Rank revealing factorizations, and low rank approximations

L. Grigori

Inria Paris, UPMC

February 2017

Plan

Low rank matrix approximation

Rank revealing QR factorization

LU_CRTP: Truncated LU factorization with column and row tournament pivoting

Experimental results, LU_CRTP

Randomized algorithms for low rank approximation

Plan

Low rank matrix approximation

Rank revealing QR factorization

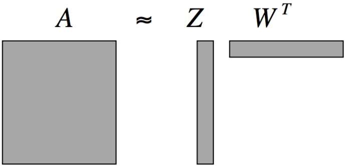
LU_CRTP: Truncated LU factorization with column and row tournament pivoting

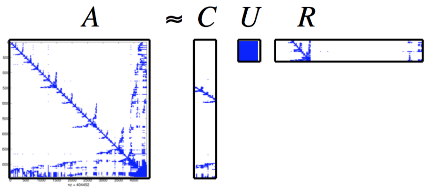
Experimental results, LU_CRTP

Randomized algorithms for low rank approximation

Low rank matrix approximation

- Problem: given $m \times n$ matrix A, compute rank-k approximation ZW^T, where Z is $m \times k$ and W^T is $k \times n$.





- Problem with diverse applications
  - from scientific computing: fast solvers for integral equations, H-matrices
  - to data analytics: principal component analysis, image processing, ...

$$Ax \rightarrow ZW^T x$$

$$Flops \quad 2mn \rightarrow 2(m+n)k$$

Singular value decomposition

Given $A \in \mathbb{R}^{m \times n}$, $m \geq n$ its singular value decomposition is

$$A = U\Sigma V^T = \begin{pmatrix} U_1 & U_2 & U_3 \end{pmatrix} \cdot \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} V_1 & V_2 \end{pmatrix}^T$$

where

- U is $m \times m$ orthogonal matrix, the left singular vectors of A ,
  U_1 is $m \times k$, U_2 is $m \times n - k$, U_3 is $m \times m - n$
- Σ is $m \times n$, its diagonal is formed by $\sigma_1(A) \geq \ldots \geq \sigma_n(A) \geq 0$
  Σ_1 is $k \times k$, Σ_2 is $n - k \times n - k$
- V is $n \times n$ orthogonal matrix, the right singular vectors of A,
  V_1 is $n \times k$, V_2 is $n \times n - k$

Norms

$$
\begin{aligned}
||A||_F &= \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{\sigma_1^2(A) + \dots \sigma_n^2(A)} \\
||A||_2 &= \sigma_{max}(A) = \sigma_1(A)
\end{aligned}
$$

Some properties:

$$||A||_2 \leq ||A||_F \leq \sqrt{min(m,n)}||A||_2$$

Orthogonal Invariance: If $Q \in \mathbb{R}^{m\times m}$ and $Z \in \mathbb{R}^{n\times n}$ are orthogonal, then

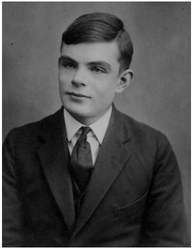
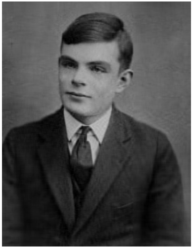
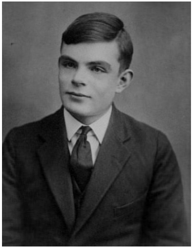
$$
\begin{aligned}
|QAZ||_F &= ||A||_F \\
||QAZ||_2 &= ||A||_2
\end{aligned}
$$

Low rank matrix approximation

- Best rank-k approximation $A_k = U_k \Sigma_k V_k$ is rank-k truncated SVD of A [Eckart and Young, 1936]

$$\min_{rank(\tilde{A}_k) \leq k} ||A - \tilde{A}_k||_2 \quad = \quad ||A - A_k||_2 = \sigma_{k+1}(A) \tag{1}$$

$$\min_{rank(\tilde{A}_k) \leq k} ||A - \tilde{A}_k||_F \quad = \quad ||A - A_k||_F = \sqrt{\sum_{j=k+1}^{n} \sigma_j^2(A)} \tag{2}$$

Original image of size 919×707	Rank-38 approximation, SVD	Rank-75 approximation, SVD
		

- Image source: `https://upload.wikimedia.org/wikipedia/commons/a/a1/Alan_Turing_Aged_16.jpg`

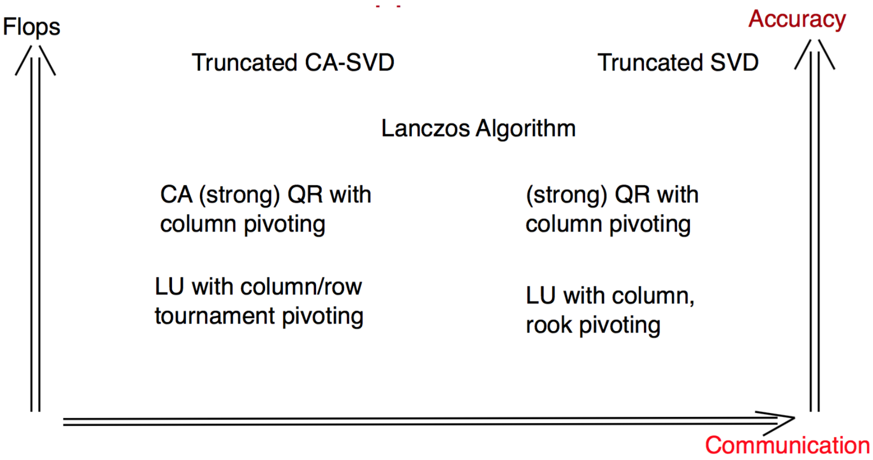
Large data sets

Matrix A might not exist entirely at a given time, rows or columns are added progressively.

- Streaming algorithm: can solve an arbitrarily large problem with one pass over the data (a row or a column at a time).
- Weakly streaming algorithm: can solve a problem with $O(1)$ passes over the data.

Matrix A might exist only implicitly, and it is never formed explicitly.

Low rank matrix approximation: trade-offs



Plan

Low rank matrix approximation

Rank revealing QR factorization

LU_CRTP: Truncated LU factorization with column and row tournament pivoting

Experimental results, LU_CRTP

Randomized algorithms for low rank approximation

Rank revealing QR factorization

Given A of size $m \times n$, consider the decomposition

$$AP_c = QR = Q \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix}, \tag{3}$$

where R_{11} is $k \times k$, P_c and k are chosen such that $||R_{22}||_2$ is small and R_{11} is well-conditioned.

- By the interlacing property of singular values [Golub, Van Loan, 4th edition, page 487],

  $$\sigma_i(R_{11}) \leq \sigma_i(A) \text{ and } \sigma_j(R_{22}) \geq \sigma_{k+j}(A)$$

  for $1 \leq i \leq k$ and $1 \leq j \leq n-k$.
- $\sigma_{k+1}(A) \leq \sigma_{max}(R_{22}) = ||R_{22}||$

Rank revealing QR factorization

Given A of size $m \times n$, consider the decomposition

$$AP_c = QR = Q \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix}. \quad (4)$$

If $||R_{22}||_2$ is small,

- $Q(:, 1:k)$ forms an approximate orthogonal basis for the range of A,

$$\begin{aligned} a(:,j) &= \sum_{i=1}^{min(j,k)} r_{ij} Q(:,i) \in span\{Q(:,1), \ldots Q(:,k)\} \\ ran(A) &\in span\{Q(:,1), \ldots Q(:,k)\} \end{aligned}$$

- $P_c \begin{bmatrix} -R_{11}^{-1} R_{12} \\ I \end{bmatrix}$ is an approximate right null space of A.

Rank revealing QR factorization

The factorization from equation (4) is rank revealing if

$$1 \le \frac{\sigma_i(A)}{\sigma_i(R_{11})}, \frac{\sigma_j(R_{22})}{\sigma_{k+j}(A)} \le q_1(n,k),$$

for $1 \le i \le k$ and $1 \le j \le \min(m,n) - k$, where

$$\sigma_{max}(A) = \sigma_1(A) \ge \ldots \ge \sigma_{min}(A) = \sigma_n(A)$$

It is **strong** rank revealing [Gu and Eisenstat, 1996] if in addition

$$||R_{11}^{-1} R_{12}||_{max} \le q_2(n,k)$$

- Gu and Eisenstat show that given k and f, there exists a P_c such that $q_1(n,k) = \sqrt{1 + f^2 k(n-k)}$ and $q_2(n,k) = f$.
- Factorization computed in $4mnk$ (QRCP) plus $O(mnk)$ flops.

QR with column pivoting [Businger and Golub, 1965]

Idea:

- At first iteration, trailing columns decomposed into parallel part to first column (or e_1) and orthogonal part (in rows $2 : m$).
- The column of maximum norm is the column with largest component orthogonal to the first column.

Implementation:

- Find at each step of the QR factorization the column of maximum norm.
- Permute it into leading position.
- If rank(A) = k, at step $k + 1$ the maximum norm is 0.
- No need to compute the column norms at each step, but just update them since

$$Q^T v = w = \begin{bmatrix} w_1 & w(2:n) \end{bmatrix}, \;\; ||w(2:n)||_2^2 = ||v||_2^2 - w_1^2$$

QR with column pivoting [Businger and Golub, 1965]

Sketch of the algorithm

column norm vector: $colnrm(j) = ||A(:,j)||_2, j = 1:n$.

for j = 1 : n **do**

Find column p of largest norm

if $colnrm[p] > \epsilon$ **then**

1. Pivot: swap columns j and p in A and modify $colnrm$.
2. Compute Householder matrix H_j s.t. $H_j A(j:m,j) = \pm||A(j:m,j)||_2 e_1$.
3. Update $A(j:m, j+1:n) = H_j A(j:m, j+1:n)$.
4. Norm downdate $colnrm(j+1:n)^2 -= A(j, j+1:n)^2$.

else Break

end if

end for

If algorithm stops after k steps

$$\sigma_{max}(R_{22}) \leq \sqrt{n-k} \max_{1 \leq j \leq n-k} ||R_{22}(:,j)||_2 \leq \sqrt{n-k}\epsilon$$

Strong RRQR [Gu and Eisenstat, 1996]

Since

$$det(R_{11}) = \prod_{i=1}^{k} \sigma_i(R_{11}) = \sqrt{det(A^T A)} / \prod_{i=1}^{n-k} \sigma_i(R_{22})$$

a stron RRQR is related to a large $det(R_{11})$. The following algorithm interchanges columns that increase $det(R_{11})$, given f and k.

Compute a strong RRQR factorization, given k:

Compute $A\Pi = QR$ by using QRCP
while there exist i and j such that $det(\tilde{R}_{11})/det(R_{11}) > f$, where
$R_{11} = R(1:k, 1:k)$, $\Pi_{i,j+k}$ permutes columns i and $j+k$,
$R\Pi_{i,j+k} = Q\tilde{R}, \tilde{R}_{11} = \tilde{R}(1:k, 1:k)$ **do**
Find i and j
Compute $R\Pi_{i,j+k} = Q\tilde{R}$ and $\Pi = \Pi\Pi_{i,j+k}$
end while

Strong RRQR (contd)

It can be shown that

$$\frac{det(\tilde{R}_{11})}{det(R_{11})} = \sqrt{\left(R_{11}^{-1}R_{12}\right)_{i,j}^{2} + \omega_i^2\left(R_{11}\right)\chi_j^2\left(R_{22}\right)} \tag{5}$$

for any $1 \leq i \leq k$ and $1 \leq j \leq n-k$ (the 2-norm of the j-th column of A is $\chi_j(A)$, and the 2-norm of the j-th row of A^{-1} is $\omega_j(A)$).

Compute a strong RRQR factorization, given k:

Compute $A\Pi = QR$ by using QRCP

while $\max_{1\leq i\leq k, 1\leq j\leq n-k} \sqrt{\left(R_{11}^{-1}R_{12}\right)_{i,j}^{2} + \omega_i^2\left(R_{11}\right)\chi_j^2\left(R_{22}\right)} > f$ **do**

Find i and j such that $\sqrt{\left(R_{11}^{-1}R_{12}\right)_{i,j}^{2} + \omega_i^2\left(R_{11}\right)\chi_j^2\left(R_{22}\right)} > f$

Compute $R\Pi_{i,j+k} = Q\tilde{R}$ and $\Pi = \Pi\Pi_{i,j+k}$

end while

Strong RRQR (contd)

- $det(R_{11})$ strictly increases with every permutation, no permutation repeats, hence there is a finite number of permutations to be performed.

Strong RRQR (contd)

Theorem

[Gu and Eisenstat, 1996] If the QR factorization with column pivoting as in equation (4) *satisfies inequality*

$$\sqrt{\left(R_{11}^{-1}R_{12}\right)_{i,j}^{2} + \omega_i^2\left(R_{11}\right)\chi_j^2\left(R_{22}\right)} < f$$

for any $1 \le i \le k$ *and* $1 \le j \le n-k$*, then*

$$1 \le \frac{\sigma_i(A)}{\sigma_i(R_{11})}, \frac{\sigma_j(R_{22})}{\sigma_{k+j}(A)} \le \sqrt{1 + f^2 k(n-k)},$$

for any $1 \le i \le k$ *and* $1 \le j \le \min(m,n) - k$.

Sketch of the proof ([Gu and Eisenstat, 1996])

Assume A is full column rank. Let $\alpha = \sigma_{max}(R_{22})/\sigma_{min}(R_{11})$, and let

$$R = \begin{bmatrix} R_{11} & \\ & R_{22}/\alpha \end{bmatrix} \begin{bmatrix} I_k & R_{11}^{-1}R_{12} \\ & \alpha I_{n-k} \end{bmatrix} = \tilde{R}_1 W_1.$$

We have

$$\sigma_i(R) \leq \sigma_i(\tilde{R}_1)||W_1||_2, 1 \leq i \leq n.$$

Since $\sigma_{min}(R_{11}) = \sigma_{max}(R_{22}/\alpha)$, then $\sigma_i(\tilde{R}_1) = \sigma_i(R_{11})$, for $1 \leq i \leq k$.

$$\begin{aligned} ||W_1||_2^2 &\leq 1 + ||R_{11}^{-1}R_{12}||_2^2 + \alpha^2 = 1 + ||R_{11}^{-1}R_{12}||_2^2 + ||R_{22}||_2^2||R_{11}^{-1}||_2^2 \\ &\leq 1 + ||R_{11}^{-1}R_{12}||_F^2 + ||R_{22}||_F^2||R_{11}^{-1}||_F^2 \\ &= 1 + \sum_{i=1}^{k}\sum_{j=1}^{n-k}\left((R_{11}^{-1}R_{12})_{i,j}^2 + \omega_i^2\left(R_{11}\right)\chi_j^2\left(R_{22}\right)\right) \leq 1 + f^2 k(n-k) \end{aligned}$$

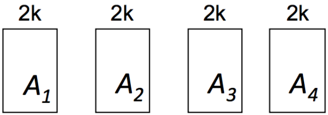
We obtain,

$$\frac{\sigma_i(A)}{\sigma_i(R_{11})} \leq \sqrt{1 + f^2 k(n-k)}$$

Tournament pivoting [Demmel et al., 2015]

One step of CA_RRQR, tournament pivoting used to select k columns

- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$, by using QR with column pivoting
- Permute A_{ji} in leading positions, compute QR with no pivoting

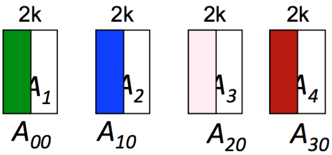


$$AP_{c1} = Q_1 \begin{pmatrix} R_{11} & * \\ & * \end{pmatrix}$$

Tournament pivoting [Demmel et al., 2015]

One step of CA_RRQR, tournament pivoting used to select k columns

- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$, by using QR with column pivoting
- Permute A_{ji} in leading positions, compute QR with no pivoting



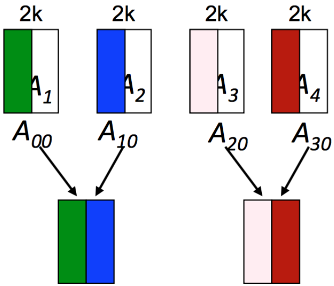
$$AP_{c1} = Q_1 \begin{pmatrix} R_{11} & * \\ & * \end{pmatrix}$$

Tournament pivoting [Demmel et al., 2015]

One step of CA_RRQR, tournament pivoting used to select k columns

- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$, by using QR with column pivoting
- Permute A_{ji} in leading positions, compute QR with no pivoting

$$AP_{c1} = Q_1 \begin{pmatrix} R_{11} & * \\ & * \end{pmatrix}$$

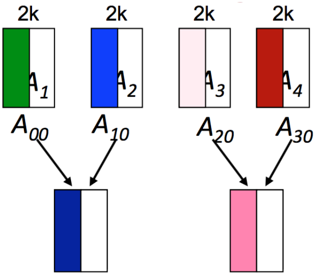


Tournament pivoting [Demmel et al., 2015]

One step of CA_RRQR, tournament pivoting used to select k columns

- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$, by using QR with column pivoting
- Permute A_{ji} in leading positions, compute QR with no pivoting

$$AP_{c1} = Q_1 \begin{pmatrix} R_{11} & * \\ & * \end{pmatrix}$$

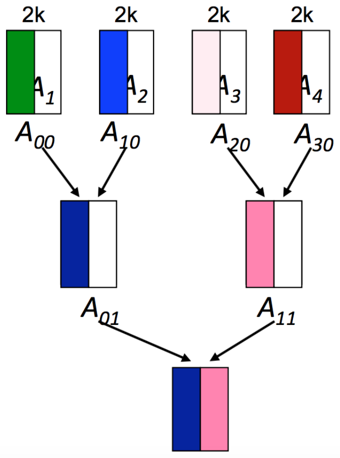


Tournament pivoting [Demmel et al., 2015]

One step of CA_RRQR, tournament pivoting used to select k columns

- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$, by using QR with column pivoting
- Permute A_{ji} in leading positions, compute QR with no pivoting

$$AP_{c1} = Q_1 \begin{pmatrix} R_{11} & * \\ & * \end{pmatrix}$$

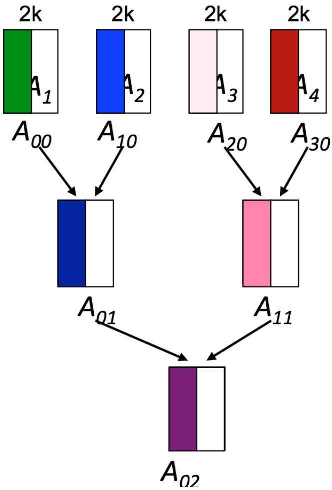


Tournament pivoting [Demmel et al., 2015]

One step of CA_RRQR, tournament pivoting used to select k columns

- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$, by using QR with column pivoting
- Permute A_{ji} in leading positions, compute QR with no pivoting

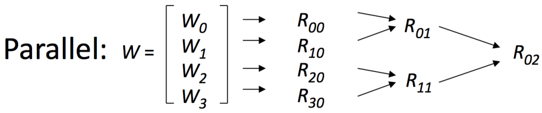
$$AP_{c1} = Q_1 \begin{pmatrix} R_{11} & * \\ & * \end{pmatrix}$$



Select k columns from a tall and skinny matrix

Given W of size $m \times 2k$, $m >> k$, k columns are selected as:

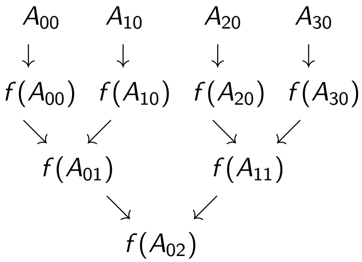
$W = QR_{02}$ using TSQR
$R_{02} P_c = Q_2 R_2$ using QRCP
Return $WP_c(:, 1:k)$



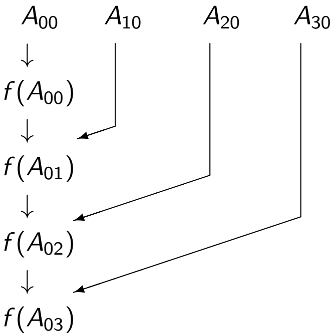
Reduction trees

Any shape of reduction tree can be used during CA_RRQR, depending on the underlying architecture.

- Binary tree:



- Flat tree:



Notation: at each node of the reduction tree, $f(A_{ij})$ returns the first b columns obtained after performing (strong) RRQR of A_{ij}.

Rank revealing properties of CA-RRQR

It is shown in [Demmel et al., 2015] that the column permutation computed by CA-RRQR satisfies

$$\chi_j^2\left(R_{11}^{-1}R_{12}\right) + \left(\chi_j\left(R_{22}\right)/\sigma_{\min}(R_{11})\right)^2 \leq F_{TP}^2, \text{ for } j = 1, \ldots, n-k. \qquad (6)$$

where F_{TP} depends on k, f, n, the shape of reduction tree used during tournament pivoting, and the number of iterations of CARRQR.

CA-RRQR - bounds for one tournament

Selecting k columns by using tournament pivoting reveals the rank of A with the following bounds:

$$1 \leq \frac{\sigma_i(A)}{\sigma_i(R_{11})}, \frac{\sigma_j(R_{22})}{\sigma_{k+j}(A)} \leq \sqrt{1 + F_{TP}^2(n-k)},$$

$$||R_{11}^{-1}R_{12}||_{max} \leq F_{TP}$$

- Binary tree of depth $\log_2(n/k)$,

$$F_{TP} \leq \frac{1}{\sqrt{2k}} (n/k)^{\log_2\left(\sqrt{2}fk\right)}. \quad (7)$$

  The upper bound is a decreasing function of k when $k > \sqrt{n/(\sqrt{2}f)}$.
- Flat tree of depth n/k,

$$F_{TP} \leq \frac{1}{\sqrt{2k}} \left(\sqrt{2}fk\right)^{n/k}. \quad (8)$$

Cost of CA-RRQR

Cost of CA-RRQR vs QR with column pivoting

$n \times n$ matrix on $\sqrt{P} \times \sqrt{P}$ processor grid, block size k

$$\begin{array}{llll} Flops: & 4n^3/P + O(n^2 klogP/\sqrt{P}) & vs & (4/3)n^3/P \\ Bandwidth: & O(n^2 \log P/\sqrt{P}) & vs & same \\ Latency: & O(n \log P/k) & vs & O(n \log P) \end{array}$$

Communication optimal, modulo polylogarithmic factors, by choosing

$$k = \frac{1}{2log^2 P} \frac{n}{\sqrt{P}}$$

Numerical results

- Stability close to QRCP for many tested matrices.
- Absolute value of diagonals of R, L referred to as R-values, L-values.
- Methods compared
  - RRQR: QR with column pivoting
  - CA-RRQR-B with tournament pivoting based on binary tree
  - CA-RRQR-F with tournament pivoting based on flat tree
  - SVD

Numerical results - devil's stairs

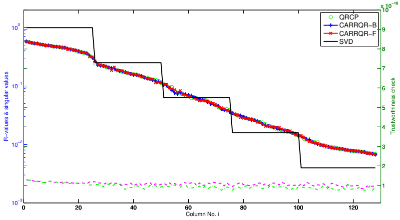
Devil's stairs (Stewart), a matrix with multiple gaps in the singular values.

Matlab code:

```
Length = 20; s = zeros(n,1); Nst = floor(n/Length);
for i = 1 : Nst do
        s(1+Length*(i-1):Length*i) = -0.6*(i-1);
end for
s(Length * Nst : end) = -0.6 * (Nst - 1);
s = 10. ∧ s;
A = orth(rand(n)) * diag(s) * orth(randn(n));
```

QLP decomposition (Stewart)

$$\begin{aligned} AP_{c_1} &= Q_1 R_1 \text{ using ca\_rrqr} \\ R_1^T &= Q_2 R_2 \end{aligned}$$



Numerical results - devil's stairs

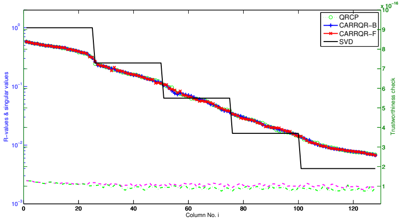
Devil's stairs (Stewart), a matrix with multiple gaps in the singular values.

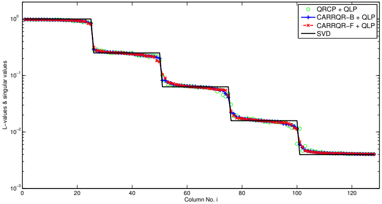
Matlab code:

Length = 20; s = zeros(n,1); Nst = floor(n/Length);
for i = 1 : Nst **do**
    s(1+Length*(i-1):Length*i) = -0.6*(i-1);
end for
$s(Length * Nst : end) = -0.6 * (Nst - 1);$
$s = 10. \wedge s;$
A = orth(rand(n)) * diag(s) * orth(randn(n));

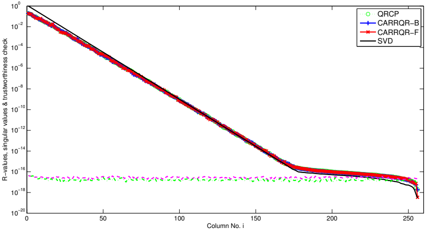
QLP decomposition (Stewart)

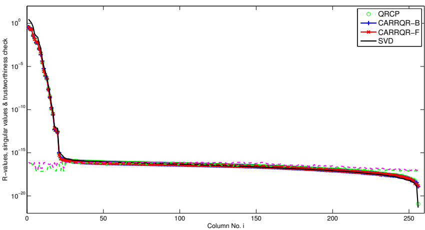
$$\begin{aligned} AP_{c_1} &= Q_1 R_1 \text{ using ca\_rrqr} \\ R_1^T &= Q_2 R_2 \end{aligned}$$





Numerical results (contd)





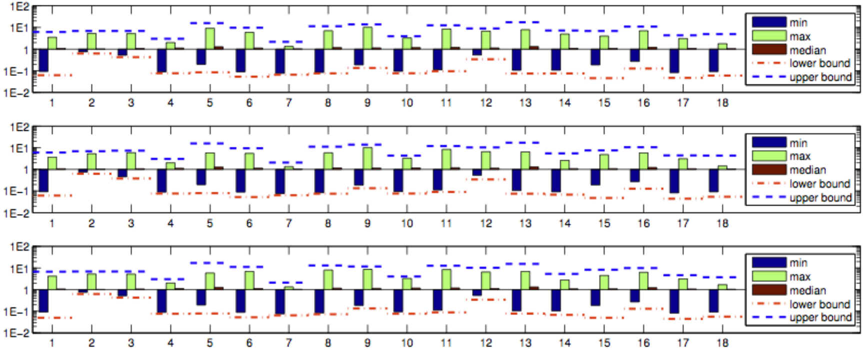
- Left: exponent - exponential Distribution, $\sigma_1 = 1$, $\sigma_i = \alpha^{i-1}$ $(i = 2, \ldots, n)$, $\alpha = 10^{-1/11}$ [Bischof, 1991]
- Right: shaw - 1D image restoration model [Hansen, 2007]

$$\epsilon \min\{||(A\Pi_0)(:,i)||_2\,, ||(A\Pi_1)(:,i)||_2\,, ||(A\Pi_2)(:,i)||_2\} \tag{9}$$

$$\epsilon \max\{||(A\Pi_0)(:,i)||_2\,, ||(A\Pi_1)(:,i)||_2\,, ||(A\Pi_2)(:,i)||_2\} \tag{10}$$

where $\Pi_j (j = 0, 1, 2)$ are the permutation matrices obtained by QRCP, CARRQR-B, and CARRQR-F, and ϵ is the machine precision.

Numerical results - a set of 18 matrices



- Ratios $|R(i,i)|/\sigma_i(R)$, for QRCP (top plot), CARRQR-B (second plot), and CARRQR-F (third plot).
- The number along x-axis represents the index of test matrices.

Plan

Low rank matrix approximation

Rank revealing QR factorization

LU_CRTP: Truncated LU factorization with column and row tournament pivoting

Experimental results, LU_CRTP

Randomized algorithms for low rank approximation

LU versus QR - filled graph $G^+(A)$

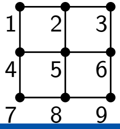
- Consider A is SPD and $A = LL^T$
- Given $G(A) = (V, E)$, $G^+(A) = (V, E^+)$ is defined as: there is an edge $(i, j) \in G^+(A)$ iff there is a path from i to j in $G(A)$ going through lower numbered vertices.
- $G(L + L^T) = G^+(A)$, ignoring cancellations.
- Definition holds also for directed graphs (LU factorization).

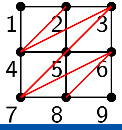
$A =$

	1	2	3	4	5	6	7	8	9
1	x	x		x					
2	x	x	x		x				
3		x	x			x			
4	x			x	x		x		
5		x		x	x	x		x	
6			x		x	x			x
7				x			x	x	
8					x		x	x	x
9						x		x	x

$L + L^T =$

	1	2	3	4	5	6	7	8	9
1	x	x		x					
2	x	x	x	x	x				
3		x	x	x	x	x			
4	x	x	x	x	x		x		
5		x	x	x	x	x	x	x	
6			x		x	x	x	x	x
7				x	x	x	x	x	
8					x	x	x	x	x
9						x		x	x





LU versus QR

Filled column intersection graph $G_\cap^+(A)$

- Graph of the Cholesky factor of $A^T A$
- $G(R) \subseteq G_\cap^+(A)$
- $A^T A$ can have many more nonzeros than A

LU versus QR

Numerical stability

- Let $\hat{L}$ and $\hat{U}$ be the computed factors of the block LU factorization. Then

$$\hat{L}\hat{U} = A + E, \quad \|E\|_{max} \leq c(n)\epsilon \left(\|A\|_{max} + \|\hat{L}\|_{max} \|\hat{U}\|_{max} \right). \tag{11}$$

- For partial pivoting, $\|L\|_{max} \leq 1$, $\|U\|_{max} \leq 2^n \|A\|_{max}$
  In practice, $\|U\|_{max} \leq \sqrt{n}\|A\|_{max}$

Low rank approximation based on LU factorization

- Given desired rank k, the factorization has the form

$$P_r A P_c = \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{pmatrix} = \begin{pmatrix} I & \\ \bar{A}_{21}\bar{A}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ & S(\bar{A}_{11}) \end{pmatrix}, \tag{12}$$

where $A \in \mathbb{R}^{m \times n}$, $\bar{A}_{11} \in \mathbb{R}^{k,k}$, $S(\bar{A}_{11}) = \bar{A}_{22} - \bar{A}_{21}\bar{A}_{11}^{-1}\bar{A}_{12}$.

- The rank-k approximation matrix $\tilde{A}_k$ is

$$\tilde{A}_k = \begin{pmatrix} I \\ \bar{A}_{21}\bar{A}_{11}^{-1} \end{pmatrix} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \end{pmatrix} = \begin{pmatrix} \bar{A}_{11} \\ \bar{A}_{21} \end{pmatrix} \bar{A}_{11}^{-1} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \end{pmatrix}. \tag{13}$$

- $\bar{A}_{11}^{-1}$ is never formed, its factorization is used when $\tilde{A}_k$ is applied to a vector.
- In randomized algorithms, $U = C^+ A R^+$, where C^+, R^+ are Moore-Penrose generalized inverses.

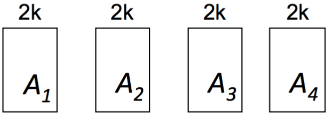
Design space

Non-exhaustive list for selecting k columns and rows:

1. Select k linearly independent columns of A (call result B), by using
   1.1 (strong) QRCP/tournament pivoting using QR,
   1.2 LU / tournament pivoting based on LU, with some form of pivoting (column, complete, rook),
   1.3 randomization: premultiply $X = ZA$ where random matrix Z is short and fat, then pick k rows from X^T, by some method from 2) below,
   1.4 tournament pivoting based on randomized algorithms to select columns at each step.
2. Select k linearly independent rows of B, by using
   2.1 (strong) QRCP / tournament pivoting based on QR on B^T, or on Q^T, the rows of the thin Q factor of B,
   2.2 LU / tournament pivoting based on LU, with pivoting (row, complete, rook) on B,
   2.3 tournament pivoting based on randomized algorithms to select rows.

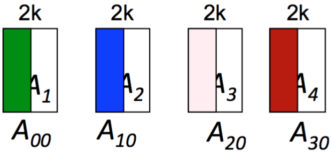
Select k cols using tournament pivoting

- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$, by using QR with column pivoting
- Return columns in A_{ji}



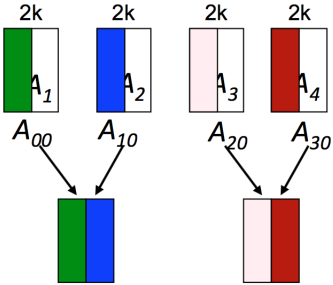
Select k cols using tournament pivoting

- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$, by using QR with column pivoting
- Return columns in A_{ji}



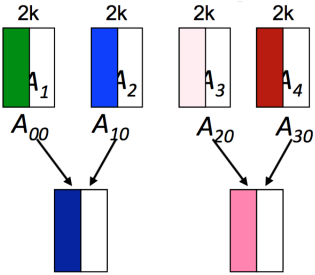
Select k cols using tournament pivoting

- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$, by using QR with column pivoting
- Return columns in A_{ji}



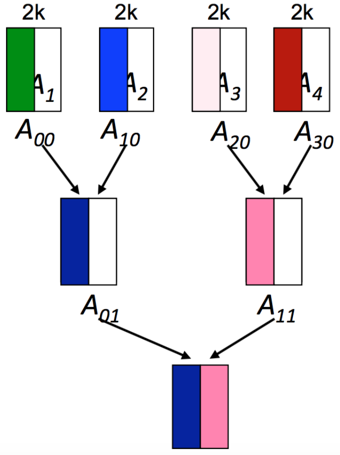
Select k cols using tournament pivoting

- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$, by using QR with column pivoting
- Return columns in A_{ji}



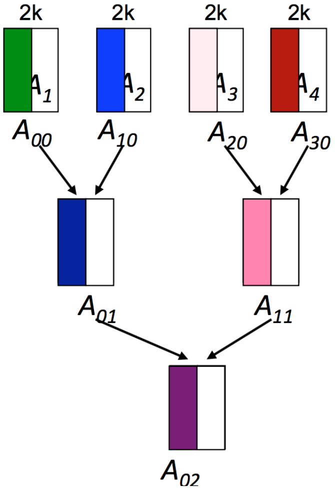
Select k cols using tournament pivoting

- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$, by using QR with column pivoting
- Return columns in A_{ji}



Select k cols using tournament pivoting

- Partition $A = (A_1, A_2, A_3, A_4)$.
- Select k cols from each column block, by using QR with column pivoting
- At each level i of the tree
  - At each node j do in parallel
    - Let $A_{v,i-1}, A_{w,i-1}$ be the cols selected by the children of node j
    - Select k cols from $(A_{v,i-1}, A_{w,i-1})$, by using QR with column pivoting
- Return columns in A_{ji}



LU_CRTP factorization - one block step

One step of truncated block LU based on column/row tournament pivoting on matrix A of size $m \times n$:

1. Select k columns by using tournament pivoting, permute them in front, bounds for s.v. governed by $q_1(n,k)$

$$AP_c = Q \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix} = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix}$$

2. Select k rows from $(Q_{11};\ Q_{21})^T$ of size $m \times k$ by using tournament pivoting,

$$P_r Q = \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ \bar{Q}_{21} & \bar{Q}_{22} \end{pmatrix}$$

such that $||\bar{Q}_{21}\bar{Q}_{11}^{-1}||_{max} \leq F_{TP}$ and bounds for s.v. governed by $q_2(m,k)$.

Orthogonal matrices

Given orthogonal matrix $Q \in \mathbb{R}^{m \times m}$ and its partitioning

$$Q = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix}, \tag{14}$$

the selection of k cols by tournament pivoting from $(Q_{11}; Q_{21})^T$ leads to the factorization

$$P_r Q = \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ \bar{Q}_{21} & \bar{Q}_{22} \end{pmatrix} = \begin{pmatrix} I & \\ \bar{Q}_{21}\bar{Q}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ & S(\bar{Q}_{11}) \end{pmatrix} \tag{15}$$

where $S(\bar{Q}_{11}) = \bar{Q}_{22} - \bar{Q}_{21}\bar{Q}_{11}^{-1}\bar{Q}_{12} = \bar{Q}_{22}^{-T}$.

Orthogonal matrices (contd)

The factorization

$$P_r Q = \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ \bar{Q}_{21} & \bar{Q}_{22} \end{pmatrix} = \begin{pmatrix} I & \\ \bar{Q}_{21}\bar{Q}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ & S(\bar{Q}_{11}) \end{pmatrix} \quad (16)$$

satisfies:

$$\rho_j(\bar{Q}_{21}\bar{Q}_{11}^{-1}) \leq F_{TP}, \quad (17)$$

$$\frac{1}{q_2(m,k)} \leq \sigma_i(\bar{Q}_{11}) \leq 1, \quad (18)$$

$$\sigma_{min}(\bar{Q}_{11}) = \sigma_{min}(\bar{Q}_{22}) \quad (19)$$

for all $1 \leq i \leq k$, $1 \leq j \leq m-k$, where $\rho_j(A)$ is the 2-norm of the j-th row of A, $q_2(m,k) = \sqrt{1 + F_{TP}^2(m-k)}$.

Sketch of the proof

$$
\begin{aligned}
P_r A P_c &= \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{pmatrix} = \begin{pmatrix} I & \\ \bar{A}_{21}\bar{A}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ & S(\bar{A}_{11}) \end{pmatrix} \\
&= \begin{pmatrix} I & \\ \bar{Q}_{21}\bar{Q}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{Q}_{11} & \bar{Q}_{12} \\ & S(\bar{Q}_{11}) \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ & R_{22} \end{pmatrix}
\end{aligned} \tag{20}
$$

where

$$
\begin{aligned}
\bar{Q}_{21}\bar{Q}_{11}^{-1} &= \bar{A}_{21}\bar{A}_{11}^{-1}, \\
S(\bar{A}_{11}) &= S(\bar{Q}_{11})R_{22} = \bar{Q}_{22}^{-T} R_{22}.
\end{aligned}
$$

Sketch of the proof (contd)

$$\bar{A}_{11} = \bar{Q}_{11} R_{11}, \tag{21}$$

$$S(\bar{A}_{11}) = S(\bar{Q}_{11}) R_{22} = \bar{Q}_{22}^{-T} R_{22}. \tag{22}$$

We obtain

$$\sigma_i(A) \geq \sigma_i(\bar{A}_{11}) \geq \sigma_{min}(\bar{Q}_{11}) \sigma_i(R_{11}) \geq \frac{1}{q_1(n,k) q_2(m,k)} \sigma_i(A),$$

We also have that

$$\begin{aligned} \sigma_{k+j}(A) \leq \sigma_j(S(\bar{A}_{11})) &= \sigma_j(S(\bar{Q}_{11}) R_{22}) \leq ||S(\bar{Q}_{11})||_2 \sigma_j(R_{22}) \\ &\leq q_1(n,k) q_2(m,k) \sigma_{k+j}(A), \end{aligned}$$

where $q_1(n,k) = \sqrt{1 + F_{TP}^2(n-k)}$, $q_2(m,k) = \sqrt{1 + F_{TP}^2(m-k)}$.

LU_CRTP factorization - bounds if $rank = k$

Given A of size $m \times n$, one step of LU_CRTP computes the decomposition

$$\bar{A} = P_r A P_c = \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{pmatrix} = \begin{pmatrix} I & \\ \bar{Q}_{21}\bar{Q}_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ & S(\bar{A}_{11}) \end{pmatrix} \tag{23}$$

where $\bar{A}_{11}$ is of size $k \times k$ and

$$S(\bar{A}_{11}) = \bar{A}_{22} - \bar{A}_{21}\bar{A}_{11}^{-1}\bar{A}_{12} = \bar{A}_{22} - \bar{Q}_{21}\bar{Q}_{11}^{-1}\bar{A}_{12}. \tag{24}$$

It satisfies the following properties:

$$\begin{aligned} \rho_l(\bar{A}_{21}\bar{A}_{11}^{-1}) &= \rho_l(\bar{Q}_{21}\bar{Q}_{11}^{-1}) \le F_{TP}, \qquad (25) \\ \|S(\bar{A}_{11})\|_{max} &\le \min((1 + F_{TP}\sqrt{k})\|A\|_{max}, F_{TP}\sqrt{1 + F_{TP}^2(m-k)}\sigma_k(A)) \end{aligned}$$

$$1 \le \frac{\sigma_i(A)}{\sigma_i(\bar{A}_{11})}, \frac{\sigma_j(S(\bar{A}_{11}))}{\sigma_{k+j}(A)} \le q(m,n,k), \tag{26}$$

for any $1 \le l \le m-k$, $1 \le i \le k$, and $1 \le j \le \min(m,n) - k$,
$q(m,n,k) = \sqrt{(1 + F_{TP}^2(n-k))(1 + F_{TP}^2(m-k))}$.

LU_CRTP factorization - bounds if $rank = K = Tk$

Consider T block steps of LU_CRTP factorization

$$P_r A P_c = \begin{pmatrix} I & & & & \\ L_{21} & I & & & \\ \vdots & \vdots & \ddots & & \\ L_{T1} & L_{T2} & \dots & I & \\ L_{T+1,1} & L_{T+1,2} & \dots & L_{T+1,T} & I \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} & \dots & U_{1T} & U_{1,T+1} \\ & U_{22} & \dots & U_{2T} & U_{2,T+1} \\ & & \ddots & \vdots & \vdots \\ & & & U_{TT} & U_{T,T+1} \\ & & & & U_{T+1,T+1} \end{pmatrix} \quad (2$$

where U_{tt} is $k \times k$ for $1 \leq t \leq T$, and $U_{T+1,T+1}$ is $(m - Tk) \times (n - Tk)$. Then:

$$\begin{aligned} \rho_l(L_{i+1,j}) &\leq F_{TP}, \\ ||U_K||_{max} &\leq \min\left((1 + F_{TP}\sqrt{k})^{K/k}||A||_{max}, q_2(m,k)q(m,n,k)^{K/k-1}\sigma_K(A)\right), \end{aligned}$$

for any $1 \leq l \leq k$. $q_2(m,k) = \sqrt{1 + F_{TP}^2(m-k)}$, and
$q(m,n,k) = \sqrt{(1 + F_{TP}^2(n-k))\,(1 + F_{TP}^2(m-k))}$.

LU_CRTP factorization - bounds if $rank = K = Tk$

Consider $T = K/k$ block steps of our LU_CRTP factorization

$$P_r A P_c = \begin{pmatrix} I & & & & \\ L_{21} & I & & & \\ \vdots & \vdots & \ddots & & \\ L_{T1} & L_{T2} & \dots & I & \\ L_{T+1,1} & L_{T+1,2} & \dots & L_{T+1,T} & I \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} & \dots & U_{1T} & U_{1,T+1} \\ & U_{22} & \dots & U_{2T} & U_{2,T+1} \\ & & \ddots & \vdots & \vdots \\ & & & U_{TT} & U_{T,T+1} \\ & & & & U_{T+1,T+1} \end{pmatrix} \quad (2$$

where U_{tt} is $k \times k$ for $1 \leq t \leq T$, and $U_{T+1,T+1}$ is $(m - Tk) \times (n - Tk)$. Then:

$$\frac{1}{\prod_{v=0}^{t-2} q(m - vk, n - vk, k)} \leq \frac{\sigma_{(t-1)k+i}(A)}{\sigma_i(U_{tt})} \leq q(m - (t-1)k, n - (t-1)k, k),$$

$$1 \leq \frac{\sigma_j(U_{T+1,T+1})}{\sigma_{K+j}(A)} \leq \prod_{v=0}^{K/k-1} q(m - vk, n - vk, k),$$

for any $1 \leq i \leq k$, $1 \leq t \leq T$, and $1 \leq j \leq \min(m, n) - K$. Here
$q_2(m, k) = \sqrt{1 + F_{TP}^2(m - k)}$, and
$q(m, n, k) = \sqrt{(1 + F_{TP}^2(n - k))\,(1 + F_{TP}^2(m - k))}$.

Arithmetic complexity - arbitrary sparse matrices

- Let d_i be the number of nonzeros in column i of A, $nnz(A) = \sum_{i=1}^{n} d_i$.
- A is permuted such that $d_1 \leq \ldots \leq d_n$.
- $A = [A_{00}, \ldots, A_{n/k,0}]$ is partitioned into n/k blocks of columns.

At first step of TP:

- Pick k cols from $A_1 = [A_{00}, A_{10}]$
  $nnz(A_1) \leq 2k \sum_{i=1}^{2k} d_i$,
  $flops_{QR}(A_1) \leq 8k^2 \sum_{i=1}^{2k} d_i$.

At the second step of TP:

- Pick k cols from A_2
  $nnz(A_2) \leq 2k \sum_{i=k+1}^{3k} d_i$
  $flops_{QR}(A_2) \leq 8k^2 \sum_{i=k+1}^{3k} d_i$

Bounds attained when:

$$A = \begin{bmatrix} * & 0 & & 0 \\ \vdots & \vdots & & \vdots \\ * & 0 & & 0 \\ 0 & * & & 0 \\ \vdots & \vdots & & \vdots \\ 0 & * & & 0 \\ & & \ddots & \\ 0 & 0 & & * \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & * \end{bmatrix}$$

Arithmetic complexity - arbitrary sparse matrices (2)

$$
\begin{aligned}
nnz_{max}(TP_{FT}) &\leq 4 d_n k^2 \\
nnz_{total}(TP_{FT}) &\leq 2k \left(\sum_{i=1}^{2k} d_i + \sum_{i=k+1}^{3k} d_i + \ldots + \sum_{i=n-2k+1}^{n} d_i \right) \leq \\
&\leq 4k \sum_{i=1}^{n} d_i = 4 nnz(A) k, \\
flops(TP_{FT}) &\leq 16 nnz(A) k^2,
\end{aligned}
$$

Tournament pivoting for sparse matrices

Arithmetic complexity

A has arbitrary sparsity structure

$$flops(TP_{FT}) \leq 16 nnz(A) k^2$$
$$flops(TP_{BT}) \leq 8 \frac{nnz(A)}{P} k^2 \log \frac{n}{k}$$

$G(A^T A)$ is an $n^{1/2}$- separable graph

$$flops(TP_{FT}) \leq O(nnz(A) k^{3/2})$$
$$flops(TP_{BT}) \leq O(\frac{nnz(A)}{P} k^{3/2} \log \frac{n}{k})$$

Randomized algorithm by Clarkson and Woodruff, STOC'13

- Given $n \times n$ matrix A, it computes LDW^T, where D is $k \times k$ such that with failure probability $1/10$ $||A - LDW^T||_F \leq (1 + \epsilon)||A - A_k||_F$, A_k is best rank-k approximation.
- The cost of this algorithm is

$$flops \leq O(nnz(A)) + nk^2\epsilon^{-4} log^{O(1)}(nk^2\epsilon^{-4})$$

- Tournament pivoting is faster if $\epsilon \leq \frac{1}{(nnz(A)/n)^{1/4}}$ or if $\epsilon = 0.1$ and $nnz(A)/n < 10^4$.

Tournament pivoting for sparse matrices

Arithmetic complexity

A has arbitrary sparsity structure

$$flops(TP_{FT}) \leq 16nnz(A)k^2$$
$$flops(TP_{BT}) \leq 8\frac{nnz(A)}{P}k^2 \log \frac{n}{k}$$

$G(A^TA)$ is an $n^{1/2}$- separable graph

$$flops(TP_{FT}) \leq O(nnz(A)k^{3/2})$$
$$flops(TP_{BT}) \leq O(\frac{nnz(A)}{P}k^{3/2} \log \frac{n}{k})$$

Randomized algorithm by Clarkson and Woodruff, STOC'13

- Given $n \times n$ matrix A, it computes LDW^T, where D is $k \times k$ such that with failure probability $1/10$ $||A - LDW^T||_F \leq (1+\epsilon)||A - A_k||_F$, A_k is best rank-k approximation.
- The cost of this algorithm is

$$flops \leq O(nnz(A)) + nk^2\epsilon^{-4} log^{O(1)}(nk^2\epsilon^{-4})$$

- Tournament pivoting is faster if $\epsilon \leq \frac{1}{(nnz(A)/n)^{1/4}}$ or if $\epsilon = 0.1$ and $nnz(A)/n < 10^4$.

Plan

Low rank matrix approximation

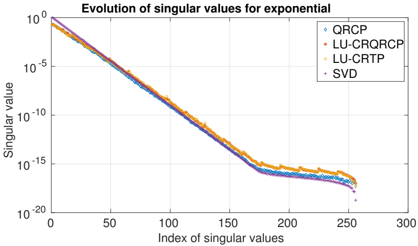
Rank revealing QR factorization

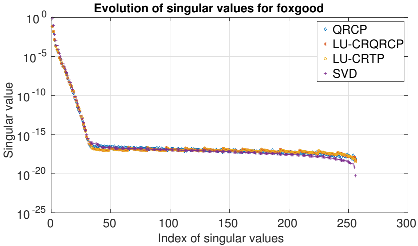
LU_CRTP: Truncated LU factorization with column and row tournament pivoting

Experimental results, LU_CRTP

Randomized algorithms for low rank approximation

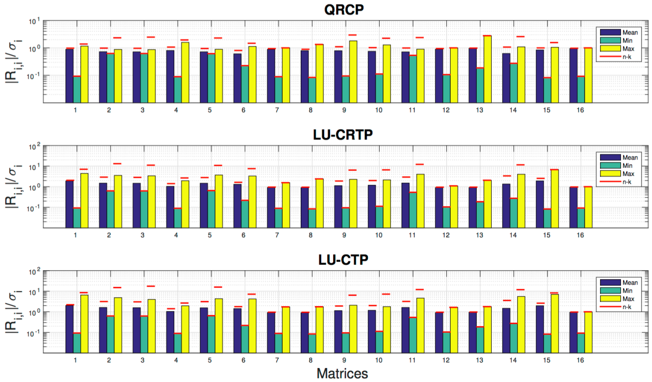
Numerical results





- Left: exponent - exponential Distribution, $\sigma_1 = 1$, $\sigma_i = \alpha^{i-1}$ $(i = 2, \ldots, n)$, $\alpha = 10^{-1/11}$ [Bischof, 1991]
- Right: foxgood - Severely ill-posed test problem of the 1st kind Fredholm integral equation used by Fox and Goodwin

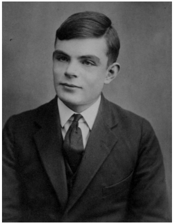
Numerical results



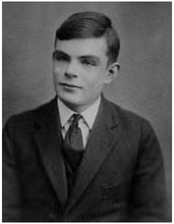
- Here $k = 16$ and the factorization is truncated at $K = 128$ (bars) or $K = 240$ (red lines).
- LU_CTP: Column tournament pivoting + partial pivoting
- All singular values smaller than machine precision, ϵ, are replaced by ϵ.
- The number along x-axis represents the index of test matrices.

Results for image of size 919 × 707

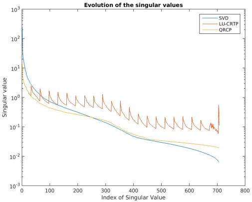
Original image



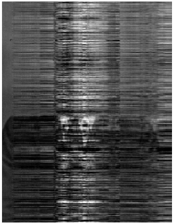
Rank-38 approx, SVD



Singular value distribution

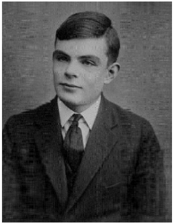


Rank-38 approx, LUPP



Rank-38 approx, LU_CRTP

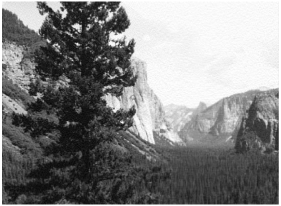
Rank-75 approx, LU_CRTP



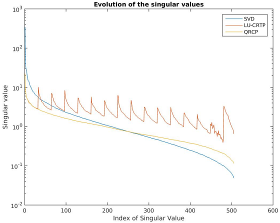
Results for image of size 691 $\times$ 505

Original image

Rank-105 approx, SVD

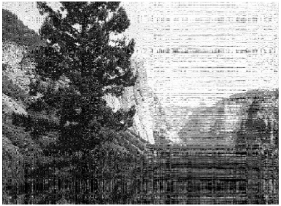


Singular value distribution

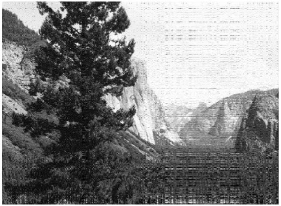


Rank-105 approx, LUPP

Rank-105 approx, LU_CRTP



Rank-209 approx, LU_CRTP



Comparing nnz in the factors L, U versus Q, R

Name/size	*Nnz* $A(:, 1:K)$	*Rank* K	*Nnz QRCP*/ *Nnz LU_CRTP*	*Nnz LU_CRTP*/ *Nnz LUPP*
*gemat*11	1232	128	2.1	2.2
4929	4895	512	3.3	2.6
	9583	1024	11.5	3.2
*wang*3	896	128	3.0	2.1
26064	3536	512	2.9	2.1
	7120	1024	2.9	1.2
Rfdevice	633	128	10.0	1.1
74104	2255	512	82.6	0.9
	4681	1024	207.2	0.0
Parab_fem	896	128	–	0.5
525825	3584	512	–	0.3
	7168	1024	–	0.2
Mac_econ	384	128	–	0.3
206500	1535	512	–	0.3
	5970	1024	–	0.2

Performance results

Selection of 256 columns by tournament pivoting

- Edison, Cray XC30 (NERSC): 2x12-core Intel Ivy Bridge (2.4 GHz)
- Tournament pivoting uses SPQR (T. Davis) + dGEQP3 (Lapack), time in secs

Matrices: dimension at leaves on 32 procs

- Parab_fem: 528825×528825 — 528825×16432
- Mac_econ: 206500×206500 — 206500×6453

	Time 2k cols	*Time leaves 32procs SPQR + dGEQP3*	*Number of MPI processes* 16	32	64	128	256	512	1024
Parab_fem	0.26	0.26 + 1129	46.7	24.5	13.7	8.4	5.9	4.8	4.4
Mac_econ	0.46	25.4 + 510	132.7	86.3	111.4	59.6	27.2	–	–

Plan

Randomized algorithms - main idea

- Construct a low dimensional subspace that captures the action of A.
- Restrict A to the subspace and compute a standard QR or SVD factorization.

Obtained as follows:

1. Compute an approximate basis for the range of A ($m \times n$)
   find Q ($m \times k$) with orthonormal columns and approximate A by the projection of its columns onto the space spanned by Q:

$$A \approx QQ^T A$$

2. Use Q to compute a standard factorization of A

Source: Halko et al, *Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decomposition*, SIREV 2011.

Why a random projection works

Johnson-Lindenstrauss Lemma

For any $0 < \epsilon < 1$, and any set of vectors $x_1, .., .x_n$ in $\mathbb{R}^m$, let $k \geq 4(\epsilon^2/2 - \epsilon^3/3)^{-1} ln(n)$. Let F be a random kxm orthogonal matrix multiplied by $\sqrt{m/k}$. Then with probability at $1/n$, for all $1 <= i, j <= n$

$$(1-\epsilon)||x_i - x_j||^2 <= ||F(x_i - x_j)||^2 <= (1+\epsilon)||x_i - x_j||^2$$

- Any m-vector can be embedded in $k = O(\log(n)/\epsilon^2)$ dimensions while incurring a distortion of at most $1 \pm \epsilon$ between all pairs of m-vectors.
- JL relies on F being uniformly distributed random orthonormal matrix.
- Such an F can be obtained by computing the QR factorization of an $m \times k$ matrix of i.i.d. $N(0, 1)$ random variables.

Source: Theorem 2.1 and proof in S. Dasgupta, A. Gupta, 2003, *An Elementary Proof of a Theorem of Johnson and Lindenstrauss*

Typical randomized truncated SVD

Algorithm

Input: $m \times n$ matrix A, desired rank k, $l = p + k$ exponent q.

1. Sample an $n \times l$ test matrix G with independent mean-zero, unit-variance Gaussian entries.
2. Compute $Y = (AA^T)^q AG$ /* Y is expected to span the column space of A */
3. Construct $Q \in \mathbb{R}^{m \times l}$ with columns forming an orthonormal basis for the range of Y.
4. Compute $B = Q^T A$
5. Compute the SVD of $B = \hat{U} \Sigma V^T$

Return the approximation $\tilde{A}_k = Q\hat{U} \cdot \Sigma \cdot V^T$

Randomized truncated SVD ($q = 0$)

The best approximation is when Q equals the first $k + p$ left singular vectors of A. Given $A = U\Sigma V^T$,

$$\begin{aligned} QQ^T A &= U(1:m, 1:k+p)\Sigma(1:k+p, 1:k+p)(V(1:n, 1:k+p) \\ ||A - QQ^T A||_2 &= \sigma_{k+p+1} \end{aligned}$$

Theorem 1.1 from Halko et al. If G is chosen to be i.i.d. N(0,1), $k, p \geq 2$, $q = 1$, then the expectation with respect to the random matrix G is

$$\mathbb{E}(||A - QQ^T A||_2) \leq \left(1 + \frac{4\sqrt{k+p}}{p-1}\sqrt{min(m,n)}\right)\sigma_{k+1}(A)$$

and the probability that the error satisfies

$$||A - QQ^T A||_2 \leq \left(1 + 11\sqrt{k+p} \cdot \sqrt{min(m,n)}\right)\sigma_{k+1}(A)$$

is at least $1 - 6/p^p$.

For $p = 6$, the probability becomes .99.

Randomized truncated SVD

Theorem 10.6, Halko et al. Average spectral norm. Under the same hypotheses as Theorem 1.1 from Halko et al.,

$$\mathbb{E}(||A - QQ^T A||_2) \leq \left(1 + \sqrt{\frac{k}{p-1}}\right) \sigma_{k+1}(A) + \frac{e\sqrt{k+p}}{p} \left(\sum_{j=k+1}^{n} \sigma_j^2(A)\right)^{1/2}$$

- Fast decay of singular values:
  If $\left(\sum_{j>k} \sigma_j^2(A)\right)^{1/2} \approx \sigma_{k+1}$ then the approximation should be accurate.
- Slow decay of singular values:
  If $\left(\sum_{j>k} \sigma_j^2(A)\right)^{1/2} \approx \sqrt{n-k}\sigma_{k+1}$ and n large, then the approximation might not be accurate.

Source: G. Martinsson's talk

Power iteration $q \geq 1$

The matrix $(AA^T)^q A$ has a faster decay in its singular values:

- has the same left singular vectors as A
- its singular values are:

$$\sigma_j((AA^T)^q A) = (\sigma_j(A))^{2q+1}$$

Cost of randomized truncated SVD

- Randomized SVD requires $2q+1$ passes over the matrix.
- The last 3 steps of the algorithms cost:
  (2) Compute $Y = (AA^T)^q AG$: $2(2q+1) \cdot nnz(A) \cdot (k+p)$
  (3) Compute QR of Y: $2m(k+p)^2$
  (4) Compute $B = Q^T A$: $2nnz(A) \cdot (k+p)$
  (5) Compute SVD of B: $O(n(k+p)^2)$
- If $nnz(A)/m \geq k+p$ and $q=1$, then (2) and (4) dominate (3).
- To be faster than deterministic approaches, the cost of (2) and (4) need to be reduced.

Fast Johnson-Lindenstrauss transform

Find sparse or structured G such that computing AG is cheap, e.g. a subsampled random Fourier trasnform (SRFT),

$$G = \sqrt{\frac{n}{k+p}} D \times F \times S, \text{ where}$$

- D is $n \times n$ diagonal with entries uniformly distributed on unit circle in $\mathbb{C}$
- F is $n \times n$ discrete Fourier transform, $F_{jk} = \frac{1}{\sqrt{n}} e^{-2\pi i (j-1)(k-1)/n}$
- S is $n \times (k+p)$ random subset of the columns of the identity (draws $k+p$ columns at random from DF).

Computational cost

(2) Compute AG in $O(mn \log(n))$ or $O(mn \log(k+p))$ via a subsampled FFT
(4) Compute $B = Q^T A$ still expensive ! – can be reduced by row sampling

References: Ailon and Chazelle (2006), Liberty, Rokhlin, Tygert and Woolfe (2006).

Summary of computation cost

Dense matrix A of size $m \times n$

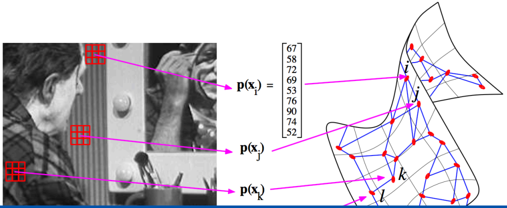
- QR with column pivoting: $4mnk$
- Randomized SVD with a Gaussian matrix: $O(mnk)$
- Randomized SVD with an SRFT: $O(mn\log(k))$

Results from image processing (from Halko et al)

- A matrix A of size 9025×9025 arising from a diffusion geometry approach.
- A is a graph Lapacian on the manifold of 3×3 patches.
- 95×95 pixel grayscale image, intensity of each pixel is an integer ≤ 4095.
- Vector $x^{(i)} \in \mathbb{R}^9$ gives the intensities of the pixels in a 3×3 neighborhood of pixel i.
- W reflects similarities between patches, $\sigma = 50$ reflects the level of sensitivity,

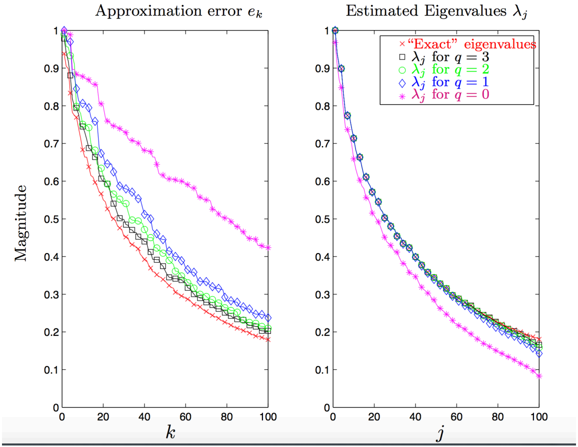
$$w_{ij} = exp\{-||x^{(i)} - x^{(j)}||^2/\sigma^2\},$$

- Sparsify W, compute dominant eigenvectors of $A = D^{-1/2} W D^{-1/2}$.



Experimental results (from Halko et al)

- Approximation error : $||A - QQ^T A||_2$
- Estimated eigenvalues for $k = 100$



Clarksson and Woodruff, STOC 2013

- Based on randomized sparse embedding
- Let S, of size $poly(k/\epsilon) \times n$ be formed such that each column has one non-zero, ± 1, randomly chosen

$$S = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- Given A of size $n \times n$ and rank k, for certain $poly(k/\epsilon)$, with probability at least $9/10$, the column space of A is preserved, that is for all $x \in \mathbb{R}^n$,

$$||SAx||_2 = (1 \pm \epsilon)||Ax||_2$$

- SA can be computed in $nnz(A)$ time

Source: Woodruff's talk, STOC 2013

Clarksson and Woodruff, STOC 2013

Main idea

- Let A be an $n \times n$ matrix
  S be an $v \times n$ sparse embedding matrix, $v = \Theta(\epsilon^{-4} k^2 \log^6(k/\epsilon))$
  R an $t \times n$ sparse embedding matrix, $t = O(k\epsilon^{-1} \log(k/\epsilon))$

$$A' = AR^T (SAR^T)^{-1} SA$$

- Extract low rank approximation from A'
- More details in Theorem 47 from STOC 2013
- Theorem 47 relies on S and R being the product of a sparse embedding and a SRHT matrix

Clarkson and Woodruff, STOC 2013

- Given $n \times n$ matrix A, it computes LDW^T, where D is $k \times k$ such that with failure probability $1/10$
  $||A - LDW^T||_F \leq (1+\epsilon)||A - A_k||_F$, A_k is best rank-k approximation.

  $$flops \leq O(nnz(A)) + (nk^2\epsilon^{-4} + k^3\epsilon^{-5})log^{O(1)}(nk^2\epsilon^{-4} + k^3\epsilon^{-5})$$

More details on CA deterministic algorithms

- [Demmel et al., 2015] Communication avoiding rank revealing QR factorization with column pivoting Demmel, Grigori, Gu, Xiang, SIAM J. Matrix Analysis and Applications, 2015.
- Low rank approximation of a sparse matrix based on LU factorization with column and row tournament pivoting, with S. Cayrols and J. Demmel, Inria TR 8910.

References (1)

Bischof, C. H. (1991).
A parallel QR factorization algorithm with controlled local pivoting.
SIAM J. Sci. Stat. Comput., 12:36–57.

Businger, P. A. and Golub, G. H. (1965).
Linear least squares solutions by Householder transformations.
Numer. Math., 7:269–276.

Demmel, J., Grigori, L., Gu, M., and Xiang, H. (2015).
Communication-avoiding rank-revealing qr decomposition.
SIAM Journal on Matrix Analysis and its Applications, 36(1):55–89.

Eckart, C. and Young, G. (1936).
The approximation of one matrix by another of lower rank.
Psychometrika, 1:211–218.

Eisenstat, S. C. and Ipsen, I. C. F. (1995).
Relative perturbation techniques for singular value problems.
SIAM J. Numer. Anal., 32(6):1972–1988.

Gu, M. and Eisenstat, S. C. (1996).
Efficient algorithms for computing a strong rank-revealing QR factorization.
SIAM J. Sci. Comput., 17(4):848–869.

Hansen, P. C. (2007).
Regularization tools: A matlab package for analysis and solution of discrete ill-posed problems.
Numerical Algorithms, (46):189–194.

Results used in the proofs

- Interlacing property of singular values [Golub, Van Loan, 4th edition, page 487]
  Let $A = [a_1| \ldots |a_n]$ be a column partitioning of an $m \times n$ matrix with $m \geq n$. If $A_r = [a_1| \ldots |a_r]$, then for $r = 1 : n-1$

$$\sigma_1(A_{r+1}) \geq \sigma_1(A_r) \geq \sigma_2(A_{r+1}) \geq \ldots \geq \sigma_r(A_{r+1}) \geq \sigma_r(A_r) \geq \sigma_{r+1}(A_{r+1}).$$

- Given $n \times n$ matrix B and $n \times k$ matrix C, then ([Eisenstat and Ipsen, 1995], p. 1977)

$$\sigma_{min}(B)\sigma_j(C) \leq \sigma_j(BC) \leq \sigma_{max}(B)\sigma_j(C), j = 1, \ldots, k.$$