Rank-revealing QR factorizations

неполный QR

(Кирилл Кулев, Батраев Радик)

Содержание

- Постановка задачи RRF (rank-revealing factorization)
- Алгоритмы
  - pivot QR
  - random QR
  - pivot LU
- Оценки (реализация, скорость, точность)

Постановка задачи

- Обычно требуется факторизация, обеспечивающая приближение к пространству диапазонов матрицы более низкого ранга.
- SVD - отличный кандидат
- Однако обновление SVD обходится дорого -добавление строки\столбца в матрицу или удаляется из нее.

Вычисление rank-revealing factorization:

$$A = XDY^T, \quad X \in \mathbb{R}^{m \times p}, \quad D \in \mathbb{R}^{p \times p}, \quad Y \in \mathbb{R}^{n \times p}, \quad p \leq min(m,n), \; D - diagonal$$

Rank revealing QR factorization

Дано LU разложение: $AP = QR = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ 0 & R22 \end{pmatrix}$

И оценки: $\sigma_{\max}(R_{22}) \leq f(k,n)\sigma_{k+1}(A), \quad \sigma_{\min}(R_{11}) \geq f(k,n)^{-1}\sigma_k(A),$

В зависимости стратегии выбора P имеем разные алгоритмы:

- Детерминированные
- Рандомизированные

В условиях изначальной задачи вычисляем:

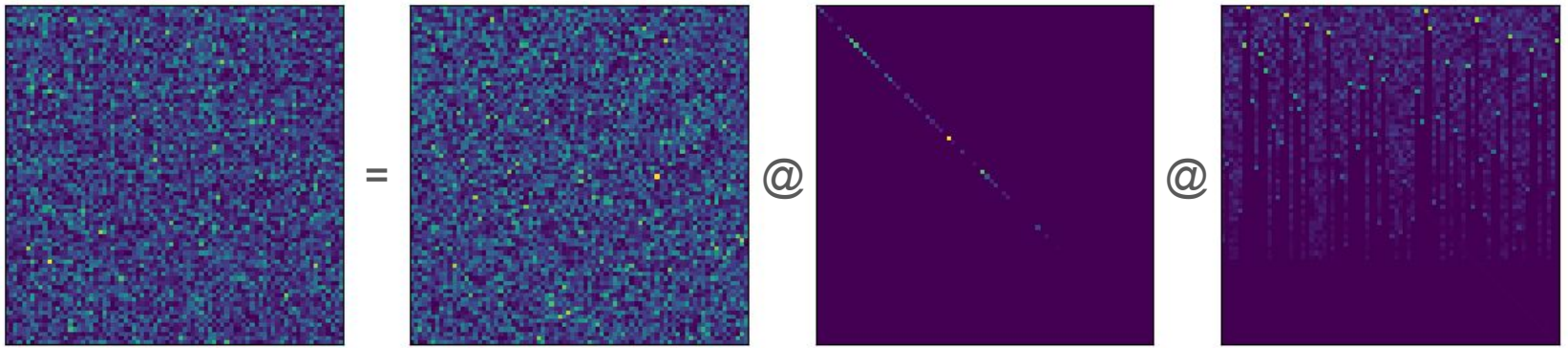
$$A = QRP^T = QDY^T,$$
$$D = diag(r_{ii}),\ Y^T = D^{-1}RP^T$$

QR with column pivoting

- На каждом шаге вычисляются нормы оставшихся столбцов
- Столбец с максимальной нормой переставляется в начало

Algorithm 1 QR with column pivoting (QRCP)

Inputs:
A is $m \times n$ matrix, k is target rank, $1 \leq k \leq \min(m, n)$
Outputs:
Q is $m \times m$ orthogonal matrix
R is $m \times n$ upper trapezoidal matrix
Π is $n \times n$ permutation matrix such that $A\Pi = QR$
Algorithm:
Initialize $\Pi^{(0)} = I_n$, $r_s = ||A(1:m, s)||_2 \ \ (1 \leq s \leq n)$
for $i = 1 : k$ **do**
  Find $j = argmax_{i \leq s \leq n} r_s$
  Swap r_i and r_j, $A(1:m, i)$ and $A(1:m, j)$
  Update permutation with last swap $\Pi^{(i)} = \Pi^{(i-1)}\Pi_{i,j}$
  Form Householder reflection H_i from $A(i:m, i)$
  Update $A(i:m, i:n) \leftarrow H_i A(i:m, i:n)$
  Update $r_s = ||A(i+1:m, s)||_2 \quad (i+1 \leq s \leq n)$
end for
$Q = H_1 H_2 \cdots H_k$ is the product of all reflections
R = upper trapezoidal part of A, $\Pi = \Pi^{(k)}$



A (80x80), rank=60

Randomized QR with column pivoting

- Генерируется случайная матрица для понижения размерности
- Вычисляется QRCP для отбора pivot'ов
- Вычисляется обычное QR на исходной матрице

Algorithm 3 SSRQRCP - Single-Sample Randomized QRCP.

Input:

A is $m \times n$.

k the desired approximation rank. $k \ll \min(m, n)$.

Output:

Q is $m \times m$ orthogonal matrix in the form of k reflectors.

R is $k \times n$ truncated upper trapezoidal matrix.

P is $n \times n$ permutation matrix such that $AP \approx Q(:, 1:k)R$.

1: **function** $[Q, R, P]$=`ssrqrcp`(A, k)

2: Set sample rank $l = k + p$ needed for acceptable sample error.

3: Generate random $l \times m$ GIID compression matrix Ω.

4: Form the sample $B = \Omega A$.

5: **Get k column pivots from sample,** $[Q_b, R_b, P] =$ `qrcp`(B).

6: **Apply permutation** $A^{(1)} = A^{(0)}P$.

7: Construct k reflectors from new leading columns, $[Q, R_{11}] =$ `qr`$(A^{(1)}($`:,1:k`$))$.

8: Finish k rows of R in remaining columns, $R_{12} = Q($`:,1:k`$)^T A^{(1)}($`:,k+1:n`$)$.

9: **end function**

Rank revealing with LU factorization

Дано LU разложение: $PA = LU = \begin{pmatrix} L_{11} & 0 \\ L_{12} & L22 \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} \\ 0 & U22 \end{pmatrix}$

И оценки $\sigma_{\max}(U_{22}) \leq f(k,n)\sigma_{k+1}(A), \quad \sigma_{\min}(L_{11}U_{11}) \geq f(k,n)^{-1}\sigma_k(A), \quad f(k,n) = k(n-k)+1.$

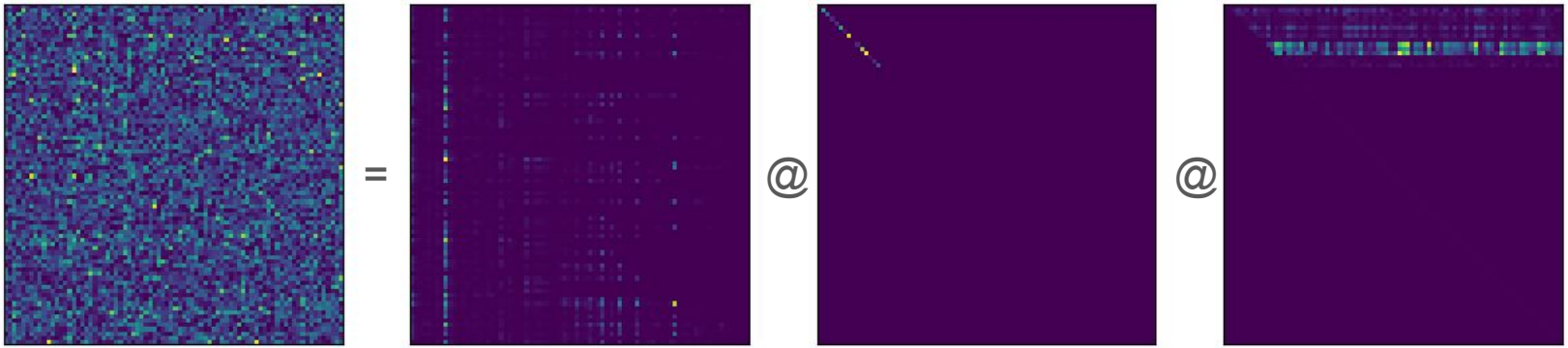
В условиях изначальной задачи вычисляем:

$X = P^T LD,\ D = diag(u_{ii}),\ Y^T = D^{-1}U$

pivoting LU with permutation matrix

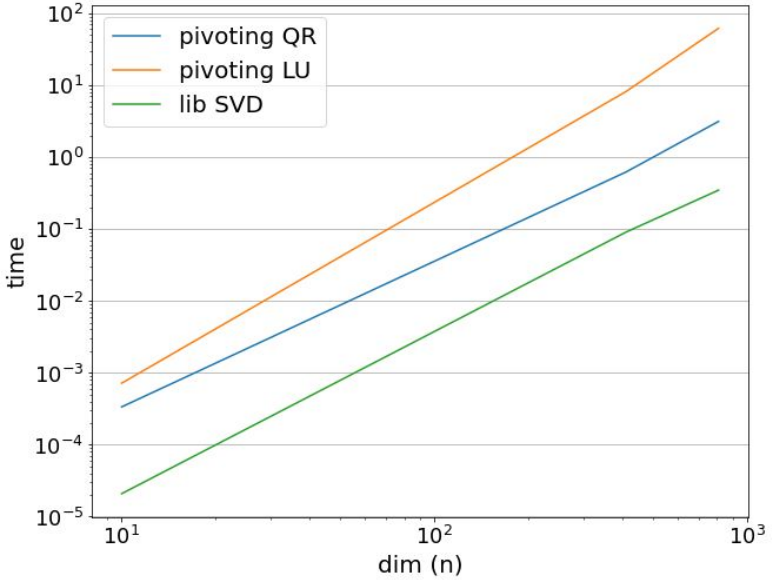
- На каждом шаге ищем в строке (столбце) максимальный элемент
- Перемещаем его на диагональ

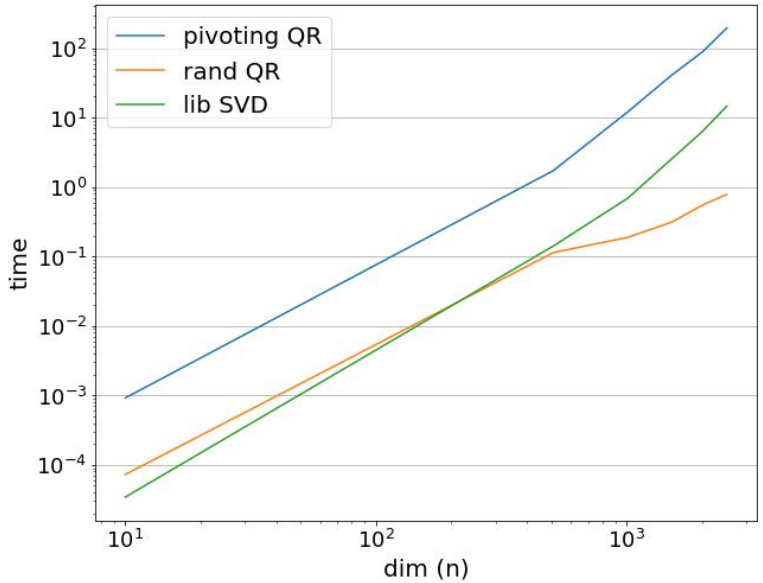
Require:
A: a matrix of size $m \times n$ to decompose, r: desired rank, and $k \geq r$: number of columns to use;
Ensure:
Matrices P, Q, L, U such that $\| PAQ - LU \|_F \leq eps$, where P and Q are permutation matrices, L and U are the lower and upper triangular matrices, respectively, and eps is a smaller error bound;
1: Create a matrix Ω of size $n \times k$, whose entries are i.i.d. Gaussian random variables with zero mean and one variance;
2: $Y \leftarrow A\Omega$;
3: Apply RRLU decomposition to Y such that $PYQ_y = L_{y_1} U_{y_1}$;
4: Truncate L_{y_1} and U_{y_1} by choosing the first r columns and the first r rows, respectively, such that $L_{y_1} \leftarrow L_{y_1}(:, 1:r)$ and $U_{y_1} \leftarrow U_{y_1}(1:r, :)$;
5: $B \leftarrow L_{y_1}^{\dagger} PA$, where $L_{y_1}^{\dagger}$ denotes the Moore-Penrose inverse of L_{y_1};
6: Apply LU decomposition to B with column pivoting $BQ = L_b U_b$;
7: $L \leftarrow L_{y_1} L_b$;
8: $U \leftarrow U_b$;
9: Return L, U, P, Q.



A (80x80), rank=60

Скорость вычислений

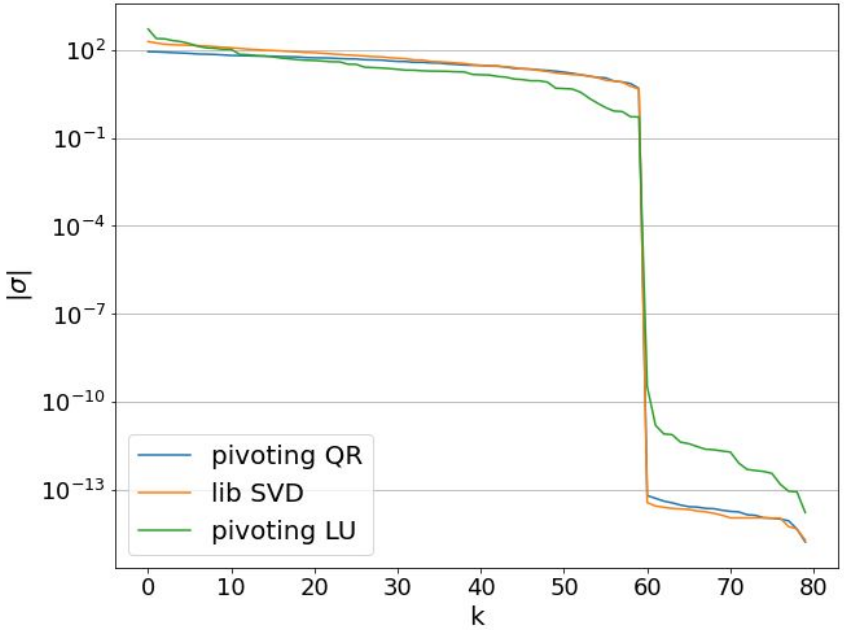


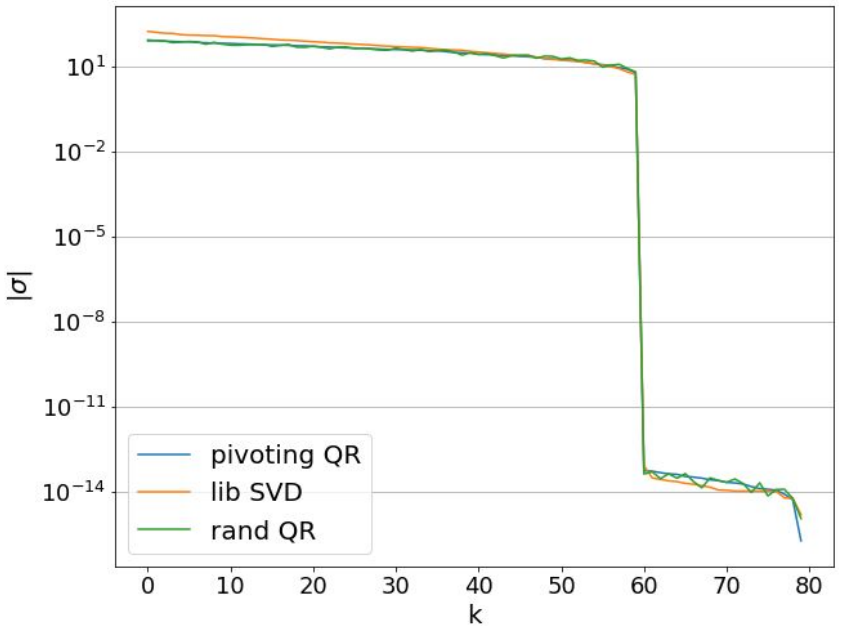


- Рандомизированный алгоритм обеспечивает прирост в скорости перед SVD
- QR pivoting быстрее SVD в теории, однако библиотечный SVD отработал быстрее нашей реализации
- LU существенно проигрывает остальным по скорости

Определение ранга

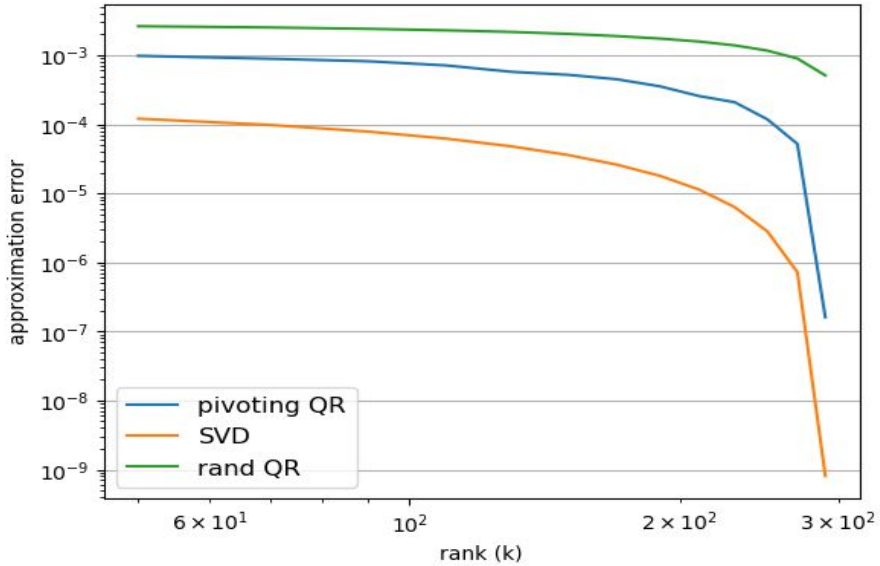
Все реализованные методы для большинства матриц позволяют определить ранг с хорошей точностью





Ранговая аппроксимация

- QR with column pivoting дает приемлемые результаты, хоть они и хуже чем у SVD
- Randomized QR обладает меньшей точностью, но также способен давать приемлемые результаты

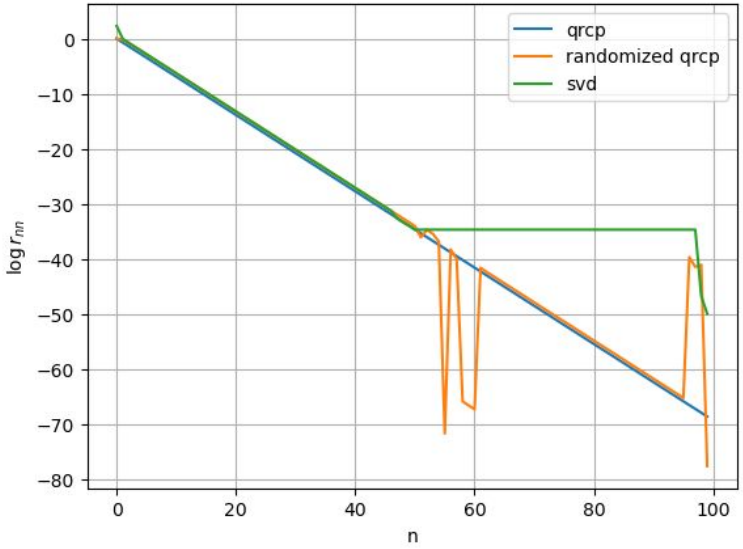


Оценки в худшем случае

- Хоть чаще всего RRQR работают лучше теоретических оценок, однако существуют матрицы, на которых они достигаются
- Пример такой матрицы - матрица Кахана

A_k	$\|A - A_k\|_2$	
$\sum_1^k \sigma_i u_i v_i^{\mathrm{T}}$	$= \sigma_{k+1}(A)$	$\sigma_{\min}(A_k) = \sigma_k(A)$
$Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & 0 \end{pmatrix}$	$\leqslant \sqrt{\tau(n,k)}\sigma_{k+1}(A)$	$\sigma_{\min}(R_{11}) \geqslant \frac{1}{\sqrt{\tau(n,k)}}\sigma_k(A)$

$$\tau(n,k) = k(n-k)\mu^2 + 1.$$



Выводы

- Rank-revealing QR алгоритмы часто (но не всегда) дают хороший результат
- Теоретически RRQR алгоритмы быстрее SVD, однако из-за различий в реализации библиотечный SVD работает быстрее
- Рандомизированные алгоритмы способны работать быстрее стандартных RR QR алгоритмов, обеспечивая при этом похожий результат

Ссылки и литература:

- https://github.com/batradiik/NLA_project
- What Is a Rank-Revealing Factorization?
- On the existence and computation of rank-revealing LU factorizations
- Rank Revealing Algorithms and its Applications