Lecture 10

LU, Cholesky

N. Bell

Department of Computer Science
University of Illinois at Urbana-Champaign

September 27, 2006

Factorization Methods

- LU factorization
- Cholesky factorization
- Use of the backslash operator

LU Factorization

Find L and U such that

$$A = LU$$

and L is lower triangular, and U is upper triangular.

$$L = \begin{bmatrix} 1 & 0 & \cdots & & 0 \\ \ell_{2,1} & 1 & 0 & & 0 \\ \ell_{3,1} & \ell_{3,2} & 1 & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ \ell_{n,1} & \ell_{n,2} & \cdots & \ell n-1, n & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \cdots & u_{1,n} \\ 0 & u_{2,2} & u_{2,3} & \cdots & u_{2,n} \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & & & u_{n-1,n} \\ 0 & 0 & & & u_{n,n} \end{bmatrix}$$

Since L and U are triangular, it is easy to apply their inverses.

Why use LU?

- Decompose once, solve many righthand sides quickly. $O(Mn^3)$ with GE vs. $O(n^3 + Mn^2)$ with LU
- Given $A = LU$ you can compute A^{-1}, $\det(A)$, $\mathrm{rank}(A)$, $\ker(A)$ etc.

LU Factorization

Since L and U are triangular, it is easy to apply their inverses.
Consider the solution to $Ax = b$.

$$A = LU \Longrightarrow (LU)x = b$$

Regroup, matrix multiplication is associative

$$L(Ux) = b$$

Let $Ux = y$, then

$$Ly = b$$

Since L is triangular it is easy (without Gaussian elimination) to compute

$$y = L^{-1}b$$

This expression should be interpreted as "Solve $Ly = b$ with a forward substitution."

LU Factorization

Now, since y is known, solve for x

$$x = U^{-1}y$$

which is interpreted as “Solve $Ux = y$ with a backward substitution.”

LU Factorization

Listing 1: Solve Ax

```
Factor A into L and U
Solve Ly = b for y          use forward substitution
Solve Ux = y for x          use backward substitution
```

LU Factorization

Consider the following matrix

$$A = \begin{bmatrix} 6 & -2 & 2 \\ 12 & -8 & 6 \\ 3 & -13 & 9 \end{bmatrix}$$

After reducing the first column we have

$$A_1 = \begin{bmatrix} 6 & -2 & 2 \\ 0 & -4 & 2 \\ 0 & -12 & 8 \end{bmatrix}$$

Notice that we can write A_1 as the product of matrices $M_1 A$

$$A_1 = \begin{bmatrix} 6 & -2 & 2 \\ 0 & -4 & 2 \\ 0 & -12 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -\frac{1}{2} & 0 & 1 \end{bmatrix} \begin{bmatrix} 6 & -2 & 2 \\ 12 & -8 & 6 \\ 3 & -13 & 9 \end{bmatrix} = M_1 A$$

LU Factorization

Seperating M_1 into two pieces shows why this is so

$$\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -\frac{1}{2} & 0 & 1 \end{bmatrix} \begin{bmatrix} 6 & -2 & 2 \\ 12 & -8 & 6 \\ 3 & -13 & 9 \end{bmatrix}$$

$$\left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{bmatrix} \right) \begin{bmatrix} 6 & -2 & 2 \\ 12 & -8 & 6 \\ 3 & -13 & 9 \end{bmatrix}$$

$$\begin{bmatrix} 6 & -2 & 2 \\ 12 & -8 & 6 \\ 3 & -13 & 9 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{bmatrix} \begin{bmatrix} 6 & -2 & 2 \\ 12 & -8 & 6 \\ 3 & -13 & 9 \end{bmatrix}$$

The subdiagonal values are simply the *negatives of the multipliers*.

LU Factorization

We can continue applying matrices to reduce the other columns

$$\begin{bmatrix} 6 & -2 & 2 \\ 12 & -8 & 6 \\ 3 & -13 & 9 \end{bmatrix} = A$$

$$\begin{bmatrix} 6 & -2 & 2 \\ 0 & -4 & 2 \\ 0 & -12 & 8 \end{bmatrix} = M_1 A = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -\frac{1}{2} & 0 & 1 \end{bmatrix} \begin{bmatrix} 6 & -2 & 2 \\ 12 & -8 & 6 \\ 3 & -13 & 9 \end{bmatrix}$$

$$\begin{bmatrix} 6 & -2 & 2 \\ 0 & -4 & 2 \\ 0 & 0 & 2 \end{bmatrix} = M_2 M_1 A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -\frac{1}{2} & 0 & 1 \end{bmatrix} \begin{bmatrix} 6 & -2 & 2 \\ 12 & -8 & 6 \\ 3 & -13 & 9 \end{bmatrix}$$

LU Factorization

In our example we determined matrices M_1 and M_2 such that $M_2 M_1 A = U$ was upper triangular. In the general case we have

$$(M_n M_{n-1} \ldots M_2 M_1) A = U$$

So it follows

$$A = (M_n M_{n-1} \ldots M_2 M_1)^{-1} U$$

Now we need to show that $(M_n M_{n-1} \ldots M_2 M_1)^{-1}$ is lower triangular.

LU Factorization

Recall M_1 and M_2 from our example

$$M_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -\frac{1}{2} & 0 & 1 \end{bmatrix} \qquad M_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3 & 1 \end{bmatrix}$$

Notice that

$$M_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{bmatrix} \qquad M_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 3 & 1 \end{bmatrix}$$

and

$$M_1^{-1} M_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ \frac{1}{2} & 3 & 1 \end{bmatrix}$$

Notice that the subdiagonals of L are the *multipliers* used in GE.

LU Factorization

Therefore we have factored $A = LU$

$$\begin{bmatrix} 6 & -2 & 2 \\ 12 & -8 & 6 \\ 3 & -13 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ \frac{1}{2} & 3 & 1 \end{bmatrix} \begin{bmatrix} 6 & -2 & 2 \\ 0 & -4 & 2 \\ 0 & 0 & 2 \end{bmatrix}$$

Caveat: Notice that the LU factorization can be stored in the same space as A. Some implementations overwrite A in the process of computing LU.

When LU goes wrong

Like GE without pivoting, LU fails for some matrices

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

However with pivoting, the LU always exists. In practice, you get a factorization

$$LU = PA$$

where P is a permutation matrix. This even works when A is singular or rectangular.

Using LU in MATLAB

Given a matrix A, MATLAB's **lu** returns LU and P such that

$$LU = PA$$

We can use this to solve $Ax = b \Leftrightarrow LUx = Pb$ as follows

```
A = rand(100,100);
b = rand(100,1);

[L,U,P] = lu(A);

x = U\(L\(P*b));
```

Cholesky Factorization

- A must be symmetric and positive definite (SPD)
- Factors A into LL^t, like LU where $U = L^t$
- For SPD matrices, pivoting is not required
- Cholesky factorization requires one half as many flops as LU factorization. Since pivoting is not required, Cholesky factorization will be more than twice as fast as LU factorization since data movement is avoided.
- Refer to the text for a view of the algorithm
- Use built-in `chol` function for routine work

SPD matrices and you

A matrix is Positive Definite (PD) if for all choices of the vector x the following holds

$$x^t A x > 0$$

A matrix is SPD if it is symmetric and positive definite.
This definition is not particularly intuitive.

SPD matrices and you

Task: find the square of the length of a vector x.
Solution: compute $x \cdot x = x^t x$ using the dot product or innerproduct.
Note that this is the same as precisely the same as

$$x^t I x$$

where I is the identity matrix. In Euclidean space, our *innerproduct* is simply I (which is SPD) and our notion of the *norm* of a vector is $\sqrt{x^t I x}$. In general, innerproducts tell us how to measure things.

SPD matrices are innerproducts

Imagine $2D$ space, except that moving in the x-direction is twice as expensive as moving in the y-direction. New innerproduct

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Taking the norm of a vector $[x, y]$ using this innerproduct represents our modified notion of length.

SPD matrices FTW!

In practice we like SPD matrices because

- They are non-singular (why?)
- Dense Cholesky is faster than LU
- Sparse Cholesky is faster than Sparse LU
- Iterative solvers like Conjugate Gradient (CG)

SPD matrices arise in the discritization of many physical problems and are by no means exotic.

Backslash Redux

The \ operator examines the coefficient matrix before attempting to solve the system.
\ uses:

- A triangular solve if A is triangular, or a permutation of a triangular matrix
- Cholesky factorization and triangular solves if A is symmetric and the diagonal elements of A are positive (*and* if the subsequent Cholesky factorization does not fail.)
- LU factorization if A is square and the preceding conditions are not met.
- QR factorization to obtain the least squares solution if A is not square.