CME 302: NUMERICAL LINEAR ALGEBRA
FALL 2005/06
LECTURE 6

GENE H. GOLUB

1. Issues with Floating-point Arithmetic

We conclude our discussion of floating-point arithmetic by highlighting two issues that frequently arise in practice.

First of all, relationships among numbers that are known to be true in exact arithmetic do not necessarily hold when using floating-point arithmetic. For example, suppose that $x > y > 0$, and that $a > 0$. Then, in exact arithmetic, $ax > ay > 0$, but in floating-point arithmetic, we can only assume that $ax \geq ay \geq 0$.

Second, the order in which floating-point arithmetic operations are performed can drastically affect the result. For example, suppose that we want to compute e^{-x}, where $x > 0$. Using the Taylor series for e^x, we know that

$$e^{-x} = 1 - x + \frac{x^2}{2} - \frac{x^3}{3!} + \cdots$$

but for sufficiently large x, this means obtaining small numbers by subtracting larger ones, and therefore the computation is susceptible to a phenomenon known as *catastrophic cancellation*, in which subtracting numbers that are nearly equal causes the loss of significant digits (since such digits in the result of the subtraction are equal to zero). An alternative approach is to compute

$$e^{-x} = \frac{1}{1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \cdots}$$

which avoids this problem.

As a rule, it is best to try to avoid subtractions altogether, instead trying to add numbers that are guaranteed to be the same sign. For example, given the quadratic equation $x^2 + bx + c = 0$, we can compute the roots by applying the quadratic formula as follows:

$$x_+ = \frac{-b + \operatorname{sgn}(-b)\sqrt{b^2 - 4c}}{2}, \quad x_- = \frac{c}{x_+}.$$

2. Rank-1 Updating and the Inverse

Suppose that we have solved the problem $A\mathbf{x} = \mathbf{b}$ and we wish to solve the perturbed problem

$$(A + \mathbf{u}\mathbf{v}^\top)\mathbf{y} = \mathbf{b}.$$

Such a perturbation is called a *rank-one update* of A, since the matrix $\mathbf{u}\mathbf{v}^\top$ has rank 1. As an example, we might find that there was an error in the element a_{11} and we update it with the value $\bar{a}_{11}$. We can accomplish this update by setting

$$\bar{A} = A + (\bar{a}_{11} - a_{11})\mathbf{e}_1\mathbf{e}_1^\top, \quad \mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Date: October 18, 2005, version 1.0.

Notes originally due to James Lambers. Minor editing by Lek-Heng Lim.

For a general rank-one update, we can use the *Sherman-Morrison formula*, which we will derive here. Multiplying through the equation $(A + \mathbf{u}\mathbf{v}^\top)\mathbf{y} = \mathbf{b}$ by A^{-1} yields

$$(I + A^{-1}\mathbf{u}\mathbf{v}^\top)\mathbf{y} = A^{-1}\mathbf{b} = \mathbf{x}.$$

We therefore need to find $(I + \mathbf{w}\mathbf{v}^\top)^{-1}$ where $\mathbf{w} = A^{-1}\mathbf{u}$. We assume that $(I + \mathbf{w}\mathbf{v}^\top)^{-1}$ is a matrix of the form $(I + \sigma\mathbf{w}\mathbf{v}^\top)$ where σ is some constant. From the relationship

$$(I + \mathbf{w}\mathbf{v}^\top)(I + \sigma\mathbf{w}\mathbf{v}^\top) = I$$

we obtain

$$\sigma\mathbf{w}\mathbf{v}^\top + \mathbf{w}\mathbf{v}^\top + \sigma\mathbf{w}\mathbf{v}^\top\mathbf{w}\mathbf{v}^\top = 0.$$

However, the quantity $\mathbf{v}^\top\mathbf{w}$ is a scalar, so this simplifies to

$$(\sigma + 1 + \sigma\mathbf{v}^\top\mathbf{w})\mathbf{w}\mathbf{v}^\top = 0$$

which yields

$$\sigma = -\frac{1}{1 + \mathbf{v}^\top\mathbf{w}}.$$

It follows that the solution $\mathbf{y}$ to the perturbed problem is given by

$$\mathbf{y} = (I + \sigma\mathbf{w}\mathbf{v}^\top)\mathbf{x} = \mathbf{x} + \sigma\mathbf{v}^{\mathbf{x}}\mathbf{w}$$

and the perturbed inverse is given by

$$\begin{aligned}(A + \mathbf{u}\mathbf{v}^\top)^{-1} &= (I + A^{-1}\mathbf{u}\mathbf{v}^\top)^{-1}A \\ &= \left(I - \frac{1}{1 + \mathbf{v}^\top\mathbf{w}}\mathbf{w}\mathbf{v}^\top\right)A^{-1} \\ &= A^{-1} - \frac{1}{1 + \mathbf{v}^\top A^{-1}\mathbf{u}}A^{-1}\mathbf{u}\mathbf{v}^\top A^{-1}.\end{aligned}$$

An efficient algorithm for solving the perturbed problem $(A + \mathbf{u}\mathbf{v}^\top)\mathbf{y} = \mathbf{b}$ can therefore proceed as follows:

(1) Solve $A\mathbf{x} = \mathbf{b}$
(2) Solve $A\mathbf{w} = \mathbf{u}$
(3) Compute $\sigma = -\frac{1}{1+\mathbf{v}^\top\mathbf{w}}$
(4) Compute $\mathbf{y} = \mathbf{x} + \sigma(\mathbf{v}^\top\mathbf{x})\mathbf{w}$

An alternative approach is to note that

$$\begin{aligned}(A + \mathbf{u}\mathbf{v}^\top)^{-1} &= [A(I + A^{-1}\mathbf{u}\mathbf{v}^\top)]^{-1} \\ &= (I + \sigma A^{-1}\mathbf{u}\mathbf{v}^\top)A^{-1} \\ &= A^{-1} + \sigma A^{-1}\mathbf{u}\mathbf{v}^\top A^{-1}\end{aligned}$$

which yields

$$\begin{aligned}(A + \mathbf{u}\mathbf{v}^\top)^{-1}\mathbf{b} &= A^{-1}(I + \sigma\mathbf{u}\mathbf{v}^\top A^{-1})\mathbf{b} \\ &= A^{-1}(\mathbf{b} + \sigma(\mathbf{v}^\top A^{-1}\mathbf{b})\mathbf{u})\end{aligned}$$

and therefore we can solve $(A + \mathbf{u}\mathbf{v}^\top)\mathbf{y} = \mathbf{b}$ by solving a problem of the form $A\mathbf{x} = \mathbf{b}$ where the right-hand side $\mathbf{b}$ is perturbed.

3. Gaussian Elimination

We often wish to solve

$$A\mathbf{x} = \mathbf{b}$$

where A is an $m \times n$ matrix and $\mathbf{b}$ is an m-vector. For now, we assume that $m = n$ and that A has rank n. If we can write

$$A = PQ$$

then we can solve the system $A\mathbf{x} = PQ\mathbf{x} = \mathbf{b}$ by solving

$$P\mathbf{y} = \mathbf{b}$$
$$Q\mathbf{x} = \mathbf{y}$$

Therefore we would like to find such a decomposition where the above systems are simple to solve. We now discuss a few scenarios where this is the case.

(1) If the matrix A is diagonal, then the system $A\mathbf{x} = \mathbf{b}$ has the solution $x_i = b_i/a_{ii}$ for $i = 1, \ldots, n$. The solution can be computed in only n divisions. Furthermore, each component of $\mathbf{x}$ can be computed independently, and therefore the algorithm can be parallelized.

(2) If $AA^\top = I$, then $A\mathbf{x} = \mathbf{b}$ can be solved simply by computing the matrix-vector product $\mathbf{x} = A^\top\mathbf{b}$. This requires only $O(n^2)$ multiplications and additions, and can also be parallelized.

(3) If A is a lower triangular matrix, i.e. if $a_{ij} = 0$ for $i < j$, then the system of equations $A\mathbf{x} = \mathbf{b}$ takes the form

$$\begin{array}{ccccccc} a_{11}x_1 & & & & & = & b_1 \\ a_{12}x_1 & + & a_{22}x_2 & & & = & b_2 \\ \vdots & & & \ddots & & & \vdots \\ a_{n1}x_1 & + & \cdots & + & a_{nn}x_n & = & b_n \end{array}$$

which can be solved by the process of *forward substitution*

$$\begin{aligned} x_1 &= b_1/a_{11} \\ x_2 &= (b_2 - a_{21}x_1)/a_{22} \\ &\vdots \\ x_n &= \left(b_n - \textstyle\sum_{j=1}^{n-1} a_{nj}x_j\right)/a_{nn} \end{aligned}$$

This algorithm cannot be parallelized, since each component x_i depends on x_j for $j < i$, but it is still efficient, as it requires only $O(n^2)$ multiplications and additions. In the case where A is an *upper triangular matrix*, i.e. $a_{ij} = 0$ whenever $i > j$, a similar process known as *back substitution* can be used.

Note that the solution method for the problem $A\mathbf{x} = \mathbf{b}$ depends on the structure of A. A may be a sparse or dense matrix, or it may have one of many well-known structures, such as being a banded matrix, or a Hankel matrix. We will consider the general case of a dense, unstructured matrix A, and obtain a decomposition $A = LU$, where L is lower triangular and U is upper triangular.

This decomposition is achieved using Gaussian elimination. We write out the system $Ax = b$ as

$$\begin{array}{ccccccc} a_{11}x_1 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ \vdots & & & & \vdots & & \vdots \\ a_{n1}x_1 & + & \cdots & + & a_{nn}x_n & = & b_n \end{array}$$

We proceed by multiplying the first equation by $-a_{21}/a_{11}$ and adding it to the second equation, and in general multiplying the first equation by $-a_{i1}/a_{11}$ and adding it to equation i. We obtain

the following equivalent system

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ 0x_1 & + & a'_{22}x_2 & + & \cdots & + & a'_{2n}x_n & = & b'_2 \\ \vdots & & & & & & \vdots & & \vdots \\ 0x_1 & + & a'_{n2}x_2 & + & \cdots & + & a'_{nn}x_n & = & b'_n \end{array}$$

Continuing in this fashion, adding multiples of the second equation to each subsequent equation to make all elements below the diagonal equal to zero, we obtain an upper triangular system.

This process of transforming A to an upper triangular matrix U is equivalent to multiplying A by a sequence of matrices to obtain U. Specifically, we have $M_1 A = A_2$ where

$$A_2 = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix}$$

and

$$M_1 = \begin{bmatrix} 1 & & & 0 \\ -\ell_{21} & 1 & & \\ \vdots & 0 & \ddots & \\ -\ell_{n1} & & & 1 \end{bmatrix}, \quad \ell_{i1} = \frac{a_{i1}}{a_{11}}.$$

Similarly, if we define M_2 by

$$M_2 = \begin{bmatrix} 1 & & & & \\ 0 & 1 & & & \\ 0 & -\ell_{32} & 1 & & \\ \vdots & \vdots & & \ddots & \\ 0 & -\ell_{n2} & & & 1 \end{bmatrix}, \quad \ell_{i2} = \frac{a_{i2}^{(2)}}{a_{22}^{(2)}}$$

then

$$M_2 A_2 = A_3 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & a_{n3}^{(3)} & \cdots & a_{nn}^{(3)} \end{bmatrix}$$

In general, we have

$$M_k = \begin{bmatrix} 1 & & & & & \\ 0 & \ddots & & & & \\ \vdots & \ddots & 1 & & & \\ \vdots & & -\ell_{k+1,k} & 1 & & \\ \vdots & & \vdots & & \ddots & \\ 0 & & -\ell_{nk} & & & 1 \end{bmatrix}$$

and

$$M_{n-1} M_{n-2} \cdots M_1 A = A_n \equiv \begin{bmatrix} u_{11} & \cdots & & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & u_{nn} \end{bmatrix}$$

or, equivalently,

$$A = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} U.$$

It turns out that M_j^{-1} is very easy to compute. We claim that

$$M_1^{-1} = \begin{bmatrix} 1 & & & 0 \\ \ell_{21} & 1 & & \\ \vdots & 0 & \ddots & \\ \ell_{n1} & & & 1 \end{bmatrix}$$

To see this, consider the product

$$M_1 M_1^{-1} = \begin{bmatrix} 1 & & & 0 \\ -\ell_{21} & 1 & & \\ \vdots & 0 & \ddots & \\ -\ell_{n1} & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ \ell_{21} & 1 & & \\ \vdots & 0 & \ddots & \\ \ell_{n1} & & & 1 \end{bmatrix}$$

which can easily be verified to be equal to the identity matrix. In general, we have

$$M_k^{-1} = \begin{bmatrix} 1 & & & & & \\ 0 & \ddots & & & & \\ \vdots & \ddots & 1 & & & \\ \vdots & & \ell_{k+1,k} & 1 & & \\ \vdots & & \vdots & & \ddots & \\ 0 & & \ell_{nk} & & & 1 \end{bmatrix}$$

Now, consider the product

$$M_1^{-1} M_2^{-1} = \begin{bmatrix} 1 & & & 0 \\ \ell_{21} & 1 & & \\ \vdots & 0 & \ddots & \\ \ell_{n1} & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & \\ 0 & 1 & & & \\ 0 & \ell_{32} & 1 & & \\ \vdots & \vdots & & \ddots & \\ 0 & \ell_{n2} & & & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & & & & \\ \ell_{21} & 1 & & & \\ \vdots & \ell_{32} & 1 & & \\ \vdots & \vdots & & \ddots & \\ \ell_{n1} & \ell_{n2} & & & 1 \end{bmatrix}$$

It can be shown that

$$M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} = \begin{bmatrix} 1 & & & & \\ \ell_{21} & \ddots & & & \\ \vdots & \ell_{32} & \ddots & & \\ \vdots & \vdots & \ddots & \ddots & \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{n,n-1} & 1 \end{bmatrix}$$

It follows that under proper circumstances, we can write $A = LU$ where

$$L = \begin{bmatrix} 1 & & & & \\ \ell_{21} & \ddots & & & \\ \vdots & \ell_{32} & \ddots & & \\ \vdots & \vdots & \ddots & \ddots & \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{n,n-1} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} u_{11} & \cdots & & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & u_{nn} \end{bmatrix}$$

Given this decomposition, we can easily compute the determinant of A:

$$\det A = \det LU = \det L \det U = 1 \cdot \prod_{i=1}^{n} u_{ii}$$

What exactly are proper circumstances? We must have $a_{kk}^{(k)} \neq 0$, or we cannot proceed with the decomposition. For example, if

$$A = \begin{bmatrix} 0 & 1 & 11 \\ 3 & 7 & 2 \\ 2 & 9 & 3 \end{bmatrix} \quad \text{or} \quad A = \begin{bmatrix} 1 & 3 & 4 \\ 2 & 6 & 4 \\ 7 & 1 & 2 \end{bmatrix}$$

Gaussian elimination will fail. In the first case, it fails immediately; in the second case, it fails after the subdiagonal entries in the first column are zeroed, and we find that $a_{22}^{(k)} = 0$. In general, we must have $\det A_{ii} \neq 0$ for $i = 1, \ldots, n$ where

$$A_{ii} = \begin{bmatrix} a_{11} & \cdots & a_{1i} \\ \vdots & & \vdots \\ a_{i1} & \cdots & a_{ii} \end{bmatrix}$$

for the LU factorization to exist.

How can we obtain the LU factorization for a general non-singular matrix? If A is nonsingular, then *some* element of the first column must be nonzero. If $a_{i1} \neq 0$, then we can interchange row i with row 1 and proceed. This is equivalent to multiplying A by a *permutation matrix* Π_1 that interchanges row 1 and row i:

$$\Pi_1 = \begin{bmatrix} 0 & \cdots & \cdots & 0 & 1 & 0 & \cdots & 0 \\ & 1 & & & & & & \\ & & \ddots & & & & & \\ & & & 1 & & & & \\ 1 & 0 & \cdots & \cdots & 0 & \cdots & \cdots & 0 \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{bmatrix}$$

Thus $M_1 \Pi_1 A = A_2$. Then, since A_2 is nonsingular, some element of column 2 of A_2 below the diagonal must be nonzero. Proceeding as before, we compute $M_2 \Pi_2 A_2 = A_3$, where Π_2 is another permutation matrix. Continuing, we obtain

$$A = (M_{n-1} \Pi_{n-1} \cdots M_1 \Pi_1)^{-1} U.$$

It can easily be shown that $\Pi A = LU$ where Π is a permutation matrix.

Most often, Π_i is chosen so that row i is interchanged with row j, where $a_{ij}^{(i)} = \max_{i \leq j \leq n} |a_{ij}^{(i)}|$. This guarantees that $|\ell_{ij}| \leq 1$. This strategy is known as *partial pivoting*. Another common strategy, *complete pivoting*, uses both row and column interchanges to ensure that at step i of the algorithm, the element a_{ii} is the largest element in absolute value from the entire submatrix obtained by deleting the first $i - 1$ rows and columns. Often, however, other criteria is used to guide pivoting, due to considerations such as preserving sparsity.

Department of Computer Science, Gates Building 2B, Room 280, Stanford, CA 94305-9025
E-mail address: golub@stanford.edu