

A. Cortinovis, D. Kressner, Low-rank approximation in the Frobenius norm by column and row subset selection. SIAM Journal on Matrix Analysis and Applications, 41.4 (2020): 1651-1673.

A. Osinsky, Close to optimal column approximations with a single SVD. Linear Algebra and its Applications, 2025.

A (non-exhaustive) list of column subset selection techniques

Deterministic strategies:

- QR with column pivoting
- Strong rank-revealing QR
- Osinsky's algorithm

Randomized strategies:

- Uniform sampling
- Sampling with respect to squared norms of columns of A
- Leverage scores
- Volume sampling / determinantal point processes
- Randomized pivoted Cholesky and randomized column-pivoted QR
- Adaptive Randomized Pivoting

Mixed strategies:

- Uniform sampling + refinement via rank-revealing algorithm
- Iterative volume maximization

An old idea: QR with column pivoting

Input: matrix A, rank k
Output: column indices $j_1, \dots, j_k$

1: Set $R \leftarrow A$
2: **for** $h = 1, \dots, k$ **do**
3: Select a new index j_h corresponding to the column of $R(h+1:n,:)$ of maximum norm
4: Find a Householder (orthogonal) matrix Q_h such that $Q_h R(h+1:n,:)$ introduces zeros in the j_h-th column of R
5: Update

$$R \leftarrow \begin{bmatrix} I_{h-1} & 0 \\ 0 & Q_h \end{bmatrix} R$$

6: **end for**

Theorem ([Gu/Eisenstat, 1996])

The index set J returned by QR with column pivoting satisfies

$$\|A - \Pi_{A(:,J)} A\|_2 \leq 2^k \sqrt{n-k} \cdot \sigma_{k+1}(A).$$

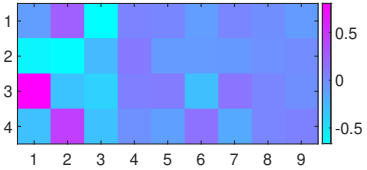
Randomized version: [Chen/Epperly/Tropp/Webber 2024]

Sampling with respect to leverage scores

Let V be the matrix containing the first k right singular vectors of A.

Some information about which columns of A are the most "important" can be extracted from V.

Simplest example:



$$\rightsquigarrow V_3 \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Leverage score sampling (e.g. [Boutsidis/Mahoney/Drineas 2009]):
Define probability distribution proportional to

$$\text{leverage scores } \ell_j := \|V(j,:)\|_2^2,$$

and take $\approx k \log k$ i.i.d. samples.

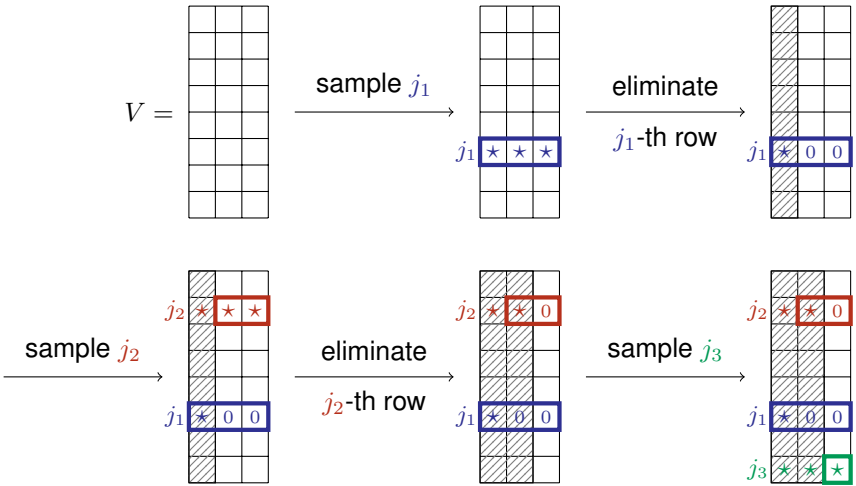
Our proposal: Adaptive Randomized Pivoting (ARP)

Input: Matrix $V \in \mathbb{R}^{n \times k}$ with orthonormal columns s.t. $A \approx AVV^T$. (more on this in the next slides)

Strategy: Perform a randomized version of column pivoted QR on V^T to select indices. At the first step, pick a column with probability proportional to the squared column norms of V^T. At the next steps, update the probability distribution.

1: Initialize $J = ()$ and $V_0 = V$
2: **for** $r = 1, \ldots, k$ **do**
3: Set $p_j = \|V_{r-1}(j, r:k)\|_2^2/(k-r+1)$ for $j = 1, \ldots, n$
4: Sample index j_r according to probabilities p_j
5: Update $J \leftarrow (J, j_r)$
6: Update $V_r \leftarrow V_{r-1}Q_r$ with Householder reflector Q_r that annihilates $V_{r-1}(j_r, r+1:k)$
7: **end for**

A visual illustration of ARP



Accuracy guarantees

Theorem ([C./Kressner'2026])

Given a matrix $V \in \mathbb{R}^{n\times k}$ with orthonormal columns, the random index set J returned by ARP satisfies

$$\mathbb{E}[\|A - \Pi_{A(:,J)}A\|_F^2] \leq \mathbb{E}[\|A - A(:,J)V(J,:)^{-T}V^T\|_F^2] = (k+1)\|A - AVV^T\|_F^2.$$

- If V is the matrix of the first k right singular vectors, the result matches the best existence result in the Frobenius norm.
- We get a low-rank approximation (the one in the middle) which is cheap to compute.
- How to get a good/decent V?

Note: bounds on the expected value can be turned into high-probability bounds via Markov/Chernoff inequalities.

For example, we have $\|A - \Pi_{A(:,J)}A\|_F \leq 10(k+1)\|A - AVV^T\|_F$ with probability at least 99%.

Choosing a cheaper V

Possible solution: randomized rangefinder!

Input: matrix A, rank k
Output: column indices $j_1, \ldots, j_{k+2}$

1: Compute a sketch $A^T\Omega$ for some Gaussian matrix $\Omega \in \mathbb{R}^{n\times(k+2)}$
2: Compute an orthonormal basis Q of the sketch
3: Apply randomly pivoted QR on Q^T to select the indices

Then, using our result + analysis of randomized rangefinder, we have

$$\mathbb{E}[\|A - \Pi_C A\|_F^2] \leq (k+3)(k+1)\|A - A_k\|_F^2.$$

Relation to existing column subset selection strategies

- ARP is a randomized version of Osinsky's algorithm. ARP could fail to get a good column subset, albeit with a low probability which is under control. ARP is faster because Osinsky's algorithm needs to deal with additional $n \times n$ matrices that should be updated at each step.

Relation to existing column subset selection strategies

- ARP is a randomized version of Osinsky's algorithm. ARP could fail to get a good column subset, albeit with a low probability which is under control. ARP is faster because Osinsky's algorithm needs to deal with additional $n \times n$ matrices that should be updated at each step.
- ARP can be seen as an *adaptive* leverage score sampling strategy. The first column is chosen sampling wrt leverage scores, but the distribution is updated after each step. ARP does not need oversampling like leverage score sampling does!

Relation to existing column subset selection strategies

- ARP is a randomized version of Osinsky's algorithm. ARP could fail to get a good column subset, albeit with a low probability which is under control. ARP is faster because Osinsky's algorithm needs to deal with additional $n \times n$ matrices that should be updated at each step.
- ARP can be seen as an *adaptive* leverage score sampling strategy. The first column is chosen sampling wrt leverage scores, but the distribution is updated after each step. ARP does not need oversampling like leverage score sampling does!
- It also makes sense to perform randomly pivoted QR on the matrix A directly (instead of V^T) [Chen/Epperly/Tropp/Webber'2024]. ARP needs the matrix V but has stronger theoretical guarantees and does not need oversampling.

Relation to existing column subset selection strategies

- ARP is a randomized version of Osinsky's algorithm. ARP could fail to get a good column subset, albeit with a low probability which is under control. ARP is faster because Osinsky's algorithm needs to deal with additional $n \times n$ matrices that should be updated at each step.
- ARP can be seen as an *adaptive* leverage score sampling strategy. The first column is chosen sampling wrt leverage scores, but the distribution is updated after each step. ARP does not need oversampling like leverage score sampling does!
- It also makes sense to perform randomly pivoted QR on the matrix A directly (instead of V^T) [Chen/Epperly/Tropp/Webber'2024]. ARP needs the matrix V but has stronger theoretical guarantees and does not need oversampling.
- ARP is equivalent to taking a sample from a k-determinantal point process corresponding to the kernel VV^T. This is closely connected to volume sampling, which plays a fundamental role in the accuracy of column subsets [Epperly'2026].

Numerical example #1

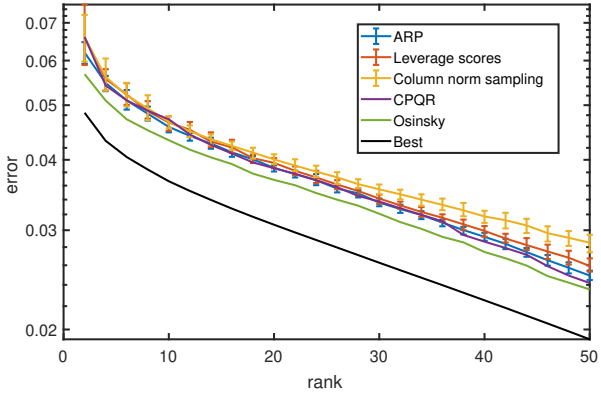
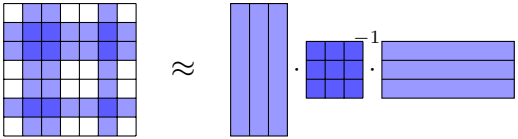


Figure: GSE10072 cancer genetics data set from the National Institutes of Health, downloaded from `https://ftp.ncbi.nlm.nih.gov/geo/series/GSE10nnn/GSE10072/matrix/`. The matrix A has size $22,283 \times 107$.

Symmetric positive semi-definite (SPSD) matrices

For A SPSD we consider the Nyström approximation $A \approx A(:, J)A(J, J)^{-1}A(J, :)$:



Key insight: Nyström for $A \iff$ column subset selection for B s.t. $B^T B = A$ (e.g. Cholesky factor)

Corollary [C./Kressner'2026]

Let A be SPSD and let $V \in \mathbb{R}^{n \times k}$ be a matrix with orthonormal columns.
Then the random index set J returned by ARP satisfies

$$\mathbb{E}[\|A - A(:, J)A(J, J)^{-1}A(J, :)\|_*] \leq (k + 1)\|(I_n - VV^T)A(I_n - VV^T)\|_*,$$

where $\|\cdot\|_*$ denotes the nuclear norm of a matrix.

This can be derandomized similarly to the column subset selection problem; however, it is cheaper!

Numerical example #2

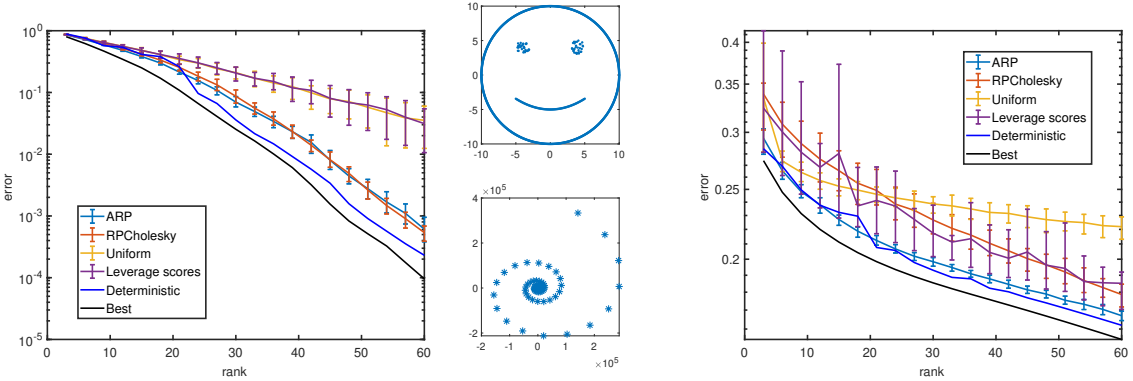
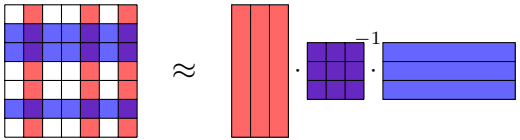


Figure: Selection of column indices for SPSD matrices: Gaussian kernel matrices associated to points distributed on a smile (left) and a spiral (right).

Cross approximation

Cross approximation:



Middle matrix: $A(I,J)^{-1}$

ARPcross algorithm

Input: Matrix A, orthonormal basis $V \in \mathbb{R}^{n\times r}$
Output: Index sets $I = (i_1, \ldots, i_k)$ and $J = (j_1, \ldots, j_k)$ defining a cross approximation

- Obtain index set J by applying ARP to V
- Compute an orthonormal basis Q_J of $A(:,J)$ by a QR decomposition
- Obtain index set I by applying ARP to Q_J

Theorem ([C./Kressner'2026])

The random index sets I and J returned by ARPcross algorithm satisfy

$$\mathbb{E}\left[\|A - A(:,J)A(I,J)^{-1}A(I,:)\|_F^2\right] \leq (k+1)^2\|A - AVV^T\|_F^2.$$

Numerical example #3

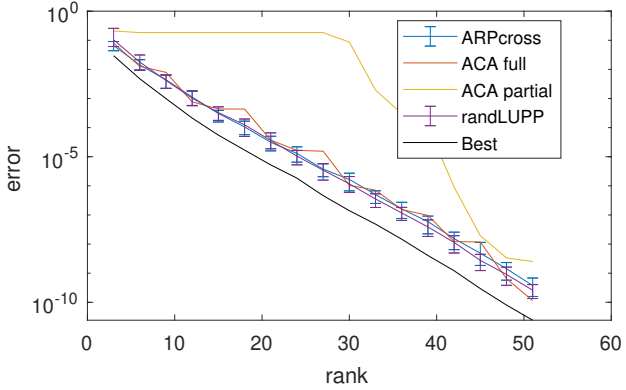


Figure: Nonsymmetric kernel matrix $A \in \mathbb{R}^{2000\times 2000}$ obtained from discretizing the kernel function $K(\alpha,\beta) = \exp\left(-15\sqrt{\alpha^2+\beta^2}\right) + \exp\left(-75\sqrt{(\alpha-1)^2+(\beta-1)^2}\right)$ by choosing 2000 equispaced samples for $\alpha \in [0,1]$ and 2000 uniformly random samples for $\beta \in [0,1]$.

Application to Discrete Empirical Interpolation Method (DEIM)

Setting: Expensive function $f \equiv f(\xi) \in \mathbb{R}^n$, but good orthonormal basis $V \in \mathbb{R}^{n\times r}$ s.t. $f \approx VV^T f$

Goal of DEIM: Construct approximation of f that does not require evaluating the whole function.

How: Select set J of r indices and return the approximation $f \approx V(V(J,:))^{-1} f(J)$
Quality is related to $\|V(J,:)^{-1}\|_2$ because

$$\|f - V(V(J,:))^{-1} f(J)\|_2 \leq \|(V(J,:))^{-1}\|_2 \|f - VV^T f\|_2.$$

We can apply ARP to V^T!

Corollary

Given an orthonormal basis $V \in \mathbb{R}^{n\times r}$, the index set returned by ARP satisfies

$$\mathbb{E}[\|f - V(V(J,:))^{-1} f(J)\|_2^2] \leq (r+1)\|f - VV^T f\|_2^2$$

for an arbitrary but fixed vector $f \in \mathbb{R}^n$. Moreover,

$$\mathbb{E}[\|(V(J,:))^{-1}\|_F^2] = r(n-r+1), \quad \mathbb{E}[\|(V(J,:))^{-1}\|_2^2] \leq r(n-r)+1.$$

Note: This recovers a result from [Dereziński/Warmuth'2018]; see also [Epperly'2025].

Conclusions

ARP is a versatile randomized algorithm that can be used for several low-rank approximation problems: column subset selection, DEIM, Nyström approximation, CUR approximation, cross approximation.

Easy to code, strong theoretical guarantees!

 Alice Cortinovis and Daniel Kressner. Adaptive randomized pivoting for column subset selection, DEIM, and low-rank approximation, SIMAX (2026)

Thank you for listening!