Lecture 13: Givens Rotations

June 26, 2025

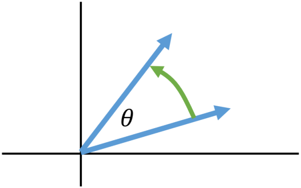
Outline

Givens Rotations - Rotation Matrices in 2D

- First consider rotating a vector in **two dimensions**.
- This can be described as multiplication with a 2×2 matrix

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}.$$

- That is, $a = Rb$ rotates the vector b *counterclockwise* by θ as shown in the figure below.



Givens Rotations - Rotation Matrices in 2D

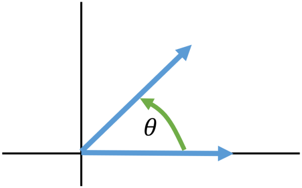
- The columns of $R(\theta)$ are orthonormal: using trigonometric identities we have $\cos^2\theta + \sin^2\theta = 1$ and $\cos\theta\sin\theta - \cos\theta\sin\theta = 0$.
- Hence it is easy to see that $R(\theta)$ is an orthogonal matrix.
- The transpose of the matrix $R(\theta)$ gives a *clockwise* rotation (i.e., the inverse operation).

Givens Rotations - Rotation Matrices in 2D

- As an example, if we want to rotate a vector by $\theta = \frac{\pi}{4}$ (45 degrees) the rotation matrix is

$$R = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}.$$

- If $b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ then, $a = Rb = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$ as shown in the figure below.



Givens Rotations

- A **Givens rotation** zeros individual elements “selectively” by an orthogonal matrix operation that performs rotation in the **(i,k)-plane only**.

Givens rotation matrices have the form

$$
G(i,k,\theta)^T = \begin{array}{cc}
\left[\begin{array}{ccccccccc}
1 & & & & & & & & \\
 & \ddots & & & & & & & \\
 & & c & & & & -s & & \\
 & & & 1 & & & & & \\
 & & & & \ddots & & & & \\
 & & & & & 1 & & & \\
 & & s & & & & c & & \\
 & & & & & & & \ddots & \\
 & & & & & & & & 1
\end{array}\right]
&
\begin{array}{l}
\\ \\ \text{row}(i) \\ \\ \\ \\ \text{row}(k) \\ \\ \\
\end{array}
\\
\begin{array}{ccccccccc}
 & & \text{col}(i) & & & & \text{col}(k) & &
\end{array}
&
\end{array}
$$

where $c = \cos\theta, s = \sin\theta$.

Givens Rotations

- The matrix mostly resembles the identity except for two rows/columns describing a rotation.
- $G(i, k, \theta)$ is always an orthogonal matrix.
- We choose i, k so that left multiplication by $G(i, k, \theta)$ uses the entry on row i, to zero out the entry on row k, in the column in which we are working.

Givens Rotations

Explanation of the Following Setup

- Let x, y be vectors in $\mathbb{R}^m$.
- Both x and y will be columns of the matrix we are about to process:
  - x is the input column, and
  - y is the output column.
- It is implicit that we have already chosen a column on which to work. This provides us with our input, x, and shows what y we are pursuing.

Givens Rotations

- Consider $y = G(i, k, \theta)^T x$, then

$$y_j = \begin{cases} cx_i - sx_k & \text{for } j = i \\ sx_i + cx_k & \text{for } j = k \\ x_j & \text{for } j \neq i, k \end{cases}$$

Givens Rotations

- To force $y_k = 0$ we must let

$$\begin{aligned} c &= \frac{x_i}{\sqrt{x_i^2 + x_k^2}}, \text{ and} \\ s &= -\frac{x_k}{\sqrt{x_i^2 + x_k^2}}. \end{aligned}$$

- Exercise: confirm $y_k = 0$ with the above values of c and s by substituting into the definition of y_j.
- Note that the value of θ itself is not needed when computing $y = G(i, k, \theta)^T x$.
- Also, note that computing the product $G(i, k, \theta)^T A$ affects only row(i) and row(k).

Givens Rotations

Solution to Exercise: Recall that y_k (i.e. y_j, where $j = k$) is defined by

$$y_k = sx_i + cx_k.$$

So we compute

$$\begin{aligned}
y_k &= sx_i + cx_k \\
&= \left(-\frac{x_k}{\sqrt{x_i^2 + x_k^2}}\right) x_i + \left(\frac{x_i}{\sqrt{x_i^2 + x_k^2}}\right) x_k \\
&= -\frac{x_i x_k}{\sqrt{x_i^2 + x_k^2}} + \frac{x_i x_k}{\sqrt{x_i^2 + x_k^2}} \\
&= \frac{x_i x_k - x_i x_k}{\sqrt{x_i^2 + x_k^2}} \\
&= 0.
\end{aligned}$$

Givens QR Factorization Process

- To perform a QR factorization we zero entries one at a time working upwards along columns.
- For example, the process performed on a 4×3 matrix is

$$\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \xrightarrow{G(3,4)^T} \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ 0 & \times & \times \end{bmatrix} \xrightarrow{G(2,3)^T} \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix} \xrightarrow{G(1,2)^T}$$

$$\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix} \xrightarrow{G(3,4)^T} \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \end{bmatrix} \xrightarrow{G(2,3)^T} \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \end{bmatrix} \xrightarrow{G(3,4)^T} R$$

Remarks on Matrix Dimensions:

1. A is $m \times n$, for some $m \geq n$.
2. The G_is are all $m \times m$.

Givens QR Factorization Process

- We are, step-by-step, turning A into R.
- The output will be a reduced QR factorization: R is $m \times n$; Q is $m \times m$, orthogonal.
- We can recover Q later, from the G's.
- Each rotation is computed based on the current matrix, not based on the original matrix.

Givens QR Factorization Process

Explanation:

1. x is a column of our coefficient matrix.
2. y is the same column of the coefficient matrix, after we have applied a Givens rotation to zero out the k^{th} entry.
3. We construct $G(i,k)^T$, to zero out the k^{th} entry of x. This is why we set $y_k = 0$ to determine what c and s have to be.
4. Think of $G(i,k)^T$ as the matrix which carries out the needed rotation in the (i,k)-plane to zero out the k^{th} entry of x.
5. As pointed out above, $G(i,k)^T$ affects only rows i and k.
6. Also as pointed out above, to perform a QR factorization we zero entries one at a time, **working upwards along columns**.

Givens QR Factorization Process

6. This should now explain the indices in the diagrams:
   1. The first line works upwards through column 1:
      1. Perform the Givens rotation on rows 3 and 4 that zeroes out the (4,1) entry of the matrix ($G(3,4)^T$).
      2. Perform the Givens rotation on rows 2 and 3 that zeroes out the (3,1) entry of the matrix ($G(2,3)^T$).
      3. Perform the Givens rotation on rows 1 and 2 that zeroes out the (2,1) entry of the matrix ($G(1,2)^T$).
   2. The second line then does the analogous steps to create the needed zeroes in columns 2 and 3.
   3. Note that $G(3,4)^T$ on line 2 is **not** the same as $G(3,4)^T$ on line 1: they are operating on different columns. The column number is implicit (because we always know which column we are processing). The (i,k) indices refer to the **row entries in the current column**.

Givens QR Factorization Process

- To obtain Q we let G_ℓ^T denote the ℓ^{th} Givens rotation.
- We can assemble Q from

$$\begin{aligned} G_\ell^T G_{\ell-1}^T \cdots G_2^T G_1^T A &= R, \\ \Rightarrow A &= G_1 G_2 \cdots G_{\ell-1} G_\ell R, \quad (G_i \text{ orthogonal}) \\ \Rightarrow Q &= G_1 G_2 \cdots G_{\ell-1} G_\ell, \quad (\text{because } A = QR). \end{aligned}$$

- Quick Reminder About Dimensions:
  1. A is $m \times n$, for some $m \geq n$.
  2. The G_is are all $m \times m$.
- **Remark:** We do not require R's "diagonal" entries to be positive. So this QR factorization might not agree with the unique QR factorization described in Lecture 10.

Givens QR Factorization Process

Example: Let $A = \begin{bmatrix} 1 & -1 \\ 0 & 2 \\ 1 & 1 \end{bmatrix}$. We will compute a QR factorization of A, via Givens rotations.

<u>Column #1</u>

- There is no need to zero out the $(2, 1)$ entry.
- Compute $G(1, 3)^T$ to use the $(1, 1)$ entry to zero out the $(3, 1)$ entry.

$$\begin{aligned} c &= \frac{x_1}{\sqrt{x_1^2 + x_3^2}} \\ &= \frac{1}{\sqrt{1^2 + 1^2}} \\ &= \frac{1}{\sqrt{2}} \\ &= \frac{\sqrt{2}}{2}, \text{ and} \end{aligned}$$

Givens QR Factorization Process

$$
\begin{aligned}
s &= -\frac{x_3}{\sqrt{x_1^2 + x_3^2}} \\
&= -\frac{1}{\sqrt{1^2 + 1^2}} \\
&= -\frac{1}{\sqrt{2}} \\
&= -\frac{\sqrt{2}}{2}.
\end{aligned}
$$

Hence we get

$$
G(1,3)^T = \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{bmatrix}.
$$

Givens QR Factorization Process

We check that left mulitplying A by $G(1,3)^T$ has the desired effect.

$$\begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 2 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{2} & 0 \\ 0 & 2 \\ 0 & \sqrt{2} \end{bmatrix}.$$

This completes Column #1.

Column #2

- Compute $G(2,3)^T$ to use the $(2,2)$ entry to zero out the $(3,2)$ entry.

Givens QR Factorization Process

$$
\begin{aligned}
c &= \frac{x_2}{\sqrt{x_2^2 + x_3^2}} \\
&= \frac{2}{\sqrt{2^2 + \sqrt{2}^2}} \\
&= \frac{2}{\sqrt{6}} \\
&= \frac{2\sqrt{6}}{6} \\
&= \frac{\sqrt{6}}{3}, \text{ and}
\end{aligned}
$$

Givens QR Factorization Process

$$
\begin{aligned}
s &= -\frac{x_3}{\sqrt{x_2^2 + x_3^2}} \\
&= -\frac{\sqrt{2}}{\sqrt{2^2 + \sqrt{2}^2}} \\
&= -\frac{\sqrt{2}}{\sqrt{6}} \\
&= -\frac{1}{\sqrt{3}} \\
&= -\frac{\sqrt{3}}{3}.
\end{aligned}
$$

Givens QR Factorization Process

Hence we get

$$G(2,3)^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \\ 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \end{bmatrix}.$$

We check that left mulitplying A by $G(2,3)^T$ has the desired effect.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \\ 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & 2 \\ 0 & \sqrt{2} \end{bmatrix} = \begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{6} \\ 0 & 0 \end{bmatrix}.$$

This completes Column #2.

Givens QR Factorization Process

This gives immediately that $R = \begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{6} \\ 0 & 0 \end{bmatrix}$.

We compute Q as the product of the G's in the reverse order:

$$
\begin{aligned}
Q &= \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{6}}{3} & -\frac{\sqrt{3}}{3} \\ 0 & \frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \end{bmatrix} \\
&= \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{6}}{6} & -\frac{\sqrt{3}}{3} \\ 0 & \frac{\sqrt{6}}{3} & -\frac{\sqrt{3}}{3} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{6}}{6} & \frac{\sqrt{3}}{3} \end{bmatrix}.
\end{aligned}
$$

One can verify that

$$
QR = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{6}}{6} & -\frac{\sqrt{3}}{3} \\ 0 & \frac{\sqrt{6}}{3} & -\frac{\sqrt{3}}{3} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{6}}{6} & \frac{\sqrt{3}}{3} \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{6} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 2 \\ 1 & 1 \end{bmatrix} = A.
$$

Givens QR Factorization Process

- In terms of complexity,
  flops(Givens QR) $\approx 3mn^2 - n^3 = 1.5 \times$ flops(Householder QR).
- So why bother with this the Givens QR if it is slower then Householder QR?
- The reason is because it is more **flexible** than Householder QR.
- Givens QR factorization can be useful when only a few elements need to be eliminated.

Hessenberg via Givens

- For example, consider an **upper Hessenberg** matrix, which has nonzeros only above the first subdiagonal.
- Performing QR factorization on an upper Hessenberg matrix involves the following

$$
\begin{bmatrix}
\times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times \\
 & \times & \times & \times & \times \\
 & & \times & \times & \times \\
 & & & \times & \times
\end{bmatrix}
\xrightarrow{G(1,2)^T}
\begin{bmatrix}
\times & \times & \times & \times & \times \\
0 & \times & \times & \times & \times \\
 & \times & \times & \times & \times \\
 & & \times & \times & \times \\
 & & & \times & \times
\end{bmatrix}
\xrightarrow{G(2,3)^T}
$$

$$
\begin{bmatrix}
\times & \times & \times & \times & \times \\
0 & \times & \times & \times & \times \\
 & 0 & \times & \times & \times \\
 & & \times & \times & \times \\
 & & & \times & \times
\end{bmatrix}
\xrightarrow{G(3,4)^T}
\begin{bmatrix}
\times & \times & \times & \times & \times \\
0 & \times & \times & \times & \times \\
 & 0 & \times & \times & \times \\
 & & 0 & \times & \times \\
 & & & \times & \times
\end{bmatrix}
\xrightarrow{G(4,5)^T}
$$

$$
\begin{bmatrix}
\times & \times & \times & \times & \times \\
0 & \times & \times & \times & \times \\
 & 0 & \times & \times & \times \\
 & & 0 & \times & \times \\
 & & & 0 & \times
\end{bmatrix}
$$

Hessenberg via Givens

- That is, we only need to zero out the first subdiagonal entries at a cost of $\approx 3n^2$ flops.
- This is less than if one used Householder QR factorization, which operates columnwise on all the entries below the main diagonal.

Least Squares: Normal Equations vs QR

- We have now seen three ways to compute the QR factorization (Gram-Schmidt, Householder, Givens).
- Recall that, as explained in Lecture 12, the shape of the output of the Householder QR factorization is different from the shapes of the other outputs.
- However, the steps to solving the least square problem are the same after the $A = QR$ is computed.
- Therefore, the two main approaches to solving least squares problems are using:
  - the normal equations or
  - the QR factorization.
- QR factorization can also be used to (exactly) solve the system $Ax = b$, when A is square. In this case, we solve $Rx = Q^T b$.

Least Squares: Normal Equations vs QR

- The drawback of using normal equations ($A^TAx = A^Tb$) when solving for the least squares problems is that is it poorly conditioned!
- Recall that the accuracy of a (square) linear system solution is dictated by the condition number $\kappa(A)$.
- However, with the normal equations the accuracy depends on $\kappa(A^TA)$ instead of $\kappa(A)$, which is often much worse.
- For example,

$$A = \begin{bmatrix} 1+10^{-8} & -1 \\ -1 & 1 \end{bmatrix} \Rightarrow \kappa(A) = 4 \times 10^8,$$

$$A^TA = \begin{bmatrix} 2+10^{-8}+10^{-16} & -2-10^{-8} \\ -2-10^{-8} & 2 \end{bmatrix} \Rightarrow \kappa(A) \approx 16 \times 10^{16}.$$

Least Squares: Normal Equations vs QR

- The QR factorization approach to least squares involves solving the system $Rx = Q^T b$.
- Therefore, the solution's accuracy depends on $\kappa(R)$ because

$$\kappa_2(A) = \kappa_2(QR) = \kappa_2(R),$$

  since $\|Q\|_2 = 1$: Q has orthonormal columns.
- **Remark:** For this to make sense, R must be square.

Least Squares: Normal Equations vs QR

- The main point is, we prefer the QR approach, despite its extra cost, because of the potential to encounter ill-conditioned problems (for which the normal equations roughly square the condition number).
- However, the normal equations approach can be used if it is known that A is well-conditioned.