Matrix Factorization

We have learned how to solve a system of linear equations $A\mathbf{x} = \mathbf{b}$ by applying Gaussian elimination to the augmented matrix $\tilde{A} = [A \ b]$, and then performing back substitution on the resulting upper-triangular matrix. However, this approach is not practical if the right-hand side $\mathbf{b}$ of the system is changed, while $A$ is not. This is due to the fact that the choice of $\mathbf{b}$ has no effect on the row operations needed to reduce $A$ to upper-triangular form. Therefore, it is desirable to instead apply these row operations to $A$ only once, and then “store” them in some way in order to apply them to any number of right-hand sides.

To accomplish this, we first note that subtracting $m_{ij}$ times row $j$ from row $i$ to eliminate $a_{ij}$ is equivalent to multiplying $A$ by the matrix

$$M_{ij} = \begin{bmatrix}
1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 1 & 0 & \cdots & \cdots & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & \cdots & \cdots & 0 & 1 & 0
\end{bmatrix},$$

where the entry $-m_{ij}$ is in row $i$, column $j$. More generally, if we let $A^{(1)} = A$ and let $A^{(k+1)}$ be the matrix obtained by eliminating elements of column $k$ in $A^{(k)}$, then, if no pivoting is required, we have, for $k = 1, 2, \ldots, n-1$,

$$A^{(k+1)} = M^{(k)}A^{(k)}$$
where

\[
M^{(k)} = \begin{bmatrix}
1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & m_{nk} & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & \cdots & \cdots & 1 & 0 & \cdots & \cdots & 0 \\
0 & \cdots & 0 & \cdots & \cdots & 0 & 1 & \cdots & \cdots & 0 \\
\end{bmatrix},
\]

with the elements \(-m_{k+1,k}, \ldots, -m_{nk}\) occupying column \(k\). It follows that the matrix

\[
U = A^{(n)} = M^{(n-1)}A^{(n-1)} = M^{(n-1)}M^{(n-2)} \cdots M^{(1)}A
\]

is upper triangular, and the vector

\[
y = M^{(n-1)}M^{(n-2)} \cdots M^{(1)}b,
\]

being the result of applying the same row operations to \(b\), is the right-hand side for the upper-triangular system that is to be solved by back substitution.

Now, we note that each matrix \(M^{(k)}\), \(k = 1, 2, \ldots, n - 1\), is not only a lower-triangular matrix, but a unit lower triangular matrix, because all of its diagonal entries are equal to 1. Next, we note two important properties of unit lower triangular matrices:

- The product of two unit lower triangular matrices is unit lower triangular.
- A unit lower triangular matrix is nonsingular, and its inverse is unit lower triangular.

In fact, the inverse of each \(M^{(k)}\) is easily computed. We have

\[
L^{(k)} = [M^{(k)}]^{-1} = \begin{bmatrix}
1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & m_{nk} & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & \cdots & \cdots & 1 & 0 & \cdots & \cdots & 0 \\
0 & \cdots & 0 & \cdots & \cdots & 0 & 1 & \cdots & \cdots & 0 \\
\end{bmatrix},
\]

2
It follows that if we define \( M = M^{(n-1)} \cdots M^{(1)} \), then \( M \) is unit lower triangular, and \( MA = U \), where \( U \) is upper triangular. It follows that \( A = M^{-1}U = LU \), where

\[
L = L^{(1)} \cdots L^{(n-1)} = [M^{(1)}]^{-1} \cdots [M^{(n-1)}]^{-1}
\]
is also unit lower triangular. Furthermore, from the structure of each matrix \( L^{(k)} \), it can readily be determined that

\[
L = \begin{bmatrix}
1 & 0 & \cdots & \cdots & 0 \\
m_{21} & 1 & 0 & \vdots & \\
\vdots & m_{32} & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & 1 & 0 \\
m_{n1} & m_{n2} & \cdots & m_{n,n-1} & 1
\end{bmatrix}
\]

That is, \( L \) stores all of the multipliers used during Gaussian elimination. The factorization of \( A \) that we have obtained,

\[
A = LU,
\]
is called the \textit{LU decomposition} of \( A \).

Now, suppose that pivoting is performed. Then, if row \( j \) is interchanged with row \( p \), for \( p > j \), before entries in column \( j \) are eliminated, the matrix \( A^{(j)} \) is effectively multiplied by a \textit{permutation matrix} \( P^{(j)} \). A permutation matrix is a matrix obtained by permuting the rows (or columns) of the identity matrix \( I \). In \( P^{(j)} \), rows \( j \) and \( p \) of \( I \) are interchanged, so that multiplying \( A^{(j)} \) on the left by \( P^{(j)} \) interchanges these rows of \( A^{(j)} \). It follows that the process of Gaussian elimination with pivoting can be described in terms of the matrix multiplications

\[
M^{(n-1)} P^{(n-1)} M^{(n-2)} P^{(n-2)} \cdots M^{(1)} P^{(1)} A = U,
\]

where \( P^{(k)} = I \) if no interchange is performed before eliminating entries in column \( k \).

However, because each permutation matrix \( P^{(k)} \) at most interchanges row \( k \) with row \( p \), where \( p > k \), there is no difference between applying all of the row interchanges “up front”, instead of applying \( P^{(k)} \) immediately before applying \( M^{(k)} \) for each \( k \). It follows that

\[
[M^{(n-1)} M^{(n-2)} \cdots M^{(1)}] [P^{(n-1)} P^{(n-2)} \cdots P^{(1)}] A = U,
\]

and because a product of permutation matrices is a permutation matrix, we have

\[
PA = LU,
\]

where \( L \) is defined as before, and \( P = P^{(n-1)} P^{(n-2)} \cdots P^{(1)} \). This decomposition exists for any nonsingular matrix \( A \).

Once the \textit{LU} decomposition \( PA = LU \) has been computed, we can solve the system \( Ax = b \) by first noting that if \( x \) is the solution, then

\[
PAx = LUx = Pb.
\]
Therefore, we can obtain $x$ by first solving the system

$$Ly = Pb,$$

and then solving

$$Ux = y.$$ 

Then, if $b$ should change, then only these last two systems need to be solved in order to obtain the new solution; the $LU$ decomposition does not need to be recomputed.

The system $Ux = y$ can be solved by back substitution, since $U$ is upper-triangular. To solve $Ly = Pb$, we can use forward substitution, since $L$ is unit lower triangular. In the following description of the algorithm, we let $c = Pb$.

\begin{verbatim}
for i = 1, 2, ..., n do
    $y_i = c_i$
    for j = 1, 2, ..., i - 1 do
        $y_i = y_i - \ell_{ij}y_j$
    end
end
\end{verbatim}

Like back substitution, this algorithm requires $O(n^2)$ floating-point operations. Unlike back substitution, there is no division of the $i$th component of the solution by a diagonal element of the matrix, but this is only because in this context, $L$ is unit lower triangular, so $\ell_{ii} = 1$. When applying forward substitution to a general lower triangular matrix, such a division is required.

Because both forward and back substitution require only $O(n^2)$ operations, whereas Gaussian elimination requires $O(n^3)$ operations, changes in the right-hand side $b$ can be handled quite efficiently by computing the factors $L$ and $U$ once, and storing them. This can be accomplished quite efficiently, because $L$ is unit lower triangular. It follows from this that $L$ and $U$ can be stored in a single $n \times n$ matrix by storing $U$ in the upper triangular part, and the multipliers $m_{ij}$ in the lower triangular part. The permutations encoded in $P$ must also be stored, but this can also be accomplished efficiently by applying the permutations to a vector containing the indices $1, 2, \ldots, n$, thus obtaining the final ordering of the rows.