

Gram-Schmidt in 9 Lines of MATLAB

The Gram-Schmidt algorithm starts with n independent vectors a_1, \dots, a_n (the columns of A). It produces n orthonormal vectors q_1, \dots, q_n (the columns of Q). To find q_j , start with a_j and subtract off its projections onto the previous q 's—and then divide by the length of that vector v to produce a unit vector.

The inner products $q_i^T a_j$ go into a square matrix R that satisfies $A = QR$. This R is upper triangular, because $q_i^T a_j = 0$ when i is larger than j (later q 's are orthogonal to earlier a 's, that is the point of the algorithm).

Here is a 9-line MATLAB code to build Q and R from A . Start with $[m, n] = \text{size}(A)$; $Q = \text{zeros}(m, n)$; $R = \text{zeros}(n, n)$; to get the shapes correct.

```
for j=1:n                                % Gram-Schmidt orthogonalization
    v=A(:,j);                             %  $v$  begins as column  $j$  of  $A$ 
    for i=1:j-1
        R(i,j)=Q(:,i)' $\ast$ A(:,j);          % modify  $A(:,j)$  to  $v$  for more accuracy
        v=v-R(i,j) $\ast$ Q(:,i);             % subtract the projection  $(q_i^T a_j)q_i = (q_i^T v)q_i$ 
    end                                     %  $v$  is now perpendicular to all of  $q_1, \dots, q_{j-1}$ 
    R(j,j)=norm(v);
    Q(:,j)=v/R(j,j);                       % normalize  $v$  to be the next unit vector  $q_j$ 
end
```

If you undo the last step and the middle steps, you find column j :

$$R(j, j)q_j = (v \text{ minus its projections}) = (\text{column } j \text{ of } A) - \sum_{i=1}^{j-1} R(i, j)q_i.$$

Moving the sum to the far left, this is column j in the multiplication $A = QR$.

That crucial change from a_j to v in line 4 gives “*modified Gram-Schmidt*.” In exact arithmetic, the number $R(i, j) = q_i^T a_j$ is the same as $q_i^T v$. (The current v has subtracted from a_j its projections onto earlier q_1, \dots, q_{i-1} . But the new q_i is orthogonal to those directions.) In real arithmetic this orthogonality is not perfect, and computations show a difference in Q . Everybody uses v at that step in the code.

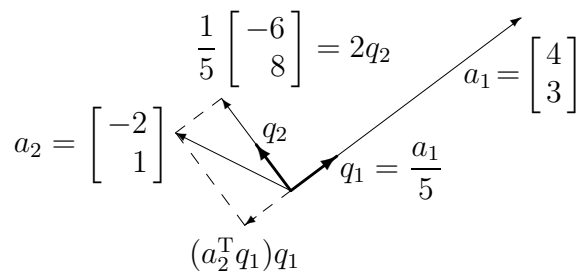
EXAMPLE A is 2 by 2. The columns of Q , normalized by $\frac{1}{5}$, are q_1 and q_2 :

$$A = \begin{bmatrix} 4 & -2 \\ 3 & 1 \end{bmatrix} = \frac{1}{5} \begin{bmatrix} 4 & -3 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & -1 \\ 0 & 2 \end{bmatrix} = QR.$$

Starting with the columns a_1 and a_2 of A , Gram-Schmidt normalizes a_1 to q_1 and subtracts from a_2 its projection in the direction of q_1 . Here are the steps to the q 's:

$$a_1 = \begin{bmatrix} 4 \\ 3 \end{bmatrix} \quad q_1 = \frac{1}{5} \begin{bmatrix} 4 \\ 3 \end{bmatrix} \quad a_2 = \begin{bmatrix} -2 \\ 1 \end{bmatrix} \quad v = a_2 - (q_1^T a_2)q_1 = \frac{1}{5} \begin{bmatrix} -6 \\ 8 \end{bmatrix} \quad q_2 = \frac{1}{5} \begin{bmatrix} -3 \\ 4 \end{bmatrix}$$

Along the way, we divided by $\|a_1\| = 5$ and $\|v\| = 2$. Then 5 and 2 go on the diagonal of R , and $q_1^T a_2 = -1$ is $R(1, 2)$. This figure shows every vector:



MIT OpenCourseWare
<http://ocw.mit.edu>

18.06 Linear Algebra
Spring 2010

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.