Lecture 11: Gram-Schmidt Orthogonalization

June 18, 2025

Outline

Gram-Schmidt Orthogonalization - Introduction

- We have already seen a variation of Gram-Schmidt orthogonalization in constructing A-orthogonal search directions for the conjugate gradient method (see Lecture 09).
- The same general idea applies here to construct the QR factorization:
  - Use columns of A as proposed vectors to be orthogonalized into Q.
  - Build each new vector q_j by orthogonalizing a_j with respect to all previous q vectors, $\{ q_1, q_2, \ldots, q_{j-1} \}$, and then normalizing it.

Gram-Schmidt Orthogonalization - QR factorization via Gram-Schmidt - Orthonormalization for Q

- For a vector a_j, we can orthogonalize it against all previous vectors q_i for $i = 1, \ldots, j-1$, using

$$v_j = a_j - (q_1^T a_j) q_1 - (q_2^T a_j) q_2 - \cdots - (q_{j-1}^T a_j) q_{j-1}. \quad (1)$$

- This removes a_j's components in the orthogonal directions constructed so far.
- We can then directly normalize, giving us

$$q_j = \frac{v_j}{\|v_j\|_2}.$$

- Equation (1) was derived previously in Lecture 09 assuming the form $v_j = a_j + \sum_{i=1}^{j-1} \beta_i q_i$.

Gram-Schmidt Orthogonalization - QR factorization via Gram-Schmidt - Orthonormalization for Q

- To recap, the coefficients β_j were then derived as follows.
- Since we want v_j to be orthogonal to all previous q_i's, we get the equation (for some $1 \leq k \leq j-1$):

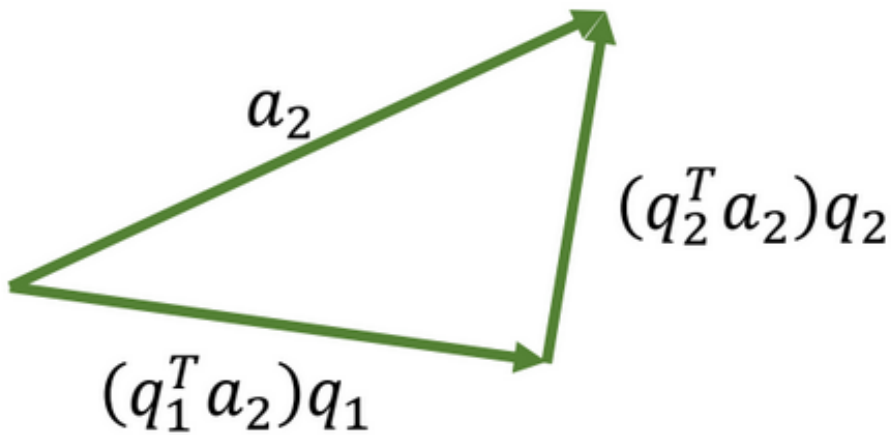
$$\begin{aligned} 0 &= q_k^T v_j, \\ &= q_k^T a_j + \sum_{i=1}^{j-1} \beta_i (q_k^T q_i), \\ &= q_k^T a_j + \beta_k (q_k^T q_k). \end{aligned}$$

- Since $q_k^T q_k = 1$ we have that $\beta_k = -q_k^T a_j$ giving the equation in (1)

$$v_j = a_j - \sum_{i=1}^{j-1} (q_i^T a_j) q_i.$$

Gram-Schmidt Orthogonalization - QR factorization via Gram-Schmidt - Orthonormalization for Q

- Consider the following example in 2D.
- We are given a_2 and the (previous) unit vector q_1.
- We want to find q_2 that is orthonormal to q_1.



Gram-Schmidt Orthogonalization - QR factorization via Gram-Schmidt - Orthonormalization for Q

- We apply the following steps:
  1. Orthogonalize using $v_2 = a_2 - (q_1^T a_2) q_1$,
  2. Normalize to get $q_2 = \frac{v_2}{\|v_2\|_2}$.
- So we have that

$$q_2 = \frac{a_2 - (q_1^T a_2) q_1}{\|a_2 - (q_1^T a_2) q_1\|_2}.$$

- But what are the entries of R in our QR factorization?

Gram-Schmidt Orthogonalization - QR factorization via Gram-Schmidt - Upper Triangular Matrix R

To get the entries of R consider the general form of the QR factorization

$$\begin{bmatrix} | & | & & | \\ a_1 & a_2 & \dots & a_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ q_1 & q_2 & \dots & q_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} r_{11} & \dots & r_{1n} \\ & \ddots & \vdots \\ & & r_{nn} \end{bmatrix}.$$

Gram-Schmidt Orthogonalization - QR factorization via Gram-Schmidt - Upper Triangular Matrix R

Written out componentwise we have

$$
\begin{aligned}
a_1 &= r_{11}q_1 && \Rightarrow && q_1 = \frac{a_1}{r_{11}},\\
a_2 &= r_{12}q_1 + r_{22}q_2 && \Rightarrow && q_2 = \frac{a_2 - r_{12}q_1}{r_{22}},\\
a_3 &= r_{13}q_1 + r_{23}q_2 + r_{33}q_3 && \Rightarrow && q_3 = \frac{a_3 - r_{13}q_1 - r_{23}q_2}{r_{33}},\\
&\vdots && && \vdots\\
a_n &= r_{1n}q_1 + r_{2n}q_2 + \ldots + r_{nn}q_n && \Rightarrow && q_n = \frac{a_n - \sum_{i=1}^{n-1} r_{in}q_i}{r_{nn}}.
\end{aligned}
$$

Gram-Schmidt Orthogonalization - QR factorization via Gram-Schmidt - Upper Triangular Matrix R

- Now we compare the q_i above with the result from our Gram-Schmidt orthogonalization expressions.
- For the 2D example above we have

$$\begin{aligned} \text{Gram-Schmidt: } q_2 &= \frac{a_2 - (q_1^T a_2) q_1}{\|a_2 - (q_1^T a_2) q_1\|_2}, \\ \text{Factorization: } q_2 &= \frac{a_2 - r_{12} q_1}{r_{22}}. \end{aligned}$$

So

$$\begin{aligned} r_{12} &= q_1^T a_2, \text{ and} \\ r_{22} &= \|a_2 - (q_1^T a_2) q_1\|_2. \end{aligned}$$

Gram-Schmidt Orthogonalization - QR factorization via Gram-Schmidt - Upper Triangular Matrix R

- In general (higher dimensions), the entries of R can be written as

$$r_{ij} = (q_i^T a_j),$$
$$r_{jj} = \left\| a_j - \sum_{i=1}^{j-1} r_{ij} q_i \right\|_2 .$$

1. The off-diagonal entries r_{ij} correspond to the lengths of components of a_j in previous directions $q_1, \ldots, q_{j-1}$.
2. The diagonal entries r_{jj} correspond to the length of v_j, required to normalize to q_j.

Gram-Schmidt Orthogonalization - QR factorization via Gram-Schmidt - Upper Triangular Matrix R

- The classic Gram-Schmidt (CGS) algorithm for QR factorization is given in Algorithm 12.
- Note that the classic Gram-Schmidt is numerically **unstable**.
- That is, it is sensitive to round off error (and can even yield non-orthogonal q's).

Gram-Schmidt Orthogonalization - QR factorization via Gram-Schmidt - Upper Triangular Matrix R

Algorithm 1 : Gram-Schmidt Algorithm (Classic)

for $j = 1, 2, \ldots, n$
  $v_j = a_j$ ▷ get next column
  for $i = 1, 2, \ldots, j-1$ ▷ Orthogonalize
    $r_{ij} = q_i^T a_j$
    $v_j = v_j - r_{ij} q_i$
  end for
  $r_{jj} = \|v_j\|_2$ ▷ Normalize
  $q_j = v_j / r_{jj}$
end for

Gram-Schmidt Orthogonalization - Modified Gram-Schmidt

We can alter the classic Gram-Schmidt algorithm in the inner loop for better numerical stability.

Algorithm 2 : QR Factorization With Modified Gram-Schmidt Algorithm

1: **for** $j = 1 : n$
2: $\quad v_j = a_j$ ▷ Get next column
3: **end for**
4: **for** $j = 1 : n$
5: $\quad r_{jj} = \|v_j\|_2$
6: $\quad q_j = \left(\frac{1}{r_{jj}}\right) v_j$ ▷ Normalize
7: $\quad$ **for** $k = j + 1 : n$ ▷ Orthogonalize
8: $\quad\quad r_{jk} = q_j^T v_k$
9: $\quad\quad v_k = v_k - r_{jk} q_j$
10: $\quad$ **end for**
11: **end for**

Gram-Schmidt Orthogonalization - Modified Gram-Schmidt

The modified Gram-Schmidt algorithm (see Algorithm 13) gives an identical result as the classic Gram-Schmidt algorithm **in exact arithmetic**.
In floating point arithmetic, the modified Algorithm 13 is more numerically stable than the classic Algorithm 12.
Think carefully about these pseudocodes.

1. In classical Gram-Schmidt, we take each vector, one at a time, and make it orthogonal to all previous vectors.
2. In modified Gram-Schmidt, we take each vector, and modify all forthcoming vectors to be orthogonal to it.

Once you argue this way, it is clear that both methods are performing the same operations, and are mathematically equivalent.

Gram-Schmidt Orthogonalization - Modified Gram-Schmidt

- But, importantly, modified Gram-Schmidt suffers from round-off instability to a significantly lesser degree.
- This can be explained, in part, from the formulas for Gram-Schmidt, without QR-factorization:

$$\begin{aligned} CGS: v_k &= a_k - \left(q_j^T a_k\right) q_j, \text{ and} \\ MGS: v_k &= v_k - \left(q_j^T v_k\right) q_j. \end{aligned}$$

Gram-Schmidt Orthogonalization - Modified Gram-Schmidt

If an error is made in computing q_2 in CGS, so that $q_1^T q_2 = \delta$ is small, but non-zero, this will not be corrected for in any of the computations that follow:

$$\begin{aligned} v_3 &= a_3 - (q_1^T a_3) q_1 - (q_2^T a_3) q_2, \\ q_2^T v_3 &= q_2^T a_3 - q_2^T (q_1^T a_3) q_1 - q_2^T (q_2^T a_3) q_2, \\ &= q_2^T a_3 - (q_1^T a_3) \underbrace{q_2^T q_1}_{=\delta} - (q_2^T a_3) \underbrace{q_2^T q_2}_{=1}, \\ &= q_2^T a_3 - (q_1^T a_3)\delta - (q_2^T a_3) \\ &= -(q_1^T a_3)\delta. \end{aligned}$$

Gram-Schmidt Orthogonalization - Modified Gram-Schmidt

Similarly,

$$\begin{aligned} v_3 &= a_3 - (q_1^T a_3)q_1 - (q_2^T a_3)q_2, \\ q_1^T v_3 &= q_1^T a_3 - q_1^T(q_1^T a_3)q_1 - q_1^T(q_2^T a_3)q_2, \\ &= q_1^T a_3 - (q_1^T a_3)\underbrace{q_1^T q_1}_{=1} - (q_2^T a_3)\underbrace{q_1^T q_2}_{=\delta}, \\ &= q_1^T a_3 - (q_1^T a_3) - (q_2^T a_3)\delta \\ &= -(q_2^T a_3)\delta. \end{aligned}$$

We see that v_3 is not orthogonal to either of q_1 or q_2.

Gram-Schmidt Orthogonalization - Modified Gram-Schmidt

On the other hand, assume the same for MGS ($n = 3$): $q_1^T q_2 = \delta$ is small, but non-zero. Let's examine how the third vector v_3 changes:

Initially, $v_3^{(0)} = a_3$.

- $\underline{j = 1, k = 3}$

$$\begin{aligned} r_{13} &= q_1^T v_3^{(0)} \\ v_3^{(1)} &= v_3^{(0)} - r_{13} q_1 \\ &= v_3^{(0)} - (q_1^T v_3^{(0)}) q_1 \end{aligned}$$

- $\underline{j = 2, k = 3}$

$$\begin{aligned} r_{23} &= q_2^T v_3^{(1)} \\ v_3^{(2)} &= v_3^{(1)} - r_{23} q_2 \\ &= v_3^{(1)} - (q_2^T v_3^{(1)}) q_2 \end{aligned}$$

Gram-Schmidt Orthogonalization - Modified Gram-Schmidt

Hence we obtain

$$
\begin{aligned}
q_2^T v_3^{(2)} &= q_2^T v_3^{(1)} - q_2^T (q_2^T v_3^{(1)}) q_2 \\
&= q_2^T v_3^{(1)} - (q_2^T v_3^{(1)}) \underbrace{q_2^T q_2}_{=1} \\
&= 0 \qquad (2) \\
q_1^T v_3^{(2)} &= q_1^T v_3^{(1)} - q_1^T (q_2^T v_3^{(1)}) q_2 \\
&= q_1^T v_3^{(1)} - (q_2^T v_3^{(1)}) \underbrace{q_1^T q_2}_{=\delta} \\
&= q_1^T v_3^{(1)} - q_2^T v_3^{(1)} \delta \qquad (3)
\end{aligned}
$$

Gram-Schmidt Orthogonalization - Modified Gram-Schmidt

Computing each of the terms on line (3) gives

$$
\begin{aligned}
q_1^T v_3^{(1)} &= q_1^T [v_3^{(0)} - (q_1^T v_3^{(0)}) q_1] \\
&= q_1^T v_3^{(0)} - q_1^T (q_1^T v_3^{(0)}) q_1] \\
&= q_1^T v_3^{(0)} - (q_1^T v_3^{(0)}) \underbrace{q_1^T q_1}_{=1} \\
&= q_1^T v_3^{(0)} - q_1^T v_3^{(0)} \\
&= 0, \text{ and} \\
q_2^T v_3^{(1)} &= q_2^T [v_3^{(0)} - (q_1^T v_3^{(0)}) q_1] \\
&= q_2^T v_3^{(0)} - q_2^T (q_1^T v_3^{(0)}) q_1 \\
&= q_2^T v_3^{(0)} - (q_1^T v_3^{(0)}) \underbrace{q_2^T q_1}_{=\delta} \\
&= q_2^T v_3^{(0)} - (q_1^T v_3^{(0)}) \delta
\end{aligned}
$$

Gram-Schmidt Orthogonalization - Modified Gram-Schmidt

Putting it all together gives

$$\begin{aligned} q_1^T v_3^{(2)} &= q_1^T v_3^{(1)} - q_2^T v_3^{(1)} \delta \\ &= 0 - [q_2^T v_3^{(0)} - (q_1^T v_3^{(0)})\delta]\delta \\ &= -q_2^T v_3^{(0)} \delta + (q_1^T v_3^{(0)})\delta^2 \end{aligned} \tag{4}$$

Gram-Schmidt Orthogonalization - Modified Gram-Schmidt

- Recall that $v_3 = v_3^{(2)}$ is the final form of the third vector (before normalization).
- Let's check orthogonality, assuming no more errors are made.
- First for q_2:

$$q_2^T v_3^{(2)} \quad = \quad 0, \text{ from equation (2).}$$

  So, we perserve orthogonality to q_2.
- Second for q_1:

$$q_1^T v_3^{(2)} \quad = \quad -q_2^T v_3^{(0)} \delta + (q_1^T v_3^{(0)}) \delta^2, \text{ from equation (4).}$$

Gram-Schmidt Orthogonalization - Modified Gram-Schmidt

Summarized Comparison of CGS versus MGS

Inner Product	CGS	MGS
$q_2^T v_3$	$-(q_1^T a_3)\delta$	0
$q_1^T v_3$	$-(q_2^T a_3)\delta$	$-q_2^T v_3^{(0)}\delta + (q_1^T v_3^{(0)})\delta^2$ $\underbrace{=}_{v_3^{(0)}=a_3}$ $-q_2^T a_3\delta + q_1^T a_3\delta^2$

Remarks:

1. Since δ is very small, therefore δ^2 is much smaller.
2. Because of the opposite signs of the terms involved, it is likely, but not guaranteed, that $|-q_2^T a_3\delta + q_1^T a_3\delta^2| \leq |-(q_2^T a_3)\delta|$.
3. The error in $q_1^T v_3$ is likely no worse than in CGS, but we have eliminated the errors in $q_2^T v_3$, an improvement.

(Reference: `https://www.math.uci.edu/~ttrogdon/105A/html/Lecture23.html`)

Gram-Schmidt Orthogonalization - Modified Gram-Schmidt

Explanation for the Computation of r_{jk}

- Note that the Modified Gram-Schmidt algorithm uses $r_{jk} = q_j^T v_k$, where one might instead expect $r_{jk} = q_j^T a_k$.
- Here I will prove that these two choices must always yield identical results.
- Use the notation of the algorithm throughout the explanation.

Gram-Schmidt Orthogonalization - Modified Gram-Schmidt

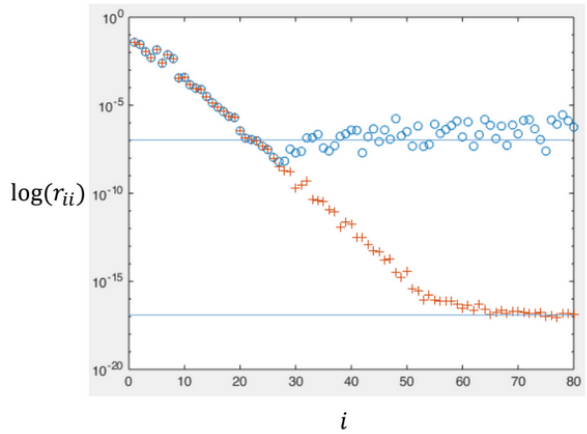
- Let $1 \leq j \leq n$ be arbitrary.
- Let $j+1 \leq k \leq n$ be arbitrary.
- I claim that $q_j^T v_k = q_j^T a_k$.
- Note that v_k is updated once per (outer) j-loop.
- Since that update takes place AFTER the computation of the $q_j^T v_k$ inner product, at the time of that computation,

$$
\begin{aligned}
v_k &= a_k - \sum_{\ell=1}^{j-1} r_{\ell k} q_\ell, \text{ so that} \\
q_j^T v_k &= q_j^T \left[a_k - \sum_{\ell=1}^{j-1} r_{\ell k} q_\ell \right] \\
&= q_j^T a_k - \sum_{\ell=1}^{j-1} r_{\ell k} \underbrace{q_j^T q_\ell}_{=0, \text{ since } \ell<j} \\
&= q_j^T a_k, \text{ as claimed.}
\end{aligned}
$$

Gram-Schmidt Orthogonalization - Modified Gram-Schmidt

- The example below computes the QR factorization of a random matrix with hugely varying magnitudes of r_{ii} (diagonal) entries.

Gram-Schmidt Orthogonalization - Modified Gram-Schmidt



- The blue points are the result using the **classic** Gram-Schmidt algorithm.
- It runs out of accuracy at around $\sqrt{E_{\text{machine}}}$.
- The orange points are the result with the **modified** Gram-Schmidt algorithm.
- The modified algorithm runs out of accuracy at around E_{machine}.
- For more information see Lecture 9, Experiment 2 in Trefethen & Bau.

Gram-Schmidt Orthogonalization - Modified Gram-Schmidt

Another exercise is to find the QR factorization of

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

via Gram-Schmidt orthogonalization. The solution to this problem is

$$Q = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{3}}{3} & -\frac{\sqrt{6}}{6} \\ 0 & \frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{6} \end{bmatrix} \quad \text{and} \quad R = \begin{bmatrix} \sqrt{2} & \sqrt{2} & \frac{\sqrt{2}}{2} \\ 0 & \sqrt{3} & 0 \\ 0 & 0 & \frac{\sqrt{6}}{2} \end{bmatrix}.$$

See Lecture Notes for the details.

Gram-Schmidt Orthogonalization - Complexity of Gram-Schmidt

- The inner i-loop involves
  - $r_{ij} = q_i^T a_j$ (CGS) or $q_i^T v_j$ (MGS) $\Rightarrow m$ (scalar) multiplications and $m-1$ additions for the inner products,
  - $v_j = v_j - r_{ij} q_i$ (CGS) or $v_j = v_j - r_{jk} q_j$ (MGS) $\Rightarrow m$ multiplications, and m subtractions.
- Hence, the flops per inner loop $\approx 4m$.
- An approximation of the total flops is therefore

$$\begin{aligned}
\sum_{j=1}^{n} \sum_{i=1}^{j-1} 4m &= 4m \sum_{j=1}^{n} (j-1), \\
&\approx 4m \sum_{j=1}^{n} = 4m \frac{n(n+1)}{2}, \\
&\approx 2mn^2.
\end{aligned}$$

Gram-Schmidt Orthogonalization - Complexity of Gram-Schmidt

- If the matrix is square ($m = n$),
  flops(Gram-Schmidt) $= 2n^3 + O(n^2) \approx 3 \times$ flops(LU).
- We could use QR factorization to solve linear systems also, but at a cost $3\times$ greater than LU factorization.
- As was pointed out earlier, if A is square and $A = QR$, then solving $Ax = b$ for x is equivalent to solving $Rx = Q^T b$ for x.