

Matrix Factorization

Reading: Lay 2.5

October 1, 2013

You have seen that if we know the inverse A^{-1} of a matrix A , we can easily solve the equation $A\mathbf{x} = \mathbf{b}$. Solving a large number of equations $A\mathbf{x}_1 = \mathbf{b}_1$, $A\mathbf{x}_2 = \mathbf{b}_2, \dots, A\mathbf{x}_k = \mathbf{b}_k$, where A is the same in each equation, is easy if we know A^{-1} . Indeed, rather than writing down k different augmented matrices and then computing the RREF for each (this can be time-consuming), we can just multiply each \mathbf{b}_i by A^{-1} to get the solutions.

In a real-life situation, where the matrices that come up are often extremely complicated, computing an inverse is not always the most computationally efficient method, because computing the inverse of a matrix can take a fair amount of time. Today, we will discuss an alternative approach which is generally faster.

1 LU factorization

We call a matrix whose entries below the main diagonal are all zero an **upper triangular matrix**, and make a similar definition for **lower triangular**. The idea of the LU factorization is to write a general matrix as a product of a lower triangular and an upper triangular matrix.

So let A be an $m \times n$ matrix, and assume we can write $A = LU$, where L is $m \times m$, lower triangular, and has 1's on the diagonal; and where U is an $m \times n$ echelon form of A . For instance, one possible set of shapes is

$$L = \begin{bmatrix} 1 & 0 & 0 \\ * & 1 & 0 \\ * & * & 1 \end{bmatrix}, \quad U = \begin{bmatrix} * & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Why would such a decomposition be important? Say we want to solve $A\mathbf{x} = \mathbf{b}$. Then we can rewrite this as

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

We can solve this equation by solving instead the pair of equations

$$\begin{aligned} U\mathbf{x} &= \mathbf{y} \\ L\mathbf{y} &= \mathbf{b}. \end{aligned}$$

You can think of this as being a “substitution trick” of the form you may have seen in, for instance, calculus class. The reason that a pair of equations like this will generally be easier to solve has to do with the particular structure of L and U . The plan of attack is as follows: first solve $L\mathbf{y} = \mathbf{b}$ for \mathbf{y} , then solve $U\mathbf{x} = \mathbf{y}$.

Example 1.1. Let

$$\mathbf{b} = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}.$$

Say A is the matrix

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

To begin solving $L\mathbf{y} = \mathbf{b}$, we write the augmented matrix

$$[L \quad \mathbf{b}] = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 2 \\ 1 & 3 & 1 & 2 \end{bmatrix} \tag{1}$$

One very fast way to solve this system above is to compute the RREF. This is made very easy by the structure of L (try it yourself and you will see what I mean), and gives

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -2 \end{bmatrix}.$$

Another way to solve the system with augmented matrix (1) is to note that we can just “read off” the value $y_1 = 1$ from the empty first row, followed by

$$y_2 = 2 - x_1 = 1$$

from the second, and

$$y_3 = 2 - y_2 - 3y_1 = -2$$

from the third. As you can see, the shape of L makes it very easy to deal with.

Solving $U\mathbf{x} = \mathbf{y}$ is similar. The augmented matrix is

$$[L \quad \mathbf{b}] = \begin{bmatrix} 4 & 3 & 3 & 1 \\ 0 & -1 & -1 & 1 \\ 0 & 0 & 2 & -2 \end{bmatrix}. \quad (2)$$

The RREF of (2) is

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}.$$

Alternatively, we could have solved the system with augmented matrix (2) using “back-substitution” as before: use the bottom row to see that

$$2x_3 = -2, \quad \text{so} \quad x_3 = -1,$$

and gone up level-by-level. Thus, we see that the original problem is solved by

$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}.$$

There is a good example in Lay with a bigger matrix A , as well, which also demonstrates the method.

2 Finding L and U

Now that you have seen how to solve linear systems given the LU factorization, I should show you an algorithm to compute L and U . It is a sad fact of life that an LU factorization of A is not possible for all matrices A , but we will see that a particular condition on A is enough to guarantee that a factorization exists.

Let A be a matrix which can be reduced to an echelon form U using only operations that add a multiple of one row to another *lower* row—that is, a row below it. Then we have

$$E_1 E_2 \dots E_p A = U$$

for some elementary matrices E_1, \dots, E_p , which are lower triangular. It can be proved that the inverse of a lower triangular matrix is always lower triangular, and that the product of two lower triangular matrices is lower triangular. Thus, if we rewrite the above as

$$A = E_p^{-1} \dots E_1^{-1}U,$$

we see that $L = E_p^{-1} \dots E_1^{-1}$ is lower triangular, and we have our desired LU decomposition.

The above also tells us how to compute U , but what about L ? Note that by our description, we have

$$E_1 \dots E_p L = I,$$

so the same row operations that reduce A to U also reduce L to I . This gives rise to our two-step algorithm for computing the LU factorization:

1. Reduce A to a row echelon form U using only operations of the type mentioned above.
2. Place entries in L so that the same sequence of operations reduces L to I .

Example 2.1. Consider the A from our previous example,

$$A = \begin{bmatrix} 4 & 3 & 3 \\ 4 & 2 & 2 \\ 4 & 0 & 2 \end{bmatrix}.$$

To compute a row echelon form of A , we add -1 times the first row of A to the second row of A , then add -1 times the first row of A to the third row of A . These two operations are of the form that we want, and the computation proceeds next by adding (-3) times row two to row three:

$$\begin{aligned} A = \begin{bmatrix} 4 & 3 & 3 \\ 4 & 2 & 2 \\ 4 & 0 & 2 \end{bmatrix} &\longrightarrow \begin{bmatrix} 4 & 3 & 3 \\ 0 & -1 & -1 \\ 4 & 0 & 2 \end{bmatrix} \\ &\longrightarrow \begin{bmatrix} 4 & 3 & 3 \\ 0 & -1 & -1 \\ 0 & -3 & -1 \end{bmatrix} \\ &\longrightarrow \begin{bmatrix} 4 & 3 & 3 \\ 0 & -1 & -1 \\ 0 & 0 & 2 \end{bmatrix} = U. \end{aligned}$$

Now we want L to be a matrix such that these row operations reduce L to I . So first, we want the first column of L to have a 1 in its topmost entry and then be such that when we do the same operations we would use to clear out the entries below the pivot of A , we get zeros in L too. After a moment's reflection, you can see that this is satisfied if we take the first column of L to be the first column of A divided by the value at the first pivot position (4), so we proceed:

$$L = \begin{bmatrix} 1 & * & * \\ 1 & * & * \\ 1 & * & * \end{bmatrix}$$

We want to have 0 in the topmost entry of the second column, 1 in the middle entry, and we want the bottom entry to be something that gets reduced to zero after performing the same row operations as on A . The way that we do this is look at the part in the computation of U *after* we have cleared the first column of A ; i.e., the matrix

$$\begin{bmatrix} 4 & 3 & 3 \\ 0 & -1 & -1 \\ 0 & -3 & -1 \end{bmatrix} \quad (3)$$

From this point, the only operations we will do on A that will affect the second column of L are the ones which clear below the pivot in the center of the above matrix. So again, the way we find the entries of the second column of L is to start with a 0 up top, and then the rest of the column is made up of the part of the middle column of (3) lying below the pivot, all divided by the value at the pivot (-1). Repeating this procedure gives

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 3 & 1 \end{bmatrix}.$$

3 Applications to circuits

Here I will talk about applications of the decomposition to circuits. I will follow closely the presentation in Lay, so I will not comment further except to say that you should read this section.