***Class Notes*, Math 554, Autumn 2012**

Lecture XXIII: LU Factorization; QR Algorithm

Alternately, the Moore-Penrose pseudo-inverse can be computed using the condensed SVD: $A^\dagger = V_r \Sigma_r^{-1} U_r^*$.

A few more things about the Moore-Penrose pseudo-inverse. In practice, one never computes it. If one wants to compute *a* minimizer, or the value of the minimum of $||b - Ax||_2^2$, one generally uses QR. If one wants the minimizer of minimal norm, one can find the SVD of A, and then find

$$x^\dagger = V_r(\Sigma_r^{-1}(U_r^* b)) \ .$$

Some of the theoretical properties of the inverse are of interest; for example

$$AA^\dagger = U_r \Sigma_r V_r^* V_r \Sigma_r^{-1} U_r^* = \sum_{i=1}^{r} u_r u_r^* \ .$$

This is the orthogonal projection onto $\mathcal{R}(A)$! Similarly, $A^\dagger A = \sum_{i=1}^r v_i v_i^*$ is the orthogonal projection onto $\mathcal{N}(A)$.

Finally, the pseudo-inverse can be used to define a condition number for singular or rectangular matrices: $\kappa(A) = ||A|| \cdot ||A^\dagger|| = \sigma_1/\sigma_r$.

1 LU factorization

The last factorization we see is the result of the process of Gaussian elimination *on square matrices*: the LU factorization. This is the standard method of solving equations $Ax = b$; it is much faster, for example, than QR, and–although in theory it may lead to significant loss of accuracy; much more than QR–in practice it is almost always accurate and stable. (It should be said, however, that in practice one always does some sort of pivoting to account for dependent columns; we will see what this means below.)

Note that if $A = LU$, where L is a lower triangular matrix with ones on the diagonal and U is an upper triangular matrix, then solving $Ax = b$ amounts to solving two triangular systems, which can be done very fast through forward, respectively backward, substitution.

We explain how to find the matrices L and U below.

Recall the matrices $I - \alpha E_{ij}$ mentioned before; E_{ij} is the matrix that has a 1 in entry (i, j), and 0s everywhere else. Previously, in showing that the Schur form can be reduced to block-triangular form, we have used only E_{ij}s for which $i < j$; now we will only be interested in E_{ij}s for which $i > j$.

Gaussian elimination zeroes out each column of a matrix, via elementary row operations (like subtraction of a multiple of a row, from a different row), starting with the lowest, and working one's way up to the element just under the diagonal. This amounts to left multiplication by a sequence of matrices $I - \alpha E_{ij}$, where each α is the ratio of the (current) entry a_{ij} to the (current) diagonal entry a_{ii}. (Recall that left multiplication by E_{ij} "extracts" the row i and places it in row j.)

In fact, for $A \in \mathcal{M}_n$, if $A = \begin{bmatrix} a_{11} & v \\ u^T & B \end{bmatrix}$, with $v = [a_{12}, \ldots, a_{1n}]$ and $u = [a_{21}, \ldots, a_{2n}]$, and B being the $(n-1) \times (n-1)$ lower right corner of A, we can do all of the operations meant to zero out the first column of A *together*.

The reason is simple: each multiplication by $I - a_{i1}/a_{11}E_{i1}$ *affects only the ith row*! It follows that

$$\prod_{i=2}^{n}(I - a_{i1}/a_{11}E_{i1}) = \begin{bmatrix} 1 & 0 \\ -u^T/a_{11} & I \end{bmatrix} =: L_1 \ . \tag{1}$$

Thus, $L_1 A$ is matrix whose first column, under the diagonal, is 0.

Continuing the process, one obtains similar matrices L_k for $k = 2, \ldots, n$, each of which zeroes out the next column of A, such that

$$L_n L_{n-1} \ldots L_1 A = U \ ,$$

where U is an upper triangular matrix. Moreover, due to the way and purpose to which we define L_k, each such matrix is a lower triangular one with a diagonal of ones and the only other non-zero entries being $L_{k(k+1)}$ through L_{kn}. Moroever, recalling that for each factor of the product on the left hand side of (1), $(I - \alpha E_{ij})^{-1} = (I + \alpha E_{ij})$, it follows trivially that the inverse of each L_k looks exactly like L_k, except that all non-diagonal entries have the opposite sign as in L_k. Therefore,

$$(L_n L_{n-1} \ldots L_1)^{-1} = L_1^{-1} \ldots L_{n-1}^{-1} L_n^{-1} =: L$$

is a lower triangular matrix with a diagonal of all 1s, and $A = LU$.

Remark 1. *Note that L is always invertible; if A is not, then U is not. In fact,* $\det(A) = \det(U)$*!*

Remark 2. *Not every matrix has an LU factorization (think of what can go wrong with the algorithm above: any diagonal matrix being* 0*. In practice, pivoting is done: before zeroing out the subdiagonal part of the ith column, one searches for the largest non-zero entry in that column, and swaps the row in which it is found with row i. The algorithm then keeps track of all the "pivoting" in a permutation matrix P ($P^T = P^{-1}$). Then PA has a unique factorization, so $A = P^T LU$ is a "PLU" factorization of A.* Any matrix has a PLU factorization; not necessarily a unique one.