

CSE 6643 Homework 3

Karl Hiner

1 One-upping [25 pts]

Let $\mathbf{A} \in \mathbb{R}^{m \times m}$ have full rank. Assume that we have already computed the QR decomposition of \mathbf{A} . For $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$, we call the matrix $\mathbf{B} = \mathbf{A} + \mathbf{u}\mathbf{v}^T$ a rank-1 update of \mathbf{A} .

(a) [5 pts]

Prove that if $\mathbf{v}^T \mathbf{A}^{-1} \mathbf{u} \neq -1$, then \mathbf{B} is invertible.

Let $\mathbf{x} \in \mathbb{R}^m$ be a non-zero vector. Then,

$$\begin{aligned} \mathbf{v}^T \mathbf{A}^{-1} \mathbf{u} &\neq -1 \\ 1 &\neq -\mathbf{v}^T \mathbf{A}^{-1} \mathbf{u} \\ \mathbf{v}^T \mathbf{x} &\neq -(\mathbf{v}^T \mathbf{A}^{-1} \mathbf{u})(\mathbf{v}^T \mathbf{x}) \\ \mathbf{x} &\neq -\mathbf{A}^{-1} \mathbf{u}(\mathbf{v}^T \mathbf{x}) \\ \mathbf{A} \mathbf{x} &\neq -\mathbf{u} \mathbf{v}^T \mathbf{x} \\ (\mathbf{A} + \mathbf{u} \mathbf{v}^T) \mathbf{x} &\neq \mathbf{0}. \\ \mathbf{B} \mathbf{x} &\neq \mathbf{0} \end{aligned}$$

Thus, if $\mathbf{v}^T \mathbf{A}^{-1} \mathbf{u} \neq -1$, there are no nontrivial solutions for $\mathbf{B} \mathbf{x} = \mathbf{0}$, and so \mathbf{B} is invertible.

(b) [10 pts]

Design an algorithm that provably solves the system of equations $\mathbf{B} \mathbf{x} = \mathbf{b}$ in $O(m^2)$ operations.

$$\begin{aligned} \mathbf{x} &= \mathbf{B}^{-1} \mathbf{b} && \text{(assume } \mathbf{B} \text{ is invertible)} \\ &= (\mathbf{A} + \mathbf{u} \mathbf{v}^T)^{-1} \mathbf{b} && \text{(since } \mathbf{B} = \mathbf{A} + \mathbf{u} \mathbf{v}^T) \\ &= \left(\mathbf{A}^{-1} - \frac{\mathbf{A}^{-1} \mathbf{u} \mathbf{v}^T \mathbf{A}^{-1}}{1 + \mathbf{v}^T \mathbf{A}^{-1} \mathbf{u}} \right) \mathbf{b} && \text{(Sherman-Morrison formula)} \\ &= \mathbf{A}^{-1} \mathbf{b} - \frac{\mathbf{A}^{-1} \mathbf{u} \mathbf{v}^T}{1 + \mathbf{v}^T \mathbf{A}^{-1} \mathbf{u}} \mathbf{A}^{-1} \mathbf{b} && \text{(note: (1a above), } \mathbf{B} \text{ invertible} \implies \text{den} \neq 0) \\ &= \tilde{\mathbf{x}} - \frac{\tilde{\mathbf{u}} \mathbf{v}^T}{1 + \mathbf{v}^T \tilde{\mathbf{u}}} \tilde{\mathbf{x}} && \text{(let } \tilde{\mathbf{x}} \equiv \mathbf{A}^{-1} \mathbf{b}, \tilde{\mathbf{u}} \equiv \mathbf{A}^{-1} \mathbf{u}) \\ &= \left[\mathbf{I} + \left(\frac{-1}{1 + \mathbf{v}^T \tilde{\mathbf{u}}} \right) \tilde{\mathbf{u}} \mathbf{v}^T \right] \tilde{\mathbf{x}} && \text{(rearrange)} \\ &= (\mathbf{I} + \alpha \tilde{\mathbf{u}} \mathbf{v}^T) \tilde{\mathbf{x}} && \text{(let } \alpha \equiv \frac{-1}{1 + \mathbf{v}^T \tilde{\mathbf{u}}}) \end{aligned}$$

Thus, we have derived an expression for \mathbf{x} in terms of a scalar α and vectors $\tilde{\mathbf{u}}, \tilde{\mathbf{x}}$.

Vectors $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{x}}$ are both defined in terms of $\mathbf{A}^{-1} = \mathbf{R}^{-1}\mathbf{Q}^T$, where we assume \mathbf{R} and \mathbf{Q} have already been computed. Thus, both vectors can be computed using back substitution in $O(m^2)$. α can then be computed in $O(m)$, and the final expression for \mathbf{x} involves a vector-scalar product, a vector outer product, an identity addition, and a matrix-vector multiplication, for a total of

$$O(m) + O(m^2) + O(m) + O(m^2) = O(2m) + O(m^2) = O(m^2)$$

operations.

Here is the algorithm:

1. Solve $\mathbf{R}\tilde{\mathbf{x}} = \mathbf{Q}^T\mathbf{b}$ for $\tilde{\mathbf{x}}$ using back substitution. ($O(m^2)$)
2. Solve $\mathbf{R}\tilde{\mathbf{u}} = \mathbf{Q}^T\mathbf{u}$ for $\tilde{\mathbf{u}}$ using back substitution. ($O(m^2)$)
3. Compute the scalar $\alpha = \frac{-1}{1 + \mathbf{v}^T\tilde{\mathbf{u}}}$. ($O(m)$)
4. Finally, compute the solution $\mathbf{x} = (\mathbf{I} + ((\alpha\tilde{\mathbf{u}})\mathbf{v}^T))\tilde{\mathbf{x}}$. ($O(m^2)$)

Since the highest-order term across all steps is $O(m^2)$, the total number of operations is thus $O(m^2)$.

(c) r-upping [10 pts]

Extend the algorithm from the previous exercise to the case of $\mathbf{B} = \mathbf{A} + \mathbf{U}\mathbf{V}^T$, for $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{m \times r}$ and $r \ll m$. Calculate the asymptotic complexity of the resulting algorithm.

The Sherman-Morrison formula was the key step above. Here, we will use the generalization of that formula, the Woodbury matrix identity. Given the definitions above, the Woodbury matrix identity states that if $(\mathbf{I} + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U})$ is invertible,

$$\mathbf{B}^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{I} + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}^T\mathbf{A}^{-1}.$$

Thus, we can express the solution \mathbf{x} as

$$\begin{aligned} \mathbf{x} &= (\mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{I} + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}^T\mathbf{A}^{-1})\mathbf{b} \\ &= \mathbf{A}^{-1}\mathbf{b} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{I} + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}^T\mathbf{A}^{-1}\mathbf{b} \\ &= \tilde{\mathbf{x}} - \tilde{\mathbf{U}}(\mathbf{I} + \mathbf{V}^T\tilde{\mathbf{U}})^{-1}\mathbf{V}^T\tilde{\mathbf{x}} && (\text{let } \tilde{\mathbf{x}} \equiv \mathbf{A}^{-1}\mathbf{b}, \tilde{\mathbf{U}} \equiv \mathbf{A}^{-1}\mathbf{U}) \\ &= [\mathbf{I} - \tilde{\mathbf{U}}(\mathbf{I} + \mathbf{V}^T\tilde{\mathbf{U}})^{-1}\mathbf{V}^T]\tilde{\mathbf{x}} && (\text{rearrange}) \\ &= [\mathbf{I} - (\mathbf{I} + \tilde{\mathbf{U}}\mathbf{V}^T)^{-1}\tilde{\mathbf{U}}\mathbf{V}^T]\tilde{\mathbf{x}} && (\text{"push-through identity"}[1]) \\ &= (\mathbf{I} + \alpha\tilde{\mathbf{U}}\mathbf{V}^T)\tilde{\mathbf{x}} && (\text{let } \alpha = -(\mathbf{I} + \tilde{\mathbf{U}}\mathbf{V}^T)^{-1}) \end{aligned}$$

Note that, since $\tilde{\mathbf{U}} \in \mathbb{R}^{m \times r}$, and $\mathbf{V}^T \in \mathbb{R}^{r \times m}$, we can compute the matrix product $\tilde{\mathbf{U}}\mathbf{V}^T$ using $O(m^2r)$ operations.

TODO need to finish after step 2. <https://people.clas.ufl.edu/hager/files/update-1.pdf> p224 is close.

Here is the algorithm:

1. Solve $\mathbf{R}\tilde{\mathbf{x}} = \mathbf{Q}^T\mathbf{b}$ for $\tilde{\mathbf{x}}$ using back substitution (same as step 1 in (b)). ($O(m^2)$)

2. Compute $\tilde{U} = A^{-1}U$ by solving each of the linear systems $R\tilde{u}_i = Q^T u_i$ for \tilde{u}_i , where \tilde{u}_i is the i th column of \tilde{U} and u_i is the i th column of U . This can be done using back substitution (as in step 2 in (b)) for each of the r columns. ($O(m^2 r)$)
3. Compute the scalar $\alpha = \frac{-1}{1 + v^T \tilde{u}}$. ($O(m)$)
4. Finally, compute the solution $x = (I + ((\alpha \tilde{u})v^T))\tilde{x}$. ($O(m^2)$)

2 You Factor [25 pts]

In class we have seen that if $\tilde{x} \in \mathbb{R}^m$ is the solution to the system $Ax = b$, as computed by unpivoted LU factorization, we have

$$(A + E)\tilde{x} = b. \quad (1)$$

For u the unit roundoff error and \tilde{L}, \tilde{U} the LU factors computed in finite precision, we have

$$|E| \leq mu(2|A| + 4|\tilde{L}||\tilde{U}|) + O(u^2). \quad (2)$$

Here, $|\cdot|$ signifies the element-wise absolute values and \leq is interpreted element-wise, as well. In this problem, we investigate the conclusions from this bound in the case of row-pivoted LU factorization.

Generally useful things:

- p174: Because each pivot selection involves maximization over a column, this algorithm produces a matrix L with entries of absolute value ≤ 1 everywhere below the diagonal. This implies $\|L\| = O(1)$ in any norm. ... so the algorithm is backward stable if $\|U\| = O(\|A\|)$.

(a) [7.5 pts]

Deduce that under row-pivoted LU factorization and taking $\|\cdot\|_\infty$ to signify the vector-infinity norm, we have

$$\|E\|_\infty \leq mu(2\|A\|_\infty + 4m\|\tilde{U}\|_\infty) + O(u^2). \quad (3)$$

This prompts us to investigate the growth factor $\rho := \frac{\|U\|_\infty}{\|A\|_\infty}$ of row-pivoted LU factorization.

(b) [5 pts]

Verify that the rows u_i^T, a_i^T of U, A satisfy

$$u_i^T = a_i^T - \sum_{j=1}^{i-1} L_{ij} u_j^T. \quad (4)$$

Given the definition of row-pivoted LU factorization, we can write the system of linear equations as $Ax = b \implies PAx = Pb \implies LUx = b$. Since L is a lower triangular matrix, U is an upper triangular matrix, with the diagonal elements equal to 1. So, we can write the i -th row of U as $u_i^T = a_i^T - \sum_{j=1}^{i-1} L_{ij} u_j^T$ which satisfies the equation (3).

(c) [5 pts]

Use part (b) to show that $\|\tilde{U}\|_\infty \leq 2^{m-1}\|A\|_\infty$.

This is the same as asking to prove that $\max \rho = 2^{m-1}$ for an $m \times m$ matrix A (Exercise 22.1 in the book). Note that (d) below is an example of this worst case. Useful here: p174: Because

each pivot selection involves maximization over a column, this algorithm produces a matrix L with entries of absolute value ≤ 1 everywhere below the diagonal. Probably want in our proof that there are no pivots in this worst case (as in the example below).

From canvas: We can first bound each element of u_1^T using (b), and then bound each element of u_2^T using (b) and the bound for each element of u_1^T , and then bound each element of u_3^T using (b) and the bounds for u_1^T, u_2^T , and so on...

(d) [7.5 pts]

Consider matrices of the form

$$\mathbf{A} = \begin{pmatrix} 1 & & & & 1 \\ -1 & 1 & & & 1 \\ -1 & -1 & 1 & & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{pmatrix}. \quad (5)$$

Derive the growth factor in this case as a function of m .

The $\mathbf{PA} = \mathbf{LU}$ factorization for \mathbf{A} is:

$$\begin{pmatrix} 1 & & & & 1 \\ -1 & 1 & & & 1 \\ -1 & -1 & 1 & & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ -1 & 1 & & & \\ -1 & -1 & 1 & & \\ -1 & -1 & -1 & 1 & \\ -1 & -1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & & & & 1 \\ & 1 & & & 2 \\ & & 1 & & 4 \\ & & & 1 & 8 \\ & & & & 16 \end{pmatrix} \quad (6)$$

The growth factor is then

$$\rho = \frac{\|\mathbf{U}\|_{\infty}}{\|\mathbf{A}\|_{\infty}} = \frac{16}{1} = 16 = 2^4 = 2^{m-1}.$$

How does this relate to part (c)?

3 [25 pts]

Suppose that $\mathbf{A} \in \mathbb{K}^{m \times m}$ is strictly column diagonally dominant, meaning that for all $1 \leq k \leq m$,

$$|\mathbf{A}_{kk}| > \sum_{j \neq k} |\mathbf{A}_{jk}|. \quad (7)$$

Show that if LU factorization with row pivoting is applied to \mathbf{A} , no row interchange takes place.

To show that now row interchange takes place, we can show the following:

1. Show that \mathbf{A} 's strict column diagonal property implies no row interchange takes place during the first factorization $k = 1$. Then the first permutation matrix $\mathbf{P}^1 = \mathbf{I}$, and we have $\mathbf{L}^1 \mathbf{P}^1 \mathbf{A} = \mathbf{L}^1 \mathbf{A} = \mathbf{U}^1$, with \mathbf{U}^1 being the first intermediate matrix on the way to \mathbf{U} after completing factorization step $k = 1$. Express \mathbf{U}^1 as

$$\mathbf{U}^1 = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{1j>1} \\ \mathbf{0} & \mathbf{A}^1 \end{bmatrix},$$

with $\mathbf{A}^1 = \mathbf{A}_{2:m,2:m} \in \mathbb{K}^{m-1 \times m-1}$.

2. Given (1), show that \mathbf{A}^1 is also strictly diagonally dominant. Then, apply LU facorization with row pivoting to \mathbf{A}^1 , with $\hat{\mathbf{P}}\mathbf{A}^1 = \hat{\mathbf{L}}\hat{\mathbf{U}}$. We conclude from (1) that $\hat{\mathbf{P}}^1 = \mathbf{I}$, and we have $\hat{\mathbf{L}}^1\mathbf{A}^1 = \hat{\mathbf{U}}^1$, where

$$\hat{\mathbf{U}}^1 = \begin{bmatrix} \mathbf{A}_{11}^1 & \mathbf{A}_{1,j>1}^1 \\ \mathbf{0} & \mathbf{A}^2 \end{bmatrix},$$

with $\mathbf{A}^2 = \mathbf{A}_{2:,2:}^1 \in \mathbb{K}^{m-2 \times m-2}$.

If we show (1) and (2) hold, then we can apply step (1) again to matrix \mathbf{A}^2 to show that no row interchange takes place at step $k = 2$. Then we can apply step again (2) to show \mathbf{A}^3 is also strictly diagonally dominant, etc. Thus, if we show (1) and (2), we can inductively infer that no row interchanges take place during all factorization steps $1 \leq k \leq m - 1$.

1. Show that $|\mathbf{A}_{kk}| > \sum_{j \neq k} |\mathbf{A}_{jk}| \implies \mathbf{P}^1 = \mathbf{I}$:

During the first factorization step, we choose a pivot row $j_1 = \operatorname{argmax}_j |\mathbf{A}_{j1}|$.

$$\begin{aligned} |\mathbf{A}_{kk}| > \sum_{j \neq k} |\mathbf{A}_{jk}| &> \max_{j \neq k} |\mathbf{A}_{jk}| \implies \\ |\mathbf{A}_{kk}| &= \max_j |\mathbf{A}_{jk}| \implies \\ |\mathbf{A}_{11}| &= \max_j |\mathbf{A}_{j1}| \implies \\ 1 &= \operatorname{argmax}_j |\mathbf{A}_{j1}|, \end{aligned}$$

and so we choose the first pivot row $j_1 = 1$. In other words, no rows are swapped during this first step, and $\mathbf{P}^1 = \mathbf{I}$.

2. Show that \mathbf{A}^1 is also strictly diagonally dominant.

Let $\alpha \equiv \frac{\mathbf{A}_{1j}}{\mathbf{A}_{11}}$. Here is what the j th column of \mathbf{A}^1 looks like:

$$\begin{bmatrix} \mathbf{A}_{2j} - \alpha \mathbf{A}_{21} \\ \dots \\ \mathbf{A}_{jj} - \alpha \mathbf{A}_{j1} \\ \dots \\ \mathbf{A}_{mj} - \alpha \mathbf{A}_{m1} \end{bmatrix}$$

Thus, for \mathbf{A}^1 to be strictly diagonally dominant, we must have

$$\begin{aligned} |\mathbf{A}_{kk}^1| &> \sum_{j \neq k} |\mathbf{A}_{jk}^1| \implies \\ |\mathbf{A}_{jj} - \alpha \mathbf{A}_{j1}| &> \sum_{i \geq 2, i \neq j} |\mathbf{A}_{ij} - \alpha \mathbf{A}_{i1}|. \end{aligned}$$

Starting with the LHS:

$$\begin{aligned}
|A_{jj} - \alpha A_{j1}| &\geq |A_{jj}| - |\alpha A_{j1}| \\
&> \sum_{i \neq j} |A_{ij}| - |\alpha A_{j1}| && (\mathbf{A} \text{ is SDD}) \\
&= \sum_{i \geq 2, i \neq j} |A_{ij}| - |A_{1j}| - |\alpha A_{j1}| && (\text{move } i = 1 \text{ out of sum}) \\
&= \sum_{i \geq 2, i \neq j} |A_{ij}| - |\alpha A_{11}| - |\alpha A_{j1}| && (\text{using def. of } \alpha) \\
&> \sum_{i \geq 2, i \neq j} |A_{ij}| - \sum_{i \neq 1} |\alpha A_{i1}| - |\alpha A_{j1}| && (\mathbf{A} \text{ is SDD}) \\
&= \sum_{i \geq 2, i \neq j} |A_{ij}| - \sum_{i \geq 2, i \neq j} |\alpha A_{i1}| && (\text{combine right two terms}) \\
&= \sum_{i \geq 2, i \neq j} |A_{ij} - \alpha A_{i1}| && (\text{QED})
\end{aligned}$$

4 Pivoting [25 pts]

(a) [5 pts]

Go to section (a) of the file `HW3_your_code.jl` and implement a function that takes in a matrix $\mathbf{LU} \in \mathbb{K}^{m \times m}$ containing the upper triangular part of \mathbf{U} as well as the strict lower triangular part of \mathbf{L} , as well as an array $\mathbf{P} \in \{1, \dots, m\}^m$ that encodes the permutation matrix \mathbf{P} by $\mathbf{P}[j] = i \Leftrightarrow \mathbf{P}_{ij} = 1$. Your function should not allocate any memory.

(b) [5 pts]

Go to section (b) of the file `HW3_your_code.jl` and implement the unpivoted LU factorization. Check your code by ensuring that the assertions in section a + b of `HW3_driver.jl` do not produce any errors.

(c) [5 pts]

Generate families of random $m \times m$ matrices and vectors of length m . Plot as a function of the size m , the relative error of the solution obtained from your code in parts (a,b) and the growth factor introduced in problem 2. Report the floating point type used by your program. You can use the code provided in the second homework as a starting point for creating and saving plots.

(d) [5 pts]

Go to section (d) of the file `HW3_your_code.jl` and implement the unpivoted LU factorization. Your code should pass the assertions in section (c) of `HW3_driver.jl`. Your function should take the matrix \mathbf{A} as an input to modify in place, and return an integer array \mathbf{P} according to the specifications of (a). Repeat the experiment of (c) using the pivoted LU factorization.

(e) [5 pts]

Go to section (e) of the file `HW3_driver.jl` and implement a function that takes an integer m as an input and returns an $m \times m$ matrix as introduced in problem 2 (d). Plot the error of the solution when solving equations in this matrix as a function of m . Compare the error to the built-in solution (the `\` operator). Draw your conclusions from this comparison.

References

- [1] H. V. Henderson and S. R. Searle. On Deriving the Inverse of a Sum of Matrices. *SIAM Review*, 23(1):53–60, 1981. Publisher: Society for Industrial and Applied Mathematics.