1. Matrix rank

- subspaces, dimension, rank
- QR factorization with pivoting
- properties of matrix rank
- low-rank matrices
- pseudo-inverse

Subspace

a nonempty subset $\mathcal{V}$ of $\mathbf{R}^m$ is a *subspace* if

$$\alpha x + \beta y \in \mathcal{V}$$

for all vectors $x, y \in \mathcal{V}$ and scalars α, β

- all linear combinations of elements of $\mathcal{V}$ are in $\mathcal{V}$
- $\mathcal{V}$ is nonempty and closed under scalar multiplication and vector addition

Examples

- $\{0\}$, $\mathbf{R}^m$
- the *span* of a set $\mathcal{S} \subseteq \mathbf{R}^m$: all linear combinations of elements of $\mathcal{S}$

$$\operatorname{span}(\mathcal{S}) = \{\beta_1 a_1 + \cdots + \beta_k a_k \mid a_1, \ldots, a_k \in \mathcal{S},\ \beta_1, \ldots, \beta_k \in \mathbf{R}\}$$

  if $\mathcal{S} = \{a_1, \ldots, a_n\}$ is a finite set, we write $\operatorname{span}(\mathcal{S}) = \operatorname{span}(a_1, \ldots, a_n)$

  (the span of the empty set is defined as $\{0\}$)

Operations on subspaces

three common operations on subspaces ($\mathcal{V}$ and $\mathcal{W}$ are subspaces)

- *intersection:*

$$\mathcal{V} \cap \mathcal{W} = \{x \mid x \in \mathcal{V}, x \in \mathcal{W}\}$$

- *sum:*

$$\mathcal{V} + \mathcal{W} = \{x + y \mid x \in \mathcal{V}, y \in \mathcal{W}\}$$

  if $\mathcal{V} \cap \mathcal{W} = \{0\}$ this is called the *direct sum* and written as $\mathcal{V} \oplus \mathcal{W}$

- *orthogonal complement:*

$$\mathcal{V}^{\perp} = \{x \mid y^T x = 0 \text{ for all } y \in \mathcal{V}\}$$

the result of each of the three operations is a subspace

Range of a matrix

suppose A is an $m \times n$ matrix with columns $a_1, \ldots, a_n$ and rows $b_1^T, \ldots, b_m^T$:

$$A = \left[\begin{array}{ccc} a_1 & \cdots & a_n \end{array}\right] = \left[\begin{array}{c} b_1^T \\ \vdots \\ b_m^T \end{array}\right]$$

Range (column space): the span of the column vectors (a subspace of $\mathbf{R}^m$)

$$\begin{aligned} \text{range}(A) &= \text{span}(a_1, \ldots, a_n) \\ &= \{x_1 a_1 + \cdots + x_n a_n \mid x \in \mathbf{R}^n\} \\ &= \{Ax \mid x \in \mathbf{R}^n\} \end{aligned}$$

the range of A^T is called the *row space* of A (a subspace of $\mathbf{R}^n$):

$$\begin{aligned} \text{range}(A^T) &= \text{span}(b_1, \ldots, b_m) \\ &= \{y_1 b_1 + \cdots + y_m b_m \mid y \in \mathbf{R}^m\} \\ &= \{A^T y \mid y \in \mathbf{R}^m\} \end{aligned}$$

Nullspace of a matrix

suppose A is an $m \times n$ matrix with columns $a_1, \ldots, a_n$ and rows $b_1^T, \ldots, b_m^T$:

$$A = \left[\begin{array}{ccc} a_1 & \cdots & a_n \end{array} \right] = \left[\begin{array}{c} b_1^T \\ \vdots \\ b_m^T \end{array} \right]$$

Nullspace: the orthogonal complement of the row space (a subspace of $\mathbf{R}^n$)

$$\begin{aligned} \mathrm{null}(A) &= \mathrm{range}(A^T)^\perp \\ &= \{x \in \mathbf{R}^n \mid b_1^T x = \cdots = b_m^T x = 0\} \\ &= \{x \in \mathbf{R}^n \mid Ax = 0\} \end{aligned}$$

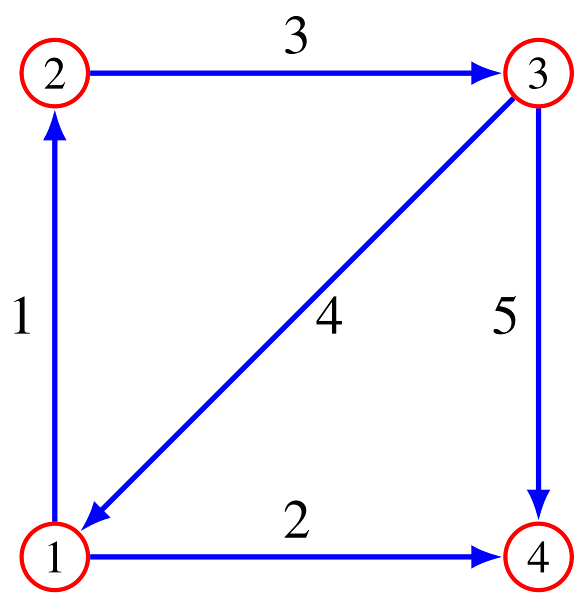
the nullspace of A^T is the orthogonal complement of $\mathrm{range}(A)$ (a subspace of $\mathbf{R}^m$)

$$\begin{aligned} \mathrm{null}(A^T) &= \mathrm{range}(A)^\perp \\ &= \{y \in \mathbf{R}^m \mid a_1^T y = \cdots = a_n^T y = 0\} \\ &= \{y \in \mathbf{R}^m \mid A^T y = 0\} \end{aligned}$$

Exercise

- directed graph with m vertices, n arcs (directed edges)
- node–arc incidence matrix is $m \times n$ matrix A with

$$A_{ij} = \begin{cases} 1 & \text{if arc } j \text{ enters node } i \\ -1 & \text{if arc } j \text{ leaves node } i \\ 0 & \text{otherwise} \end{cases}$$



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

describe in words the subspaces $\text{null}(A)$ and $\text{range}(A^T)$

Linearly independent vectors

vectors $a_1, \ldots, a_n$ are *linearly independent* if

$$x_1 a_1 + x_2 a_2 + \cdots + x_n a_n = 0 \qquad \Longrightarrow \qquad x_1 = x_2 = \cdots = x_n = 0$$

- the zero vector cannot be written as a nontrivial linear combination of $a_1, \ldots, a_n$
- no vector a_i is a linear combination of the other vectors
- in matrix–vector notation: $Ax = 0$ holds only if $x = 0$, where

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

- linear (in)dependence is a property of the set of vectors $\{a_1, \ldots, a_n\}$

  (by convention, the empty set is linearly independent)

recall from 133A: if $a_1, \ldots, a_n$ are linearly independent m-vectors, then $n \leq m$

Basis of a subspace

$\{v_1, \ldots, v_k\}$ is a *basis* for the subspace $\mathcal{V}$ if two conditions are satisfied

1. $\mathcal{V} = \mathrm{span}(v_1, \ldots, v_k)$
2. $v_1, \ldots, v_k$ are linearly independent

- condition 1 means that every $x \in \mathcal{V}$ can be expressed as

$$x = \beta_1 v_1 + \cdots + \beta_k v_k$$

- condition 2 means that the coefficients $\beta_1, \ldots, \beta_k$ are unique:

$$\left.\begin{array}{l} x = \beta_1 v_1 + \cdots + \beta_k v_k \\ x = \gamma_1 v_1 + \cdots + \gamma_k v_k \end{array}\right\} \implies (\beta_1 - \gamma_1)v_1 + \cdots + (\beta_k - \gamma_k)v_k = 0$$

$$\implies \beta_1 = \gamma_1, \quad \ldots, \quad \beta_k = \gamma_k$$

Dimension of a subspace

- every basis of a subspace $\mathcal{V}$ contains the same number of vectors
- this number is called the *dimension* of $\mathcal{V}$ (notation: $\dim(\mathcal{V})$)

Proof: let $\{v_1, \ldots, v_k\}$ be a basis of $\mathcal{V}$ containing k vectors and define

$$B = \begin{bmatrix} v_1 & \cdots & v_k \end{bmatrix}$$

- suppose $a_1, \ldots, a_n$ are linearly independent vectors in $\mathcal{V}$
- each a_i can be expressed as a linear combination of the basis vectors:

$$a_1 = Bx_1, \qquad a_2 = Bx_2, \qquad \ldots, \qquad a_n = Bx_n$$

- the k-vectors $x_1, \ldots, x_n$ are linearly independent:

$$\begin{aligned} \beta_1 x_1 + \cdots + \beta_n x_n = 0 \quad &\Longrightarrow \quad B(\beta_1 x_1 + \cdots + \beta_n x_n) = \beta_1 a_1 + \cdots + \beta_n a_n = 0 \\ &\Longrightarrow \quad \beta_1 = \cdots = \beta_n = 0 \end{aligned}$$

- therefore $n \leq k$; this shows that no basis can contain more than k vectors

Completing a basis

- suppose $\{v_1, \ldots, v_j\} \subset \mathcal{V}$ is a linearly independent set (possibly empty)
- then there exists a basis of $\mathcal{V}$ of the form $\{v_1, \ldots, v_j, v_{j+1}, \ldots, v_k\}$

we *complete* the basis by adding the vectors $v_{j+1}, \ldots, v_k$

Proof

- if $\{v_1, \ldots, v_j\}$ is not already a basis, its span is not $\mathcal{V}$
- then there exist vectors in $\mathcal{V}$ that are not linear combinations of $v_1, \ldots, v_j$
- choose one of those vectors, call it v_{j+1}, and add it to the set
- the set $\{v_1, \ldots, v_{j+1}\}$ is a linearly independent subset of $\mathcal{V}$ with $j+1$ elements
- repeat this process until it terminates
- it terminates because a linearly independent set in $\mathbf{R}^m$ has at most m elements

Consequence: every subspace of $\mathbf{R}^m$ has a basis

Rank of a matrix

Rank: the *rank* of a matrix is the dimension of its range

$$\mathrm{rank}(A) = \dim(\mathrm{range}(A))$$

this is also the maximal number of linearly independent columns

Example: a 4×4 matrix with rank 3

$$A = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 3 & 1 \\ -1 & 2 & 0 & 0 \\ 1 & -1 & 3 & 0 \\ -1 & 2 & 0 & 1 \end{bmatrix}$$

- $\{a_1\}$ is linearly independent (a_1 is not zero)
- $\{a_1, a_2\}$ is linearly independent
- $\{a_1, a_2, a_3\}$ is linearly dependent: $a_3 = 6a_1 + 3a_2$
- $\{a_1, a_2, a_4\}$ is a basis for $\mathrm{range}(A)$: linearly independent and spans $\mathrm{range}(A)$

Rank-r matrices in factored form

we will often encounter matrices in the product form $A = BC$, where

- B is $m \times r$ with linearly independent columns
- C is $r \times n$ with linearly independent rows

the matrix A has rank r

- $\mathrm{range}(A) \subseteq \mathrm{range}(B)$: each column of A is a linear combination of columns of B
- $\mathrm{range}(B) \subseteq \mathrm{range}(A)$:

$$y = Bx \qquad \Longrightarrow \qquad y = B(CD)x = A(Dx)$$

  where D is a right inverse of C (for example, $D = C^\dagger$)
- therefore $\mathrm{range}(A) = \mathrm{range}(B)$ and $\mathrm{rank}(A) = \mathrm{rank}(B)$
- since the columns of B are linearly independent, $\mathrm{rank}(B) = r$

Exercises

Exercise 1

$\mathcal{V}$ and $\mathcal{W}$ are subspaces in $\mathbf{R}^m$; show that

$$\dim(\mathcal{V} + \mathcal{W}) + \dim(\mathcal{V} \cap \mathcal{W}) = \dim(\mathcal{V}) + \dim(\mathcal{W})$$

Exercise 2

- A and B are matrices with the same number of rows; find a matrix C with

$$\mathrm{range}(C) = \mathrm{range}(A) + \mathrm{range}(B)$$

- A and B are matrices with the same number of columns; find a matrix C with

$$\mathrm{null}(C) = \mathrm{null}(A) \cap \mathrm{null}(B)$$

Outline

- subspaces, dimension, rank
- **QR factorization with pivoting**
- properties of matrix rank
- low-rank matrices
- pseudo-inverse

QR factorization

A is an $m \times n$ matrix with linearly independent columns (hence, $m \geq n$)

QR factorization

$$A = QR$$

- R is $n \times n$, upper triangular, with positive diagonal elements
- Q is $m \times n$ with orthonormal columns ($Q^T Q = I$)
- several algorithms, including Gram–Schmidt algorithm

Full QR factorization (QR decomposition)

$$A = \begin{bmatrix} Q & \tilde{Q} \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}$$

- R is $n \times n$, upper triangular, with positive diagonal elements
- $\begin{bmatrix} Q & \tilde{Q} \end{bmatrix}$ is orthogonal: square with orthonormal columns
- several algorithms, including Householder triangularization

QR factorization with column pivoting

A is an $m \times n$ matrix (may be wide or have linearly dependent columns)

QR factorization with column pivoting (column reordering)

$$A = QRP$$

- Q is $m \times r$ with orthonormal columns
- R is $r \times n$, leading $r \times r$ submatrix is upper triangular with positive diagonal:

$$R = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1r} & R_{1,r+1} & \cdots & R_{1n} \\ 0 & R_{22} & \cdots & R_{2r} & R_{2,r+1} & \cdots & R_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & R_{rr} & R_{r,r+1} & \cdots & R_{rn} \end{bmatrix}$$

- can be chosen to satisfy $R_{11} \geq R_{22} \geq \cdots \geq R_{rr} > 0$
- P is an $n \times n$ permutation matrix
- r is the rank of A: apply the result on page 1.27 with $B = Q$, $C = RP$

Interpretation

- columns of $AP^T = QR$ are the columns of A in a different order
- the columns are divided in two groups:

$$AP^T = \left[\begin{array}{cc} \hat{A}_1 & \hat{A}_2 \end{array}\right] = Q \left[\begin{array}{cc} R_1 & R_2 \end{array}\right] \qquad \hat{A}_1 \text{ is } m \times r,\ R_1 \text{ is } r \times r$$

- $\hat{A}_1 = QR_1$ is $m \times r$ with linearly independent columns:

$$\hat{A}_1 x = QR_1 x = 0 \qquad \Longrightarrow \qquad R_1^{-1} Q^T \hat{A}_1 x = x = 0$$

- $\hat{A}_2 = QR_2$ is $m \times (n - r)$: columns are linear combinations of columns of $\hat{A}_1$

$$\hat{A}_2 = QR_2 = \hat{A}_1 R_1^{-1} R_2$$

the factorization provides two useful bases for $\text{range}(A)$

- columns of Q are an orthonormal basis
- columns of $\hat{A}_1$ are a basis selected from the columns of A

Computing the pivoted QR factorization

we first describe the *modified Gram–Schmidt algorithm*

- a variation of the standard Gram–Schmidt algorithm for QR factorization

$$A = QR$$

  of a matrix A with linearly independent columns

- has better numerical properties than the standard Gram–Schmidt algorithm

we then extend the modified GS method to the pivoted QR factorization

Modified Gram–Schmidt algorithm

computes QR factorization of an $m \times n$ matrix A with linearly independent columns

$$\begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1n} \\ 0 & R_{22} & \cdots & R_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{nn} \end{bmatrix}$$

- after k steps, the algorithm has computed a partial factorization

$$A = \begin{bmatrix} q_1 \cdots q_k & B_k \end{bmatrix} \begin{bmatrix} R_{11} & \cdots & R_{1k} & R_{1,k+1} & \cdots & R_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & R_{kk} & R_{k,k+1} & \cdots & R_{kn} \\ & 0 & & & I & \end{bmatrix}$$

  where B_k has size $m \times (n-k)$ with columns orthogonal to $q_1, \ldots, q_k$

- we start the algorithm with $B_0 = A$

Modified Gram–Schmidt update

careful inspection of the update at step k shows that

$$B_{k-1} = \left[\begin{array}{cc} q_k & B_k \end{array}\right] \left[\begin{array}{cc} R_{kk} & R_{k,(k+1):n} \\ 0 & I \end{array}\right]$$

partition B_{k-1} as $B_{k-1} = \begin{bmatrix} b & \hat{B} \end{bmatrix}$ with b the first column and $\hat{B}$ of size $m \times (n-k)$:

$$b = q_k R_{kk}, \qquad \hat{B} = q_k R_{k,(k+1):n} + B_k$$

- from the first equation, and the required properties $\|q_k\| = 1$ and $R_{kk} > 0$:

$$R_{kk} = \|b\|, \qquad q_k = \frac{1}{R_{kk}} b$$

- from the second equation, and the requirement that $q_k^T B_k = 0$:

$$R_{k,(k+1):n} = q_k^T \hat{B}, \qquad B_k = \hat{B} - q_k R_{k,(k+1):n}$$

Summary: modified Gram–Schmidt algorithm

Algorithm

define $B_0 = A$ and repeat for $k = 1$ to n:

- compute $R_{kk} = \|b\|$ and $q_k = (1/R_{kk})b$ where b is the first column of B_{k-1}
- compute

$$\begin{bmatrix} R_{k,k+1} & \cdots & R_{kn} \end{bmatrix} = q_k^T \hat{B}, \qquad B_k = \hat{B} - q_k \begin{bmatrix} R_{k,k+1} & \cdots & R_{kn} \end{bmatrix}$$

  where $\hat{B}$ is B_{k-1} with first column removed

MATLAB code

```
Q = A;  R = zeros(n,n);
for k = 1:n
    R(k,k) = norm(Q(:,k));
    Q(:,k) = Q(:,k) / R(k,k);
    R(k,k+1:n) = Q(:,k)' * Q(:,k+1:n);
    Q(:,k+1:n) = Q(:,k+1:n)  - Q(:,k) * R(k,k+1:n);
end;
```

Modified Gram–Schmidt algorithm with pivoting

with minor changes the modified GS algorithm computes the pivoted factorization

$$AP^T = \left[\begin{array}{cccc} q_1 & q_2 & \cdots & q_r \end{array}\right] \left[\begin{array}{ccccccc} R_{11} & R_{12} & \cdots & R_{1r} & R_{1,r+1} & \cdots & R_{1n} \\ 0 & R_{22} & \cdots & R_{1r} & R_{1,r+1} & \cdots & R_{1n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & R_{rr} & R_{r,r+1} & \cdots & R_{rn} \end{array}\right]$$

- partial factorization after k steps

$$AP_k^T = \left[\begin{array}{cccc} q_1 & \cdots & q_k & B_k \end{array}\right] \left[\begin{array}{cccccc} R_{11} & \cdots & R_{1k} & R_{1,k+1} & \cdots & R_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & R_{kk} & R_{k,k+1} & \cdots & R_{kn} \\ & 0 & & & I & \end{array}\right]$$

- if $B_k = 0$, the factorization is complete ($r = k$, $P = P_k$)
- before step k, we reorder columns of B_{k-1} to place the largest column first
- this requires reordering columns $k, \ldots, n$ of R, and modifying P_{k-1}

Example

$$A = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & -1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & -1 & -1 \end{bmatrix}$$

Step 1

- a_2 and a_4 have the largest norms; we move a_2 to the first position
- find first column of Q, first row of R

$$\begin{aligned} \begin{bmatrix} a_2 & a_1 & a_3 & a_4 \end{bmatrix} &= \left[\begin{array}{c|ccc} 1/2 & 1/2 & 0 & 1 \\ 1/2 & -1/2 & 1 & -1 \\ 1/2 & 1/2 & 0 & 1 \\ 1/2 & -1/2 & -1 & -1 \end{array}\right] \left[\begin{array}{c|ccc} 2 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] \\ &= \begin{bmatrix} q_1 & B_1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{1,2:4} \\ 0 & I \end{bmatrix} \end{aligned}$$

Example

Step 2

- move column 3 of B_1 to first position in B_1

$$
\left[\begin{array}{cccc} a_2 & a_4 & a_1 & a_3 \end{array}\right] = \left[\begin{array}{c|ccc} 1/2 & 1 & 1/2 & 0 \\ 1/2 & -1 & -1/2 & 1 \\ 1/2 & 1 & 1/2 & 0 \\ 1/2 & -1 & -1/2 & -1 \end{array}\right] \left[\begin{array}{c|ccc} 2 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right]
$$

- find second column of Q, second row or R

$$
\begin{aligned}
\left[\begin{array}{cccc} a_2 & a_4 & a_1 & a_3 \end{array}\right] &= \left[\begin{array}{cc|cc} 1/2 & 1/2 & 0 & 0 \\ 1/2 & -1/2 & 0 & 1 \\ 1/2 & 1/2 & 0 & 0 \\ 1/2 & -1/2 & 0 & -1 \end{array}\right] \left[\begin{array}{cc|cc} 2 & 0 & 1 & 0 \\ 0 & 2 & 1 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] \\
&= \left[\begin{array}{cc|c} q_1 & q_2 & B_2 \end{array}\right] \left[\begin{array}{cc|c} R_{11} & R_{12} & R_{1,3:4} \\ 0 & R_{22} & R_{2,3:4} \\ \hline 0 & 0 & I \end{array}\right]
\end{aligned}
$$

Example

Step 3

- move column 2 of B_2 to first position in B_2

$$\begin{bmatrix} a_2 & a_4 & a_3 & a_1 \end{bmatrix} = \left[\begin{array}{rr|rr} 1/2 & 1/2 & 0 & 0 \\ 1/2 & -1/2 & 1 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1/2 & -1/2 & -1 & 0 \end{array}\right] \left[\begin{array}{cc|cc} 2 & 0 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

- find third column of Q, third row of R

$$\begin{aligned} \begin{bmatrix} a_2 & a_4 & a_3 & a_1 \end{bmatrix} &= \left[\begin{array}{rrr|r} 1/2 & 1/2 & 0 & 0 \\ 1/2 & -1/2 & 1/\sqrt{2} & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1/2 & -1/2 & -1/\sqrt{2} & 0 \end{array}\right] \left[\begin{array}{ccc|c} 2 & 0 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & \sqrt{2} & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \\ &= \left[\begin{array}{ccc|c} q_1 & q_2 & q_3 & B_3 \end{array}\right] \left[\begin{array}{ccc|c} R_{11} & R_{12} & R_{13} & R_{14} \\ 0 & R_{22} & R_{23} & R_{24} \\ 0 & 0 & R_{33} & R_{34} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \end{aligned}$$

Example

Result: since B_3 is zero, the algorithm terminates with the factorization

$$\begin{bmatrix} a_2 & a_4 & a_3 & a_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & -1 & -1 & 0 \end{bmatrix}$$

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

Exercise

the modified Gram–Schmidt update on page 1.20 is

$$R_{kk} = \|b\|, \qquad q_k = \frac{1}{R_{kk}} b,$$

$$\begin{bmatrix} R_{k,k+1} & \cdots & R_{kn} \end{bmatrix} = q_k^T \hat{B}, \qquad B_k = \hat{B} - q_k \begin{bmatrix} R_{k,k+1} & \cdots & R_{kn} \end{bmatrix}$$

where $\begin{bmatrix} b & \hat{B} \end{bmatrix} = B_{k-1}$

1. denote column i of $\hat{B}$ by $\hat{b}_i$ and column i of B_k by b_i; show that

$$\|b_i\|^2 = \|\hat{b}_i\|^2 - R_{k,k+i}^2$$

2. in the pivoting algorithm, $\|b\| \geq \|\hat{b}_i\|$ for $i = 1, \ldots, n-k$; show that therefore

$$R_{kk} \geq R_{k+1,k+1}$$

Outline

- subspaces, dimension, rank
- QR factorization with pivoting
- **properties of matrix rank**
- low-rank matrices
- pseudo-inverse

Factorization theorem

an $m \times n$ matrix A has rank r if and only if it can be factored as

$$A = BC$$

- B is $m \times r$ with linearly independent columns
- C is $r \times n$ with linearly independent rows

this is called a *full-rank factorization* of A

- "if" statement was shown on page 1.12
- the pivoted QR factorization proves the "only if" statement
- other algorithms will be discussed later

Rank of transpose

an immediate and important consequence of the factorization theorem:

$$\mathrm{rank}(A^T) = \mathrm{rank}(A)$$

the column space (range) of a matrix has the same dimension as its row space:

$$\dim(\mathrm{range}(A^T)) = \dim(\mathrm{range}(A))$$

Full-rank matrices

for any $m \times n$ matrix

$$\operatorname{rank}(A) \leq \min\{m, n\}$$

Full rank: A has *full rank* if $\operatorname{rank}(A) = \min\{m, n\}$

- $\operatorname{rank}(A) = n < m$: tall and left-invertible (linearly independent columns)
- $\operatorname{rank}(A) = m < n$: wide and right-invertible (linearly independent rows)
- $\operatorname{rank}(A) = m = n$: square and invertible (nonsingular)

Full column rank: A has *full column rank* if $\operatorname{rank}(A) = n$

- A has linearly independent columns (is left-invertible)
- must be tall or square

Full row rank: A has *full row rank* if $\operatorname{rank}(A) = m$

- A has linearly independent rows (is right-invertible)
- must be wide or square

Dimension of nullspace

if A is $m \times n$ then

$$\dim(\text{null}(A)) = n - \text{rank}(A)$$

we show this by constructing a basis containing $n - \text{rank}(A)$ vectors

Basis for nullspace: a basis for the nullspace of A is given by the columns of

$$P^T \begin{bmatrix} -R_1^{-1}R_2 \\ I \end{bmatrix}$$

where P, R_1, R_2 are the matrices in the pivoted QR factorization

$$AP^T = Q \begin{bmatrix} R_1 & R_2 \end{bmatrix}$$

- P is a $n \times n$ permutation matrix
- Q is $m \times r$ with orthonormal columns, where $r = \text{rank}(A)$
- R_1 is $r \times r$ upper triangular and nonsingular, R_2 is $r \times (n - r)$

(proof on next page)

Proof:

$$
\begin{aligned}
AP^T x = 0 \quad &\Longleftrightarrow \quad Q \begin{bmatrix} R_1 & R_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0 \\
&\Longleftrightarrow \quad \begin{bmatrix} R_1 & R_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0 \\
&\Longleftrightarrow \quad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -R_1^{-1} R_2 \\ I \end{bmatrix} x_2
\end{aligned}
$$

therefore, x is in the nullspace of AP^T if and only if it is in the range of

$$
\begin{bmatrix} -R_1^{-1} R_2 \\ I \end{bmatrix}
$$

the columns of this matrix are linearly independent, so they are a basis for

$$
\operatorname{range}\left(\begin{bmatrix} -R_1^{-1} R_2 \\ I \end{bmatrix}\right) = \operatorname{null}(AP^T)
$$

Outline

- subspaces, dimension, rank
- QR factorization with pivoting
- properties of matrix rank
- **low-rank matrices**
- pseudo-inverse

Low-rank matrix

an $m \times n$ matrix has *low rank* if

$$\operatorname{rank}(A) \ll \min\{m, n\}$$

if $r = \operatorname{rank}(A) \ll \min\{m, n\}$, a factorization

$$A = BC \qquad \text{(with } B \in \mathbf{R}^{m \times r} \text{ and } C \in \mathbf{R}^{r \times n}\text{)}$$

gives an efficient representation of A

- memory: B and C have $r(m+n)$ entries, compared with mn for A
- fast matrix–vector product: $2r(m+n)$ flops if we compute $y = Ax$ as

  $$y = B(Cx)$$

  compare with $2mn$ for general product $y = Ax$

Low-rank approximation

(approximate) low-rank representations

$$A \approx BC$$

are useful in many applications

Singular value decomposition (SVD)

finds the best approximation (in Frobenius norm or 2-norm) of a given rank

Heuristic algorithms

- less expensive than SVD but not guaranteed to find a good approximation
- *e.g.*, in the pivoted QR factorization, terminate at step k when R_{kk} is small

Optimization algorithms

add additional constraints on B, C (for example, entries must be nonnegative)

Example: document analysis

a collection of documents is modeled by a *term–document* matrix A

- columns correspond to documents, rows to words in a dictionary
- column vectors represent word histograms

in a low-rank approximation $A \approx BC$,

$$A_{ij} \approx \sum_{k=1}^{r} B_{ik} C_{kj}$$

- r dimensions represent “concepts”
- C_{kj} indicates strength of association between document j and concept k
- A_{ik} indicates strength of association between term i and concept k

a well-known SVD-based method of this type is called *Latent Semantic Indexing*

Example: multiview geometry

- n objects at positions $x_j \in \mathbf{R}^3$, $j = 1, \ldots, n$, are viewed by l cameras
- $y_{ij} \in \mathbf{R}^2$ is the location of object j in the image acquired by camera i
- each camera is modeled as an affine mapping:

$$y_{ij} = P_i x_j + q_i, \quad i = 1, \ldots, l, \quad j = 1, \ldots, n$$

define a $2l \times n$ matrix with the observations y_{ij}:

$$A = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & & \vdots \\ y_{l1} & y_{l2} & \cdots & y_{ln} \end{bmatrix} = \begin{bmatrix} P_1 & q_1 \\ P_2 & q_2 \\ \vdots & \\ P_l & q_l \end{bmatrix} \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

- a matrix of rank 4 (or less)
- equality assumes noise-free observations and perfectly affine cameras
- a rank-4 approximation of A gives estimates of positions and camera models

Outline

- subspaces, dimension, rank
- QR factorization with pivoting
- properties of matrix rank
- low-rank matrices
- **pseudo-inverse**

Pseudo-inverse

suppose A is $m \times n$ with rank r and rank factorization

$$A = BC$$

- B is $m \times r$ with linearly independent columns; its pseudo-inverse is defined as

$$B^\dagger = (B^T B)^{-1} B^T$$

- C is $r \times n$ with linearly independent rows; its pseudo-inverse is defined as

$$C^\dagger = C^T (C C^T)^{-1}$$

we define the **pseudo-inverse** of A as

$$A^\dagger = C^\dagger B^\dagger$$

- this extends the definition of pseudo-inverse to matrices that are not full rank
- $A^\dagger$ is also known as the *Moore–Penrose (generalized) inverse*

Uniqueness

$A^\dagger = C^\dagger B^\dagger$ does not depend on the particular rank factorization $A = BC$ used

- suppose $A = \tilde{B}\tilde{C}$ is another rank factorization
- the columns of B and $\tilde{B}$ are two bases for $\mathrm{range}(A)$; therefore

$$\tilde{B} = BM \quad \text{for some nonsingular } r \times r \text{ matrix } M$$

- hence $BC = \tilde{B}\tilde{C} = BM\tilde{C}$; multiplying with $B^\dagger$ on the left shows that $C = M\tilde{C}$
- the pseudo-inverses of $\tilde{B} = BM$ and $\tilde{C} = M^{-1}C$ are

$$\tilde{B}^\dagger = (\tilde{B}^T\tilde{B})^{-1}\tilde{B}^T = M^{-1}(B^TB)^{-1}B^T = M^{-1}B^\dagger$$

  and

$$\tilde{C}^\dagger = \tilde{C}^T(\tilde{C}\tilde{C}^T)^{-1} = C^T(CC^T)^{-1}M = C^\dagger M$$

- we conclude that $\tilde{C}^\dagger\tilde{B}^\dagger = C^\dagger M M^{-1} B^\dagger = C^\dagger B^\dagger$

Example: pseudo-inverse of diagonal matrix

- the rank of a diagonal matrix A is the number of nonzero diagonal elements
- pseudo-inverse $A^\dagger$ is the diagonal matrix with

$$(A^\dagger)_{ii} = \begin{cases} 1/A_{ii} & \text{if } A_{ii} \neq 0 \\ 0 & \text{if } A_{ii} = 0 \end{cases}$$

Example

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3 \end{bmatrix}, \qquad A^\dagger = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1/3 \end{bmatrix}$$

this follows, for example, from the factorization $A = BC$ with

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad C = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & -3 \end{bmatrix}$$

Rank-deficient least squares

least squares problem with $m \times n$ matrix A and $\text{rank}(A) = r$ (possibly $r < n$)

$$\text{minimize} \quad \|Ax - b\|^2 \tag{1}$$

- substitute rank factorization $A = BC$:

$$\text{minimize} \quad \|BCx - b\|^2$$

- $\hat{y} = B^\dagger b = (B^T B)^{-1} B^T b$ is the solution of the full-rank least squares problem

$$\text{minimize} \quad \|By - b\|^2$$

- every x that satisfies $Cx = \hat{y}$ is a solution of the least squares problem (1)
- $\hat{x} = C^\dagger \hat{y} = C^T (CC^T)^{-1} \hat{y}$ is the least norm solution of the equation $Cx = \hat{y}$

therefore the solution of (1) with the smallest norm is

$$\hat{x} = A^\dagger b = C^\dagger B^\dagger b$$

other solutions of (1) are the vectors $\hat{x} + v$, for nonzero $v \in \text{null}(A)$

Meaning of $AA^\dagger$ and $A^\dagger A$

if A does not have full rank, $A^\dagger$ is not a left or a right inverse of A

Interpretation of $AA^\dagger$

$$AA^\dagger = BCC^\dagger B^\dagger = BB^\dagger = B(B^TB)^{-1}B^T$$

- $BB^\dagger x$ is the orthogonal projection of x on $\mathrm{range}(B)$ (see 133A, slide 6.12)
- hence, $AA^\dagger x$ is the orthogonal projection of x on $\mathrm{range}(A) = \mathrm{range}(B)$

Interpretation of $A^\dagger A$

$$A^\dagger A = C^\dagger B^\dagger BC = C^\dagger C = C^T(CC^T)^{-1}C$$

- $C^\dagger Cx$ is the orthogonal projection of x on $\mathrm{range}(C^T)$
- hence, $A^\dagger Ax$ is orthogonal projection on row space $\mathrm{range}(A^T) = \mathrm{range}(C^T)$

Exercise

show that $A^{\dagger}$ satisfies the following properties

- $AA^{\dagger}A = A$
- $A^{\dagger}AA^{\dagger} = A^{\dagger}$
- $AA^{\dagger}$ is a symmetric matrix
- $A^{\dagger}A$ is a symmetric matrix