STAT 309: MATHEMATICAL COMPUTATIONS I
FALL 2015
LECTURE 13

1. need for pivoting

- last time we showed that under proper circumstances, we can write $A = LU$ where

$$L = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \ell_{21} & 1 & 0 & \cdots & 0 \\ \ell_{31} & \ell_{32} & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{n,n-1} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} u_{11} & u_{12} & \cdots & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & \cdots & u_{2n} \\ 0 & 0 & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & u_{nn} \end{bmatrix}$$

- 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; note that both matrices are nonsingular
- 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 $i = 1, \dots, n$ 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
- the existence of LU factorization (without pivoting) can be guaranteed by several conditions, one example is *column*[1] *diagonal dominance*: if a nonsingular $A \in \mathbb{R}^{n\times n}$ satisfies

$$|a_{jj}| \geq \sum_{\substack{i=1 \\ i\neq j}}^{n} |a_{ij}|, \quad j = 1, \dots, n,$$

  then one can guarantee that Gaussian elimination as described above produces $A = LU$ with $|\ell_{ij}| \leq 1$
- there are necessary and sufficient conditions guaranteeing the existence of LU decomposition but those are difficult to check in practice and we do not state them here
- 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

Date: November 12, 2015, version 1.0. Comments, bug reports: `lekheng@galton.uchicago.edu`.

[1]the usual type of diagonal dominance, i.e., $|a_{ii}| \geq \sum_{j=1, j\neq i}^{n} |a_{ij}|$, $i = 1, \dots, n$, is called row diagonal dominance

- 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 $M_1\Pi_1 A = A_2$ (refer to earlier lecture notes for more information about permutation matrices)
- 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 — easy but a bit of a pain because notation is cumbersome
- so we will be informal but you'll get the idea
- for example if after two steps we get (recall that permutation matrices or orthogonal matrices),

$$\begin{aligned} A &= (M_2\Pi_2 M_1\Pi_1)^{-1}A_2 \\ &= \Pi_1^{\mathsf{T}} M_1^{-1}\Pi_2^{\mathsf{T}} M_2^{-1}A_2 \\ &= \Pi_1^{\mathsf{T}}\Pi_2^{\mathsf{T}}(\Pi_2 M_1^{-1}\Pi_2^{\mathsf{T}})M_2^{-1}A_2 \\ &= \Pi^{\mathsf{T}} L_1 L_2 A_2 \end{aligned}$$

  then
  - $\Pi = \Pi_2\Pi_1$ is a permutation matrix
  - $L_2 = M_2^{-1}$ is a unit lower triangular matrix
  - $L_1 = \Pi_2 M_1^{-1}\Pi_2^{\mathsf{T}}$ will always be a unit lower triangular matrix because M_1^{-1} is of the form in (**??**)

$$M_1^{-1} = \begin{bmatrix} 1 & \\ \boldsymbol{\ell} & I \end{bmatrix}$$

    whereas Π_2 must be of the form

$$\Pi_2 = \begin{bmatrix} 1 & \\ & \widehat{\Pi}_2 \end{bmatrix}$$

    for some $(n-1)\times(n-1)$ permutation matrix $\widehat{\Pi}_2$ and so

$$\Pi_2 M_1^{-1}\Pi_2^{\mathsf{T}} = \begin{bmatrix} 1 & 0 \\ \widehat{\Pi}_2\boldsymbol{\ell} & I \end{bmatrix}$$

    in other words $\Pi_2 M_1^{-1}\Pi_2^{\mathsf{T}}$ also has the form in (**??**)

- if we do one more steps we get

$$\begin{aligned}
A &= (M_3\Pi_3 M_2\Pi_2 M_1\Pi_1)^{-1}A_3 \\
&= \Pi_1^{\mathsf{T}} M_1^{-1}\Pi_2^{\mathsf{T}} M_2^{-1}\Pi_3^{\mathsf{T}} M_3^{-1} A_2 \\
&= \Pi_1^{\mathsf{T}}\Pi_2^{\mathsf{T}}\Pi_3^{\mathsf{T}}(\Pi_3\Pi_2 M_1^{-1}\Pi_2^{\mathsf{T}}\Pi_3^{\mathsf{T}})(\Pi_3 M_2^{-1}\Pi_3^{\mathsf{T}})M_3^{-1}A_2 \\
&= \Pi^{\mathsf{T}} L_1 L_2 L_3 A_2
\end{aligned}$$

  where
  - $\Pi = \Pi_3\Pi_2\Pi_1$ is a permutation matrix
  - $L_3 = M_3^{-1}$ is a unit lower triangular matrix
  - $L_2 = \Pi_3 M_2^{-1}\Pi_3^{\mathsf{T}}$ will always be a unit lower triangular matrix because M_2^{-1} is of the form in (**??**)

$$M_2^{-1} = \begin{bmatrix} 1 & & \\ & 1 & \\ & \boldsymbol{\ell} & I \end{bmatrix}$$

    whereas Π_3 must be of the form

$$\Pi_3 = \begin{bmatrix} 1 & & \\ & 1 & \\ & & \widehat{\Pi}_3 \end{bmatrix}$$

    for some $(n-2)\times(n-2)$ permutation matrix $\widehat{\Pi}_3$ and so

$$\Pi_3 M_2^{-1}\Pi_3^{\mathsf{T}} = \begin{bmatrix} 1 & & \\ & 1 & 0 \\ & \widehat{\Pi}_3\boldsymbol{\ell} & I \end{bmatrix}$$

    in other words $\Pi_3 M_2^{-1}\Pi_3^{\mathsf{T}}$ also has the form in (**??**)
  - $L_1 = \Pi_3\Pi_2 M_1^{-1}\Pi_2^{\mathsf{T}}\Pi_3^{\mathsf{T}}$ will always be a unit lower triangular matrix for the same reason above because $\Pi_3\Pi_2$ must have the form

$$\Pi_3\Pi_2 = \begin{bmatrix} 1 & & \\ & 1 & \\ & & \widehat{\Pi}_3 \end{bmatrix}\begin{bmatrix} 1 & \\ & \widehat{\Pi}_2 \end{bmatrix} = \begin{bmatrix} 1 & \\ & \Pi_{32} \end{bmatrix}$$

    for some $(n-1)\times(n-1)$ permutation matrix

$$\Pi_{32} = \begin{bmatrix} 1 & \\ & \widehat{\Pi}_3 \end{bmatrix}\widehat{\Pi}_2$$

- more generally if we keep doing this, then

$$A = \Pi^{\mathsf{T}} L_1 L_2 \cdots L_{n-1} A_{n-1}$$

  where
  - $\Pi = \Pi_{n-1}\Pi_{n-2}\cdots\Pi_1$ is a permutation matrix
  - $L_{n-1} = M_{n-1}^{-1}$ is a unit lower triangular matrix
  - $L_k = \Pi_{n-1}\cdots\Pi_{k+1}M_k^{-1}\Pi_{k+1}^{\mathsf{T}}\cdots\Pi_{n-1}^{\mathsf{T}}$ is a unit lower triangular matrix for all $k = 1,\dots,n-2$
  - $A_{n-1} = U$ is an upper triangular matrix
  - $L = L_1L_2\cdots L_{n-1}$ is a unit lower triangular matrix
- this algorithm with the row permutations is called *Gaussian elimination with partial pivoting* or GEPP for short; we will say more in the next section

2. PIVOTING STRATEGIES

- the (k,k) entry at step k during Gaussian elimination is called the *pivoting entry* or just *pivot* for short
- in the preceding section, we said that if the pivoting entry is zero, i.e., $a_{kk}^{(k)} = 0$, then we just need to find an entry below it in the same column, i.e., $a_{kj}^{(k)}$ for some $j > k$, and then permute this entry into the pivoting position, before carrying on with the algorithm
- but it is really better to choose the *largest* entry below the pivot, and not just any non-zero entry
- that is, the permutation Π_k is chosen so that row k is interchanged with row j, where $a_{kj}^{(k)} = \max_{j=k,k+1,\dots,n}|a_{kj}^{(k)}|$
- this guarantees that $|\ell_{kj}| \leq 1$ for all k and j
- this strategy is known as *partial pivoting*, which is guaranteed to produce an LU factorization if $A \in \mathbb{R}^{m\times n}$ has full row-rank, i.e., $\operatorname{rank}(A) = m \leq n$
- another common strategy, *complete pivoting*, which uses both row and column interchanges to ensure that at step k of the algorithm, the element $a_{kk}^{(k)}$ is the largest element in absolute value from the entire submatrix obtained by deleting the first $k-1$ rows and columns, i.e.,

$$a_{ij}^{(k)} = \max_{\substack{i=k,k+1,\dots,n\\ j=k,k+1,\dots,n}} |a_{ij}^{(k)}|$$

- in this case we need both row and column permutation matrices, i.e., we get

$$\Pi_1 A \Pi_2 = LU$$

  when we do complete pivoting
- complete pivoting is necessary when $\operatorname{rank}(A) < \min\{m,n\}$
- there are yet other pivoting strategies due to considerations such as preserving sparsity (if you're interested, look up *minimum degree algorithm* or *Markowitz algorithm*) or a tradeoff between partial and complete pivoting (e.g., *rook pivoting*)

3. UNIQUENESS OF THE LU FACTORIZATION

- the LU decomposition of a nonsingular matrix, if it exists (i.e., without row or column permutations), is unique
- if A has two LU decompositions, $A = L_1U_1$ and $A = L_2U_2$
- from $L_1U_1 = L_2U_2$ we obtain $L_2^{-1}L_1 = U_2U_1^{-1}$
- the inverse of a unit lower triangular matrix is a unit lower triangular matrix, and the product of two unit lower triangular matrices is a unit lower triangular matrix, so $L_2^{-1}L_1$ must be a unit lower triangular matrix
- similarly, $U_2U_1^{-1}$ is an upper triangular matrix
- the only matrix that is both upper triangular and unit lower triangular is the identity matrix I, so we must have $L_1 = L_2$ and $U_1 = U_2$

4. GAUSS–JORDAN ELIMINATION

- a variant of Gaussian elimination is called *Gauss–Jordan elimination*
- it entails zeroing elements above the diagonal as well as below, transforming an $m \times n$ matrix into *reduced row echelon form*, i.e., a form where all pivoting entries in U are 1 and all entries above the pivots are zeros
- this is what you probably learnt in your undergraduate linear algebra class, e.g.,

$$A = \begin{bmatrix} 1 & 3 & 1 & 9\\ 1 & 1 & -1 & 1\\ 3 & 11 & 5 & 35 \end{bmatrix} \to \begin{bmatrix} 1 & 3 & 1 & 9\\ 0 & -2 & -2 & -8\\ 0 & 2 & 2 & 8 \end{bmatrix} \to \begin{bmatrix} 1 & 3 & 1 & 9\\ 0 & -2 & -2 & -8\\ 0 & 0 & 0 & 0 \end{bmatrix} \to \begin{bmatrix} 1 & 0 & -2 & -3\\ 0 & 1 & 1 & 4\\ 0 & 0 & 0 & 0 \end{bmatrix} = U$$

- the main drawback is that the elimination process can be numerically unstable, since the multipliers can be large
- furthermore the way it is done in undergraduate linear algebra courses is that the elimination matrices (i.e., the L and Π) are not stored

5. CONDENSED LU FACTORIZATION

- just like QR and SVD, LU factorization with complete pivoting has a condensed form too
- let $A \in \mathbb{R}^{m\times n}$ and $\operatorname{rank}(A) = r \le \min\{m, n\}$, recall that GEPP yields

$$\begin{aligned}\Pi_1 A \Pi_2 &= LU \\ &= \begin{bmatrix} L_{11} & 0 \\ L_{21} & I_{m-r}\end{bmatrix}\begin{bmatrix} U_{11} & U_{12} \\ 0 & 0\end{bmatrix} \\ &= \begin{bmatrix} L_{11} \\ L_{21}\end{bmatrix}\begin{bmatrix} U_{11} & U_{12}\end{bmatrix} =: \widetilde{L}\widetilde{U}\end{aligned}$$

  where $L_{11} \in \mathbb{R}^{r\times r}$ is unit lower triangular (thus nonsingular) and $U_{11} \in \mathbb{R}^{r\times r}$ is also nonsingular
- note that $\widetilde{L} \in \mathbb{R}^{m\times r}$ and $\widetilde{U} \in \mathbb{R}^{n\times r}$ and so

$$A = (\Pi_1^{\mathsf{T}}\widetilde{L})(\widetilde{U}\Pi_2^{\mathsf{T}})$$

  is a rank-retaining factorization

6. LDU AND LDL^{T} FACTORIZATIONS

- if $A \in \mathbb{R}^{n\times n}$ has nonsingular principal submatrices $A_{1:k,1:k}$ for $k = 1, \dots, n$, then there exists a unit lower triangular matrix $L \in \mathbb{R}^{n\times n}$, a unit upper triangular matrix $U \in \mathbb{R}^{n\times n}$, and a diagonal matrix $D = \operatorname{diag}(d_{11}, \dots, d_{nn}) \in \mathbb{R}^{n\times n}$ such that

$$A = LDU = \begin{bmatrix} 1 & & & 0 \\ \ell_{21} & 1 & & \\ \vdots & & \ddots & \\ \ell_{n1} & \ell_{n2} & \cdots & 1\end{bmatrix}\begin{bmatrix} d_{11} & & & \\ & d_{22} & & \\ & & \ddots & \\ & & & d_{nn}\end{bmatrix}\begin{bmatrix} 1 & u_{12} & \cdots & u_{1n} \\ & 1 & & u_{2n} \\ & & \ddots & \vdots \\ & & & 1\end{bmatrix}$$

- this is called the LDU *factorization* of A
- if A is furthermore symmetric, then $L = U^{\mathsf{T}}$ and this called the LDL^{T} *factorization*
- if they exist, then both LDU and LDL^{T} factorizations are unique (exercise)
- if a symmetric A has an LDL^{T} factorization and if $d_{ii} > 0$ for all $i = 1, \dots, n$, then A is positive definite
- in fact, even though $d_{11}, \dots, d_{nn}$ are not the eigenvalues of A (why not?), they must have the same signs as the eigenvalues of A, i.e., if A has p positive eigenvalues, q negative eigenvalues, and z zero eigenvalues, then there are exactly p, q, and z positive, negative, and zero entries in $d_{11}, \dots, d_{nn}$ — a consequence of the Sylvester law of inertia
- unfortunately, both LDU and LDL^{T} factorizations are difficult to compute because
  - the condition on the principal submatrices is difficult to check in advance
  - algorithms for computing them are invariably unstable because size of multipliers cannot be bounded in terms of the entries of A
- for example, the LDL^{T} factorization of a 2×2 symmetric matrix is

$$\begin{bmatrix} a & c \\ c & d\end{bmatrix} = \begin{bmatrix} 1 & 0 \\ c/a & 1\end{bmatrix}\begin{bmatrix} a & c \\ 0 & d-(c/a)c\end{bmatrix} = \begin{bmatrix} 1 & 0 \\ c/a & 1\end{bmatrix}\begin{bmatrix} a & 0 \\ 0 & d-(c/a)c\end{bmatrix}\begin{bmatrix} 1 & c/a \\ 0 & 1\end{bmatrix}$$

- so
$$\begin{bmatrix} \varepsilon & 1 \\ 1 & \varepsilon \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1/\varepsilon & 1 \end{bmatrix} \begin{bmatrix} \varepsilon & 0 \\ 0 & \varepsilon - 1/\varepsilon \end{bmatrix} \begin{bmatrix} 1 & 1/\varepsilon \\ 0 & 1 \end{bmatrix}$$
the elements of L and D are arbitrarily large when $|\varepsilon|$ is small
- nonetheless there is one special case when LDL^{T} factorization not only exists but can be computed in an efficient and stable way — when A is positive definite

7. POSITIVE DEFINITE MATRICES

- a symmetric matrix $A \in \mathbb{R}^{n \times n}$ is *positive definite* if $\mathbf{x}^{\mathsf{T}} A \mathbf{x} > 0$ for all nonzero $\mathbf{x}$
- a symmetric positive definite matrix has real and positive eigenvalues, and its leading principal submatrices all have positive determinants
- from the definition, it is easy to see that all diagonal elements are positive
- to solve the system $A\mathbf{x} = \mathbf{b}$ where A is symmetric positive definite, we can compute the *Cholesky factorization*
$$A = R^{\mathsf{T}} R$$
where R is upper triangular
- this factorization exists if and only if A is symmetric positive definite
- in fact, attempting to compute the Cholesky factorization of A is an efficient method for checking whether A is symmetric positive definite
- it is important to distinguish the Cholesky factorization from the *square root factorization*
- a square root of a matrix A is defined as a matrix S such that
$$S^2 = SS = A$$
- note that the matrix R in $A = R^{\mathsf{T}} R$ is not the square root of A, since it does not hold that $R^2 = A$ unless A is a diagonal matrix
- the square root of a symmetric positive definite A can be computed by using the fact that A has an eigendecomposition $A = U\Lambda U^{\mathsf{T}}$ where Λ is a diagonal matrix whose diagonal elements are the positive eigenvalues of A and U is an orthogonal matrix whose columns are the eigenvectors of A
- it follows that
$$A = U\Lambda U^{\mathsf{T}} = (U\Lambda^{1/2} U^{\mathsf{T}})(U\Lambda^{1/2} U^{\mathsf{T}}) = SS$$
and so $S = U\Lambda^{1/2} U^{\mathsf{T}}$ is a square root of A

8. CHOLESKY FACTORIZATION

- the Cholesky factorization can be computed directly from the matrix equation $A = R^{\mathsf{T}} R$ where R is upper-triangular
- while it is conventional to write Cholesky factorization in the form $A = R^{\mathsf{T}} R$, it will be more natural later when we discuss the vectorized version of the algorithm to write $F = R^{\mathsf{T}}$ and $A = FF^{\mathsf{T}}$
- we can derive the algorithm for computing F by examining the matrix equation $A = R^{\mathsf{T}} R = FF^{\mathsf{T}}$ on an element-by-element basis, writing
$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{12} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} f_{11} & & & \\ f_{21} & f_{22} & & \\ \vdots & \vdots & \ddots & \\ f_{n1} & f_{n2} & \cdots & f_{nn} \end{bmatrix} \begin{bmatrix} f_{11} & f_{21} & \cdots & f_{n1} \\ & f_{22} & & f_{n2} \\ & & \ddots & \vdots \\ & & & f_{nn} \end{bmatrix}$$
- from the above matrix multiplication we see that $f_{11}^2 = a_{11}$, from which it follows that
$$f_{11} = \sqrt{a_{11}}$$

- from the relationship $f_{11}f_{i1} = a_{1i}$ and the fact that we already know f_{11}, we obtain

$$f_{i1} = \frac{a_{1i}}{f_{11}}, \quad i = 2, \dots, n$$

- proceeding to the second column of F, we see that $f_{21}^2 + f_{22}^2 = a_{22}$
- since we already know f_{21}, we have

$$f_{22} = \sqrt{a_{22} - f_{21}^2}$$

- if you know the fact that a positive definite matrix must have positive leading principal minors, then you could deduce the term above in the square root is positive by examining the 2×2 principal minor:

$$a_{11}a_{22} - a_{12}^2 > 0$$

  and therefore

$$a_{22} > \frac{a_{12}^2}{a_{11}} = f_{21}^2$$

- next, we use the relation $f_{21}f_{i1} + f_{22}f_{i2} = a_{2i}$ to compute

$$f_{i2} = \frac{a_{2i} - f_{21}f_{i1}}{f_{22}}$$

- hence we get

$$\begin{aligned} a_{11} &= f_{11}^2, & \\ a_{i1} &= f_{11}f_{i1}, & i &= 2, \dots, n \\ &\vdots & \\ a_{kk} &= f_{k1}^2 + f_{k2}^2 + \cdots + f_{kk}^2, & \\ a_{ik} &= f_{k1}f_{i1} + \cdots + f_{kk}f_{ik}, & i &= k+1, \dots, n \end{aligned}$$

- the resulting algorithm that runs for $k = 1, \dots, n$ is

$$\begin{aligned} f_{kk} &= \left(a_{kk} - \textstyle\sum_{j=1}^{k-1} f_{kj}^2\right)^{1/2}, & \\ f_{ik} &= \frac{\left(a_{ik} - \sum_{j=1}^{k-1} f_{kj}f_{ij}\right)}{f_{kk}}, & i &= k+1, \dots, n \end{aligned}$$

- you could use induction to show that the term in the square root is always positive but we'll soon see a more elegant vectorized version showing that this algorithm doesn't ever require taking square roots of negative numbers
- this algorithm requires roughly half as many operations as Gaussian elimination