Lecture 15: Complexity

Amos Ron

University of Wisconsin - Madison

March 12, 2021

Outline

Outline

1. Complexity of factorization algorithms
   - Complexity of QR
2. LU factorization

Blank page

LU-factorization

- LU-factorization applies to A $m \times m$:

  $$A = LU.$$

  Here, L is lower triangular, and U is upper triangular.
- LU-factorization is not used to decompose rectangular matrices, or to solve least squares.
- It is used in order to solve square, invertible systems.

Q: QR- or LU-? Which method to prefer?

LU-factorization

Q: QR- or LU-? Which method to prefer?

We will compare the two methods based on:

- Complexity
- Stability: To recall, whenever $A = QR$,

$$\mathrm{cond}(A) = \mathrm{cond}(Q)\mathrm{cond}(R),$$

  which is best possible. However, for LU, it is possible to have, if $A = LU$,

$$\mathrm{cond}(A) << \mathrm{cond}(L)\mathrm{cond}(U).$$

- So, unless LU has better complexity, there is no motivation to use, not even to learn it!!

LU-factorization

Q: QR- or LU-? Which method to prefer?
In order to compare the complexity of QR and LU we need

- To learn LU-factorization...
- To understand the complexity of LU.
- To understand the complexity of QR.

Exact understanding of 'QR'

A square, $A = QR$. Output of QR is

- The matrix R.
- The vectors $w_1, \ldots, w_{m-1}$ that make the matrices $H_1, \ldots, H_{m-1}$.
- We do not compute the Householder matrices, let alone Q.

Exact understanding of 'QR'

How to solve then the linear system $Ax = b$?

- We need to solve
$$Rx = Q'b,$$
so we need to know $Q'b$ without computing Q.
- $Q = H_1 \ldots H_{m-1}$, hence
$$Q'b = H_{m-1}(....H_2(H_1 b))....).$$
- So, we just need to know how to compute Hv, when $H = I - 2ww'$.

Exact understanding of 'QR'

The key computation:

- Input: Vectors $v, w \in R^m$, such $w(i) = 0,\ 1 \leq i \leq k$.
- Output: The vector $(I - 2ww')v$.

Exact understanding of 'QR'

The key computation:

- Input: Vectors $v, w \in R^m$, such $w(i) = 0,\ 1 \le i \le k$.
- Output: The vector $(I - 2ww')v$.

$$(I - 2ww')v = v - 2w(w'v) = v - 2(v, w)w.$$

Counting multiplication:

- Computing (v, w): $m - k$.
- Computing $2(v, w)$: $m - k + 1$.
- Computing $2(v, w)w$: $2(m - k) + 1 \approx 2(m - k)$.

Exact understanding of 'QR'

$$(I - 2ww')v = v - 2w(w'v) = v - 2(v,w)w.$$

Counting multiplication:

- Computing (v,w): $m-k$.
- Computing $2(v,w)$: $m-k+1$.
- Computing $2(v,w)w$: $2(m-k)+1 \approx 2(m-k)$.

In Step I, w_1 has not zeros ($k=0$).
We need to

- Compute w_1 - This will require $O(m)$ operations.
- Compute $(I - 2w_1w_1')A_0$: Need to update $m-1$ columns in A_0.
- Investment: $2(m-0)(m-1) = 2m(m-1)$

Exact understanding of 'QR'

$$(I - 2ww')v = v - 2w(w'v) = v - 2(v,w)w.$$

Counting multiplication:

- Computing (v,w): $m-k$.
- Computing $2(v,w)$: $m-k+1$.
- Computing $2(v,w)w$: $2(m-k)+1 \approx 2(m-k)$.

In Step II, w_2 has one zero ($k=1$). We need to

- Compute w_2 - This will require $O(m)$ operations.
- Compute $(I - 2w_2w_2')A_1$: Need to update $m-2$ columns in A_1.
- Investment: $2(m-1)(m-2)$

So, total investment is

$$2(m(m-1) + (m-1)(m-2) + \ldots + 2 \times 1) = \frac{2(m-1)m(m+1)}{3}.$$

The complexity of QR-factorization

F

or a matrix of order m, efficient QR-factorization consumes $2m^3/3$ multiplications (+ a few lower order terms).

We will see later that LU-factorization consumes $m^3/3$ operations.

Outline

1. Complexity of factorization algorithms
   - Complexity of QR
2. LU factorization

Outline

Goal:

Factor A $m \times m$ into $A = LU$, where

- L is lower triangular
- U is upper triangular

Outline

Process:

Create a sequence of matrices

$$A = A_0 \mapsto A_1 \mapsto \ldots A_{m-1} = U.$$

Here:

$$A_j = L_j A_{j-1},$$

with L_j lower triangular. Then

$$U = L_{m-1} L_{m-2} \ldots L_1 A,$$

and L is then the inverse of the above product of lower triangular matrices.

Outline

Process:

Create a sequence of matrices

$$A = A_0 \mapsto A_1 \mapsto \ldots A_{m-1} = U.$$

Here:

$$A_j = L_j A_{j-1},$$

with L_j lower triangular. Then

$$U = L_{m-1} L_{m-2} \ldots L_1 A,$$

and L is then the inverse of the above product of lower triangular matrices.

Like in QR, the matrix L_j is a rank-1 perturbation of I:

$$L_j = I - l_j e_j'$$

Outline

Like in QR, the matrix L_j is a rank-1 perturbation of I:

$$L_j = I - l_j e_j',$$

with $l_j(i) = 0$, for $i \leq j$.

If $B = I - le_j'$, where $l(i) = 0$, $i \leq j$, then

$$B^{-1} = I + le_j'.$$

Outline

If $B = I - le'_j$, where $l(i) = 0$, $i \le j$, then

$$B^{-1} = I + le'_j.$$

Proof:

$$(I - le'_j)(I + le'_j) = I - le'_j + le'_j - le'_j le'_j = I - le'_j le'_j.$$

However:

Outline

If $B = I - le_j'$, where $l(i) = 0$, $i \le j$, then

$$B^{-1} = I + le_j'.$$

Proof:

$$(I - le_j')(I + le_j') = I - le_j' + le_j' - le_j'le_j' = I - le_j'le_j'.$$

However:

$$le_j'le_j' = l(e_j'l)e_j' = l(j)le_j' = 0.$$

How to compute the L-matrix?

We have that

$$U = L_{m-1} \dots L_1 A.$$

Therefore,

$$A = L_1^{-1} \dots L_{m-1}^{-1} U.$$

We have just observed that inverting L_j is trivial.
But: How to multiply the inverses?

How to compute the L-matrix?

If $B_j = I + l_j e_j'$, where $l_j(i) = 0$, $i \leq j$, then

$$B_1 B_2 \ldots B_{m-1} = I + l_1 e_1' + l_2 e_2' + \ldots l_{m-1} e_{m-1}'.$$

How to compute the L-matrix?

If $B_j = I + l_j e_j'$, where $l_j(i) = 0$, $i \leq j$, then

$$B_1 B_2 \dots B_{m-1} = I + l_1 e_1' + l_2 e_2' + \dots l_{m-1} e_{m-1}'.$$

Proof: When expanding, every term other than those listed above contains a sequence

$$l_j e_j' l_k e_k',$$

where $j < k$. But:

$$l_j e_j' l_k e_k' = l_j (e_j' l_k) e_k' = l_k(j) l_j e_k' = 0.$$