CS412: Lecture #18

Mridul Aanjaneya

March 24, 2015

Solving $Ax = b$:

- Without pivoting:

$$Ax = L\underbrace{Ux}_{=y} = b$$

1. Solve $Ly = b$ through forward substitution.
2. Solve $Ux = y$ through back substitution to obtain the solution x.

Note that if we have multiple systems $Ax_i = b_i$, we only need to incur the cost of computing an LU decomposition of A once.

- With pivoting:

$$Ax = b \Leftrightarrow PAx = Pb$$

1. Solve $Ly = Pb$ using forward substitution.
2. Solve $Ux = y$ using backward substitution to obtain the solution x.

Note that switching two rows twice puts the rows back, so P is its own inverse. Also note that P is an *orthogonal* matrix, i.e., $P^{-1} = P^T$, so in general $P^{-1} = P^T = P$. The process shown above is called *partial pivoting* because it switches rows to always get the largest diagonal element. This is in contrast to *full pivoting* (see below) which can switch both rows and columns to obtain the largest diagonal element. Partial pivoting gives $A = LU$ where $U = M_{n-1}P_{n-1}\dots M_1P_1A$ and $L = P_1L_1\dots P_{n-1}L_{n-1}$ where U is upper triangular, but L is a permutation of a lower triangular matrix. It turns out that we can write L as $L = P_1\dots P_{n-1}L_1^P\dots L_{n-1}^P$ where each $L_k^P = I + (P_{n-1}\dots P_{k+1}m_k)e_k^T$ has the same form as L_k. Thus, we can write $PA = L^PU$ where $L^P = L_1^P\dots L_{n-1}^P$ is *lower triangular* and $P = P_{n-1}\dots P_1$ is the total permutation matrix.

Full pivoting

In this case, when we are in the kth step of the Gaussian Elimination/LU procedure, we pick the pivot element among the *entire* $(n-k+1)\times(n-k+1)$ lower rightmost submatrix of A. For example, if $k = 2$ and $Ax = b$

$$\begin{bmatrix} 1 & 2 & 5 & -1 \\ 0 & \boxed{0} & 3 & 1 \\ 0 & 4 & 1 & \boxed{-8} \\ 0 & -6 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 4 \\ 7 \\ 8 \\ 2 \end{bmatrix}$$

In this case, we can bring (-8) to the pivot position a_{22} by permuting *both* rows $2-3$ *and* columns $2-4$. Naturally, we will respectively swap rows $2-3$ of the RHS, and rows $2-4$ of the vector of unknowns. Thus, the equivalent system becomes

$$\begin{bmatrix} 1 & -1 & 5 & 2 \\ 0 & -8 & 1 & 4 \\ 0 & 1 & 3 & 0 \\ 0 & 3 & 0 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_4 \\ x_3 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 8 \\ 7 \\ 2 \end{bmatrix}$$

This process is encoded in the LU factorization using *two* permutation matrices P and Q such that $\boxed{PAQ = LU}$. The solution is then computed via

$$Ax = b \Rightarrow PA\underbrace{QQ^T}_{=I}x = Pb \Rightarrow (LU)(Q^Tx) = Pb$$

1. Solve $Ly = Pb$ using forward substitution.
2. Solve $Uz = y$ using back substitution.
3. Finally, $Q^Tx = z \Rightarrow QQ^Tx = Qz \Rightarrow \boxed{x = Qz}$ gives the solution!

To summarize:

- *Partial pivoting* permutes rows, such that the pivot element in the kth iteration is the largest number in the $(n-k+1)$ lower entries of the kth column. It is written, in the context of LU decomposition as

$$PA = LU \qquad (P = \text{permutation})$$

- *Full pivoting* selects the pivot element in the kth iteration as the largest element of the $(n-k+1)\times(n-k+1)$ lower rightmost sub-matrix of A. It operates by permuting rows *and* columns and leads to an LU decomposition of

$$PAQ = LU$$

However, there are certain categories of matrices for which we can safely use Gaussian elimination or LU decomposition *without* the need for pivoting (i.e., the pivot elements will never be problematically small).

Definition: A matrix A is called *diagonally dominant by columns* if the magnitude of every diagonal element is larger than the sum of the magnitudes of all other entries in the same column, i.e., for every $i = 1, 2, \ldots, n$ we have

$$|a_{ii}| > \sum_{j \neq i} |a_{ji}|$$

If the diagonal element exceeds in magnitude the sum of magnitudes of all other elements in its *row*, i.e., for every $i = 1, 2, \ldots, n$ we have

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|$$

then the matrix is called *diagonally dominant by rows.*

Definition: A symmetric matrix $A \in \mathbb{R}^{n \times n}$ is called *positive definite* (in short SPD for "symmetric positive definite"), if for any $x \in \mathbb{R}^n, x \neq 0$ we have $x^T Ax > 0$. If for any $x \in \mathbb{R}^n, x \neq 0$ we have $x^T Ax \geq 0$, the matrix is called positive semi-definite. If the respective properties are $x^T Ax < 0$ (or $x^T Ax \leq 0$) the matrix is called negative (semi) definite.

Definition: The kth *leading principal minor* of a matrix $A \in \mathbb{R}^{n \times n}$ is the determinant of the top-leftmost $k \times k$ sub-matrix of A. Thus, if we denote this minor by M_k:

$$M_1 = |a_{11}|, \quad M_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, \quad \ldots \quad M_k = \begin{vmatrix} a_{11} & \ldots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \ldots & a_{kk} \end{vmatrix}$$

Theorem 1. *If* all *leading principal minors (i.e., for $k = 1, 2, 3, \ldots, n$) of the symmetric matrix A are positive, then A is positive definite. If $M_k < 0$ for $k = odd$ and $M_k > 0$ for $k = even$, then A is negative definite.*

Theorem 2. *Pivoting is* not *necessary when A is diagonally dominant by columns, or symmetric and positive (or negative) definite.*

These "special" classes of matrices (which appear quite often in engineering and applied sciences) not only make LU decomposition more robust, but also open some additional possibilities for solving $Ax = b$.

Iterative methods for linear systems

The general idea is similar to the philosophy of iterative methods we saw for nonlinear equations, i.e., we proceed as follows:

- We write a (matrix) equation

$$x = Tx + c$$

in such a way that this equation is equivalent to $Ax = b$.

- We start with an initial guess $x^{(0)}$ for the solution of $Ax = b$.

- We iterate

$$x^{(k+1)} = Tx^{(k)} + c$$

- If properly designed the sequence $x^{(0)}, x^{(1)}, \dots, x^{(k)}, \dots$ *converges* to $x^\star$, which satisfies $x^\star = Tx^\star + c$ and consequently $Ax^\star = b$.

The Jacobi Method

We decompose

$$A = \underbrace{D}_{\text{diagonal}} - \underbrace{L}_{\text{lower triangular}} - \underbrace{U}_{\text{upper triangular}}$$

$$\begin{aligned} Ax &= b \\ \Rightarrow (D - L - U)x &= b \\ \Rightarrow Dx &= (L + U)x + b \\ \Rightarrow x = \underbrace{D^{-1}(L+U)}_{T}\, x + \underbrace{D^{-1}b}_{c} \quad (x = Tx + c) & \end{aligned}$$

Iteration: $\boxed{x^{(k+1)} = D^{-1}(L+U)x^{(k)} + D^{-1}b}$ or $\boxed{Dx^{(k+1)} = (L+U)x^{(k)} + b}$