Generalities

These direct matrix factorization algorithms can be understood as n iterations which work on block matrices. Quite a bit of steup is required before they can "look easy".

On step i, we split each matrix (the input A and the two output factors) at the ith row and the ith column.

We'll build on the notation a_{ij}, which means the entry in row i and column j of the matrix A.

Let's use capital letters (A) to refer to matrices, bold letters ($\mathbf{a}$) to refer to vectors, and small letters (a) to refer to scalars. A "$-$" will be shorthand for "less than i"; a $+$ will stand for "greater than i".

On each step, assume: A submatrix with a $-$ in its name has already been computed. A submatrix with an i in its name, but no $-$, must be computed now.

So the matrix A splits into these submatrices:

$$A = \left[\begin{array}{c|c|c} A_{--} & \mathbf{a}_{-i} & A_{-+} \\ \hline \mathbf{a}_{i-} & a_{ii} & \mathbf{a}_{i+} \\ \hline A_{+-} & \mathbf{a}_{+i} & A_{++} \end{array}\right]$$

$$\left(\left[\begin{array}{c|c|c} (i-1)\times(i-1) & (i-1)\times 1 & (i-1)\times(n-i) \\ \hline 1\times(i-1) & 1\times 1 & 1\times(n-i) \\ \hline (n-i)\times(i-1) & (n-i)\times 1 & (n-i)\times(n-i) \end{array}\right]\right)$$

And progress looks like this:

$$\left(\left[\begin{array}{c|c|c} \text{done} & \text{done} & \text{done} \\ \hline \text{done} & \text{now} & \text{now} \\ \hline \text{done} & \text{now} & \text{later} \end{array}\right]\right)$$

Algorithm 6.4: LU Decomposition (Derivation)

Write down $A = LU$:

$$\left[\begin{array}{c|c|c} A_{--} & \mathbf{a}_{-i} & A_{-+} \\ \hline \mathbf{a}_{i-} & a_{ii} & \mathbf{a}_{i+} \\ \hline A_{+-} & \mathbf{a}_{+i} & A_{++} \end{array}\right] = \left[\begin{array}{c|c|c} L_{--} & & \\ \hline \mathbf{l}_{i-} & l_{ii} & \\ \hline L_{+-} & \mathbf{l}_{+i} & L_{++} \end{array}\right] \left[\begin{array}{c|c|c} U_{--} & \mathbf{u}_{-i} & U_{-+} \\ \hline & u_{ii} & \mathbf{u}_{i+} \\ \hline & & U_{++} \end{array}\right]$$

$$= \left[\begin{array}{c|c|c} L_{--}U_{--} & L_{--}\mathbf{u}_{-i} & L_{--}U_{-+} \\ \hline \mathbf{l}_{i-}U_{--} & \mathbf{l}_{i-}\mathbf{u}_{-i} + l_{ii}u_{ii} & \mathbf{l}_{i-}U_{-+} + l_{ii}\mathbf{u}_{i+} \\ \hline L_{+-}U_{--} & L_{+-}\mathbf{u}_{-i} + \mathbf{l}_{+i}u_{ii} & L_{+-}U_{-+} + \mathbf{l}_{+i}\mathbf{u}_{i+} + L_{++}U_{++} \end{array}\right]$$

Equate the submatrices in the "now" positions:

$$\begin{aligned} a_{ii} &= \mathbf{l}_{i-}\mathbf{u}_{-i} + l_{ii}u_{ii} \\ \mathbf{a}_{i+} &= \mathbf{l}_{i-}U_{-+} + l_{ii}\mathbf{u}_{i+} \\ \mathbf{a}_{+i} &= L_{+-}\mathbf{u}_{-i} + \mathbf{l}_{+i}u_{ii} \end{aligned}$$

Algorithm 6.4: LU Decomposition (Result)

A step in the factorization

$$\left[\begin{array}{c|c|c} A_{--} & \mathbf{a}_{-i} & A_{-+} \\ \hline \mathbf{a}_{i-} & a_{ii} & \mathbf{a}_{i+} \\ \hline A_{+-} & \mathbf{a}_{+i} & A_{++} \end{array}\right] = \left[\begin{array}{c|c|c} L_{--} & & \\ \hline \mathbf{l}_{i-} & l_{ii} & \\ \hline L_{+-} & \mathbf{l}_{+i} & L_{++} \end{array}\right] \left[\begin{array}{c|c|c} U_{--} & \mathbf{u}_{-i} & U_{-+} \\ \hline & u_{ii} & \mathbf{u}_{i+} \\ \hline & & U_{++} \end{array}\right]$$

goes like this:

$$\begin{array}{rrl} \text{Factor} & l_{ii}u_{ii} = & a_{ii} - \mathbf{l}_{i-}\mathbf{u}_{-i} \qquad \text{(arbitrarily)} \\ \text{Set} & \mathbf{u}_{i+} = & (\mathbf{a}_{i+} - \mathbf{l}_{i-}U_{-+})/l_{ii} \\ \text{Set} & \mathbf{l}_{+i} = & (\mathbf{a}_{+i} - L_{+-}\mathbf{u}_{-i})/u_{ii} \end{array}$$

If $a_{ii} - \mathbf{l}_{i-}\mathbf{u}_{-i}$ is zero at any step, this could fail.

There are three common choices for the factorization $l_{ii}u_{ii} = 1$: setting $l_{ii} = 1$ (Doolittle's), $u_{ii} = 1$ (Crout's), and…

Cholesky (LL^*) factorization

…the last one, Cholesky's, works when A is a symmetric positive definite matrix: $l_{ii} = u_{ii} \in \mathbb{R}^+$. But in that case the L and U you produce will be each other's transpose. So one needn't compute U; the algorithm simplifies to:

$$\begin{aligned} l_{ii} =& (a_{ii} - \mathbf{l}_{i-}\mathbf{l}^*_{i-})^{1/2} \\ \mathbf{l}_{+i} =& (\mathbf{a}_{+i} - L_{+-}\mathbf{l}^*_{i-})/l^*_{ii} \end{aligned}$$

Avoiding square roots: LDL^* factorization

Let's try another factorization. (A is still positive definite.)

$$\left[\begin{array}{c|c|c} L_{--} & & \\ \hline \mathbf{l}_{i-} & l_{ii} & \\ \hline L_{+-} & \mathbf{l}_{+i} & L_{++} \end{array}\right] \left[\begin{array}{c|c|c} D_{--} & & \\ \hline & d_{ii} & \\ \hline & & D_{++} \end{array}\right] \left[\begin{array}{c|c|c} L_{--}^* & \mathbf{l}_{i-}^* & L_{+-}^* \\ \hline & l_{ii}^* & \mathbf{l}_{+i}^* \\ \hline & & L_{++}^* \end{array}\right]$$

$$= \left[\begin{array}{c|c|c} \dots & \dots & \dots \\ \hline \dots & \mathbf{l}_{i-} D_{--} \mathbf{l}_{i-}^* + l_{ii} d_{ii} l_{ii}^* & \mathbf{l}_{i-} D_{--} L_{+-}^* + l_{ii} d_{ii} \mathbf{l}_{+i}^* \\ \hline \dots & L_{+-} D_{--} \mathbf{l}_{i-}^* + \mathbf{l}_{+i} d_{ii} l_{ii}^* & \dots \end{array}\right]$$

(We only have to write down entries in the "now" positions when we multiply this out. Since A and the product are symmetric, we could have even skipped the bottom-center block.) Result:

$$\begin{aligned} d_{ii} =& a_{ii} - \mathbf{l}_{i-} D_{--} \mathbf{l}_{i-}^* \ ; \ l_{ii} = 1 \\ \mathbf{l}_{+i} =& (\mathbf{a}_{+i} - L_{+-} D_{--} \mathbf{l}_{i-}^*)/d_{ii} \end{aligned}$$

Remember that D_{--} is diagonal, so multiplying by it is very easy.