IE170: Algorithms in Systems Engineering: Lecture 30

Jeff Linderoth

Department of Industrial and Systems Engineering
Lehigh University

April 18, 2007

Solving Linear Systems

- Last time we learned about how to solve systems $Ax = b$, when A was symmetric and positive-definite.
- The key was to factor the matrix into two triangular matrices $A = LU$
- In the case that A is SPD, then we can always do this, and in fact $U = L^T$.
- What if A is *not* SPD?
- The workhorse in this case is the LU-decomposition
- LU-decomposition is very related to (the well-known) Gaussian elimination, a fact we will try to make clear today...

Gaussian Elimination

- An example for today. Let's solve it...

$$\begin{aligned} x_1 + x_2 + 2x_3 &= 3 \\ 2x_1 + 3x_2 + x_3 &= 2 \\ 3x_1 - x_2 - x_3 &= 6 \end{aligned}$$

  - Subtract twice first equation from the second
  - Subtract 3 times the first equation from the third
  - Then add 4 times second equation to the third
- You've made a triangular system!
- What were the matrices that produce this?

Elementary Dear Watson!

- We reduced the columns by taking linear combinations of the rows of the matrix.
- This implies that the reduction process can be thought of as a multiplication of A on the left by some matrix
- What does the matrix look like?
- It is an elementary matrix of the form

$$E = I - uv^T$$

- In fact, it's a special form of an elementary matrix: It will be a unit lower triangular matrix with multipliers only in one column

The Elimination Matrix

- Let's find a matrix M_1 that reduces the first column of A.

$$M_1 = \begin{pmatrix} 1 & & & & \\ -m_{21} & 1 & & & \\ -m_{31} & 1 & 1 & & \\ \vdots & \vdots & \ddots & & \\ -m_{n1} & 0 & 0 & \cdots & 1 \end{pmatrix}$$

- By our properties of matrix multiplication, this matrix
  - Leaves the first row of A alone
  - Takes $-m_{21}$ times the first row, adds the second row, and puts this in the second row of the new matrix M_1A
  - Takes $-m_{31}$ times the first row of A, adds the third row, and puts this in the third row of the new matrix M_1A
  - (And So On...)

The UpShot

- To eliminate the first column, we want

$$m_{i1} = \frac{a_{i1}}{a_{11}}$$

- Note: These were exactly the multipliers we used in our simple example
- a_{11} is called the pivot element, and this reduction only works if the pivot element is $\neq 0$
- Next time: What happens if pivot element is 0 (or small)

Let's Carry On

$$M_1A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{pmatrix}$$

- To reduce the second column, we would like a matrix

$$M_1 = \begin{pmatrix} 1 & & & & \\ 0 & 1 & & & \\ 0 & -m_{32} & 1 & & \\ \vdots & \vdots & \ddots & & \\ 0 & -m_{n2} & 0 & \cdots & 1 \end{pmatrix}$$

Matrix Effect

- Again, the matrix M_2 will...
  - Leave first row of M_1A unchanged in M_2M_1A
  - Leave second row of M_1A unchanged M_2M_1A
  - Take $-m_{32}$ times second row + third row in M_2M_1A

Lather Rinse Repeat

- Repeat this $n-1$ times
- In the end, we get $M_{n-1} \cdots M_2M_1A = U$
- Fact: The product of unit lower triangular matrices is unit lower triangular
- So in the end we have $MA = U$, with M unit lower triangular, and U upper traiangular
- This process is known as Gaussian Elimination, and the matrix M is known as the product form of the LU factorization

Finding LU Directly

- Here is a recursive method for finding the LU factorization
- We'll divide the matrix A into four pieces:

$$A = \left[\begin{array}{c|ccc} a_{11} & a_{12} & \cdots & a_{1n} \\ \hline a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{array}\right] \quad (1)$$

$$= \begin{bmatrix} a_{11} & w^T \\ v & A' \end{bmatrix} \quad (2)$$

- Next, we'll use use row operations to change v into the zero vector and record the operations in another matrix.

Finding the LU Decomposition (cont.)

- By simple multiplication, you can verify the following factorization of A:

$$A = \begin{bmatrix} a_{11} & w^T \\ v & A' \end{bmatrix} \quad (3)$$

$$= \begin{bmatrix} 1 & 0 \\ v/a_{11} & I \end{bmatrix} \begin{bmatrix} a_{11} & w^T \\ 0 & A' - vw^T/a_{11} \end{bmatrix} \quad (4)$$

- We can show that if A is nonsingular, then so is $A' - vw^T/a_{11}$.
- So we can recursively call the method to factor the $(n-1) \times (n-1)$ matrix $A' - vw^T/a_{11}$.
- Applying this recursion n times yields the desired factorization

Finding the LU Decomposition (cont.)

- To see how to get the factorization from the recursive application of the algorithm, we have the following.

$$A = \begin{bmatrix} 1 & 0 \\ v/a_{11} & I \end{bmatrix} \begin{bmatrix} a_{11} & w^T \\ 0 & A' - vw^T/a_{11} \end{bmatrix} \quad (5)$$

$$= \begin{bmatrix} 1 & 0 \\ v/a_{11} & I \end{bmatrix} \begin{bmatrix} a_{11} & w^T \\ 0 & L'U' \end{bmatrix} \quad (6)$$

$$= \begin{bmatrix} 1 & 0 \\ v/a_{11} & L' \end{bmatrix} \begin{bmatrix} a_{11} & w^T \\ 0 & U' \end{bmatrix} \quad (7)$$

- This shows how to obtain the factorization recursively.
- This can also be done iteratively and "in place."

The Algorithm

```
LU-Decomposition(A)
n ← rows[L]
for k ← 1 to n
do
    u_kk ← a_kk
    for i ← 1 to n
    do
        ℓ_ik ← a_ik / u_kk
        u_ki ← a_ki
    for i ← k + 1 to n
    do
        for j ← k + 1 to n
        do
            a_ij ← a_ij − ℓ_ik u_kj
```

LU ≈ Gaussian Elimination

- We either have $A = LU$ or we have $MA = U$
- M is unit lower triangular, and in fact the inverse of a unit lower triangular matrix is unit lower-triangular, so $A = M^{-1}U$, and since the elements of L and U are unique, it must be that $L = M^{-1}$
- Because of the special structure of M, we have a (fairly) remarkable relationship

LU ≈ Gaussian Elimination

- The relationship is the following:

$$M^{-1} = (M_{n-1} \cdots M_2 M_1)^{-1} = M_1^{-1} M_2^{-1} \cdots = L$$

$$L = \begin{pmatrix} 1 & & & & \\ m_{21} & 1 & & & \\ m_{31} & m_{32} & 1 & & \\ \vdots & & \ddots & & \\ m_{n1} & m_{n2} & m_{n3} & \cdots & 1 \end{pmatrix}$$

where the m_{ik} are the multipliers from Gaussian elimination!

- So L and U can be derived directly from the elimination process:

$$\ell_{ik} = m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \qquad u_{kj} = a_{kj}^{(k)}$$