Lecture 04

Paul Scemama

OCW: *This session explains inverses, transposes and permutation matrices. We also learn how elimination leads to a useful factorization $A = LU$ and how hard a computer will work to invert a very large matrix.*

1 Outline

1. Inverse of AB, A^T
2. $A = LU$ (no row exchanges)
3. $A = LU$ (row exchanges)
   - a great way to look at Gaussian elimination.
4. Compute cost of elimination.

2 Inverse of AB, A^T

2.1 Inverse of AB

Suppose matrices A and B are invertible,

$$(AB)(B^{-1}A^{-1}) = I \tag{1}$$

due to associativity of matrix multiplication. To illustrate this, we just change the order of parentheses,

$$\begin{gathered}(AB)(B^{-1}A^{-1}) \\ A\underbrace{(BB^{-1})}_{I}A^{-1} \\ AA^{-1} = I\end{gathered}$$

So we've proved that (1) is true. Next, we notice that (1) implies that

$$(AB)^{-1} = (B^{-1}A^{-1}). \tag{2}$$

In words: *the inverse of AB is the inverse of B multiplied to the inverse of A* – the order gets switched.

By the definition of the inverse, it is also true that

$$B^{-1}A^{-1}AB = I.$$

2.2 Inverse of A^T

Consider the fact that

$$AA^{-1} = I.$$

Now transpose both sides,

$$(AA^{-1})^T = I^T \tag{3}$$

It can be shown (proof in the appendix) that

$$(AB)^T = B^T A^T \tag{4}$$

And so applying (4) to the LHS of (3) we get

$$(AA^{-1})^T = (A^{-1})^T A^T = I$$

And so by the definition of the inverse, we get

$$(A^T)^{-1} = (A^{-1})^T \tag{5}$$

or in words: *the inverse of A transpose is the transpose of A inverse.*

3 $A = LU$ (no row exchanges)

Notable assumptions:

- *no rows exchanges involved in elimination*

This is the first of many useful factorizations. The goal is to represent *gaussian* elimination in the best possible way, where by *best* we mean *most practical*.

3.1 Showing $A = LU$

The first thing to note is that L and U are *triangular* matrices.

- *Triangular* matrices are useful in that we can easily solve and substitute one row after the other.

We already know U – the result of elimination on A. It has the pivots on its diagonal, and back-substitution can be performed to solve for $\boldsymbol{x}$. We will show taking U back to A is achieved by a lower triangular L. In other words, if E is the elimination matrix such that $EA = U$, L is the inverse of E such that applying L to U bring us back to A. Importantly, **the entries of L are exactly the multipliers** ℓ_{ij} – which multiplied the pivot row j when it was subtracted from row i (a lot of jumbled words yes, but after an example it should be clear what this means).

3.1.1 Example

Consider 2×2 matrix A containing $2, 1, 6, 8$. The number to eliminate is 6. Therefore, we begin by *subtracting* 3 *times row* 1 *from row* 2. This corresponds to E_{21} in the forward direction with multiplier $\ell_{21} = 3$. To get back to A from here we must add back 3; and so the *inverse* (or return step) from U to A is $L = E_{21}^{-1}$ (adding 3 times row 1 back to row 2):

$$\textit{forward from } A \textit{ to } U: \quad E_{21}A = \begin{bmatrix} 1 & 0 \\ -3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 6 & 8 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 0 & 5 \end{bmatrix} = U \tag{6}$$

$$\textit{back from } U \textit{ to } A: \quad E_{21}^{-1}U = \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 0 & 5 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 6 & 8 \end{bmatrix} = A \tag{7}$$

(7) is our factorization $LU = A$ – instead of E_{21}^{-1} we write L. If it takes many E_{ij} to take A to U, then L is the inverse of all those E_{ij} combined. For example,

$$(E_{32}E_{31}E_{21})A = U \;\; \textit{becomes} \;\; A = (E_{21}^{-1}E_{31}^{-1}E_{32}^{-1})U \;\; \textit{which is} \;\; A = LU \tag{8}$$

Notice that the inverses go in opposite order (recall (2)). We've shown that $A = LU$ and how L is derived from elimination. In fact, it holds all the information of the elimination steps done to A.

3.1.2 Another Example

Consider

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

We do elimination on A by first subtracting $\frac{1}{2}$ of row 1 from row 2. The second (and last) step subtracts $\frac{2}{3}$ of row 2 from row 3. Recall that **the entries of L are exactly the multipliers ℓ_{ij}** – which multiplied the pivot row j when it was subtracted from row i. And so $A = LU$ can written as

$$A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix} = LU = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ 0 & \frac{2}{3} & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 0 \\ 0 & \frac{3}{2} & 1 \\ 0 & 0 & \frac{4}{3} \end{bmatrix}$$

3.2 Practicality of $A = LU$

How does $A = LU$ help us in solving $A\boldsymbol{x} = \boldsymbol{b}$? In many engineering applications, A is static. Instead, we are given various $\boldsymbol{b}$, and we need to solve $A\boldsymbol{x} = \boldsymbol{b}$ for new $\boldsymbol{b}$! To solve the system as we had previous to this lecture, we would need to do elimination on $\begin{bmatrix} A & \boldsymbol{b} \end{bmatrix}$. This is slow and laborious for computers to do. We need to do the same elimination to the same A each time.

What we can do is *factorize* A into L and U. Then we have

$$LU\boldsymbol{x} = \boldsymbol{b}.$$

What we can do is set $U\boldsymbol{x} = \boldsymbol{c}$ and solve,

$$L\boldsymbol{c} = \boldsymbol{b}$$

This is easy since L is triangular - no need to do any elimination! Then once $\boldsymbol{c}$ is found, we can do

$$U\boldsymbol{x} = \boldsymbol{c}$$

which can simply be solved with back-substitution since U is also triangular.

This is all to show that saving the memory of elimination in L and the result of elimination of A into U, we can save time and memory when we encounter a new $\boldsymbol{b}$.

3.2.1 Why not just use E

A natural question is why not just save the history of elimination in E? Recall that it is also true that

$$EA = U$$

where E is all the elimination steps combined into one matrix. Then, once we have E, we can just apply E to any new $\boldsymbol{b}$ and transform it the same way we did A - by left multiplying it by E:

$$E\boldsymbol{b} = \boldsymbol{b}'$$

And then we can back-substitute

$$U\boldsymbol{x} = \boldsymbol{b}'$$

Yes - this does solve the problem of having to do elimination for A for every new $\boldsymbol{b}$ we see. *However*, creating E from elimination steps actually takes more computation than creating L from elimination steps. And we will show this now. Remember the very special property about L...that the elements of L (other than the 1's on the diagonal) are **exactly the multipliers** ℓ_{ij} – which multiplied the pivot row j when it was subtracted from row i. So we can just fill in L as we do elimination. However, this is not the case with E. To show this:

$$\overbrace{\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & -5 & 1\end{bmatrix}}^{E_{32}}\overbrace{\begin{bmatrix}1 & 0 & 0\\ -2 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}}^{E_{21}} = \begin{bmatrix}1 & 0 & 0\\ -2 & 1 & 0\\ \boxed{10} & -5 & 1\end{bmatrix} = E$$

On the other hand,

$$\overbrace{\begin{bmatrix}1 & 0 & 0\\ 2 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}}^{E_{21}^{-1}}\overbrace{\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 5 & 1\end{bmatrix}}^{E_{32}^{-1}} = \begin{bmatrix}1 & 0 & 0\\ 2 & 1 & 0\\ 0 & 5 & 1\end{bmatrix} = L$$

What does this show: this shows that we cannot just create E by noticing the multipliers used in elimination. We need to construct each single E_i and then matrix multiply them together. This is not the case with L! We can put in the multipliers 2 and 5 directly into a matrix as we go. Therefore there is no need for extraneous matrix multiplications.

3.2.2 Remark

One of the ideas of $A = LU$ is to *decouple* $\boldsymbol{b}$ from A in the context of elimination. *Decoupling* is a very important "goal" in many areas of engineering. It allows us to construct systems dynamically since we have independent building blocks, and so we can perform actions on one block without affecting other blocks.

4 $A = LU$ (row exchanges)

We've ignored the possibility of exchanging rows in elimination. In fact, this detail introduces a new object that can help us.

Here we introduce *permutation matrices*. These matrices will contain the information of *exchanging rows* while doing elimination. $A = LU$ then becomes $PA = LU$.

A nice property about *permutation matrices* P is that

$$P^{-1} = P^T$$

For a $3x3$ A, permutation matrices are all the permutations of the identity matrix I:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}; etc.$$

5 Cost of elimination

Q: how many operations on an $n \times n$ matrix A does it take to do elimination? I.e. to get to U?

Let's define *operation* = *multiply* + *subtract*. Consider $n = 100$.

- Change all rows under the first pivot.
  - cost $\simeq 100^2$ since there are 100^2 numbers in the matrix.
- Change all rows under the next pivot.
  - cost $\simeq 99^2$ since there are 99^2 numbers we need to change.
- and so on...

And so in total: $n^2 + (n-1)^2 + ... + 3^2 + 2^2 + 1^2 \simeq \frac{1}{3}n^3$.

Q: now what about the number of operations on $\boldsymbol{b}$?

$\boldsymbol{b}$ is like doing elimination for the first pivot. So it is $= n^2$. However if we have a lot of $\boldsymbol{b}$ we need to evaluate it can be a lot.