Lecture Companion: Cholesky Decomposition

The Art of Taking a Matrix Square Root

MFML Companion Guide

Contents

---

1 Introduction: Beyond the Diagonal

In **Lecture 3**, we explored **Symmetric Positive-Definite (SPD)** matrices. We learned that these matrices act as the "engine" for custom inner products, defining how we measure length and angles.

But there is a practical question we haven't answered: *If we have a complex covariance matrix or a custom inner product engine, can we break it down into something simpler?*

The answer is **yes**. Just as we can take the square root of a positive number ($9 \to 3$), we can take the "square root" of an SPD matrix. This operation is called the **Cholesky Decomposition**.

—

2 The Prerequisites: When Can We Use It?

The Cholesky Decomposition is special because it is **exclusive**. It does not work for every matrix. To use it, a matrix A must satisfy two conditions (from Lecture 3):

1. **Symmetric:** $A = A^T$. (The matrix is a mirror image across the diagonal).
2. **Positive-Definite:** For any non-zero vector x, $x^T Ax > 0$. (The matrix represents a "bowl" shape opening upwards; it has strictly positive eigenvalues).

The Intuition: $A = LL^T$

If A is our complex "combined" transformation (like a covariance matrix), the Cholesky decomposition finds a simple **Lower Triangular matrix** L such that:

$$A = LL^T$$

Since L is triangular, it represents a sequence of simple steps (scaling and adding previous results). This makes L much easier to work with than A.

—

3 The Algorithm: Constructing L

We want to find the entries of L. Since L is lower triangular, we know its structure. For a 3×3 matrix:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}$$

By multiplying the matrices on the right and equating them to A on the left, we can solve for l_{ij} row by row.

3.1 The Formulas

- **Diagonal Entries:** Square root of the diagonal element minus the "squared" elements before it.

$$l_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2}$$

- **Off-Diagonal Entries:** (Row i, Col j):

$$l_{ij} = \frac{1}{l_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right)$$

—

4 Example: A Narrated Calculation

Let's decompose the following symmetric positive-definite matrix A:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 8 \\ 3 & 8 & 14 \end{bmatrix}$$

We want to find L such that $A = LL^T$.

Step 1: Column 1 (The easy part)

We start at the top-left corner $(1,1)$.

$$l_{11} = \sqrt{a_{11}} = \sqrt{1} = \mathbf{1}$$

Now we calculate the rest of the first column (l_{21} and l_{31}) by dividing the A entries by the diagonal l_{11}.

$$l_{21} = \frac{a_{21}}{l_{11}} = \frac{2}{1} = \mathbf{2}$$

$$l_{31} = \frac{a_{31}}{l_{11}} = \frac{3}{1} = \mathbf{3}$$

Current L:

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & ? & 0 \\ 3 & ? & ? \end{bmatrix}$$

Step 2: Column 2 (Subtracting the "shadow")

Now we move to the diagonal entry $(2,2)$. We must subtract the contribution from the first column (l_{21}^2).

$$l_{22} = \sqrt{a_{22} - l_{21}^2} = \sqrt{5 - 2^2} = \sqrt{5-4} = \sqrt{1} = \mathbf{1}$$

Now for the entry below it (l_{32}). We subtract the product of the previous column's values ($l_{31} \times l_{21}$) from a_{32}, then divide by the diagonal l_{22}.

$$l_{32} = \frac{1}{l_{22}}(a_{32} - l_{31}l_{21}) = \frac{1}{1}(8 - (3)(2)) = 8 - 6 = \mathbf{2}$$

Current L:

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & ? \end{bmatrix}$$

Step 3: Column 3 (The final diagonal)

Finally, the bottom-right corner $(3,3)$. We subtract the squared contributions from both previous columns (l_{31}^2 and l_{32}^2).

$$l_{33} = \sqrt{a_{33} - (l_{31}^2 + l_{32}^2)}$$

$$l_{33} = \sqrt{14 - (3^2 + 2^2)} = \sqrt{14 - (9+4)} = \sqrt{14 - 13} = \sqrt{1} = \mathbf{1}$$

Final Result:

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix}$$

Verification: Calculate LL^T.

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 8 \\ 3 & 8 & 14 \end{bmatrix} = A$$

It works perfectly!

—

5 Why Do We Care? (Applications)

5.1 1. Solving Linear Systems Efficiently

If you want to solve $Ax = b$ and A is SPD, you generally **do not** calculate A^{-1}. Instead:

- Compute $A = LL^T$ (Cholesky).
- Solve $Ly = b$ (Forward Substitution - easy because L is triangular).
- Solve $L^T x = y$ (Backward Substitution - easy because L^T is triangular).

This is twice as fast as the standard LU decomposition!

5.2 2. Monte Carlo Simulations

In finance and data science, we often need to generate random variables that are **correlated** (e.g., stock prices that move together).

- We start with uncorrelated random noise z (unit variance).
- If we want variables with covariance matrix Σ, we perform Cholesky: $\Sigma = LL^T$.
- We calculate $x = Lz$.

The new vector x now has the exact correlation structure defined by Σ. This is the standard way to simulate correlated systems.