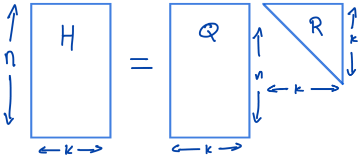
8: Gram-Schmidt orthogonalization

- successive orthogonalization
- Gram-Schmidt algorithm (classic)
- Projection onto a subspace

QR factorization

recall the **reduced** QR factorization (p.7-13) of a tall matrix: $A = QR$



- Q's columns, $\{q_i\}_{i=1}^{k}$, form an **orthonormal set**
- if $\mathbf{rank}(A) = k$, the the factorization is unique and $r_{ii} > 0$
- when $n = k$, Q is **orthogonal**, i.e., $Q^T Q = QQ^T = I$
- it was shown (p.7-20) that $\{q_i\}_{i=1}^{k}$ provide an orthonormal basis for $\mathbf{range}(A)$
- workhorse of numerical linear algebra – how do we compute Q and R?

Successive orthogonalization

given a tall matrix A, find a sequence of vectors that span the column **spaces** of A

$$A = \begin{bmatrix} a_1 & a_2 & a_3 & \dots & a_n \end{bmatrix}$$

define a growing set of matrices

$$A^{(1)} = a_1, \quad A^{(2)} = \begin{bmatrix} a_1 & a_2 \end{bmatrix}, \quad A^{(3)} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}, \quad \dots, \quad A^{(n)} = A$$

further, define $\langle a_1, a_2, a_3 \rangle$ as the subspace spanned by columns a_1, a_2, a_3, then

$$\langle a_1 \rangle \quad \subseteq \quad \langle a_1, a_2 \rangle \quad \subseteq \quad \langle a_1, a_2, a_3 \rangle \quad \subseteq \quad \dots$$

QR factorization can be thought of as constructing an orthonormal sequence of vectors, $\{q_i\}_{i=1}^n$, that span the above spaces

$$\langle q_1, q_2, \dots, q_j \rangle = \langle a_1, a_2, \dots, a_n \rangle, \quad j = 1, \dots, n$$

Column interpretation

$A = QR$ expressed in terms of the columns (c.f. p.4-19)

$$\begin{bmatrix} \vert & \vert & & \vert \\ a_1 & a_2 & \dots & a_n \\ \vert & \vert & & \vert \end{bmatrix} = \begin{bmatrix} \vert & \vert & & \vert \\ q_1 & q_2 & \dots & q_n \\ \vert & \vert & & \vert \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ & r_{22} & \dots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{1n} \end{bmatrix}$$

- $a_1, \dots, a_k$ expressed as **linear combinations** of $q_1, \dots, q_k$

$$\begin{aligned} a_1 &= r_{11} q_1 \\ a_2 &= r_{12} q_1 + r_{22} q_2 \\ a_3 &= r_{13} q_1 + r_{23} q_2 + r_{33} q_3 \\ &\vdots \\ a_n &= r_{1n} q_1 + r_{2n} q_2 + \dots + r_{nn} q_n \end{aligned} \qquad (2)$$

the sequence of equations (2) suggests a sequential process for computing q_i

Objective: at iteration j

- find $q_j \in \langle a_1, \ldots, a_j \rangle$ such that $q_j \perp q_i$ for $i = 1, \ldots j-1$, with $\|q_j\|_2 = 1$

Idea:

- suppose $\{q_1, q_2, \ldots, q_n\}$ is an **orthornormal set** and v is arbitrary
- form a new vector r

$$r = v - (q_1^T v)q_1 - (q_2^T v)q_2 - \ldots - (q_n^T v)q_n \tag{3}$$

- it follows that $r \perp \{q_1, q_2, \ldots, q_n\}$, to verify, observe that

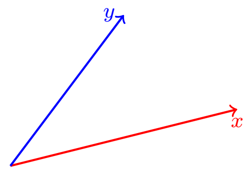
$$q_i^T r = q_i^T v - (q_1^T v)(q_i^T q_1) - \ldots - (q_n^T v)(q_i^T q_n)$$

$$= q_i^T v - (q_i^T v)(q_i^T q_i) = 0$$

Projection

vector-on-vector projection

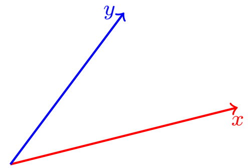
given two non-zero vectors x and y find the scalar multiple of x that is closest to y



Projection

vector-on-vector projection

given two non-zero vectors x and y find the scalar multiple of x that is closest to y

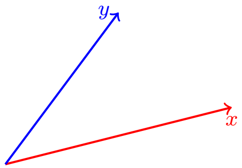


- denote the projected vector as $\mathbf{proj}_x\, y$, we the projection of y onto x

$$\mathbf{proj}_x\, y = \left(\frac{y^T x}{x^T x}\right) x$$

- the projection is **orthogonal**

derivation of the projection



formulate as an unconstrained optimization problem:

$$\underset{\alpha}{\text{minimize}} \quad \|y - \alpha x\|_2^2$$

1. expand the objective function

$$\|y - \alpha x\|_2^2 \quad = \quad y^T y - 2\alpha y^T x + \alpha^2 x^T x$$

2. differentiate with respect to α and set to zero

$$-2y^T x + 2\alpha x^T x = 0$$

3. rearrange to get $\alpha = \frac{y^T x}{x^T x}$

Gram-Schmidt orthogonalization

the iterates in eq. (2) combined with the idea on p. 8-5 suggest a method for computing Q and R

Objective

at iteration j produce $q_j \in \langle a_1, \ldots, a_j \rangle$ with $q_i \perp q_j$ for $i = 1, \ldots, j-1$ with $\|q_j\|_2 = 1$

- adapting (3) to this setting, gives

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

- it has already been shown that $v_j \perp \{q_1, \ldots q_{j-1}\}$
- to make $\|v_j\|_2 = 1$, normalize by multiplying by $1/\|v_j\|_2$

- rewriting this procedure iteratively, we have

$$q_1 = \frac{a_1}{r_{11}}$$

$$q_2 = \frac{a_2 - r_{12}q_1}{r_{22}}$$

$$q_3 = \frac{a_3 - r_{13}q_1 - r_{23}q_2}{r_{33}}$$

$$\vdots$$

$$q_n = \frac{a_n - \sum_{i=1}^{n-1} r_{in}q_i}{r_{nn}}$$

- from which it is clear that

$$r_{ij} = q_i^T a_j \quad (i \neq j) \quad \text{and} \quad r_{jj} = \left\| a_j - \sum_{i=1}^{j-1} r_{ij} q_i \right\|_2$$

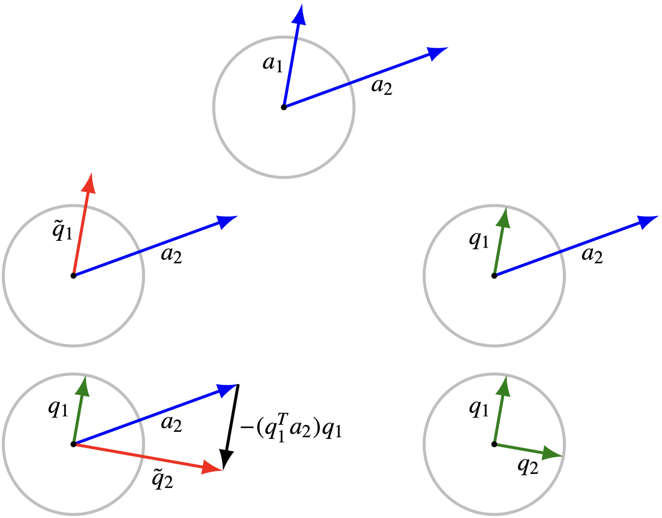
Classical Gram-Schmidt

```
Data: a_1, a_2, ..., a_n                                      /* columns of A */
Result: q_1, q_2, ..., q_n and r_ij    /* columns of Q, non-zero R elements */

for j = 1 to n do
    v_j = a_j
    for i = 1 to j − 1 do
        r_ij = q_i^T a_j
        v_j = v_j − r_ij q_i              /* orthogonal but orthormormal yet */
    end
    r_jj = ‖v_j‖_2
    q_j = v_j / r_jj                                           /* normalize */
end
```

- referred to as "classical Gram-Schmidt" to differentiate from "modified Gram-Schmidt"
- classical Gram-Schmidt is numerically unstable

Example



[Image from Boyd & Vandenberghe, Ch.5]

Numerical example

recall example from p.7-14

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 2 \\ 1 & 5 & 2 \\ -1 & -5 & -4 \\ -1 & 2 & -2 \end{bmatrix}$$

Iteration 1:

$$v_1 = a_1 = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}, \qquad r_{11} = \|v_1\|_2 = 2, \qquad q_1 = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$$

$$v_j = a_j - \underbrace{(q_1^T a_j)}_{r_{12}} q_1 - \underbrace{(q_2^T a_j)q_2 - \ldots - (q_{j-1}^T a_j)q_{j-1}}_{=0}$$

Iteration 2: (Phase 1: produce an orthogonal vector v_2)

$$v_2 = a_2 = \begin{bmatrix} -2 \\ 5 \\ -5 \\ 2 \end{bmatrix}, \qquad r_{12} = q_1^T a_2 = 3,$$

$$v_2 \leftarrow v_2 - r_{12} q_1 = \begin{bmatrix} -2 \\ 5 \\ -5 \\ 2 \end{bmatrix} - \frac{3}{2} \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = 3.5 \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix}$$

$$v_j = a_j - \underbrace{(q_1^T a_j)}_{r_{12}} q_1 - \underbrace{(q_2^T a_j) q_2 - \ldots - (q_{j-1}^T a_j) q_{j-1}}_{=0}$$

Iteration 2: (Phase 1: produce an orthogonal vector v_2)

$$v_2 = a_2 = \begin{bmatrix} -2 \\ 5 \\ -5 \\ 2 \end{bmatrix}, \qquad r_{12} = q_1^T a_2 = 3,$$

$$v_2 \leftarrow v_2 - r_{12} q_1 = \begin{bmatrix} -2 \\ 5 \\ -5 \\ 2 \end{bmatrix} - \frac{3}{2} \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = 3.5 \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix}$$

Iteration 2: (Phase 2: normalize v_2)

$$r_{22} = \|v_2\|_2 = 7 \qquad \Longrightarrow \qquad q_2 = \frac{v_2}{7} = \frac{1}{2} \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix}$$

$$v_j = a_j - \underbrace{(q_1^T a_j)}_{r_{13}} q_1 - \underbrace{(q_2^T a_j)}_{r_{23}} q_2 - \underbrace{\ldots - (q_{j-1}^T a_j) q_{j-1}}_{=0}$$

Iteration 3: (Phase 1: produce an orthogonal vector v_3)

$$v_3 = a_3 = \begin{bmatrix} 2 \\ 2 \\ -4 \\ -2 \end{bmatrix}, \qquad r_{13} = q_1^T a_3 = 5, \qquad r_{23} = q_2^T a_3 = 1$$

$$v_3 \leftarrow v_3 = r_{13} q_1 - r_{23} q_2 = \begin{bmatrix} 0 \\ -1 \\ -1 \\ 0 \end{bmatrix}$$

$$v_j = a_j - \underbrace{(q_1^T a_j)}_{r_{13}} q_1 - \underbrace{(q_2^T a_j)}_{r_{23}} q_2 - \underbrace{\ldots - (q_{j-1}^T a_j) q_{j-1}}_{=0}$$

Iteration 3: (Phase 1: produce an orthogonal vector v_3)

$$v_3 = a_3 = \begin{bmatrix} 2 \\ 2 \\ -4 \\ -2 \end{bmatrix}, \qquad r_{13} = q_1^T a_3 = 5, \qquad r_{23} = q_2^T a_3 = 1$$

$$v_3 \leftarrow v_3 = r_{13} q_1 - r_{23} q_2 = \begin{bmatrix} 0 \\ -1 \\ -1 \\ 0 \end{bmatrix}$$

Iteration 3: (Phase 2: normalize v_3)

$$r_{33} = \|v_3\|_2 = \sqrt{2} \qquad \Longrightarrow \qquad q_3 = \frac{v_3}{\sqrt{2}} = -\begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}$$

Complexity analysis

computing a QR factorization using Gram-Schmidt takes $\sim 2mn^2$ flops

- each inner loop requires $j-1$ inner products of n-vectors

$$q_1^T a_j, \quad q_2^T a_j, \quad \ldots, \quad q_{j-1}^T q_j$$

  $(j-1)(2n-1)$ flops

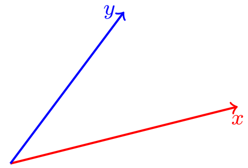
- $j-1$ multiplications for $r_{ij}q_i$: $n(j-1)$ flops
- substracting $j-1$ vectors from v_i: $n(j-1)$ flops
- total flop count for one inner loop is

$$(j-1)(2n-1) + 2n(j-1) = (4n-1)(j-1)$$

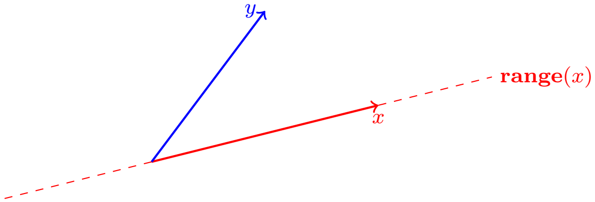
- computing $\|v_j\|_2$ is approximately $2n$ flops
- normalizing a vector is approximately n flops
- summing over all $j = 1, \ldots, k$ iterations gives the stated result

Projection revisited

recall the setup of projecting one vector onto another



looking for a point αx close to y, instead, look for a point in $\mathbf{range}(x)$



generalize this to arbitrary subspaces

Projectors

a **projector** is a square matrix P that satisfies $P^2 = P$

- if we were to shine a light onto the subspace $\mathbf{range}(P)$ from the right direction, then Pv is the shadow projected by the vector v

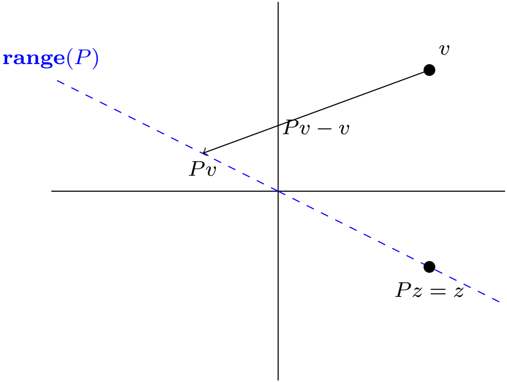
- if $v \in \mathbf{range}(P)$, then it lies on its own shadow and so Pv returns v, let $v = Px$ for some x, then

$$Pv \quad = \quad P^2 x \quad = \quad Px \quad = \quad v$$

- from where does the light shine when $v \neq Pv$?

Projection onto a subspaces

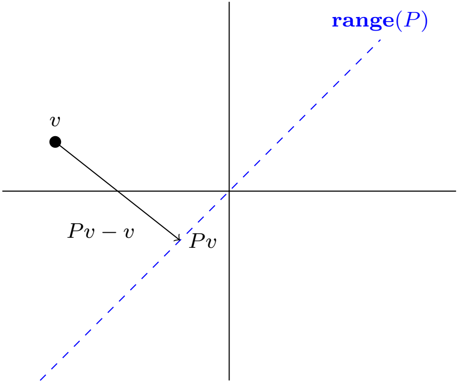
geometric interpretation



- $Pz = z$ because $z \in \textbf{range}(P)$
- $P(Pv) = P^2 v = Pv$ because $Pv \in \textbf{range}(P)$
- $P(Pv - v) = 0$, i.e., $Pv - p \in \textbf{null}(P)$

Orthogonal projectors

an **orthogonal projector** is a square matrix P such that $P^2 = P$ and $P = P^T$



- **do not confuse an orthogonal projector with an orthogonal matrix**

Complementary projectors

if P is a projector (not necessarily orthogonal), then so is $I - P$

$$(I - P)^2 = I - 2P + P^2 = I - P$$

- $I - P$ is called a complementary projector
- projects onto $\mathbf{null}(P)$ - to see this, think about $Pv - v$ from p.8-18

Vector decomposition

let $\{q_1, q_2, \ldots, q_n\}$ be a set of orthonormal vectors in $\mathbb{C}^m$ and let v be an arbitrary vector

$$v = r + \sum_{i=1}^{n} (q_i^* v) q_i = r + \sum_{i=1}^{n} (q_i q_i^*) v$$

- r is the part of v that is orthogonal to $\{q_1, q_2, \ldots, q_n\}$
- actually, r is orthogonal to $\textbf{span}(q_1, q_2, \ldots, q_n)$

if $q_1, q_2, \ldots, q_n$ is a basis for $\mathbb{C}^m$, then

- $n = m$ and $r = 0$
- v is decomposed into m orthogonal components in the direction q_i:

$$v = \sum_{i=1}^{m} (q_i^* v) q_i = \sum_{i=1}^{m} (q_i q_i^*) v$$

Projection with an orthonormal basis

Let $A = QR$, where A is tall with linearly independent columns, then

$$P = QQ^* \text{ projects onto } \mathbf{range}(A)$$

recall that v can be expressed as

$$v = r + \sum_{i=1}^{n} (q_i q_i^*) v$$

then we have

$$v = \underbrace{(I - P)v}_{\text{orthog. } \mathbf{range}(Q)} + \underbrace{Pv}_{\in \mathbf{range}(Q)}$$

and finally, note that that $\mathbf{range}(Q) = \mathbf{range}(A)$ (p. 7-20)

Further topics

- orthogonal subspaces
- numerical stability