Lec 6: Column Spaces and QR

15-369/669/769: Numerical Computing

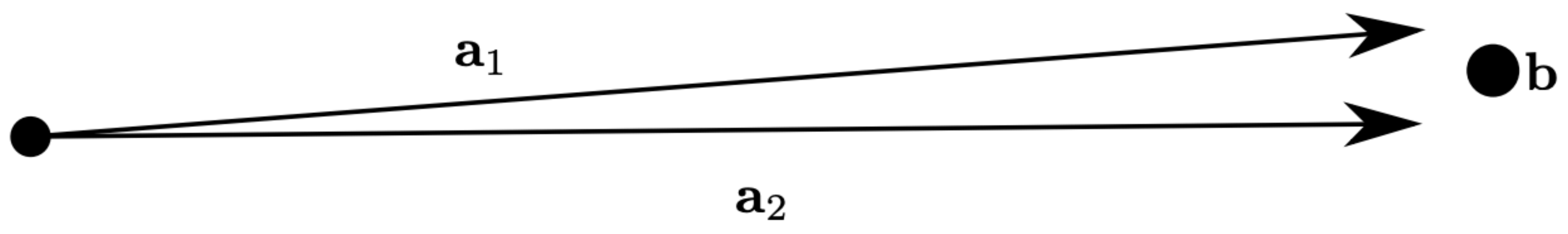
Instructor: Minchen Li

Table of Content

Table of Content

QR Factorization

Motivation



- Solving least squares problems is equivalent to solving the normal equation $A^TA\mathbf{x} = A^T\mathbf{b}$.

- Suppose A is square and invertible, we have

$$\begin{aligned} \text{cond } A^\top A &= \|A^\top A\| \|(A^\top A)^{-1}\| \\ &\approx \|A^\top\| \|A\| \|A^{-1}\| \|(A^\top)^{-1}\| \text{ for many choices of } \|\cdot\| \\ &= \|A\|^2 \|A^{-1}\|^2 \\ &= (\text{cond } A)^2. \end{aligned}$$

- Thus, when the columns of A are nearly linearly dependent, solving $A^TA\mathbf{x} = A^T\mathbf{b}$ is likely to exhibit considerable error compared to **directly working on** A.

QR Factorization

Motivation (cont.)

- The easiest linear system to solve is $I_{n\times n}\mathbf{x} = \mathbf{b}$, where I is the identity matrix.

- For a least squares problem, an ideal setup would be $A^T A = I$.

We can examine the case $Q^\top Q = I_{n\times n}$ to see how it becomes so favorable. Write the columns of Q as vectors $\mathbf{q}_1, \cdots, \mathbf{q}_n \in \mathbb{R}^m$. Then, the product $Q^\top Q$ has the following structure:

$$Q^\top Q = \begin{pmatrix} - & \mathbf{q}_1^\top & - \\ - & \mathbf{q}_2^\top & - \\ & \vdots & \\ - & \mathbf{q}_n^\top & - \end{pmatrix} \begin{pmatrix} | & | & & | \\ \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \\ | & | & & | \end{pmatrix} = \begin{pmatrix} \mathbf{q}_1 \cdot \mathbf{q}_1 & \mathbf{q}_1 \cdot \mathbf{q}_2 & \cdots & \mathbf{q}_1 \cdot \mathbf{q}_n \\ \mathbf{q}_2 \cdot \mathbf{q}_1 & \mathbf{q}_2 \cdot \mathbf{q}_2 & \cdots & \mathbf{q}_2 \cdot \mathbf{q}_n \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{q}_n \cdot \mathbf{q}_1 & \mathbf{q}_n \cdot \mathbf{q}_2 & \cdots & \mathbf{q}_n \cdot \mathbf{q}_n \end{pmatrix}.$$

$$\mathbf{q}_i \cdot \mathbf{q}_j = \begin{cases} 1 & \text{when } i = j \\ 0 & \text{when } i \neq j. \end{cases}$$

The columns of Q are unit-length and orthogonal to one another.

QR Factorization

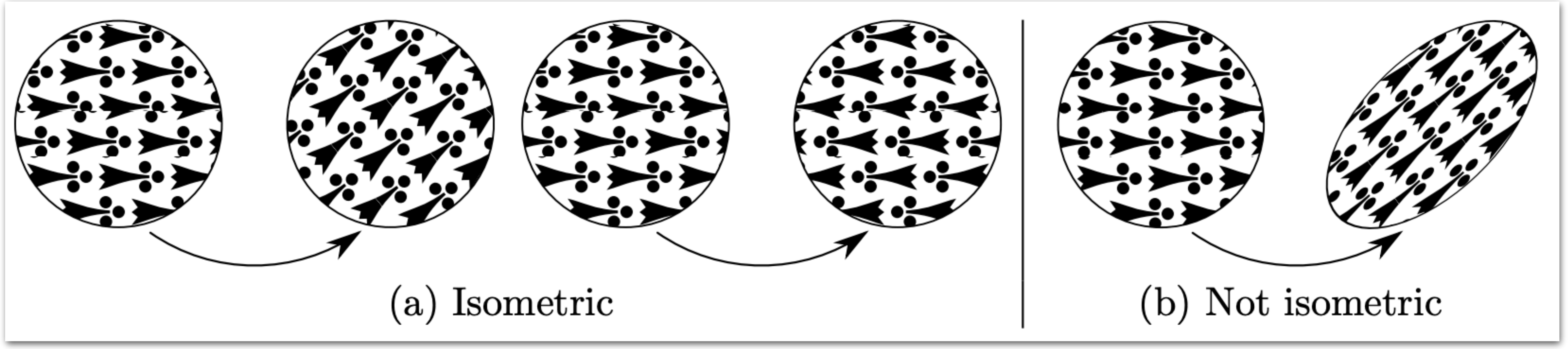
Orthogonal Matrix

> **Definition 5.1** (Orthonormal; orthogonal matrix)**.** A set of vectors $\{\mathbf{v}_1, \cdots, \mathbf{v}_k\}$ is *orthonormal* if $\|\mathbf{v}_i\|_2 = 1$ for all i and $\mathbf{v}_i \cdot \mathbf{v}_j = 0$ for all $i \neq j$. A square matrix whose columns are orthonormal is called an *orthogonal* matrix.

- Identity matrix is orthogonal.
- If Q is square and invertible with $Q^T Q = I$, multiplying both sides by Q^{-1} shows $Q^{-1} = Q^T$.
- If Q is orthogonal, its action does not affect the length and angle of vectors:

$$\|Q\mathbf{x}\|_2^2 = \mathbf{x}^\top Q^\top Q\mathbf{x} = \mathbf{x}^\top I_{n\times n}\mathbf{x} = \mathbf{x}\cdot\mathbf{x} = \|\mathbf{x}\|_2^2. \qquad (Q\mathbf{x})\cdot(Q\mathbf{y}) = \mathbf{x}^\top Q^\top Q\mathbf{y} = \mathbf{x}^\top I_{n\times n}\mathbf{y} = \mathbf{x}\cdot\mathbf{y}.$$

— the map $\mathbf{x} \to Q\mathbf{x}$ is isometric:



QR Factorization

The Idea

- For a general matrix, we can do some computations and connect it to an orthogonal matrix.

> **Proposition 5.1** (Column space invariance)**.** For any $A \in \mathbb{R}^{m\times n}$ and invertible $B \in \mathbb{R}^{n\times n}$, col $A =$ col AB.

> *Proof.* Suppose $\mathbf{b} \in$ col A. By definition, there exists $\mathbf{x}$ with $A\mathbf{x} = \mathbf{b}$. If we take $\mathbf{y} = B^{-1}\mathbf{x}$, then $AB\mathbf{y} = (AB)\cdot(B^{-1}\mathbf{x}) = A\mathbf{x} = \mathbf{b}$, so $\mathbf{b} \in$ col AB. Conversely, take $\mathbf{c} \in$ col AB, so there exists $\mathbf{y}$ with $(AB)\mathbf{y} = \mathbf{c}$. In this case, $A\cdot(B\mathbf{y}) = \mathbf{c}$, showing that $\mathbf{c} \in$ col A.

- We can find a product $Q = AE_1E_2\ldots E_k$ starting from A and applying invertible operation matrices E_i such that Q has orthonormal columns (assuming A has full column rank).

- Proposition 5.1 shows that col Q = col A. Inverting these operations yields a factorization $A = QR$ for $R = E_k^{-1}E_{k-1}^{-1}\ldots E_1^{-1}$. With careful design we can make R upper triangular.

QR Factorization

Application to Least Squares Systems

- When A = QR, by orthogonality of Q we have $A^T A = R^T Q^T Q R = R^T R$.

- Then, the normal equations $A^T A\mathbf{x} = A^T\mathbf{b}$ imply $R^T R\mathbf{x} = R^T Q^T\mathbf{b}$, or $R\mathbf{x} = Q^T\mathbf{b}$.

- If we design R to be a triangular matrix, then $A^T A\mathbf{x} = A^T\mathbf{b}$ can be solved efficiently by back-substitution,

    - without computing $A^T A$ and suffering from a squared condition number!

Table of Content

Gram-Schmidt Orthogonalization

Projections

- Given two vectors $\mathbf{a}$ and $\mathbf{b}$, with $\mathbf{a} \neq \mathbf{0}$. Which multiple of $\mathbf{a}$ is closest to $\mathbf{b}$?
- Solve by minimizing $\|c\mathbf{a} - \mathbf{b}\|^2$ over all possible $c \in \mathbb{R}$.
- Normal equations show $\mathbf{a}^T\mathbf{a}c = \mathbf{a}^T\mathbf{b}$, or

$$c = \frac{\mathbf{a} \cdot \mathbf{b}}{\mathbf{a} \cdot \mathbf{a}} = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|_2^2}.$$

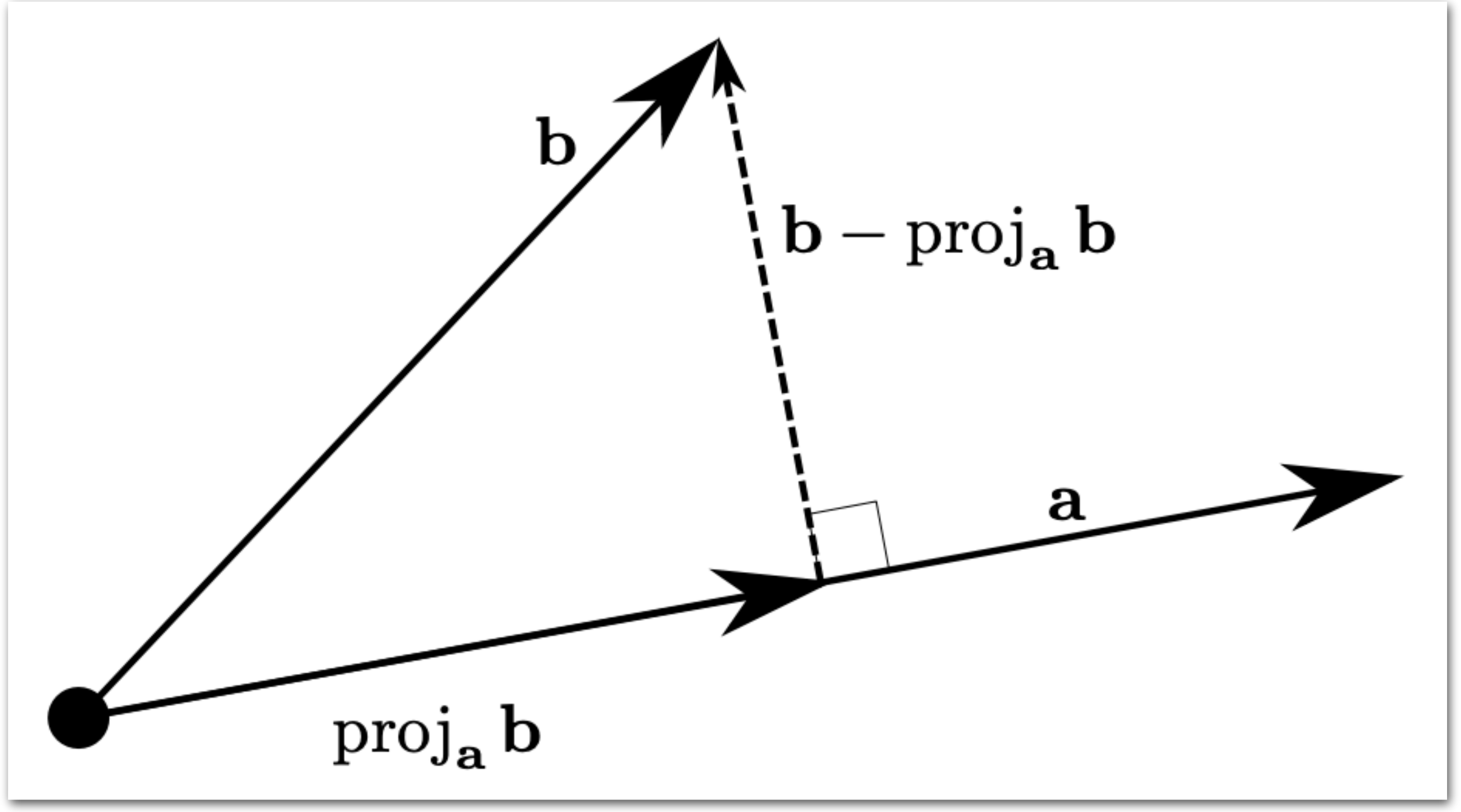
We denote the resulting *projection* of $\mathbf{b}$ onto $\mathbf{a}$ as:

$$\operatorname{proj}_{\mathbf{a}} \mathbf{b} := c\mathbf{a} = \frac{\mathbf{a} \cdot \mathbf{b}}{\mathbf{a} \cdot \mathbf{a}}\mathbf{a} = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|_2^2}\mathbf{a}.$$

By design, $\operatorname{proj}_{\mathbf{a}} \mathbf{b}$ is parallel to $\mathbf{a}$. What about the remainder $\mathbf{b} - \operatorname{proj}_{\mathbf{a}}\mathbf{b}$?

Gram-Schmidt Orthogonalization

Vector Decomposition via Projection



$$\begin{aligned}
\mathbf{a} \cdot (\mathbf{b} - \operatorname{proj}_{\mathbf{a}} \mathbf{b}) &= \mathbf{a} \cdot \mathbf{b} - \mathbf{a} \cdot \left(\frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|_2^2} \mathbf{a} \right) \text{ by definition of } \operatorname{proj}_{\mathbf{a}} \mathbf{b} \\
&= \mathbf{a} \cdot \mathbf{b} - \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|_2^2} (\mathbf{a} \cdot \mathbf{a}) \text{ by moving the constant outside the dot product} \\
&= \mathbf{a} \cdot \mathbf{b} - \mathbf{a} \cdot \mathbf{b} \text{ since } \mathbf{a} \cdot \mathbf{a} = \|\mathbf{a}\|_2^2 \\
&= 0.
\end{aligned}$$

- We have decomposed $\mathbf{b}$ into a component $\operatorname{proj}_{\mathbf{a}}\mathbf{b}$ parallel to $\mathbf{a}$ and another component $\mathbf{b} - \operatorname{proj}_{\mathbf{a}}\mathbf{b}$ orthogonal to $\mathbf{a}$.

- This extends to projection onto the span of a set of vectors.

Gram-Schmidt Orthogonalization

Projection onto a Span

- Suppose that $\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2, \ldots, \hat{\mathbf{a}}_k$ are orthonormal, $\operatorname{proj}_{\hat{\mathbf{a}}_i} \mathbf{b} = (\hat{\mathbf{a}}_i \cdot \mathbf{b})\hat{\mathbf{a}}_i$.

- We can project $\mathbf{b}$ onto span $\{ \hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2, \ldots, \hat{\mathbf{a}}_k \}$ by minimizing

$$
\begin{aligned}
E(c_1, c_2, \ldots, c_k) &:= \|c_1\hat{\mathbf{a}}_1 + c_2\hat{\mathbf{a}}_2 + \cdots + c_k\hat{\mathbf{a}}_k - \mathbf{b}\|_2^2 \\
&= \left(\sum_{i=1}^{k} \sum_{j=1}^{k} c_i c_j (\hat{\mathbf{a}}_i \cdot \hat{\mathbf{a}}_j) \right) - 2\mathbf{b} \cdot \left(\sum_{i=1}^{k} c_i \hat{\mathbf{a}}_i \right) + \mathbf{b} \cdot \mathbf{b} \\
&\qquad \text{by applying and expanding } \|\mathbf{v}\|_2^2 = \mathbf{v} \cdot \mathbf{v} \\
&= \sum_{i=1}^{k} \left(c_i^2 - 2c_i \mathbf{b} \cdot \hat{\mathbf{a}}_i \right) + \|\mathbf{b}\|_2^2 \text{ since the } \hat{\mathbf{a}}_i\text{'s are orthonormal.}
\end{aligned}
$$

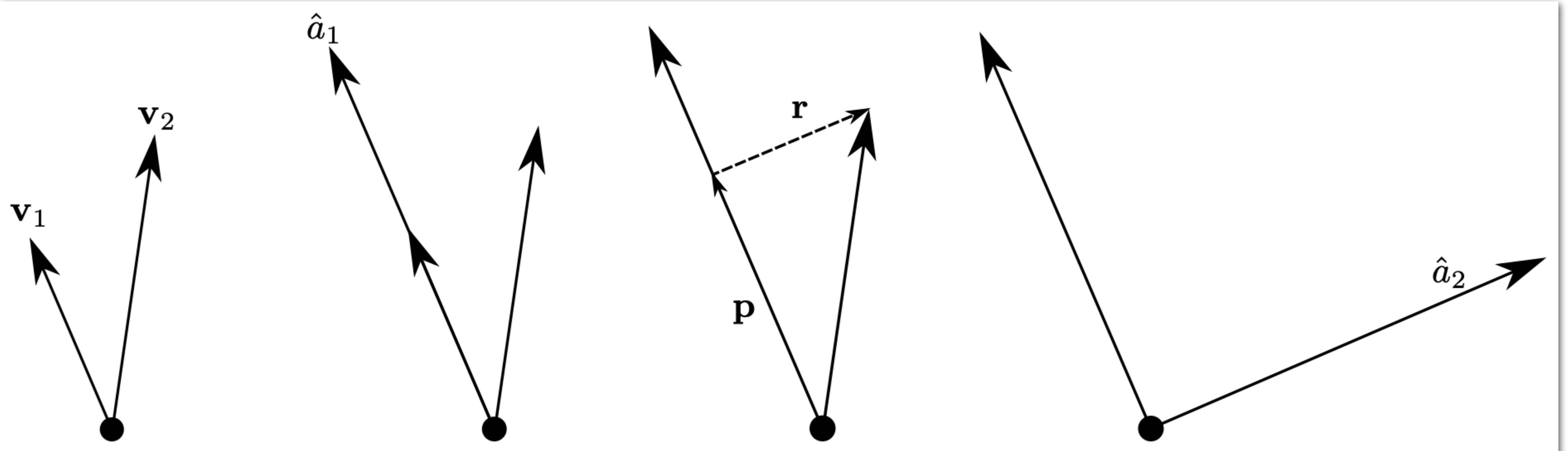
$$0 = \frac{\partial E}{\partial c_i} = 2c_i - 2\mathbf{b} \cdot \hat{\mathbf{a}}_i \implies c_i = \hat{\mathbf{a}}_i \cdot \mathbf{b}.$$

$$\boxed{\operatorname{proj}_{\operatorname{span}\{\hat{\mathbf{a}}_1, \cdots, \hat{\mathbf{a}}_k\}} \mathbf{b} = (\hat{\mathbf{a}}_1 \cdot \mathbf{b})\hat{\mathbf{a}}_1 + \cdots + (\hat{\mathbf{a}}_k \cdot \mathbf{b})\hat{\mathbf{a}}_k.}$$

$$\boxed{\hat{\mathbf{a}}_i \cdot (\mathbf{b} - \operatorname{proj}_{\operatorname{span}\{\hat{\mathbf{a}}_1, \cdots, \hat{\mathbf{a}}_k\}} \mathbf{b}) = 0.}$$

Gram-Schmidt Orthogonalization

Gram-Schmidt Algorithm



(a) Input (b) Rescaling (c) Projection (d) Normalization

function GRAM-SCHMIDT$(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n)$

$\triangleright$ Computes an orthonormal basis $\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_k$ for span $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$

$\triangleright$ Assumes $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are linearly independent.

$\hat{\mathbf{a}}_1 \leftarrow \mathbf{v}_1 / \|\mathbf{v}_1\|_2$ $\triangleright$ Nothing to project out of the first vector

for $i \leftarrow 2, 3, \ldots, n$

$\quad \mathbf{p} \leftarrow \mathbf{0}$ $\triangleright$ Projection of $\mathbf{v}_i$ onto span $\{\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_{i-1}\}$

$\quad$ **for** $j \leftarrow 1, 2, \ldots, i-1$

$\quad\quad \mathbf{p} \leftarrow \mathbf{p} + (\mathbf{v}_i \cdot \hat{\mathbf{a}}_j)\hat{\mathbf{a}}_j$ $\triangleright$ Projecting onto orthonormal basis

$\quad \mathbf{r} \leftarrow \mathbf{v}_i - \mathbf{p}$ $\triangleright$ Residual is orthogonal to current basis

$\quad \hat{\mathbf{a}}_i \leftarrow \mathbf{r} / \|\mathbf{r}\|_2$ $\triangleright$ Normalize this residual and add it to the basis

return $\{\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_n\}$

Gram-Schmidt Orthogonalization

Example

Example 5.1 (Gram-Schmidt orthogonalization). Suppose we are given $\mathbf{v}_1 = (1,0,0)$, $\mathbf{v}_2 = (1,1,1)$, and $\mathbf{v}_3 = (1,1,0)$. The Gram-Schmidt algorithm proceeds as follows:

1. The first vector $\mathbf{v}_1$ is already unit-length, so we take $\hat{\mathbf{a}}_1 = \mathbf{v}_1 = (1,0,0)$.

2. Now, we remove the span of $\hat{\mathbf{a}}_1$ from the second vector $\mathbf{v}_2$:

$$\mathbf{v}_2 - \operatorname{proj}_{\hat{\mathbf{a}}_1} \mathbf{v}_2 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \left[\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right] \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}.$$

$\hat{\mathbf{a}}_2 = (0, 1/\sqrt{2}, 1/\sqrt{2})$.

3. Finally, we remove $\operatorname{span}\{\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2\}$ from $\mathbf{v}_3$:

$$\begin{aligned} &\mathbf{v}_3 - \operatorname{proj}_{\operatorname{span}\{\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2\}} \mathbf{v}_3 \\ &= \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} - \left[\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \right] \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} - \left[\begin{pmatrix} 0 \\ 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \right] \begin{pmatrix} 0 \\ 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ 1/2 \\ -1/2 \end{pmatrix}. \end{aligned}$$

$\hat{\mathbf{a}}_3 = (0, 1/\sqrt{2}, -1/\sqrt{2})$.

Gram-Schmidt Orthogonalization

Computing QR Factorization

- Start with a matrix $A \in \mathbb{R}^{m \times n}$ whose columns are $\mathbf{v}_1, \ldots, \mathbf{v}_n$, we can implement Gram-Schmidt using a series of column operations on A:
  - Dividing column i of A by its norm $\Leftrightarrow$ post-multiplying A by a **diagonal** matrix.
  - The projection step for column i involves subtracting only multiples of columns $j < i$, and thus can be implemented with an **upper-triangular** elimination matrix.
- Thus, we can use Gram-Schmidt to obtain a factorization $A = QR$, where $Q \in \mathbb{R}^{m \times n}$ has orthonormal columns and $R \in \mathbb{R}^{n \times n}$ is upper triangular.
- When the columns of A are linearly independent, one way to find R is as the product $R = Q^T A$; a more stable approach is to keep track of operations as we did for Gaussian elimination.

Gram-Schmidt Orthogonalization

Computing QR Factorization (cont.)

- Computing QR Factorization using Gram-Schmidt Orthogonalization may result in a non-square matrix Q with orthonormal columns, which implies that $Q^T Q = I_{n\times n}$.
  - But if Q is non-square, we do not know whether QQ^T also equals the identity.
- Due to the division step, the algorithm will fail if the columns of A are linearly dependent, which means the dimension of the column space of A is less than n.
- The Gram-Schmidt algorithm is well known to be numerically unstable, partly because $\hat{\mathbf{a}}_i$'s may not be completely orthogonal after the projection step.

Gram-Schmidt Orthogonalization

Modified Gram-Schmidt Algorithm

function MODIFIED-GRAM-SCHMIDT($\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$)
  ▷ Computes an orthonormal basis $\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_n$ for span $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$
  ▷ Assumes $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are linearly independent.

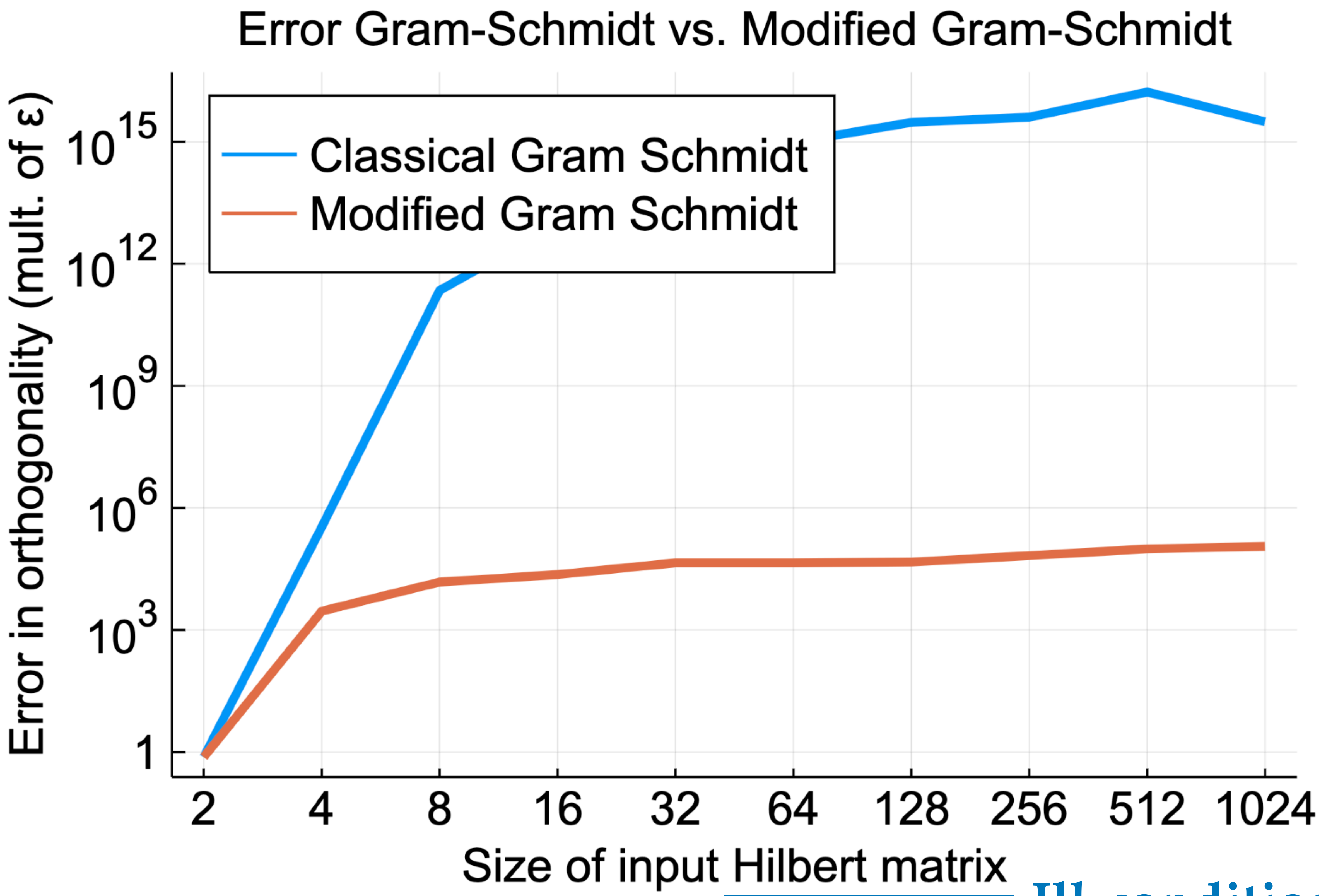
  for $i \leftarrow 1, 2, \ldots, n$
    $\hat{\mathbf{a}}_i \leftarrow \mathbf{v}_i / \|\mathbf{v}_i\|_2$ ▷ Normalize the current vector and store in the basis
    for $j \leftarrow i+1, i+2, \ldots, n$
      $\mathbf{v}_j \leftarrow \mathbf{v}_j - (\mathbf{v}_j \cdot \hat{\mathbf{a}}_i)\hat{\mathbf{a}}_i$ ▷ Project $\hat{\mathbf{a}}_i$ out of the remaining vectors
  return $\{\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_n\}$

- Once $\hat{\mathbf{a}}_i$ is computed, it is projected out of $\mathbf{v}_{i+1}, \ldots, \mathbf{v}_n$, and we never have to consider $\hat{\mathbf{a}}_i$ again.
- This way, even if the basis globally is not completely orthogonal due to rounding, the projection is valid.
- In the absence of rounding, modified Gram-Schmidt and classical Gram-Schmidt generate identical output.

Gram-Schmidt Orthogonalization

Gram-Schmidt v.s. Modified Gram-Schmidt

- *Image source: https://laurenthoeltgen.name/post/gram-schmidt/*



Ill-conditioned but invertible

$$H_{ij} = \frac{1}{i+j-1}.$$

For example, this is the 5 × 5 Hilbert matrix:

$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{bmatrix}.$$

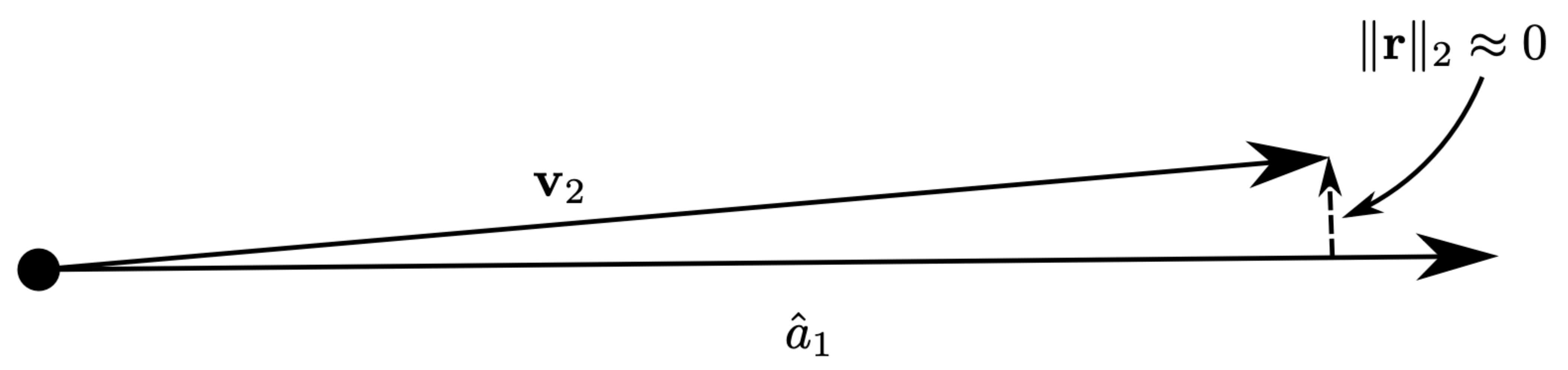
Gram-Schmidt Orthogonalization

Issues of (Modified) Gram-Schmidt Algorithm

- Given vectors $\mathbf{v}_1 = (1,1)$ and $\mathbf{v}_2 = (1+\epsilon,1)$ as input to Gram-Schmidt for some $0 < \epsilon \ll 1$.

- A reasonable basis for span $\{\mathbf{v}_1, \mathbf{v}_2\}$ might be $\{(1,0),(0,1)\}$, but we would get:

$$\begin{aligned}
\hat{\mathbf{a}}_1 &= \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
\mathbf{p} &= \frac{2+\varepsilon}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
\mathbf{r} = \mathbf{v}_2 - \mathbf{p} &= \begin{pmatrix} 1+\varepsilon \\ 1 \end{pmatrix} - \frac{2+\varepsilon}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
&= \frac{1}{2}\begin{pmatrix} \varepsilon \\ -\varepsilon \end{pmatrix}.
\end{aligned}$$



- Computing $\hat{\mathbf{a}}_2$ requires division by a small number.

Table of Content

Householder Transformations

Motivation

- Rather than post-multiplying A by column operations to obtain $Q = AE_1 \ldots E_k$, we can also pre-multiply A by orthogonal matrices Q_i to obtain $Q_k \ldots Q_1 A = R$.
- These Q's act like row operations to eliminate elements of A until it becomes upper triangular.
- Due to orthogonality, we can obtain the QR factorization as $A = (Q_1^T \ldots Q_k^T)R$.
- But this QR will be different from the one given by Gram-Schmidt algorithm. Here:
  - $Q \in \mathbb{R}^{m \times m}$ is invertible,
  - $R \in \mathbb{R}^{m \times n}$ will be potentially non-square and not invertible.
- We will introduce a common **orthogonal** row operation by Householder [1958].

Householder Transformations

Choice of Orthogonal Elimination Matrices

- The space of orthogonal n × n matrices is very large, so we seek a smaller set of possible Q_i's that is easier to work with for elimination.

- From geometric intuition, we know that orthogonal matrices must preserve **angles** and **lengths**, so intuitively they only can **rotate** and **reflect** vectors.

- Householder proposed using only reflection operations to reduce A to upper triangular.

- A well-known alternative by Givens uses only rotations to accomplish the same task *(to be explored in the assignment)*.

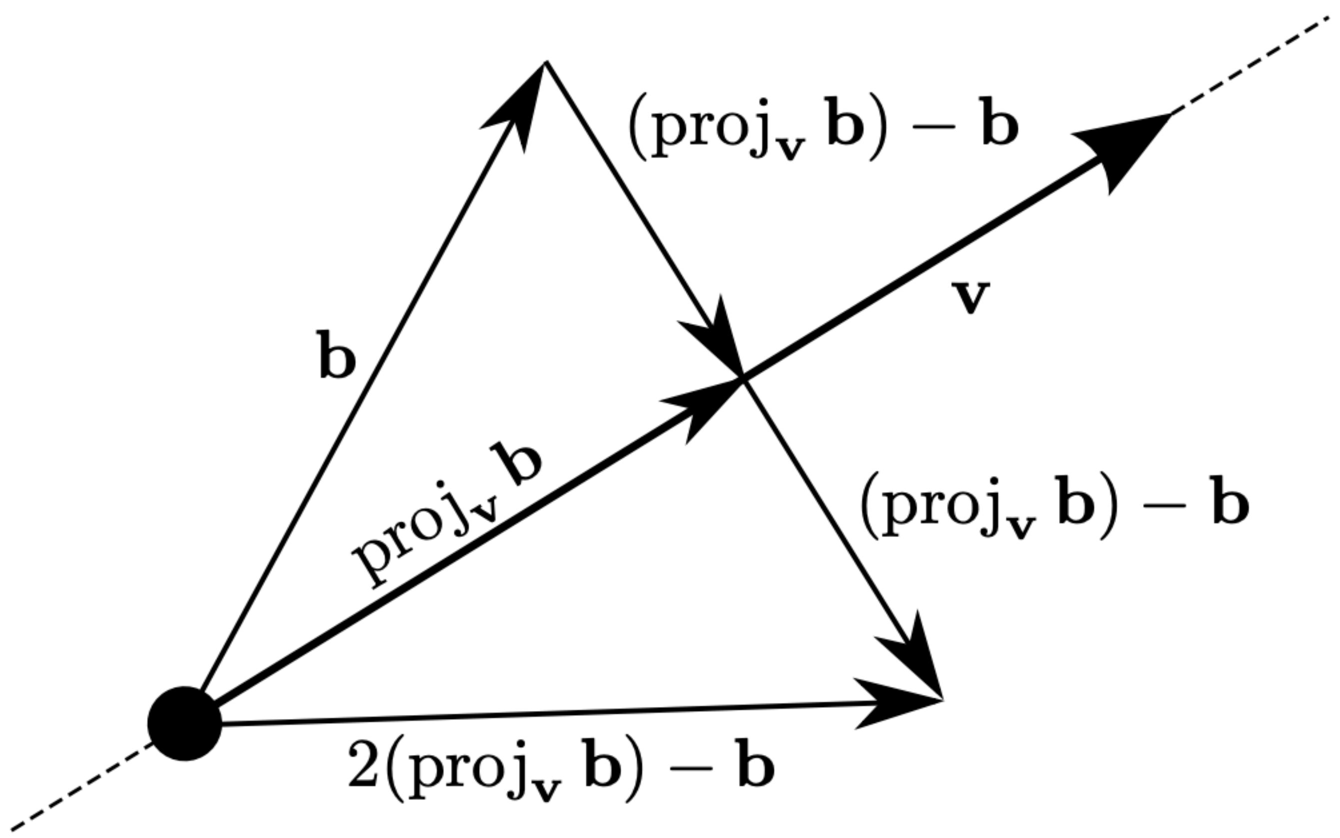
Householder Transformations

Reflection via Projection

- One way to write an orthogonal reflection matrix is in terms of projections.

- To reflect a vector $\mathbf{b}$ over a vector $\mathbf{v}$, we've shown that the residual $\mathbf{r} := \mathbf{b} - \mathrm{proj}_{\mathbf{v}}\mathbf{b}$ is perpendicular to $\mathbf{v}$. Following the reverse of this direction twice reflects $\mathbf{b}$ over $\mathbf{v}$:

$$
\begin{aligned}
2\mathrm{proj}_{\mathbf{v}}\,\mathbf{b} - \mathbf{b} &= 2\frac{\mathbf{v}\cdot\mathbf{b}}{\mathbf{v}\cdot\mathbf{v}}\mathbf{v} - \mathbf{b} \text{ by definition of projection} \\
&= 2\mathbf{v}\cdot\frac{\mathbf{v}^\top\mathbf{b}}{\mathbf{v}^\top\mathbf{v}} - \mathbf{b} \text{ using matrix notation} \\
&= \left(\frac{2\mathbf{v}\mathbf{v}^\top}{\mathbf{v}^\top\mathbf{v}} - I_{n\times n}\right)\mathbf{b} \\
&:= -H_{\mathbf{v}}\mathbf{b}, \text{ where we define } H_{\mathbf{v}} := I_{n\times n} - \frac{2\mathbf{v}\mathbf{v}^\top}{\mathbf{v}^\top\mathbf{v}}.
\end{aligned}
$$



Householder Transformations

Row Elimination via Reflection (1st Column)

- Like in forward-substitution, in our first step we wish to pre-multiply A by a matrix that takes the first column of A, which we will denote $\mathbf{a}$, to some multiple of the first identity vector $\mathbf{e}_1$.

- Using reflections, we need to find some $\mathbf{v}$, c such that $H_{\mathbf{v}}\mathbf{a} = c\mathbf{e}_1$:

$$\begin{aligned} c\mathbf{e}_1 &= H_{\mathbf{v}}\mathbf{a}, \text{ as explained above} \\ &= \left(I_{n\times n} - \frac{2\mathbf{v}\mathbf{v}^\top}{\mathbf{v}^\top\mathbf{v}}\right)\mathbf{a}, \text{ by definition of } H_{\mathbf{v}} \\ &= \mathbf{a} - 2\mathbf{v}\frac{\mathbf{v}^\top\mathbf{a}}{\mathbf{v}^\top\mathbf{v}}. \end{aligned}$$

$$\mathbf{v} = (\mathbf{a} - c\mathbf{e}_1)\cdot\frac{\mathbf{v}^\top\mathbf{v}}{2\mathbf{v}^\top\mathbf{a}}.$$

$$\textit{choose } \mathbf{v} = \mathbf{a} - c\mathbf{e}_1. \quad \mathbf{v} = \mathbf{v}\cdot\frac{\mathbf{v}^\top\mathbf{v}}{2\mathbf{v}^\top\mathbf{a}}.$$

$$\begin{aligned} \text{Assuming } \mathbf{v} \neq \mathbf{0}, \quad 1 &= \frac{\mathbf{v}^\top\mathbf{v}}{2\mathbf{v}^\top\mathbf{a}} \\ &= \frac{\|\mathbf{a}\|_2^2 - 2c\mathbf{e}_1\cdot\mathbf{a} + c^2}{2(\|\mathbf{a}\|_2^2 - c\mathbf{e}_1\cdot\mathbf{a})} \end{aligned}$$

$$\text{or, equivalently, } 0 = \|\mathbf{a}\|_2^2 - c^2 \implies c = \pm\|\mathbf{a}\|_2.$$

Householder Transformations

Row Elimination via Reflection (Remaining Columns)

After choosing $c = \pm\|\mathbf{a}\|_2$, our steps above are all reversible. We originally set out to find $\mathbf{v}$ such that $H_{\mathbf{v}}\mathbf{a} = c\mathbf{e}_1$. By taking $\mathbf{v} = \mathbf{a} - c\mathbf{e}_1$ with $c = \pm\|\mathbf{a}\|_2$, the steps above show:

$$H_{\mathbf{v}}A = \begin{pmatrix} c & \times & \times & \times \\ 0 & \times & \times & \times \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \times & \times & \times \end{pmatrix}.$$

- To fully reduce A to upper triangular, we repeat the steps. During the k-th step, we take $\mathbf{a}$ to be the k-th column of $Q_{k-1}\ldots Q_1A$, and split $\mathbf{a}$ into two components:

  $\mathbf{a} = \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{pmatrix}$. Here, $\mathbf{a}_1 \in \mathbb{R}^{k-1}$ and $\mathbf{a}_2 \in \mathbb{R}^{m-k+1}$. We wish to find $\mathbf{v}$ such that $H_{\mathbf{v}}\mathbf{a} = \begin{pmatrix} \mathbf{a}_1 \\ c \\ \mathbf{0} \end{pmatrix}$.

  Following a parallel derivation to the one above for the case $k = 1$ shows that $\mathbf{v} = \begin{pmatrix} \mathbf{0} \\ \mathbf{a}_2 \end{pmatrix} - c\mathbf{e}_k$ accomplishes exactly this transformation when $c = \pm\|\mathbf{a}_2\|_2$.

Householder Transformations

Householder QR Algorithm

function HOUSEHOLDER-QR(A)
  ▷ Factors $A \in \mathbb{R}^{m\times n}$ as $A = QR$.
  ▷ $Q \in \mathbb{R}^{m\times m}$ is orthogonal and $R \in \mathbb{R}^{m\times n}$ is upper triangular

  $Q \leftarrow I_{m\times m}$
  $R \leftarrow A$
  for $k \leftarrow 1, 2, \ldots, m$
    $\mathbf{a} \leftarrow R\mathbf{e}_k$ ▷ Isolate column k of R and store it in $\mathbf{a}$
    $(\mathbf{a}_1, \mathbf{a}_2) \leftarrow$ SPLIT($\mathbf{a}$,$k-1$) ▷ Separate off the first $k-1$ elements of $\mathbf{a}$
    $c \leftarrow \|\mathbf{a}_2\|_2$ ▷ Find reflection vector $\mathbf{v}$ for the Householder matrix $H_{\mathbf{v}}$
    $\mathbf{v} \leftarrow \begin{pmatrix} \mathbf{0} \\ \mathbf{a}_2 \end{pmatrix} - c\mathbf{e}_k$
    $R \leftarrow H_{\mathbf{v}} R$ ▷ Eliminate elements below the diagonal of the k-th column
    $Q \leftarrow Q H_{\mathbf{v}}^{\top}$
  return Q, R

When $m < n$, it may be preferable to store Q implicitly as a list of vectors $\mathbf{v}$, which fits in the lower triangle that otherwise would be empty in R.

Householder Transformations

Gram-Schmidt QR v.s. Householder QR

- Both algorithms can factor non-square matrices $A \in \mathbb{R}^{m \times n}$ into products QR, but:
  - For Gram-Schmidt, we do column operations on A to obtain Q by orthogonalization. Thus, the dimension of A is that of Q, yielding $Q \in \mathbb{R}^{m \times n}$ and $R \in \mathbb{R}^{n \times n}$.
    - $Q^T Q = I_{n \times n}$ if A has linearly independent columns, but it is likely that $QQ^T \neq I_{m \times m}$
  - When using Householder reflections, we obtain Q as the product of m × m reflection matrices, leaving $R \in \mathbb{R}^{m \times n}$.
    - Q is orthogonal but R might not be square;
    - Still works when A's columns are linearly dependent, but in this case the upper triangle of R may not be invertible.

Householder Transformations

Reduced QR Factorization

- In typical least-squares problems, m ≫ n. We still prefer the Householder method due to its numerical stability, but now the m × m matrix Q might be too large to store.

- To save space, we can utilize the upper-triangular structure of R. For instance, consider the structure of a 5 × 3 matrix R:

$$R = \left(\begin{array}{ccc} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right). \qquad A = QR = \left(\begin{array}{cc} Q_1 & Q_2 \end{array} \right) \left(\begin{array}{c} R_1 \\ 0 \end{array} \right) = Q_1 R_1.$$

- Here, $Q_1 \in \mathbb{R}^{m \times n}$ has orthonormal columns and $R_1 \in \mathbb{R}^{n \times n}$ still contains the upper triangle of R. The factorization $A = Q_1 R_1$ is called the "reduced" QR factorization of A.

- We can still recover least-squares solutions to $A\mathbf{x} = \mathbf{b}$ by solving $R_1\mathbf{x} = Q_1^T\mathbf{b}$

Table of Content

- QR Factorization
- Gram-Schmidt Orthogonalization
- Householder Transformations