Lecture 23. Cholesky Factorization

Hermitian positive definite matrices can be decomposed into triangular factors twice as quickly as general matrices. The standard algorithm for this, Cholesky factorization, is a variant of Gaussian elimination that operates on both the left and the right of the matrix at once, preserving and exploiting symmetry.

Hermitian Positive Definite Matrices

A real matrix $A \in \mathbb{R}^{m \times m}$ is *symmetric* if it has the same entries below the diagonal as above: $a_{ij} = a_{ji}$ for all i, j, hence $A = A^T$. Such a matrix satisfies $x^T Ay = y^T Ax$ for all vectors $x, y \in \mathbb{R}^m$.

For a complex matrix $A \in \mathbb{C}^{m \times m}$, the analogous property is that A is *hermitian*. A hermitian matrix has entries below the diagonal that are complex conjugates of those above the diagonal: $a_{ij} = \overline{a_{ji}}$, hence $A = A^*$. (These definitions appeared already in Lecture 2.) Note that this means that the diagonal entries of a hermitian matrix must be real.

A hermitian matrix A satisfies $x^*Ay = \overline{y^*Ax}$ for all $x, y \in \mathbb{C}^m$. This means in particular that for any $x \in \mathbb{C}^m$, x^*Ax is real. If in addition $x^*Ax > 0$ for all $x \neq 0$, then A is said to be *hermitian positive definite* (or sometimes just *positive definite*). Many matrices that arise in physical systems are hermitian positive definite because of fundamental physical laws.

If A is an $m \times m$ hermitian positive definite matrix and X is an $m \times n$ matrix of full rank with $m \geq n$, then the matrix X^*AX is also hermitian positive definite. It is hermitian because $(X^*AX)^* = X^*A^*X = X^*AX$, and

it is positive definite because, for any vector $x \neq 0$, we have $Xx \neq 0$ and thus $x^*(X^*AX)x = (Xx)^*A(Xx) > 0$. By choosing X to be an $m \times n$ matrix with a 1 in each column and zeros elsewhere, we can write any $n \times n$ principal submatrix of A in the form X^*AX. Therefore, any principal submatrix of A must be positive definite. In particular, every diagonal entry of A is a positive real number.

The eigenvalues of a hermitian positive definite matrix are also positive real numbers. If $Ax = \lambda x$ for $x \neq 0$, we have $x^*Ax = \lambda x^*x > 0$ and therefore $\lambda > 0$. Conversely, it can be shown that if a hermitian matrix has all positive eigenvalues, then it is positive definite.

Eigenvectors that correspond to distinct eigenvalues of a hermitian matrix are orthogonal. (As discussed in the next lecture, hermitian matrices are *normal*.) Suppose $Ax_1 = \lambda_1 x_1$ and $Ax_2 = \lambda_2 x_2$ with $\lambda_1 \neq \lambda_2$. Then

$$\lambda_2 x_1^* x_2 = x_1^* A x_2 = \overline{x_2^* A x_1} = \overline{\lambda_1 x_2^* x_1} = \lambda_1 x_1^* x_2,$$

so $(\lambda_1 - \lambda_2) x_1^* x_2 = 0$. Since $\lambda_1 \neq \lambda_2$, we have $x_1^* x_2 = 0$.

Symmetric Gaussian Elimination

We turn now to the problem of decomposing a hermitian positive definite matrix into triangular factors. To begin, consider what happens if a single step of Gaussian elimination is applied to a hermitian matrix A with a 1 in the upper-left position:

$$A = \begin{bmatrix} 1 & w^* \\ w & K \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ w & I \end{bmatrix} \begin{bmatrix} 1 & w^* \\ 0 & K - ww^* \end{bmatrix}.$$

As described in Lecture 20, zeros have been introduced into the first column of the matrix by an elementary lower-triangular operation on the left that subtracts multiples of the first row from subsequent rows.

Gaussian elimination would now continue the reduction to triangular form by introducing zeros in the second column. However, in order to maintain symmetry, Cholesky factorization first introduces zeros in the first row to match the zeros just introduced in the first column. We can do this by a right upper-triangular operation that subtracts multiples of the first column from the subsequent ones:

$$\begin{bmatrix} 1 & w^* \\ 0 & K - ww^* \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & K - ww^* \end{bmatrix} \begin{bmatrix} 1 & w^* \\ 0 & I \end{bmatrix}.$$

Note that this upper-triangular operation is exactly the adjoint of the lower-triangular operation that we used to introduce zeros in the first column.

Combining the operations above, we find that the matrix A has been factored into three terms:

$$A = \begin{bmatrix} 1 & w^* \\ w & K \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ w & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & K - ww^* \end{bmatrix} \begin{bmatrix} 1 & w^* \\ 0 & I \end{bmatrix}. \tag{23.1}$$

The idea of Cholesky factorization is to continue this process, zeroing one column and one row of A symmetrically until it is reduced to the identity.

Cholesky Factorization

In order for the symmetric triangular reduction to work in general, we need a factorization that works for any $a_{11} > 0$, not just $a_{11} = 1$. The generalization of (23.1) is accomplished by adjusting some of the elements of R_1 by a factor of $\sqrt{a_{11}}$. Let $\alpha = \sqrt{a_{11}}$ and observe:

$$A = \begin{bmatrix} a_{11} & w^* \\ w & K \end{bmatrix}$$

$$= \begin{bmatrix} \alpha & 0 \\ w/\alpha & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & K - ww^*/a_{11} \end{bmatrix} \begin{bmatrix} \alpha & w^*/\alpha \\ 0 & I \end{bmatrix} = R_1^* A_1 R_1.$$

This is the basic step that is applied repeatedly in Cholesky factorization. If the upper-left entry of the submatrix $K - ww^*/a_{11}$ is positive, the same formula can be used to factor it; we then have $A_1 = R_2^* A_2 R_2$ and thus $A = R_1^* R_2^* A_2 R_2 R_1$. The process is continued down to the bottom-right corner, giving us eventually a factorization

$$A = \underbrace{R_1^* R_2^* \cdots R_m^*}_{R^*} \underbrace{R_m \cdots R_2 R_1}_{R}. \tag{23.2}$$

This equation has the form

$$A = R^* R, \qquad r_{jj} > 0, \tag{23.3}$$

where R is upper-triangular. A reduction of this kind of a hermitian positive definite matrix is known as a *Cholesky factorization*.

The description above left one item dangling. How do we know that the upper-left entry of the submatrix $K - ww^*/a_{11}$ is positive? The answer is that it must be positive because $K - ww^*/a_{11}$ is positive definite, since it is the $(m-1)\times(m-1)$ lower-right principal submatrix of the positive definite matrix $R_1^{-*} A R_1^{-1}$. By induction, the same argument shows that all the submatrices A_j that appear in the course of the factorization are positive definite, and thus the process cannot break down. We can formalize this conclusion as follows.

Theorem 23.1. *Every hermitian positive definite matrix $A \in \mathbb{C}^{m\times m}$ has a unique Cholesky factorization* (23.3).

Proof. Existence is what we just discussed; a factorization exists since the algorithm cannot break down. In fact, the algorithm also establishes uniqueness. At each step (23.2), the value $\alpha = \sqrt{a_{11}}$ is determined by the form of

the R^*R factorization, and once α is determined, the first row of R_1^* is determined too. Since the analogous quantities are determined at each step of the reduction, the entire factorization is unique. □

The Algorithm

When Cholesky factorization is implemented, only half of the matrix being operated on needs to be represented explicitly. This simplification allows half of the arithmetic to be avoided. A formal statement of the algorithm (only one of many possibilities) is given below. The input matrix A represents the superdiagonal half of the $m \times m$ hermitian positive definite matrix to be factored. (In practical software, a compressed storage scheme may be used to avoid wasting half the entries of a square array.) The output matrix R represents the upper-triangular factor for which $A = R^*R$. Each outer iteration corresponds to a single elementary factorization: the upper-triangular part of the submatrix $R^*_{k:m,k:m}$ represents the superdiagonal part of the hermitian matrix being factored at step k.

Algorithm 23.1. Cholesky Factorization

$R = A$
for $k = 1$ **to** m
    for $j = k+1$ **to** m
        $R_{j,j:m} = R_{j,j:m} - R_{k,j:m}\overline{R}_{kj}/R_{kk}$
    $R_{k,k:m} = R_{k,k:m}/\sqrt{R_{kk}}$

Operation Count

The arithmetic done in Cholesky factorization is dominated by the inner loop. A single execution of the line

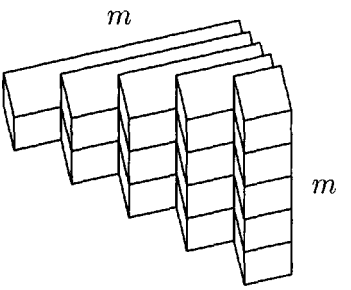
$$R_{j,j:m} = R_{j,j:m} - R_{k,j:m}\overline{R}_{kj}/R_{kk}$$

requires one division, $m-j+1$ multiplications, and $m-j+1$ subtractions, for a total of $\sim 2(m-j)$ flops. This calculation is repeated once for each j from $k+1$ to m, and that loop is repeated for each k from 1 to m. The sum is straightforward to evaluate:

$$\sum_{k=1}^{m}\sum_{j=k+1}^{m} 2(m-j) \sim 2\sum_{k=1}^{m}\sum_{j=1}^{k} j \sim \sum_{k=1}^{m} k^2 \sim \frac{1}{3}m^3 \text{ flops.}$$

Thus, Cholesky factorization involves only half as many operations as Gaussian elimination, which would require $\sim \frac{2}{3}m^3$ flops to factor the same matrix.

As usual, the operation count can also be determined graphically. For each k, two floating point operations are carried out (one multiplication and one subtraction) at each position of a triangular layer. The entire algorithm corresponds to stacking m layers:



As $m \to \infty$, the solid converges to a tetrahedron with volume $\frac{1}{6}m^3$. Since each unit cube corresponds to two floating point operations, we obtain again

$$\text{Work for Cholesky factorization:} \quad \sim \frac{1}{3}m^3 \text{ flops.} \tag{23.4}$$

Stability

All of the subtleties of the stability analysis of Gaussian elimination vanish for Cholesky factorization. This algorithm is always stable. Intuitively, the reason is that the factors R can never grow large. In the 2-norm, for example, we have $\|R\| = \|R^*\| = \|A\|^{1/2}$ (proof: SVD), and in other p-norms with $1 \le p \le \infty$, $\|R\|$ cannot differ from $\|A\|^{1/2}$ by more than a factor of $\sqrt{m}$. Thus, numbers much larger than the entries of A can never arise.

Note that the stability of Cholesky factorization is achieved without the need for any pivoting. Intuitively, one may observe that this is related to the fact that most of the weight of a hermitian positive definite matrix is on the diagonal. For example, it is not hard to show that the largest entry must appear on the diagonal, and this property carries over to the positive definite submatrices constructed in the inductive process (23.2).

An analysis of the stability of the Cholesky process leads to the following backward stability result.

Theorem 23.2. *Let $A \in \mathbb{C}^{m\times m}$ be hermitian positive definite, and let a Cholesky factorization of A be computed by Algorithm* 23.1 *on a computer satisfying* (13.5) *and* (13.7). *For all sufficiently small $\epsilon_{\text{machine}}$, this process is guaranteed to run to completion (i.e., no zero or negative corner entries r_{kk} will arise), generating a computed factor $\tilde{R}$ that satisfies*

$$\tilde{R}^*\tilde{R} = A + \delta A, \qquad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{\text{machine}}) \tag{23.5}$$

for some $\delta A \in \mathbb{C}^{m\times m}$.

Like so many algorithms of numerical linear algebra, this one would look much worse if we tried to carry out a forward error analysis rather than a backward one. If A is ill-conditioned, $\tilde{R}$ will not generally be close to R; the best we can say is $\|\tilde{R} - R\|/\|R\| = O(\kappa(A)\epsilon_{\text{machine}})$. (In other words, Cholesky factorization is in general an ill-conditioned problem.) It is only the product $\tilde{R}^*R$ that satisfies the much better error bound (23.5). Thus the errors introduced in $\tilde{R}$ by rounding are large but "diabolically correlated," just as we saw in Lecture 16 for QR factorization.

Solution of $Ax = b$

If A is hermitian positive definite, the standard way to solve a system of equations $Ax = b$ is by Cholesky factorization. Algorithm 23.1 reduces the system to $R^*Rx = b$, and we then solve two triangular systems in succession: first $R^*y = b$ for the unknown y, then $Rx = y$ for the unknown x. Each triangular solution requires just $\sim m^2$ flops, so the total work is again $\sim \frac{1}{3}m^3$ flops.

By reasoning analogous to that of Lecture 16, it can be shown that this process is backward stable.

Theorem 23.3. *The solution of hermitian positive definite systems $Ax = b$ via Cholesky factorization (Algorithm 23.1) is backward stable, generating a computed solution $\tilde{x}$ that satisfies*

$$(A + \Delta A)\tilde{x} = b, \qquad \frac{\|\Delta A\|}{\|A\|} = O(\epsilon_{\text{machine}}) \tag{23.6}$$

for some $\Delta A \in \mathbb{C}^{m\times m}$.

Exercises

23.1. Let A be a nonsingular square matrix and let $A = QR$ and $A^*A = U^*U$ be QR and Cholesky factorizations, respectively, with the usual normalizations r_{jj}, $u_{jj} > 0$. Is it true or false that $R = U$?

23.2. Using the proof of Theorem 16.2 as a guide, derive Theorem 23.3 from Theorems 23.2 and 17.1.

23.3. *Reverse Software Engineering of "\".* The following MATLAB session records a sequence of tests of the elapsed times for various computations on a workstation manufactured in 1991. For each part, try to explain: (i) Why was this experiment carried out? (ii) Why did the result come out as it did? Your

answers should refer to formulas from the text for flop counts. The MATLAB queries `help chol` and `help slash` may help in your detective work.

(a)
```
m = 200; Z = randn(m,m);
A = Z'*Z; b = randn(m,1);
tic; x = A\b; toc;
       elapsed_time = 1.0368
```

(b)
```
tic; x = A\b; toc;
       elapsed_time = 1.0303
```

(c)
```
A2 = A; A2(m,1) = A2(m,1)/2;
tic; x = A2\b; toc;
       elapsed_time = 2.0361
```

(d)
```
I = eye(m,m); emin = min(eig(A));
A3 = A - .9*emin*I;
tic; x = A3\b; toc;
       elapsed_time = 1.0362
```

(e)
```
A4 = A - 1.1*emin*I;
tic; x = A4\b; toc;
       elapsed_time = 2.9624
```

(f)
```
A5 = triu(A);
tic; x = A5\b; toc;
       elapsed_time = 0.1261
```

(g)
```
A6 = A5; A6(m,1) = A5(1,m);
tic; x = A6\b; toc;
       elapsed_time = 2.0012
```