

Strang

Malcolm

Started 5th June 2025

Contents

1	Vectors and Matrices	3
1.1	Intuition for Dot product, Cosine formula, Schwarz and Triangle inequalities	3
1.2	Intuition for column rank being equal to row rank	5
1.3	Ways to multiply $\mathbf{AB} = \mathbf{C}$	8
2	Solving linear equations $\mathbf{Ax} = \mathbf{b}$	9
2.1	Solutions to $\mathbf{Ax} = \mathbf{b}$	10
2.2	Elimination and Back Substitution	11
2.3	Elimination matrices and inverse matrices	14
2.4	Gauss-Jordan elimination	16
2.5	Proving $\mathbf{A} = \mathbf{LU}$	17
2.6	Permutation matrices	19
2.7	Transposes and symmetric matrices	21
2.7.1	The transpose of \mathbf{AB} is $\mathbf{B}^T \mathbf{A}^T$: Intuition	21
2.7.2	Showing $(\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T$	21
2.7.3	$\mathbf{A}^T \mathbf{A}$ is symmetric	21
3	The Four Fundamental Subspaces	22
3.1	Reduced Row Echelon Form	22
3.2	$\mathbf{A} = \mathbf{CR}$ factorisation	24
3.3	Systematic nullspace computation	25
3.4	The complete solution to $\mathbf{Ax} = \mathbf{b}$	27
3.4.1	Rank and solution	28
3.5	Intuition for four subspaces	30
3.6	$\mathbf{A} = \mathbf{CR}$ and Block elimination	31
4	Orthogonality	33
4.1	Orthogonal subspaces	33
4.1.1	Orthogonal complements	34
4.1.2	Intuition for $\mathbf{x} = \mathbf{x}_{\text{row}} + \mathbf{x}_{\text{null}}$	34
4.1.3	\mathbf{A} is invertible from the row space to the column space	35
4.2	$\mathbf{A}^T \mathbf{A}$ is invertible only if \mathbf{A} has linearly independent columns	36

4.3	Projections	37
4.3.1	Projection onto a line	37
4.3.2	Projection onto a subspace	38
4.4	Least Squares Approximations	40
4.4.1	Fitting a straight line	43
4.5	Orthonormal Bases	44
4.5.1	Multiplication by any orthogonal matrix preserves length and angle	45
4.5.2	Projections using orthonormal bases	45
4.6	Gram-Schmidt Process	46
4.7	$\mathbf{A} = \mathbf{QR}$ factorisation	47
4.8	Pseudoinverses	50
5	Determinants	51
5.1	Properties of determinants	51
A	Misc. topics	52
A.1	Taylor series, Difference approximations	52

Chapter 1

Vectors and Matrices

1.1 Intuition for Dot product, Cosine formula, Schwarz and Triangle inequalities

Intuition for dot product

The unit vectors $\mathbf{v} = (\cos \alpha, \sin \alpha)$ and $\mathbf{w} = (\cos \beta, \sin \beta)$ are plotted as follows



Figure 1.5: Unit vectors: $\mathbf{u} \cdot \mathbf{i} = \cos \theta$. The angle between the vectors is θ .

See first that when fixed in this form, the magnitude of both vectors is 1, with an angle $\beta - \alpha$ between them. These unit vectors have dot product

$$\mathbf{v} \cdot \mathbf{w} = \cos \alpha \cos \beta + \sin \alpha \sin \beta = \cos(\beta - \alpha)$$

We have θ as the angle between the two vectors; see that the sign of $\mathbf{v} \cdot \mathbf{w}$ tells us whether θ is below or above a right angle (due to the cosine function being negative for its argument $> \pi/2$ and positive for $< \pi/2$):



Figure 1.6: Small angle $\mathbf{v} \cdot \mathbf{w}_1 > 0$. Right angle $\mathbf{v} \cdot \mathbf{w}_2 = 0$. Large angle $\mathbf{v} \cdot \mathbf{w}_3 < 0$.

(next page)

Cont.

The idea here is that the dot product reveals the exact angle θ ; for unit vectors \mathbf{u} and \mathbf{U} , the dot product $\mathbf{u} \cdot \mathbf{U}$ is the cosine of θ . This remains true in n dimensions (not shown).

See that any \mathbf{u} and \mathbf{v} can be fixed in the above form by normalising their lengths to get $\mathbf{u} = \mathbf{v}/\|\mathbf{v}\|$ and $\mathbf{U} = \mathbf{w}/\|\mathbf{w}\|$. After which their dot product would give $\cos \theta$. This leads us to the *cosine formula*:

$$\text{Cosine formula: } \frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{v}\| \|\mathbf{w}\|} = \cos \theta \quad \text{if } \mathbf{v} \text{ and } \mathbf{w} \text{ are nonzero vectors}$$

Perpendicular vectors

See that when the angle between \mathbf{v} and \mathbf{w} is 90° , its cosine is 0; this gives us a way to test this. Also see that for perpendicular vectors:

$$\|\mathbf{v} + \mathbf{w}\|^2 = \|\mathbf{v}\|^2 + \|\mathbf{w}\|^2$$

because

$$\|\mathbf{v} + \mathbf{w}\|^2 = (\mathbf{v} + \mathbf{w}) \cdot (\mathbf{v} + \mathbf{w}) = \mathbf{v} \cdot \mathbf{v} + \mathbf{v} \cdot \mathbf{w} + \mathbf{w} \cdot \mathbf{v} + \mathbf{w} \cdot \mathbf{w}$$

where $\mathbf{v} \cdot \mathbf{w} = 0$.

Schwarz and Triangle inequalities

First, see from the cosine formula that the dot product of $\mathbf{v}/\|\mathbf{v}\|$ and $\mathbf{w}/\|\mathbf{w}\|$ never exceeds one (since $\cos \theta$ never exceeds one). This is the *Schwarz inequality*:

$$\text{Schwarz inequality: } |\mathbf{v} \cdot \mathbf{w}| \leq \|\mathbf{v}\| \|\mathbf{w}\|$$

The *Triangle inequality* comes directly from the Schwarz inequality:

$$\text{Triangle inequality: } \|\mathbf{v} + \mathbf{w}\| \leq \|\mathbf{v}\| + \|\mathbf{w}\|$$

This can be seen from

$$\|\mathbf{v} + \mathbf{w}\|^2 = \mathbf{v} \cdot \mathbf{v} + \mathbf{v} \cdot \mathbf{w} + \mathbf{w} \cdot \mathbf{v} + \mathbf{w} \cdot \mathbf{w} \leq \|\mathbf{v}\|^2 + 2\|\mathbf{v}\| \|\mathbf{w}\| + \|\mathbf{w}\|^2$$

The square root gives us the triangle equality (side 3 cannot exceed side 1 + side 2).

1.2 Intuition for column rank being equal to row rank

If all columns are in the same direction, why does it happen that all the rows are the same direction?

Consider the matrix, see that column 2 is m times column 1:

$$\mathbf{A} = \begin{bmatrix} a & ma \\ b & mb \end{bmatrix}$$

See that the second row is just b/a times the first row—if the column rank is 1, then the row rank is 1. See that transposing the matrix, we have

$$\mathbf{A} = \begin{bmatrix} a(1) & b(1) \\ a(m) & b(m) \end{bmatrix}$$

which still has one independent column. Now consider the 3x3 case:

$$\mathbf{A} = \begin{bmatrix} a & ma & pa \\ b & mb & pb \\ c & mc & pc \end{bmatrix}$$

See that a similar deduction can also be made in this case, where the row rank of A is equal to its column rank.

(next page)

An informal proof

Consider any matrix \mathbf{A} , suppose we go from left to right, looking for independent columns of \mathbf{A} using the following procedure:

1. If column 1 of \mathbf{A} is not zero, put it in matrix \mathbf{C}
2. If column 2 of \mathbf{A} is not a multiple of column 1, put it in into \mathbf{C}
3. If column 3 of \mathbf{A} is not a combination of columns 1 and 2, put it into \mathbf{C} .
continue

See that at the end \mathbf{C} will have r columns taken from \mathbf{A} , where r is the rank of \mathbf{A} and \mathbf{C} . While the n columns of \mathbf{A} are dependent, the r columns of \mathbf{C} will surely be independent.

For instance consider \mathbf{A} with rank 2

$$\mathbf{A} = \begin{bmatrix} 2 & 6 & 4 \\ 4 & 12 & 8 \\ 1 & 3 & 5 \end{bmatrix} \quad \text{leads to} \quad \mathbf{C} = \begin{bmatrix} 2 & 4 \\ 4 & 8 \\ 1 & 5 \end{bmatrix}$$

Now consider another matrix \mathbf{R} to be multiplied by \mathbf{C} such that $\mathbf{A} = \mathbf{C}\mathbf{R}$. The first and third columns of \mathbf{A} are already in \mathbf{C} , so those respective columns in \mathbf{R} make up a *identity matrix*; the second column of \mathbf{A} is a multiple of the first, so we have

$$\mathbf{A} = \mathbf{C}\mathbf{R} \quad \text{is} \quad \begin{bmatrix} 2 & 6 & 4 \\ 4 & 12 & 8 \\ 1 & 3 & 5 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 4 & 8 \\ 1 & 5 \end{bmatrix} \begin{bmatrix} 1 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(See that the i th row of \mathbf{A} can be seen as a linear combination of the rows of \mathbf{R} specified the i th row of \mathbf{C} . (or just consider $\mathbf{A}^T = \mathbf{R}^T \mathbf{C}^T$). We know that

1. \mathbf{C} contains the full set of r independent columns of \mathbf{A} .
2. $\mathbf{R} = [\mathbf{I} \mathbf{F}]$ contains the identity matrix \mathbf{I} in the same r columns that held \mathbf{C} .
3. The dependent columns of \mathbf{A} are combinations of $\mathbf{C}\mathbf{F}$ of the independent columns in \mathbf{C} .

Where the matrix \mathbf{F} goes into the other $n - r$ columns of $\mathbf{R} = [\mathbf{I} \mathbf{F}]$. ($\mathbf{A} = \mathbf{C}\mathbf{R}$ becomes $\mathbf{A} = \mathbf{C}[\mathbf{I}, \mathbf{F}] = [\mathbf{C}, \mathbf{C}\mathbf{F}] = [\text{indep cols of } \mathbf{A}, \text{ dep cols of } \mathbf{A}]$ (in correct order).

See that \mathbf{C} has the same column space as \mathbf{A} , and \mathbf{R} has the same row space as \mathbf{A} (every row of \mathbf{A} is a combination of the rows of \mathbf{R}).

(next page)

Cont.

We had the example

$$\mathbf{A} = \mathbf{C}\mathbf{R} \quad \text{is} \quad \begin{bmatrix} 2 & 6 & 4 \\ 4 & 12 & 8 \\ 1 & 3 & 5 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 4 & 8 \\ 1 & 5 \end{bmatrix} \begin{bmatrix} 1 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Here is an informal proof that the row rank of \mathbf{A} equals the column rank of \mathbf{A} (based from facts we already know)

1. The r columns of \mathbf{C} are independent (chosen that way from \mathbf{A})
2. Every column of \mathbf{A} is a combination of those r columns of \mathbf{C} (since $\mathbf{A} = \mathbf{C}\mathbf{R}$)
3. The r rows of \mathbf{R} are independent (they contain the r by r matrix \mathbf{I})
4. Every row of \mathbf{A} is a combination of the r rows of \mathbf{R}

See that for every column of \mathbf{A} that goes into \mathbf{C} , a column of \mathbf{I} goes into \mathbf{R} , where each column of \mathbf{I} in \mathbf{R} adds an independent row.

This means that the column rank of \mathbf{C} (column space of \mathbf{A}) is always equal to the row rank of \mathbf{R} (row space of \mathbf{A})—the column rank of \mathbf{A} is equal to the row rank of \mathbf{A} .

More examples

Rank 2:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 4 & 5 \\ 7 & 8 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \end{bmatrix}$$

Rank 2:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 4 & 5 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Rank 1:

$$\begin{bmatrix} 1 & 2 & 10 & 100 \\ 3 & 6 & 30 & 300 \\ 2 & 4 & 20 & 200 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} \begin{bmatrix} 1 & 2 & 10 & 100 \end{bmatrix}$$

1.3 Ways to multiply $AB = C$

Multiplication by columns of A and rows of B

A lesser known way to multiply AB is through considering the columns of A and the rows of B (contrary to the usual ideas where each entry of the result is a dot product of a row of A and column of B):

$$AB = \left[\begin{array}{c|c|c} | & & | \\ \hline a_1 & \cdots & a_n \\ \hline | & & | \end{array} \right] \left[\begin{array}{c} -b_1^* - \\ \vdots \\ -b_n^* - \end{array} \right] = a_1 b_1^* + a_2 b_2^* + \cdots + a_n b_n^*.$$

columns a_k rows b_k^* Add columns a_k times rows b_k^*

We multiply each column of A by each row of B ; this gives us n rank 1 matrices, which we then sum together; these matrices are called *outer products*

(usually we see the i th column of the result as a linear combination of the columns of A specified by the i th column of B . However in this case the i th outer product is a matrix of the same size as the result, that contains all the contributions of the i th column of A to the final product. By summing this over all n columns of A we get the result.)

Summary of methods

(3 by 2)(2 by 4) = (3 by 4)		Four Ways to Multiply $AB = C$	
		(Row i of A) \cdot (Column k of B) = Number C_{ik} $i = 1$ to 3 $k = 1$ to 4 12 numbers	
		A times (Column k of B) = Column k of C $k = 1$ to 4 4 columns	
		(Row i of A) times B = Row i of C $i = 1$ to 3 3 rows	
		(Column j of A) (Row j of B) = Rank 1 Matrix $j = 1$ to 2 2 matrices	
Dot product way, Column way, Row way, Columns times rows			

(A nice way to intuit the the third method is to consider $(AB)^T = B^T A^T$; where the columns of $(AB)^T$ are the rows of AB)

Chapter 2

Solving linear equations

$$Ax = b$$

2.1 Solutions to $A\mathbf{x} = \mathbf{b}$

Given a $n \times n$ matrix A and an $n \times 1$ column vector \mathbf{b} , there are three outcomes for the vector \mathbf{x} that solves $A\mathbf{x} = \mathbf{b}$.

First there may be *no vector* \mathbf{x} that solves $A\mathbf{x} = \mathbf{b}$, or there may be exactly *one* solution, or there may be *infinitely many* solution vectors \mathbf{x} . Here are the possibilities:

1. **Exactly one solution** to $A\mathbf{x} = \mathbf{b}$ means that A has independent columns (only one particular linear combination of the columns of A leads to \mathbf{b} . That combination is specified by \mathbf{x}). A is full rank and the only solution to $A\mathbf{x} = \mathbf{0}$ is $\mathbf{x} = \mathbf{0}$. A has an inverse matrix A^{-1} (given \mathbf{b} , we can work backward to get \mathbf{x} since only one \mathbf{x} leads to \mathbf{b}).

2. **No solution** to $A\mathbf{x} = \mathbf{b}$ means that \mathbf{b} is not in the column space of A , so A is not full rank.

3. **Infinitely many solutions.** See that when the columns of A are not independent (not full rank), then there are infinitely many ways to produce the zero vector $\mathbf{b} = \mathbf{0}$ (this is the meaning of dependent columns), and so there are infinitely many solutions to $A\mathbf{x} = \mathbf{0}$.

Also see that if A is not full rank it means that its column space is some subspace, where solutions only exist for \mathbf{b} within that subspace.

As such, if there so happens to be a solution to $A\mathbf{x} = \mathbf{b}$ then we can add any solution to $A\mathbf{x} = \mathbf{0}$:

$$A(\mathbf{x} + \alpha\mathbf{x}) = A\mathbf{x} + \alpha A\mathbf{x} = \mathbf{b} + \mathbf{0} = \mathbf{b}$$

For some constant α , which gives us \mathbf{b} again—we have infinitely many solutions.

2.2 Elimination and Back Substitution

Elimination

We want to produce an *upper triangular* matrix \mathbf{U} from a square matrix \mathbf{A} . This is done through elimination; the procedure (for a 3x3 matrix) is as follows (assuming no row exchanges):

1. Use the first equation(row) to produce zeros in column 1 below the first pivot.
2. Use the new second equation(row) to clear out column 2 below pivot 2 in row 2.
3. *Continue to column 3.* The expected result is an upper triangular matrix \mathbf{U} .

These steps can be carried out using elimination matrices \mathbf{E} .

Consider \mathbf{A} and \mathbf{b}

$$\mathbf{A} = \begin{bmatrix} 2 & 3 & 4 \\ 4 & 11 & 14 \\ 2 & 8 & 17 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 19 \\ 55 \\ 50 \end{bmatrix}$$

\mathbf{E}_{21} multiplies equation 1 by 2 and subtracts that from equation 2 to get a zero in the first column below the first pivot:

$$\mathbf{E}_{21} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \implies \mathbf{E}_{21}\mathbf{A} = \begin{bmatrix} 2 & 3 & 4 \\ 0 & 5 & 6 \\ 2 & 8 & 17 \end{bmatrix}, \quad \mathbf{E}_{21}\mathbf{b} = \begin{bmatrix} 19 \\ 17 \\ 50 \end{bmatrix}$$

(For intuition on the elimination matrices, consider the row perspective of matrix multiplication) This produced the desired zero in column 1. It changed equation 2. To make the first pivot column zero, we subtract row 1 from row 3 using \mathbf{E}_{31}

$$\mathbf{E}_{31} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \implies \mathbf{E}_{31}\mathbf{E}_{21}\mathbf{A} = \begin{bmatrix} 2 & 3 & 4 \\ 0 & 5 & 6 \\ 0 & 5 & 13 \end{bmatrix}, \quad \mathbf{E}_{31}\mathbf{E}_{21}\mathbf{b} = \begin{bmatrix} 19 \\ 17 \\ 31 \end{bmatrix}$$

This completes elimination in column 1. Moving on to column 2 row 2 (the second pivot row). We use \mathbf{E}_{32} to subtract equation 2 from equation 3

$$\mathbf{E}_{32} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \implies \mathbf{U} = \begin{bmatrix} 2 & 3 & 4 \\ 0 & 5 & 6 \\ 0 & 0 & 7 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 19 \\ 17 \\ 14 \end{bmatrix}$$

$\mathbf{E}_{32}\mathbf{E}_{31}\mathbf{E}_{21}\mathbf{A} = \mathbf{U}$ is triangular. See that the same steps were applied to the right hand side \mathbf{b} to produce a new right hand side \mathbf{c} .
(next page)

Possible breakdown of elimination

Elimination might fail. This occurs when zero appears in a pivot position—subtracting that zero from lower rows will not clear out the column below that pivot. For example:

$$\mathbf{A} = \begin{bmatrix} 2 & 3 & 4 \\ 4 & 6 & 14 \\ 2 & 8 & 17 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 3 & 4 \\ 0 & 0 & 6 \\ 0 & 5 & 13 \end{bmatrix} = \mathbf{B}$$

A possible way to get around this would be to *exchange* row 2 (with the zero pivot) for row (with the nonzero in that column), then carry out elimination as per usual. This exchange is carried out using the permutation matrix \mathbf{P} :

$$\mathbf{PB} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 3 & 4 \\ 0 & 0 & 6 \\ 0 & 5 & 13 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 4 \\ 0 & 5 & 13 \\ 0 & 0 & 6 \end{bmatrix}$$

In this example the exchange produced \mathbf{U} with nonzero pivots; normally there may be more columns eliminate before \mathbf{U} is reached.

At times this exchange strategy may not work. This occurs when there is no pivot is available. Consider \mathbf{A}' :

$$\mathbf{A}' = \begin{bmatrix} 2 & 3 & 4 \\ 4 & 6 & 14 \\ 2 & 3 & 17 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 3 & 4 \\ 0 & 0 & 6 \\ 0 & 0 & 13 \end{bmatrix} = \mathbf{U}'$$

There is no second pivot in this case. This tells us that the matrix \mathbf{A}' *did not have full rank* (intuitively see that the non-pivot row can be expressed as a linear combination of the other pivot rows, so the row space (which is equal to the row space of the original \mathbf{A}' since the elimination steps are just linear combinations of the existing rows) is not full rank, and so the column space of the original \mathbf{A}' is also not full rank.

In this case there will be nonzero solutions \mathbf{X} to $\mathbf{A}'\mathbf{X} = 0$. The columns of \mathbf{U}' (and \mathbf{A}') are not independent.

Augmented matrix

During elimination, in order to make sure that the operations on the matrix \mathbf{A} are also executed on \mathbf{b} , one can *include \mathbf{b} as an extra column of \mathbf{A}* ; this combination $[\mathbf{A}, \mathbf{b}]$ is called an *augmented matrix*:

$$[\mathbf{A} \quad \mathbf{b}] = \begin{bmatrix} 2 & 3 & 4 & 19 \\ 4 & 11 & 14 & 55 \\ 2 & 8 & 17 & 50 \end{bmatrix} \xrightarrow{E} \begin{bmatrix} 2 & 3 & 4 & 19 \\ 0 & 5 & 6 & 17 \\ 0 & 0 & 7 & 14 \end{bmatrix} = [\mathbf{U} \quad \mathbf{c}]$$

(next page)

Back Substitution to solve $U\mathbf{x} = \mathbf{c}$

Elimination (ideally) produces an upper triangular matrix U that has all zeros below the diagonal, with nonzero pivots. For instance we had, for some \mathbf{x}

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{A} = \begin{bmatrix} 2 & 3 & 4 \\ 4 & 11 & 14 \\ 2 & 8 & 17 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 19 \\ 55 \\ 50 \end{bmatrix}$$

undergo elimination to to become

$$\mathbf{U}\mathbf{x} = \mathbf{c}, \quad \mathbf{U} = \begin{bmatrix} 2 & 3 & 4 \\ 0 & 5 & 6 \\ 0 & 0 & 7 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 19 \\ 17 \\ 14 \end{bmatrix}$$

See that this form allows us to easily solve the equations by going from bottom to top in a procedural manner, finding x_3 , then x_2 , then x_1 :

1. *Back substitution*: The last equation $7x_3 = 14$ gives $x_3 = 2$
2. *Work upwards*: The next equation $5x_2 + 6(2) = 17$ gives $x_2 = 1$
3. *repeat*: The first equation $2x_1 + 3(1) + 4(2) = 19$ gives $x_1 = 4$

giving us the only solution to this example $\mathbf{x} = (4, 1, 2)$. Remember the pivots need to be nonzero(full rank) for a single specific solution to be found.

2.3 Elimination matrices and inverse matrices

Elimination matrices

The basic elimination step *subtracts* a multiple ℓ_{ij} of equation j from equation i . We always speak about *subtractions* as elimination proceeds. For instance even if the first pivot $a_{11} = 3$ and below it is $a_{21} = -3$ where we could just add equation 1 to 2, we *subtract* $\ell_{21} = -1$ times equation 1 from equation 2 (which gives us the same result).

For instance here is the matrix that subtracts 2 times row 1 from row 3:

$$\mathbf{E}_{31} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}$$

If no row exchanges are needed, then three elimination matrices $\mathbf{E}_{21}, \mathbf{E}_{31}, \mathbf{E}_{32}$ will produce three zeros below the diagonal to change \mathbf{A} to the upper triangular \mathbf{U} (this just carries out elimination using matrices to represent each step).

Inverse

See that the *inverse* of each matrix \mathbf{E}_{ij} just *adds back* $\ell_{ij} \cdot (\text{row } j)$ to row i . This leads to the inverse of their product $\mathbf{E} = \mathbf{E}_{32}\mathbf{E}_{31}\mathbf{E}_{21}$. We denote the inverse of \mathbf{E} by \mathbf{L} . For instance, say some \mathbf{E} subtracts 5 times row 1 from row 2, then \mathbf{E}^{-1} adds 5 times row 1 to row 2:

$$\mathbf{E} = \begin{bmatrix} 1 & 0 & 0 \\ -5 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{E}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 5 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(See that sequential application of these matrices, in either order, to some vector leads to no net change—as if we had multiplied by the identity.)

Now let's consider \mathbf{F} which subtracts 4 times row 2 from row 3 (which might be a next step during elimination), naturally \mathbf{F}^{-1} adds it back:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -4 & 1 \end{bmatrix}, \quad \mathbf{F}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 4 & 1 \end{bmatrix}$$

During elimination we would first apply \mathbf{E} then \mathbf{F} , which is the same as applying \mathbf{FE} . Reversing the elimination would amount to $(\mathbf{FE})^{-1}$, which is the same as applying $\mathbf{E}^{-1}\mathbf{F}^{-1}$:

$$\mathbf{FE} = \begin{bmatrix} 1 & 0 & 0 \\ -5 & 1 & 0 \\ 20 & -4 & 1 \end{bmatrix} \quad \text{is inverted by} \quad \mathbf{E}^{-1}\mathbf{F}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 5 & 1 & 0 \\ 0 & 4 & 1 \end{bmatrix}$$

See that the product \mathbf{FE} contains '20' but its inverse doesn't. In \mathbf{FE} , row 3 feels an effect of size 20 from row 1. However in $\mathbf{E}^{-1}\mathbf{F}^{-1}$ that doesn't happen. (next page)

L is the inverse of E

E is the product of all the elimination matrices E_{ij} , it turns A into its upper triangular form $EA = U$ (assuming no row exchanges). The difficulty with E is multiplying all the separate elimination steps E_{ij} does not produce a good formula; illustrating $E = E_{32}E_{31}E_{21}$:

$$E = \begin{bmatrix} 1 & & \\ 0 & 1 & \\ 0 & -\ell_{32} & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ 0 & 1 & \\ -\ell_{31} & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ -\ell_{21} & 1 & \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & \\ -\ell_{21} & 1 & \\ (\ell_{32}\ell_{21} - \ell_{31}) & -\ell_{32} & 1 \end{bmatrix}.$$

See that the bottom left corner is dependent on multiple constants (since mutating the third row requires knowledge of the mutations that already occurred to the second row).

Now consider the inverse $E_{21}^{-1}E_{31}^{-1}E_{32}^{-1} = E^{-1} = L$ (see that inverses need to be applied in reverse):

$$E^{-1} = \begin{bmatrix} 1 & & \\ \ell_{21} & 1 & \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ 0 & 1 & \\ \ell_{31} & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ 0 & 1 & \\ 0 & \ell_{32} & 1 \end{bmatrix} = \begin{bmatrix} 1 & & \\ \ell_{21} & 1 & \\ \ell_{31} & \ell_{32} & 1 \end{bmatrix} = L$$

(In the inverse matrix, the mutation of each row doesn't depend on previous mutations, so the multipliers fall into place in the lower triangular L . Also see that the final matrix is only lower triangular because the elimination algorithm specifies that we don't manipulate the first row, and that we don't manipulate the second row with the third row.)

This is why we might want to consider $A = LU$ to go back from triangular U to the original A .

2.4 Gauss-Jordan elimination

How would one compute the inverse of an $n \times n$ matrix \mathbf{A} ? Before answering that question, one might want to consider whether it is really necessary to know \mathbf{A}^{-1} ; although it is possible to find the solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$ using $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, computing \mathbf{A}^{-1} and taking $\mathbf{A}^{-1}\mathbf{b}$ is a very slow way to find \mathbf{x} .

Say we want to compute \mathbf{A}^{-1} . This is equivalent to solving for $\mathbf{A}\mathbf{X} = \mathbf{I}$. In that sense, by performing the manipulations on \mathbf{A} to make it look like \mathbf{I} . We essentially replicate the effect of \mathbf{A}^{-1} on \mathbf{A} ; by repeating those steps on the identity, it becomes as if we were taking $\mathbf{A}^{-1}\mathbf{I} = \mathbf{A}^{-1}$, allowing us to obtain the desired matrix.

This whole process can be done with an augmented matrix using *Gauss-Jordan elimination*, where we essentially take steps to reduce \mathbf{A} to reduced row echelon form, while repeating said steps on the identity to obtain the inverse:

$$[\mathbf{A} \quad \mathbf{I}] = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 \end{array} \right] \rightarrow \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 \end{array} \right]$$

Gauss-Jordan elimination
Solve $\mathbf{A}\mathbf{X} = \mathbf{I} \Rightarrow \mathbf{X} = \mathbf{A}^{-1}$
Slower than solving $\mathbf{A}\mathbf{x} = \mathbf{b}$

$$\rightarrow \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{array} \right] = [\mathbf{I} \quad \mathbf{A}^{-1}]$$

The Gauss-Jordan essentially turns $[\mathbf{A}, \mathbf{I}]$ into $[\mathbf{I}, \mathbf{A}^{-1}]$, where the elimination steps are essentially equivalent to multiplication by \mathbf{A}^{-1} .

2.5 Proving $\mathbf{A} = \mathbf{LU}$

Elimination is expressed by $\mathbf{EA} = \mathbf{U}$ and inverted by $\mathbf{LU} = \mathbf{A}$. It starts with \mathbf{A} and ends with upper triangular \mathbf{U} , with each elimination step being carried out by elimination matrices \mathbf{E}_{ij} . To invert one elimination step we add rows instead of subtracting:

$$\mathbf{E}_{31} = \begin{bmatrix} 1 & & & \\ & 0 & 1 & \\ & -\ell_{31} & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{L}_{31} = \text{inverse of } \mathbf{E}_{31} = \begin{bmatrix} 1 & & & \\ & 0 & 1 & \\ & \ell_{31} & 0 & 1 \end{bmatrix}.$$

Recall that $\mathbf{E} = \mathbf{E}_{32}\mathbf{E}_{31}\mathbf{E}_{21}$ gives us a fairly messy result:

$$\mathbf{E} = \begin{bmatrix} 1 & & & \\ 0 & 1 & & \\ 0 & -\ell_{32} & 1 & \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 0 & 1 & \\ & -\ell_{31} & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & -\ell_{21} & 1 & \\ & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & -\ell_{21} & & 1 \\ & (\ell_{32}\ell_{21} - \ell_{31}) & -\ell_{32} & 1 \end{bmatrix}.$$

while the inverse, $\mathbf{E}^{-1} = \mathbf{E}_{21}^{-1}\mathbf{E}_{31}^{-1}\mathbf{E}_{32}^{-1} = \mathbf{L}$ produces a much simpler result

$$\boxed{\mathbf{E}^{-1} = \begin{bmatrix} 1 & & & \\ \ell_{21} & 1 & & \\ 0 & 0 & 1 & \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 0 & 1 & \\ & \ell_{31} & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 0 & 1 & \\ & 0 & \ell_{32} & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ \ell_{21} & 1 & & \\ \ell_{31} & \ell_{32} & 1 & \end{bmatrix} = \mathbf{L}}$$

We then have an elegant expression in $\mathbf{A} = \mathbf{E}^{-1}\mathbf{U} = \mathbf{LU}$. We now show that this equation holds over larger matrices of size n .

Proof 1

Following each step in elimination, consider the pivot rows that are subtracted from lower rows; see that these rows are not original rows of \mathbf{A} , since they have been mutated by the previous elimination steps; they are instead rows of \mathbf{U} .

When computing the, say, third row of \mathbf{U} , we subtract multiples of earlier rows of \mathbf{U} :

$$\text{Row 3 of } \mathbf{U} = (\text{Row 3 of } \mathbf{A}) - \ell_{31}(\text{Row 1 of } \mathbf{U}) - \ell_{32}(\text{Row 2 of } \mathbf{U})$$

Rewriting, see that

$$\text{Row 3 of } \mathbf{A} = \ell_{31}(\text{Row 1 of } \mathbf{U}) + \ell_{32}(\text{Row 2 of } \mathbf{U}) + 1(\text{Row 3 of } \mathbf{U})$$

Row $[\ell_{31}, \ell_{32}, 1]$ is multiplying the matrix \mathbf{U} . This is *exactly* row 3 of $\mathbf{A} = \mathbf{LU}$. All rows look like this regardless of the size of \mathbf{A} . With no row exchanges, we have $\mathbf{A} = \mathbf{LU}$.

(next page)

Proof 2

Here is another proof. The idea here is to see elimination as removing one rank 1 matrix at a time—one column of \mathbf{L} times one row of \mathbf{U} from \mathbf{A} ; where the problem becomes one size smaller with each iteration.

Elimination begins with pivot row = row 1 of \mathbf{A} . We multiply that pivot row by the numbers ℓ_{21} , then ℓ_{31} , and eventually ℓ_{n1} ; we subtract the respective products from row 2, row 3, and eventually row n of \mathbf{A} . By choosing $\ell_{21} = a_{21}/a_{11}$, $\ell_{31} = a_{31}/a_{11}$ and so on until $\ell_{n1} = a_{n1}/a_{11}$; now consider if we also subtracted away the pivot row away from itself—this subtraction leaves zeros in column 1:

$$\text{Step 1 removes } \begin{bmatrix} 1 \text{ (row 1)} \\ \ell_{21} \text{ (row 1)} \\ \ell_{31} \text{ (row 1)} \\ \ell_{41} \text{ (row 1)} \end{bmatrix} \text{ from } \mathbf{A} \text{ to leave } \mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}.$$

The idea here is that we *removed a rank 1 matrix* with columns made up of multiples of $\ell_1 = (1, \ell_{21}, \ell_{31}, \ell_{41}, \dots)$, scaled by each respective entry of the first row of \mathbf{A} , which is also the first pivot row \mathbf{u}_1 of the final upper triangular matrix.

We continue with elimination in the second pivot column, using the second row as the pivot row; as before we also subtract the second pivot row from itself. See that this is again equivalent to subtracting a rank 1 matrix with basis $\ell_2 = (0, 1, \ell_{32}, \ell_{42}, \dots)$. Recall that the second pivot row is the second row of the upper triangular matrix \mathbf{U} in our factorisation.

$$\text{Step 2 removes } \begin{bmatrix} 0 \text{ (row 2 of } \mathbf{A}_2) \\ 1 \text{ (row 2 of } \mathbf{A}_2) \\ \ell_{32} \text{ (row 2 of } \mathbf{A}_2) \\ \ell_{42} \text{ (row 2 of } \mathbf{A}_2) \end{bmatrix} \text{ from } \mathbf{A}_2 \text{ to leave } \mathbf{A}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix}.$$

Notice that both ℓ_1 and ℓ_2 are columns of \mathbf{L} . We removed a column ℓ_2 times the second pivot row. Continuing in the same way, we successively remove columns with each step removing a column ℓ_j of \mathbf{L} times a pivot row \mathbf{u}_j of \mathbf{U} . See that this entire process can be depicted as

$$\boxed{\mathbf{A} = \underset{\text{columns times rows}}{\ell_1 \mathbf{u}_1 + \ell_2 \mathbf{u}_2 + \dots + \ell_n \mathbf{u}_n} = \begin{bmatrix} \ell_1 & \dots & \ell_n \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_n \end{bmatrix} = \mathbf{L}\mathbf{U}}$$

Notice that \mathbf{U} is upper triangular; the pivot row \mathbf{u}_k begins with $k - 1$ zeros. \mathbf{L} is lower triangular with 1's on the main diagonal. Column ℓ_k also begins with $k - 1$ zeros. (See that this proof makes use of the idea that matrix multiplication can be seen as a sum of rank 1 matrices)

2.6 Permutation matrices

Examples

Permutation matrices have a 1 in every row and a 1 in every column. All other entries are 0. When this matrix P multiplies a vector, it changes the order of its components; for instance:

$$\begin{array}{l} \mathbf{P} \text{ has the rows of } \mathbf{I} \\ \mathbf{Px} = \text{Circular shift of } \mathbf{x} \\ 1, 2, 3 \text{ to } 3, 1, 2 \end{array} \quad \mathbf{Px} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_3 \\ x_1 \\ x_2 \end{bmatrix}$$

(A nonzero i th entry on row m of P means that row i of x goes on that row m in the result) Other examples include

$$\begin{array}{ll} \mathbf{Px} \text{ reverses} & \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} & \mathbf{x}_4 \text{ stays fixed} & \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ \text{the order to} & & \text{Input } x_1, x_2, x_3, x_4 & \\ x_4, x_3, x_2, x_1 & & \text{output } x_3, x_1, x_2, x_4 & \\ \mathbf{P}^2 = \mathbf{I} & & \mathbf{P}^3 = \mathbf{I} & \end{array}$$

$$\begin{array}{ll} \text{Circular shift} & \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} & \text{Evens before odds} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ \text{Input } x_1, x_2, x_3, x_4 & & \mathbf{x}_0, \mathbf{x}_2 \text{ before } \mathbf{x}_1, \mathbf{x}_3 & \\ \text{output } x_4, x_1, x_2, x_3 & & \text{Fast Fourier Transform} & \\ \mathbf{P}^4 = \mathbf{I} & & \text{Number from 0 to 3} & \end{array}$$

Recall that elimination may require row exchanges. If A is invertible, then there is a permutation P to order its rows in advance, so that elimination on PA meets no zeros in the pivot positions. Then $PA = LU$.

Inverse

We can intuit that the inverse of P is just its transpose:

$$\begin{array}{l} \text{The rows of any } P \text{ are} \\ \text{the columns of } P^{-1} = P^T \\ P^T = \text{"transpose of } P \text{"} \end{array} \quad P^T P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} = I$$

(For instance, a nonzero entry in row 1 column 2 means ‘row 2 of x becomes row 1 of result’. Its transpose corresponds to a nonzero entry in row 2 column 1, so ‘row 1 of x becomes row 2 in the result’—reversing the change. This logic can be extrapolated to every other row.)

(next page)

$PA = LU$ factorisation

Recall that elimination may require row exchanges; consider for instance

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & a \\ 2 & 4 & b \\ 3 & 7 & c \end{bmatrix} \xrightarrow{\mathbf{E}} \begin{bmatrix} 1 & 2 & a \\ 0 & 0 & b-2a \\ 0 & 1 & c-3a \end{bmatrix} \xrightarrow{\mathbf{P}} \begin{bmatrix} 1 & 2 & a \\ 0 & 1 & c-3a \\ 0 & 0 & b-2a \end{bmatrix} = \mathbf{U}$$

To rescue elimination, \mathbf{P} exchanged row 2 with 3, bringing 1 to the second pivot so elimination could continue.

See that we could order the rows in advance, first exchanging rows 2 and 3 to get \mathbf{PA} , then LU factorisation becomes $\mathbf{PA} = \mathbf{LU}$; the matrix \mathbf{PA} sails through elimination without seeing that zero pivot:

$$\begin{array}{ccccc} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 2 & a \\ 2 & 4 & b \\ 3 & 7 & c \end{bmatrix} & = & \begin{bmatrix} 1 & 2 & a \\ 3 & 7 & c \\ 2 & 4 & b \end{bmatrix} & = & \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & a \\ 0 & 1 & c-3a \\ 0 & 0 & b-2a \end{bmatrix} \\ \mathbf{P} & \mathbf{A} & & \mathbf{PA} & & \mathbf{L} & \mathbf{U} \end{array}$$

We may require several row exchanges; in that case the an overall permutation \mathbf{P} would include them all, still producing $\mathbf{PA} = \mathbf{LU}$. A useful way to keep track of the permutations might be to add a column of indices at the end of \mathbf{A} so that the original indices are displayed:

$$\begin{bmatrix} 1 & 2 & a & 1 \\ 2 & 4 & b & 2 \\ 3 & 7 & c & 3 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & a & 1 \\ 0 & 0 & b-2a & 2 \\ 0 & 1 & c-3a & 3 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & a & 1 \\ 0 & 1 & c-3a & 3 \\ 0 & 0 & b-2a & 2 \end{bmatrix}. \mathbf{P}_{132} \text{ is } \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Column permutations

We know that we can reorder rows using

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad \mathbf{PA} = \begin{bmatrix} a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{11} & a_{12} & a_{13} \end{bmatrix}$$

See that we can also reorder columns by applying a permutation matrix on the right:

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \mathbf{PAQ} = \begin{bmatrix} a_{23} & a_{22} & a_{21} \\ a_{33} & a_{32} & a_{31} \\ a_{13} & a_{12} & a_{11} \end{bmatrix}$$

Note that these two operations don't cover all possible permutations of the 9 entries in \mathbf{A} . The first index is constant on every row and the second index is constant on every column.

2.7 Transposes and symmetric matrices

2.7.1 The transpose of AB is $B^T A^T$: Intuition

We have

$$(AB)^T = B^T A^T$$

Consider AB . Computing the first *row* of the result (the first column of $(AB)^T$); this can be seen as a linear combination of the *rows* of B (the columns of B^T) specified by the entries in the first *row* of A (the first column of A^T).

2.7.2 Showing $(A^T)^{-1} = (A^{-1})^T$

Consider taking the transpose of $A^{-1}A$,

$$\text{Transpose of inverse} \quad A^{-1}A = I \quad \text{is transposed to} \quad A^T(A^{-1})^T = I.$$

Similarly, transposing $AA^{-1} = I$ leads to $(A^{-1})^T A^T = I$. Notice especially that A^T is invertible exactly when A is invertible.

2.7.3 $A^T A$ is symmetric

Choose any matrix A , probably rectangular. Multiply A^T times A , then the product $S = A^T A$ is automatically a square symmetric matrix:

$$\text{The transpose of } A^T A \text{ is } A^T(A^T)^T \text{ which is } A^T A \text{ again.}$$

The matrix AA^T is also symmetric (see that their shapes permit multiplication in either order), note however that AA^T is a different matrix from $A^T A$.

Chapter 3

The Four Fundamental Subspaces

3.1 Reduced Row Echelon Form

Motivation

Consider reducing a 2x4 matrix \mathbf{A} to its reduced row echelon form \mathbf{R} :

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 11 & 17 \\ 3 & 7 & 37 & 57 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & 11 & 17 \\ 0 & 1 & 4 & 6 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 3 & 5 \\ 0 & 1 & 4 & 6 \end{bmatrix} = \mathbf{R}$$

See how the added upward elimination essentially inverted the a portion \mathbf{W} of \mathbf{A} :

$$\mathbf{W} = \begin{bmatrix} 1 & 2 \\ 3 & 7 \end{bmatrix}$$

to turn that part of the matrix into the identity:

$$\text{Multiply } \mathbf{W}^{-1}\mathbf{A} = \mathbf{W}^{-1} \begin{bmatrix} \mathbf{W} & \mathbf{H} \end{bmatrix} \text{ to produce } \mathbf{R} = \begin{bmatrix} \mathbf{I} & \mathbf{W}^{-1}\mathbf{H} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{F} \end{bmatrix}.$$

We have $\mathbf{H} = \mathbf{W}\mathbf{F}$, where \mathbf{H} refers to the other parts of \mathbf{A} , and \mathbf{W} the portion that was inverted—we can combine the components of \mathbf{W} (the first independent columns) to produce the other (dependent) columns. The matrix \mathbf{F} specifies the parameters to do this:

$$\text{Dependent columns } \mathbf{H} = \begin{bmatrix} 11 & 17 \\ 37 & 57 \end{bmatrix} = \mathbf{W}\mathbf{F} = \begin{bmatrix} 1 & 2 \\ 3 & 7 \end{bmatrix} \text{ times } \begin{bmatrix} 3 & 5 \\ 4 & 6 \end{bmatrix}.$$

See that the first r independent columns of \mathbf{A} locate the columns of \mathbf{R} containing \mathbf{I} . Also see that the last $m - r$ rows of \mathbf{R} will be rows of zeros (the dependent columns in the reduced form can be expressed in terms of the identity contained within that form, so there can't be any additional nonzero rows).
(next page)

Algorithm

Elimination goes a column at a time from left to right; after k columns, that part of the matrix is in the reduced form, and we move to the $(k+1)$ th column; this new column has an upper part \mathbf{u} and a lower part ℓ :

$$\text{First } k+1 \text{ columns} \quad \begin{bmatrix} I_k & F_k \\ 0 & 0 \end{bmatrix} P_k \text{ followed by } \begin{bmatrix} \mathbf{u} \\ \ell \end{bmatrix}.$$

The idea here is to decide whether this $(k+1)$ th column joins I_k or F_k .

See that if ℓ is *all zeros*, the new column is *dependent* on the first k columns; then \mathbf{u} joins with F_k to produce F_{k+1} .

If ℓ is *not* all zero, then the column is *independent* of the first k columns. Pick any nonzero in ℓ as the pivot, move that row of \mathbf{A} up into row $k+1$, then subtract multiples of that pivot row to zero out all the rest of column $k+1$ (eliminate up and down). If necessary, divide the row by its first nonzero entry to have a pivot of 1. Column $k+1$ joins I_k (see that it adds a new nonzero row to the identity) to produce I_{k+1} .

Repeat for the next column and so on.

Row operations

There are three row operations allowed in elimination from \mathbf{A} to \mathbf{R} :

1. Subtract a multiple of one row from another (below or above)
2. Divide a row by its first nonzero entry (to reach pivot 1)
3. Exchange rows (to move all zero rows to the bottom)

A different series of steps could be used reach the same \mathbf{R} . But the result \mathbf{R} can't change.

3.2 $A = CR$ factorisation

We can apply elimination to reduce A to R_0 (reduced echelon form with zero rows); then I in R_0 locates the matrix C of *independent columns* in A . Removing any zero rows from R_0 produces the row matrix R such that $A = CR$. For instance,

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 5 \\ 3 & 6 & 9 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 3 \\ 0 & 0 & 6 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = R_0 \text{ with rank 2}$$

where the independent columns of A are 1 and 3, and

$$R = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 \\ 2 & 5 \\ 3 & 9 \end{bmatrix}$$

so

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 5 \\ 3 & 6 & 9 \end{bmatrix} = CR = \begin{bmatrix} 1 & 1 \\ 2 & 5 \\ 3 & 9 \end{bmatrix} \begin{bmatrix} 1 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{matrix} \text{(column basis)} \\ \text{in } C \end{matrix} \begin{matrix} \text{(row basis)} \\ \text{in } R \end{matrix}$$

3.3 Systematic nullspace computation

Elimination gives us a systematic way to find a basis for the nullspace. Say we have

$$\mathbf{A} = \begin{bmatrix} 1 & 7 & 3 & 35 \\ 2 & 14 & 6 & 70 \\ 2 & 14 & 9 & 97 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 7 & 3 & 35 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 27 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 7 & 0 & 8 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 27 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 7 & 0 & 8 \\ 0 & 0 & 1 & 9 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \mathbf{R}_0$$

Consider attempting to find a basis for the nullspace, meaning a basis for the solutions of $\mathbf{A}\mathbf{x} = \mathbf{0}$; say that $\mathbf{W}^{-1}\mathbf{A} = \mathbf{R}_0$, then for some \mathbf{x} ,

$$\mathbf{A}\mathbf{x} = \mathbf{0} \implies \mathbf{W}^{-1}\mathbf{A}\mathbf{x} = \mathbf{W}^{-1}\mathbf{0} \implies \mathbf{R}_0\mathbf{x} = \mathbf{0}$$

See that the elimination process doesn't change the space of solutions that satisfy $\mathbf{A}\mathbf{x} = \mathbf{0}$. Also see that using \mathbf{R} instead of \mathbf{R}_0 only removes the redundant zero rows without removing any solutions.

Given some \mathbf{R} , we have a systematic way to find a basis for the nullspace: considering all the indices of \mathbf{x} corresponding to dependent indices, by letting one of those indices be 1 and the rest 0, we have a simple solution for $\mathbf{R}\mathbf{x} = \mathbf{0}$:

$$\begin{aligned} \mathbf{R}\mathbf{s}_1 = \mathbf{0} \quad & \begin{bmatrix} 1 & 7 & 0 & 8 \\ 0 & 0 & 1 & 9 \end{bmatrix} \begin{bmatrix} -7 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{Put 1 and 0} \\ & \text{in positions 2 and 4} \\ \mathbf{R}\mathbf{s}_2 = \mathbf{0} \quad & \begin{bmatrix} 1 & 7 & 0 & 8 \\ 0 & 0 & 1 & 9 \end{bmatrix} \begin{bmatrix} -8 \\ 0 \\ -9 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{Put 0 and 1} \\ & \text{in positions 2 and 4} \end{aligned}$$

See that if we were to write \mathbf{R} as $[\mathbf{I}, \mathbf{F}]$, the solutions can be found in an algorithmic manner:

<p>The two special solutions to $[\mathbf{I} \quad \mathbf{F}]\mathbf{x} = \mathbf{0}$ are the columns of $\begin{bmatrix} -\mathbf{F} \\ \mathbf{I} \end{bmatrix}$ in Example 1</p> <p>The special solutions to $[\mathbf{I} \quad \mathbf{F}]\mathbf{P}\mathbf{x} = \mathbf{0}$ are the columns of $\mathbf{P}^T \begin{bmatrix} -\mathbf{F} \\ \mathbf{I} \end{bmatrix}$ in Example 2</p>
--

The second case occurs should we have to permute \mathbf{R} to reorder the independent columns. Recall that $\mathbf{P}\mathbf{P}^T = \mathbf{I}$, and so

$$\mathbf{R}\mathbf{x} = \mathbf{0} \quad [\mathbf{I} \quad \mathbf{F}]\mathbf{P} \text{ times } \mathbf{P}^T \begin{bmatrix} -\mathbf{F} \\ \mathbf{I} \end{bmatrix} \text{ reduces to } [\mathbf{I} \quad \mathbf{F}] \begin{bmatrix} -\mathbf{F} \\ \mathbf{I} \end{bmatrix} = [\mathbf{0}]$$

(next page)

Cont.

To put these two ideas together, suppose the $m \times n$ matrix \mathbf{A} has rank r . To find the $n - r$ special solutions to $\mathbf{Ax} = \mathbf{0}$, compute the reduced row echelon form \mathbf{R}_0 of \mathbf{A} ; remove the $m - r$ zero rows of \mathbf{R}_0 to produce $\mathbf{R} = [\mathbf{I}, \mathbf{F}]\mathbf{P}$, where $\mathbf{A} = \mathbf{C}\mathbf{R}$.

Then the special solutions to $\mathbf{Ax} = 0$ are the $n - r$ columns of $\mathbf{P}^T \begin{bmatrix} -\mathbf{F} \\ \mathbf{I} \end{bmatrix}$.

3.4 The complete solution to $Ax = b$

Finding a particular solution

We can reduce $Ax = b$ to a simpler system $R_0x = d$ with the same solutions (if any). A useful aid here is to augment A with b to produce an augmented matrix $[A, b]$:

$$\begin{bmatrix} 1 & 3 & 0 & 2 \\ 0 & 0 & 1 & 4 \\ 1 & 3 & 1 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 6 \\ 7 \end{bmatrix} \quad \text{has the augmented matrix} \quad \begin{bmatrix} 1 & 3 & 0 & 2 & 1 \\ 0 & 0 & 1 & 4 & 6 \\ 1 & 3 & 1 & 6 & 7 \end{bmatrix} = [A \ b].$$

Applying elimination to reach R_0 , the augmented part also undergoes the same elimination steps to produce a d :

$$\begin{bmatrix} 1 & 3 & 0 & 2 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 6 \\ 0 \end{bmatrix} \quad \text{has the augmented matrix} \quad \boxed{\begin{bmatrix} 1 & 3 & 0 & 2 & 1 \\ 0 & 0 & 1 & 4 & 6 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = [R_0 \ d]}.$$

For an easy solution x_p we can choose the free variables to be 0. In this case $x_2 = x_4 = 0$, then the pivot variables can just be read off from d :

$$R_0x_p = \begin{bmatrix} 1 & 3 & 0 & 2 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 6 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 6 \\ 0 \end{bmatrix} = d \quad \begin{array}{l} \text{Pivot variables } 1, 6 \\ \text{Free variables } 0, 0 \\ \mathbf{x_{particular}} = (1, 0, 6, 0). \end{array}$$

See that for a solution to exist, the zero rows in R_0 must also be zero in d .

This procedure allows us to come up with a particular solution. For a complete solution, we consider any sum of the particular solution x_p with any nullspace vectors x_n :

One $x_{\text{particular}}$	<i>The particular solution solves</i>	$Ax_p = b$
All $x_{\text{nullspace}}$	<i>The $n - r$ special solutions solve</i>	$Ax_n = 0$
Complete solution one x_p + many x_n particular x_p nullspace vectors x_n	$x = x_p + x_n =$	$\begin{bmatrix} 1 \\ 0 \\ 6 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} -3 \\ 1 \\ 0 \\ 0 \end{bmatrix} + x_4 \begin{bmatrix} -2 \\ 0 \\ -4 \\ 1 \end{bmatrix}.$

The nullspace can be found as per outlined earlier (setting one variable to one and the rest to zero to get each nullspace basis vector). See that the complete solution includes all nullspace basis vectors, and that if the nullspace is just the zero vector then every particular solution x_p is the only solution to that b .

(next page)

3.4.1 Rank and solution

Full column rank

In the case where \mathbf{A} has *full column rank*, every column has a pivot and the rank is $r = n$. The matrix is tall and thin ($m \geq n$). Row reduction puts $R = I$ at the top when \mathbf{A} is reduced to \mathbf{R}_0 with rank n :

$$\text{Full column rank } r = n \quad \mathbf{R}_0 = \begin{bmatrix} I \\ 0 \end{bmatrix} = \begin{bmatrix} n \text{ by } n \text{ identity matrix} \\ m - n \text{ rows of zeros} \end{bmatrix}$$

See that in this case the matrix has the following properties:

Every matrix \mathbf{A} with **full column rank** ($r = n$) has all these properties:

1. All columns of \mathbf{A} are pivot columns (**independent**). No free variables.
2. The nullspace $\mathbf{N}(\mathbf{A})$ contains only the zero vector $\mathbf{x} = \mathbf{0}$.
3. If $\mathbf{Ax} = \mathbf{b}$ has a solution (it might not) then it has only *one solution*.

With full column rank, $\mathbf{Ax} = \mathbf{b}$ will have *one solution or no solution*. (think of a mapping from n dimensional space into m dimensional space where $m \geq n$, there aren't enough vectors in n space to encompass all of m space, so there may be no solution for a particular vector in m space; or see that the columns only span a n dimensional hyperplane in a higher m dimensional space—any m dimensional vector not on that hyperplane has no solution)

Full row rank

The other case is full row rank—now $\mathbf{Ax} = \mathbf{b}$ has *one or infinitely many solutions*. In this case \mathbf{A} must be *short and wide* ($m \leq n$); the matrix has full row rank if $r = m$, where every row has a pivot.

(If $m < n$, then a nullspace consisting of more than just the zero vector exists and there will be multiple solutions to $\mathbf{Ax} = \mathbf{b}$.) See that a matrix like this has the following properties:

Every matrix \mathbf{A} with **full row rank** ($r=m$) has all these properties:

1. All rows have pivots, and \mathbf{R}_0 has no zero rows: $\mathbf{R}_0 = \mathbf{R}$.
2. $\mathbf{Ax} = \mathbf{b}$ has a solution for every right side \mathbf{b} .
3. The column space of \mathbf{A} is the whole space \mathbf{R}^m .
4. If $m < n$ the equation $\mathbf{Ax} = \mathbf{b}$ is **underdetermined** (many solutions).

(next page)

Cont.

The four possibilities for linear equations depend on the rank r

$r = m$	and	$r = n$	<i>Square and invertible</i>	$Ax = b$	has 1 solution
$r = m$	and	$r < n$	<i>Short and wide</i>	$Ax = b$	has ∞ solutions
$r < m$	and	$r = n$	<i>Tall and thin</i>	$Ax = b$	has 0 or 1 solution
$r < m$	and	$r < n$	<i>Not full rank</i>	$Ax = b$	has 0 or ∞ solutions

If the matrix A is square and invertible, it indicates a mapping from $m = n = r$ space into that same space; the basis specified by A spans the whole of r space and there is only one linear combination of that basis that solves for $Ax = b$ (since A has linearly independent columns and therefore a trivial nullspace).

If the matrix is not full rank, then the columns span a hyperplane in the space equal to the row length, meaning some $Ax = b$ may not have a solution, and if some b does lie within that hyperplane there will be infinite solutions to it (since the columns are not independent and therefore a nontrivial nullspace exists).

Their reduced row echelon form will fall into the following categories:

Four types for R_0	$[I]$	$[I \ F]$	$\begin{bmatrix} I \\ 0 \end{bmatrix}$	$\begin{bmatrix} I & F \\ 0 & 0 \end{bmatrix}$
Their ranks	$r = m = n$	$r = m < n$	$r = n < m$	$r < m, r < n$

Cases 1 and 2 have full row rank, case 3 has full column rank but not row rank, case 4 is not full rank. See that in the last two cases, when $Ax = b$ is reduced to $R_0x = d$, d must end in $m - r$ zeros for the equation to be solvable. F refers to the top part of the dependent columns.

3.5 Intuition for four subspaces

We have

<i>Four Fundamental Subspaces</i>	Dimensions
1. The <i>row space</i> is $\mathbf{C}(A^T)$, a subspace of \mathbf{R}^n .	r
2. The <i>column space</i> is $\mathbf{C}(A)$, a subspace of \mathbf{R}^m .	r
3. The <i>nullspace</i> is $\mathbf{N}(A)$, a subspace of \mathbf{R}^n .	$n - r$
4. The <i>left nullspace</i> is $\mathbf{N}(A^T)$, a subspace of \mathbf{R}^m .	$m - r$

Column space of reduced form not the same as original matrix

Note that although the reduced row echelon form provides the pivot positions to determine the row and column rank of a matrix, while the *rows* of the rref form span the same row space as the original unreduced matrix, the columns of the rref form cannot act as substitutes for the original column space, instead only providing information on dimensionality and on which original columns act as basis vectors for the column space. That is

$$\mathbf{C}(A) \neq \mathbf{C}(R_0) \quad \text{but} \quad \mathbf{C}(A^T) = \mathbf{C}(R_0^T)$$

(none of the row operations in elimination change the row space, but they do change the column space)

Nullspace dimension is $n - r$

The intuition for the dimension of $\mathbf{N}(A)$ being $n - r$ is clear from elimination, where each dependent column provides a new basis vector for the nullspace (we can find a basis by substituting one free variable as 1 and the rest zero, repeating for all free variables), so the number of basis vectors equals the number of dependent columns equals total columns minus independent columns—so $n - r$.

Left nullspace dimensionality

We know the dimensions for A is the same as that of A^T . Transposing A gives us m columns, of which we know r are independent. So there are $m - r$ dependent columns which each provide a basis vector for the nullspace.

3.6 $A = CR$ and Block elimination

The earlier notes on the rref form introduced it as a simple way to express the entire matrix in terms of its dependent rows, where for a matrix A , with independent components W and dependent components H , reduction to rref looks like:

$$A = \begin{bmatrix} W & H \end{bmatrix} \implies W^{-1}A = W^{-1} \begin{bmatrix} W & H \end{bmatrix} = \begin{bmatrix} I & W^{-1}H \end{bmatrix}$$

See that we can recover the dependent components of H with $W^{-1}H$ by multiplying it by the independent columns W , so

$$A = \underbrace{W}_C \underbrace{\begin{bmatrix} I & W^{-1}H \end{bmatrix}}_R$$

where C means the same thing as W , that is, the independent columns of A . See that W is guaranteed to be square since row and column rank are equal.

Block elimination

The above doesn't show how the zero rows in R_0 (if they exist) are redundant. We can develop a better idea of elimination if we consider working with blocks of the original A . This is called block elimination; consider reducing a matrix A with rank r to rref using the following three steps:

Step 1: Exchange columns of A using P_C and exchange rows of A using P_R to put the r independent columns first and r independent rows first. This looks like

$$P_R A P_C = \begin{bmatrix} W & H \\ J & K \end{bmatrix}$$

$\begin{bmatrix} W \\ J \end{bmatrix}$ has full column rank and $\begin{bmatrix} W & H \end{bmatrix}$ has full row rank. See that W is a square matrix with full rank r and is therefore invertible.

Step 2: Multiply the top rows by W^{-1} :

$$\begin{bmatrix} W & H \\ J & K \end{bmatrix} \rightarrow \begin{bmatrix} I & W^{-1}H \\ J & K \end{bmatrix}$$

Step 3: Subtract $J \begin{bmatrix} I & W^{-1}H \end{bmatrix}$ from the $m - r$ rows $\begin{bmatrix} J & K \end{bmatrix}$ to produce $\begin{bmatrix} 0 & 0 \end{bmatrix}$:

$$\begin{bmatrix} I & W^{-1}H \\ J & K \end{bmatrix} \rightarrow \begin{bmatrix} I & W^{-1}H \\ 0 & 0 \end{bmatrix}$$

To understand why $JW^{-1}H = K$, we know that the first r rows $\begin{bmatrix} I & W^{-1}H \end{bmatrix}$ are linearly independent, and since A has rank r , the lower rows $\begin{bmatrix} J & K \end{bmatrix}$ must be combinations of the upper rows. These combinations must be given by J to get the first r columns correct: $J I = J$. Then J times $W^{-1}H$ must equal K to make the last columns correct.

(next page)

Cont.

We have block elimination as the following transformations:

$$\boxed{P_R A P_C = \begin{bmatrix} W & H \\ J & K \end{bmatrix} \rightarrow \begin{bmatrix} I & W^{-1}H \\ J & K \end{bmatrix} \rightarrow \begin{bmatrix} I & W^{-1}H \\ 0 & 0 \end{bmatrix} = R_0}$$

We can, in a manner similar to the steps taken, factor out parts of A as follows to acquire the reduced row echelon form; for step 2:

$$P_R A P_C = \begin{bmatrix} W & H \\ J & K \end{bmatrix} = \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & W^{-1}H \\ J & K \end{bmatrix}$$

For step 3:

$$\begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & W^{-1}H \\ J & K \end{bmatrix} = \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ J & 0 \end{bmatrix} \begin{bmatrix} I & W^{-1}H \\ 0 & 0 \end{bmatrix}$$

The result is

$$P_R A P_C = \begin{bmatrix} W & 0 \\ J & 0 \end{bmatrix} \begin{bmatrix} I & W^{-1}H \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} W \\ J \end{bmatrix} \begin{bmatrix} I & W^{-1}H \end{bmatrix}$$

See that this is just the $A = CR$ factorisation, but this time showing that the zero rows are redundant rather than just using a reduced form where zero rows don't exist. Another way of writing this result is

$$\boxed{\text{The conclusion is that } P_R A P_C = \begin{bmatrix} W \\ J \end{bmatrix} W^{-1} \begin{bmatrix} W & H \end{bmatrix} = C W^{-1} B}$$

Chapter 4

Orthogonality

4.1 Orthogonal subspaces

Recall that two vectors \mathbf{v} and \mathbf{w} are orthogonal if $\mathbf{v}^T \mathbf{w} = 0$. To say that two subspaces are orthogonal means that every vector in one space is orthogonal to every vector in the other space.

We can make the claim that for any matrix \mathbf{A} , *the nullspace of \mathbf{A} is orthogonal to the row space of \mathbf{A}* . To see this just consider $\mathbf{Ax} = \mathbf{0}$:

$$\mathbf{Ax} = \begin{bmatrix} \text{row 1 of } \mathbf{A} \\ \vdots \\ \text{row } m \text{ of } \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{x} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \quad \begin{array}{l} \leftarrow (\text{row } 1) \cdot \mathbf{x} \text{ is zero} \\ \leftarrow (\text{row } m) \cdot \mathbf{x} \text{ is zero} \end{array}$$

Every row has a zero dot product with \mathbf{x} . Then every combination of the rows is perpendicular to \mathbf{x} . That is, the whole row space $\mathbf{C}(\mathbf{A}^T)$ is orthogonal to the nullspace $\mathbf{N}(\mathbf{A})$ in n dimensional space (since \mathbf{A}^T has n rows). Another way to show this is

$$\mathbf{x} \text{ in nullspace orthogonal to } \mathbf{A}^T \mathbf{y} \text{ in row space } \mathbf{x}^T (\mathbf{A}^T \mathbf{y}) = (\mathbf{Ax})^T \mathbf{y} = \mathbf{0}^T \mathbf{y} = 0.$$

See that we can analogously reason that the *the column space $\mathbf{C}(\mathbf{A})$ and the left nullspace $\mathbf{N}(\mathbf{A}^T)$ are perpendicular inside m dimensional space.*
(next page)

Cont.

See that the only common vector between two orthogonal subspaces would be the zero vector (we can't sum the columns of a matrix to produce a vector orthogonal to all the columns). There is a very important restriction on the dimension of any two orthogonal subspaces:

If V and W are orthogonal subspaces in \mathbb{R}^n then $\dim V + \dim W \leq n$.

For instance, the row space of a matrix is a subspace of dimension r in n space, and its orthogonal complement, by construction, will have dimensionality $n - r$.

4.1.1 Orthogonal complements

When two orthogonal subspaces account for the whole space, they are called *orthogonal complements*. The orthogonal complement of V is written as V^\perp . The row space and the nullspace, as well as the column space and left nullspace, are instances of orthogonal complements:

Orthogonal complements	Row space and nullspace	$r + (n - r) = n$
	Column space and left nullspace	$r + (m - r) = m$

4.1.2 Intuition for $x = x_{\text{row}} + x_{\text{null}}$

See that because the row space and the nullspace combined span the entire n space, we can say that any vector in \mathbb{R}^n can be expressed as a sum of vectors from both subspaces. That is, any vector x in \mathbb{R}^n is the sum $x = x_{\text{row}} + x_{\text{null}}$ of its row space component and its nullspace component. Any vector y in \mathbb{R}^m is the sum $y = y_{\text{col}} + y_{\text{null}}$ of its column space component and its left nullspace component from $N(A^T)$:



(next page)

4.1.3 A is invertible from the row space to the column space

Consider again the illustration from the previous page:



Every vector in Ax is in the column space; that is obvious. Every x comes from \mathbb{R}^n space, which is made up of the the row space $C(A^T)$ and its orthogonal complement, the nullspace $N(A)$. In that context, every candidate x either comes from the nullspace, which would map to $\mathbf{0}$, or the row space, which would then map to some b in the column space—every b in the column space can be found by some Ax with x from the row space.

We can also go on to say that *every vector b in the column space comes from exactly one vector x_r in the row space*. Proof: If $Ax_r = Ax'_r$, the difference $x_r - x'_r$ is in the nullspace (since $A(x_r - x'_r) = \mathbf{0}$); however, should x_r and x'_r come from the row space, then $x_r - x'_r$ would also have to be in the row space. The only way for a vector to be in both the nullspace and the row space would be if it were the zero vector; so $x_r - x'_r = \mathbf{0}$ and $x_r = x'_r$.

So A maps every row space component to a specific column space component; see that there is an r by r invertible matrix hiding inside A if we throw away the two nullspaces. *A is invertible from the row space to the column space.*

4.2 $A^T A$ is invertible only if A has linearly independent columns

$A^T A$ is a square matrix (n by n). We prove the above by showing that $A^T A$ has the same nullspace as A ; so that when the columns of A are linearly independent, and its nullspace only contains the zero vector, then $A^T A$, with this same nullspace, is also invertible.

First we show that a vector in the nullspace of A is also in that of $A^T A$. Let A be any matrix; if \mathbf{x} is in the nullspace, then $A\mathbf{x} = \mathbf{0}$. Multiplying by A^T gives $A^T A\mathbf{x} = \mathbf{0}$, so \mathbf{x} is also in the nullspace of $A^T A$.

Now the other way, we show that a vector in the nullspace of $A^T A$ is in that of A . Starting with $A^T A\mathbf{x} = \mathbf{0}$, multiplying by \mathbf{x}^T :

$$(\mathbf{x}^T)A^T A\mathbf{x} = 0 \quad \text{or} \quad (A\mathbf{x})^T(A\mathbf{x}) = 0 \quad \text{or} \quad \|A\mathbf{x}\|^2 = 0.$$

So if $A^T A\mathbf{x} = \mathbf{0}$ then $A\mathbf{x}$ has length 0, so $A\mathbf{x} = \mathbf{0}$.

Every vector \mathbf{x} in one nullspace is in the other nullspace, so if $A^T A$ has dependent columns then so does A , and if $A^T A$ has independent columns then so does A :

When A has independent columns, $A^T A$ is square, symmetric, and invertible.

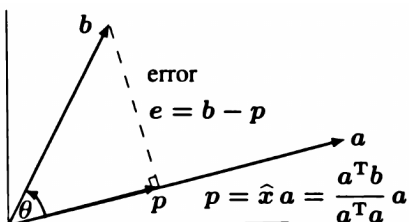
4.3 Projections

4.3.1 Projection onto a line

We want to project any \mathbf{b} onto the column space of any m by n matrix \mathbf{A} . This entails finding the point \mathbf{p} on that subspace that is closest to our arbitrary point \mathbf{b} . We start with a line.

Considering a one dimensional subspace (a line) with basis $\mathbf{a} = (a_1, \dots, a_m)$. Along that line, we want the point \mathbf{p} closest to $\mathbf{b} = (b_1, \dots, b_m)$; the key to projection is orthogonality: *the line from \mathbf{b} to \mathbf{p} , that is $\mathbf{b} - \mathbf{p}$, is perpendicular to the vector \mathbf{a} .*

We know that the projection \mathbf{p} will be some multiple of \mathbf{a} ; call it $\mathbf{p} = \hat{x}\mathbf{a}$ (so here \hat{x} denotes a constant)—computing this number \hat{x} will give us the vector \mathbf{p} . We can call the perpendicular line $\mathbf{b} - \mathbf{p}$ the ‘error’ $\mathbf{e} = \mathbf{b} - \hat{x}\mathbf{a}$, and use the fact that it is perpendicular to \mathbf{a} to derive \hat{x} :



Projecting \mathbf{b} onto \mathbf{a} with error $\mathbf{e} = \mathbf{b} - \hat{x}\mathbf{a}$ $\mathbf{a} \cdot (\mathbf{b} - \hat{x}\mathbf{a}) = 0$ or $\mathbf{a} \cdot \mathbf{b} - \hat{x}\mathbf{a} \cdot \mathbf{a} = 0$	$\hat{x} = \frac{\mathbf{a} \cdot \mathbf{b}}{\mathbf{a} \cdot \mathbf{a}} = \frac{\mathbf{a}^T \mathbf{b}}{\mathbf{a}^T \mathbf{a}}$
---	---

With that we can compute the projection $\mathbf{p} = \hat{x}\mathbf{a}$:

The projection of \mathbf{b} onto the line through \mathbf{a} is the vector $\mathbf{p} = \hat{x}\mathbf{a} = \frac{\mathbf{a}^T \mathbf{b}}{\mathbf{a}^T \mathbf{a}} \mathbf{a}$. Special case 1 : If $\mathbf{b} = \mathbf{a}$ then $\hat{x} = 1$. The projection of \mathbf{a} onto \mathbf{a} is itself. $P\mathbf{a} = \mathbf{a}$. Special case 2 : If \mathbf{b} is perpendicular to \mathbf{a} then $\mathbf{a}^T \mathbf{b} = 0$. The projection is $\mathbf{p} = \mathbf{0}$.

(next page)

Projection matrix

We had the projection of \mathbf{b} onto the line through \mathbf{a} as the vector \mathbf{p} given by

$$\mathbf{p} = \hat{x}\mathbf{a} = \frac{\mathbf{a}^T \mathbf{b}}{\mathbf{a}^T \mathbf{a}} \mathbf{a}$$

Now comes the *projection matrix*, where we want a matrix \mathbf{P} that can be applied to the arbitrary point \mathbf{b} to get the desired projection \mathbf{p} , so $\mathbf{P}\mathbf{b} = \mathbf{p}$. This can be found easily; since \hat{x} is a constant, we can rearrange the above equation:

Projection matrix \mathbf{P}	$\mathbf{p} = \mathbf{a}\hat{x} = \mathbf{a} \frac{\mathbf{a}^T \mathbf{b}}{\mathbf{a}^T \mathbf{a}} = \mathbf{P}\mathbf{b} \quad \text{when the matrix is} \quad \mathbf{P} = \frac{\mathbf{a}\mathbf{a}^T}{\mathbf{a}^T \mathbf{a}}.$
--	---

The projection matrix \mathbf{P} is a m by m rank one matrix. Notice that the line through \mathbf{a} , the subspace we are projecting onto, is the column space of \mathbf{P} .

4.3.2 Projection onto a subspace

Now we project onto a n -dimensional subspace of \mathbb{R}^m . We start with n vectors $\mathbf{a}_1, \dots, \mathbf{a}_n$ in \mathbb{R}^m , assume that they are linearly independent.

We want to find a projection of \mathbf{b} onto the subspace spanned by these vectors, that is, the combination $\mathbf{p} = \hat{x}_1 \mathbf{a}_1 + \dots + \hat{x}_n \mathbf{a}_n$ closest to a given vector \mathbf{b} . See that this can be written as $\mathbf{A}\hat{\mathbf{x}}$, where $\hat{\mathbf{x}} = (\hat{x}_1, \dots, \hat{x}_n)$.

As before, the idea here is that the error vector $\mathbf{e} = \mathbf{b} - \mathbf{p} = \mathbf{b} - \mathbf{A}\hat{\mathbf{x}}$ is perpendicular to the subspace we are projecting onto. Every basis vector is perpendicular to this error vector, so

$$\begin{array}{l} \mathbf{a}_1^T (\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}) = 0 \\ \vdots \\ \mathbf{a}_n^T (\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}) = 0 \end{array} \quad \text{or} \quad \begin{bmatrix} -\mathbf{a}_1^T & - \\ \vdots & \\ -\mathbf{a}_n^T & - \end{bmatrix} \begin{bmatrix} \mathbf{b} - \mathbf{A}\hat{\mathbf{x}} \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix} \quad \text{or} \quad \mathbf{A}^T \mathbf{A}\hat{\mathbf{x}} = \mathbf{A}^T \mathbf{b}.$$

We have

$$\mathbf{A}^T (\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}) = \mathbf{0} \implies \mathbf{A}^T \mathbf{A}\hat{\mathbf{x}} = \mathbf{A}^T \mathbf{b}$$

(next page)

Cont.

We had the projection of \mathbf{b} onto the column space of \mathbf{A} as $\mathbf{p} = \mathbf{A}\hat{\mathbf{x}}$. We showed that

$$\mathbf{A}^T(\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}) = \mathbf{0} \implies \mathbf{A}^T\mathbf{A}\hat{\mathbf{x}} = \mathbf{A}^T\mathbf{b}$$

With this we can derive the projection matrix:

The combination $\mathbf{p} = \hat{x}_1\mathbf{a}_1 + \dots + \hat{x}_n\mathbf{a}_n = \mathbf{A}\hat{\mathbf{x}}$ that is closest to \mathbf{b} comes from $\hat{\mathbf{x}}$:

$$\text{Find } \hat{\mathbf{x}} (n \times 1) \quad \mathbf{A}^T(\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}) = \mathbf{0} \quad \text{or} \quad \mathbf{A}^T\mathbf{A}\hat{\mathbf{x}} = \mathbf{A}^T\mathbf{b}. \quad (5)$$

This symmetric matrix $\mathbf{A}^T\mathbf{A}$ is n by n . It is invertible if the \mathbf{a} 's are independent. The solution is $\hat{\mathbf{x}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$. The **projection** of \mathbf{b} onto the subspace is \mathbf{p} :

$$\text{Find } \mathbf{p} (m \times 1) \quad \mathbf{p} = \mathbf{A}\hat{\mathbf{x}} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}. \quad (6)$$

The **projection matrix** \mathbf{P} is multiplying \mathbf{b} in (6). It has four \mathbf{A} 's:

$$\text{Find } \mathbf{P} (m \times m) \quad \mathbf{P} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T. \quad (7)$$

Again see that this only works if $\mathbf{A}^T\mathbf{A}$ is invertible, which is only when the column vectors of \mathbf{A} are independent. See that the final projection matrix formula is analogous to the formula derived in the one dimensional case.

The subspace we are projecting onto is the column space of \mathbf{A} , that is, $\mathbf{C}(\mathbf{A})$. See that the error $\mathbf{e} = \mathbf{b} - \mathbf{A}\hat{\mathbf{x}}$ belongs to the perpendicular subspace $\mathbf{N}(\mathbf{A}^T)$ (since $\mathbf{A}^T(\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}) = \mathbf{0}$).

A note on the projection matrix

The matrix $\mathbf{P} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ is deceptive. One might try to split $(\mathbf{A}^T\mathbf{A})^{-1}$ into \mathbf{A}^{-1} times $(\mathbf{A}^T)^{-1}$ and find that the entire formula reduces to \mathbf{I} .

This is wrong because *the matrix \mathbf{A} is rectangular and therefore has no inverse*. We cannot split $(\mathbf{A}^T\mathbf{A})^{-1}$ into \mathbf{A}^{-1} times $(\mathbf{A}^T)^{-1}$ because there is no \mathbf{A}^{-1} in the first place.

■ REVIEW OF THE KEY IDEAS ■

1. The projection of \mathbf{b} onto the line through \mathbf{a} is $\mathbf{p} = \mathbf{a}\hat{x} = \mathbf{a}(\mathbf{a}^T\mathbf{b}/\mathbf{a}^T\mathbf{a})$.
2. The rank one projection matrix $\mathbf{P} = \mathbf{a}\mathbf{a}^T/\mathbf{a}^T\mathbf{a}$ multiplies \mathbf{b} to produce \mathbf{p} .
3. Projecting \mathbf{b} onto a subspace leaves $\mathbf{e} = \mathbf{b} - \mathbf{p}$ perpendicular to the subspace.
4. When \mathbf{A} has full rank n , the equation $\mathbf{A}^T\mathbf{A}\hat{\mathbf{x}} = \mathbf{A}^T\mathbf{b}$ leads to $\hat{\mathbf{x}}$ and $\mathbf{p} = \mathbf{A}\hat{\mathbf{x}}$.
5. The projection matrix $\mathbf{P} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ has $\mathbf{P}^T = \mathbf{P}$ and $\mathbf{P}^2 = \mathbf{P}$ and $\mathbf{P}\mathbf{b} = \mathbf{p}$.

4.4 Least Squares Approximations

Illustrative example

Consider trying to find the closest line to three points $(0, 6), (1, 0), (2, 0)$. No straight line $\mathbf{b} = C + D\mathbf{t}$ goes through those three points; we want two numbers C and D that satisfy a system of three equations (this leads us to a 3 by 2 matrix):

$$\begin{array}{lll} \mathbf{t} = 0 & \text{The first point is on the line } \mathbf{b} = C + D\mathbf{t} \text{ if} & C + D \cdot 0 = 6 \\ \mathbf{t} = 1 & \text{The second point is on the line } \mathbf{b} = C + D\mathbf{t} \text{ if} & C + D \cdot 1 = 0 \\ \mathbf{t} = 2 & \text{The third point is on the line } \mathbf{b} = C + D\mathbf{t} \text{ if} & C + D \cdot 2 = 0. \end{array}$$

(This is essentially sampling a straight line at the three time points and comparing it to the desired data points at those times) This 3 by 2 system has *no solution*: $\mathbf{b} = (6, 0, 0)$ is not a combination of the columns $(1, 1, 1)$ and $(0, 1, 2)$ (of the matrix \mathbf{A}):

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} C \\ D \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{Ax} = \mathbf{b} \text{ is not solvable.}$$

The idea here is to choose a vector of parameters \mathbf{x} that makes $\mathbf{e} = \mathbf{b} - \mathbf{Ax}$ as small as possible—so finding a point in the column space of \mathbf{A} that is *closest to* \mathbf{b} —see that this is just the projection. So the best \mathbf{x} , which we denote as $\hat{\mathbf{x}}$, is found by $\mathbf{A}^T \mathbf{A} \hat{\mathbf{x}} = \mathbf{A}^T \mathbf{b}$, where the projection \mathbf{p} of \mathbf{b} onto the column space of \mathbf{A} is denoted by $\mathbf{p} = \mathbf{A} \hat{\mathbf{x}}$:

$$\mathbf{Ax} = \mathbf{b} = \mathbf{p} + \mathbf{e} \text{ is impossible} \quad \mathbf{A} \hat{\mathbf{x}} = \mathbf{p} \text{ is solvable} \quad \hat{\mathbf{x}} \text{ is } (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}.$$

See that the solution to $\mathbf{A} \hat{\mathbf{x}} = \mathbf{p}$ results in the least possible squared error:

$$\text{Squared error for any } \mathbf{x} \quad \|\mathbf{Ax} - \mathbf{b}\|^2 = \|\mathbf{Ax} - \mathbf{p}\|^2 + \|\mathbf{e}\|^2.$$

Since any \mathbf{x} other than $\hat{\mathbf{x}}$ would lead to $\|\mathbf{Ax} - \mathbf{p}\|^2 > 0$. (This expression comes from the pythagorean theorem, where the vector from some point \mathbf{Ax} to point \mathbf{b} , that is $\mathbf{Ax} - \mathbf{b}$, whose length we want to minimise, is the hypotenuse, while the the error $\mathbf{e} = \mathbf{b} - \mathbf{p}$, perpendicular to the vector $\mathbf{Ax} - \mathbf{p}$, are the opposite and adjacent sides respectively)

By choosing $\mathbf{x} = \hat{\mathbf{x}}$, we leave the smallest possible error $\mathbf{e} = (e_1, e_2, e_3)$ that we can't reduce, while reducing $\mathbf{Ax} - \mathbf{p}$ to zero. There is some ambiguity in the idea of 'smallest', but here it means that we minimise the *squared length* of $\mathbf{Ax} - \mathbf{b}$:

The least squares solution $\hat{\mathbf{x}}$ makes $E = \|\mathbf{Ax} - \mathbf{b}\|^2$ as small as possible.

(next page)

Cont.

By taking $x = \hat{x}$, where $A\hat{x}$ corresponds to the projection of b onto the column space of A , we have the x that minimises the squared error $\|Ax - b\|^2$:

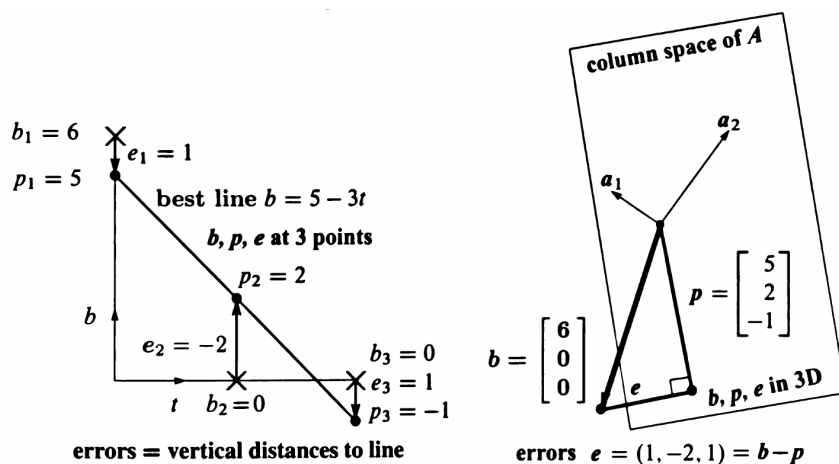


Figure 4.6: **Best line and projection: Two pictures, same problem.** The line has heights $p = (5, 2, -1)$ with errors $e = (1, -2, 1)$. The equations $A^T A \hat{x} = A^T b$ give $\hat{x} = (5, -3)$. Same answer! The best line is $b = 5 - 3t$ and the closest point is $p = 5a_1 - 3a_2$.

Projection allows us to minimise squared error.

Intuition from calculus

Consider minimising the squared error by calculus. The total squared error of the system, with parameters C and D , is

$$E = \|Ax - b\|^2 = (C + D \cdot 0 - 6)^2 + (C + D \cdot 1)^2 + (C + D \cdot 2)^2.$$

We take the partial derivative with respect to each parameter and set them to 0:

$$\partial E / \partial C = 2(C + D \cdot 0 - 6) + 2(C + D \cdot 1) + 2(C + D \cdot 2) = 0$$

$$\partial E / \partial D = 2(C + D \cdot 0 - 6)(0) + 2(C + D \cdot 1)(1) + 2(C + D \cdot 2)(2) = 0.$$

Simplifying leaves us with the system

$$\begin{array}{ll} \text{The } C \text{ derivative is zero: } 3C + 3D = 6 & \text{This matrix } \begin{bmatrix} 3 & 3 \\ 3 & 5 \end{bmatrix} \text{ is } A^T A \\ \text{The } D \text{ derivative is zero: } 3C + 5D = 0 \end{array}$$

This system is exactly $A^T A \hat{x} = A^T b$, where optimisation leads us to the exact same system we derived by projection:

$$\boxed{\text{The partial derivatives of } \|Ax - b\|^2 \text{ are zero when } A^T A \hat{x} = A^T b.}$$

(next page)

The big picture for least squares

The intuition for the four subspaces can be applied here. Given some \mathbf{b} that isn't in the column space of \mathbf{A} , we can split it into the sum of its projection \mathbf{p} in the column space and the minimum error \mathbf{e} in the orthogonal subspace; that is, we have $\mathbf{b} = \mathbf{p} + \mathbf{e}$:

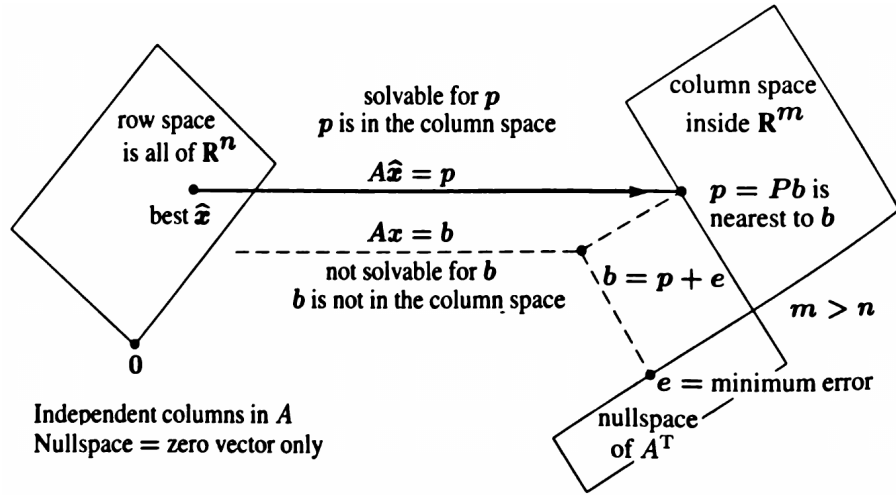


Figure 4.7: The projection $\mathbf{p} = \mathbf{A}\hat{\mathbf{x}}$ is closest to \mathbf{b} , so $\hat{\mathbf{x}}$ minimizes $E = \|\mathbf{b} - \mathbf{A}\mathbf{x}\|^2$.

As discussed earlier, see that the row space has a solution for any vector in the column space of \mathbf{A} , meaning that it contains $\hat{\mathbf{x}}$. Also notice that since we construct \mathbf{A} as having full column rank, its nullspace will just be the zero vector. (next page)

4.4.1 Fitting a straight line

Fitting a line is the clearest application of least squares. It starts with $m > 2$ points b_1, \dots, b_m , at times t_1, \dots, t_m . The best fit line $C + Dt$ misses the point by vertical distances e_1, \dots, e_m . The least squares line minimises $E = e_1^2 + \dots + e_m^2$ (note that this is not the minimum error as in the previous pages).

A line goes through all m points when we can exactly solve $A\mathbf{x} = \mathbf{b}$; generally this isn't possible so we want a best fit. Two unknowns C and D determine a line, so A only has $n = 2$ columns. To fit m points, we are trying to solve m equations:

$$A\mathbf{x} = \mathbf{b} \quad \text{is} \quad \begin{array}{l} C + Dt_1 = b_1 \\ C + Dt_2 = b_2 \\ \vdots \\ C + Dt_m = b_m \end{array} \quad \text{with} \quad A = \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_m \end{bmatrix}.$$

The column space is so thin that \mathbf{b} is almost certainly outside of it (if \mathbf{b} happens to lie in the column space then $\mathbf{b} = \mathbf{p}$ and the errors are $\mathbf{e} = \mathbf{0}$). The parameters $\hat{\mathbf{x}}$ for the line that minimises the squared error $\|A\mathbf{x} - \mathbf{b}\|^2$ is found by the projection:

The closest line $C + Dt$ has heights p_1, \dots, p_m with errors e_1, \dots, e_m .

Solve $A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$ for $\hat{\mathbf{x}} = (C, D)$. The errors are $\mathbf{e}_i = \mathbf{b}_i - C - Dt_i$.

See that the matrix $A^T A$ looks like

$$\text{Dot-product matrix } A^T A = \begin{bmatrix} 1 & \cdots & 1 \\ t_1 & \cdots & t_m \end{bmatrix} \begin{bmatrix} 1 & t_1 \\ \vdots & \vdots \\ 1 & t_m \end{bmatrix} = \begin{bmatrix} m & \sum t_i \\ \sum t_i & \sum t_i^2 \end{bmatrix}.$$

and $A^T \mathbf{b}$,

$$A^T \mathbf{b} = \begin{bmatrix} 1 & \cdots & 1 \\ t_1 & \cdots & t_m \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} = \begin{bmatrix} \sum b_i \\ \sum t_i b_i \end{bmatrix}.$$

The line $C + Dt$ minimises the error $E(x) = e_1^2 + \dots + e_m^2 = \|A\mathbf{x} - \mathbf{b}\|^2$ when $A^T A \hat{\mathbf{x}} = A^T \mathbf{b}$:

$$A^T A \hat{\mathbf{x}} = A^T \mathbf{b} \quad \begin{bmatrix} m & \sum t_i \\ \sum t_i & \sum t_i^2 \end{bmatrix} \begin{bmatrix} C \\ D \end{bmatrix} = \begin{bmatrix} \sum b_i \\ \sum t_i b_i \end{bmatrix}.$$

$$E(\mathbf{x}) = \|A\mathbf{x} - \mathbf{b}\|^2 = (C + Dt_1 - b_1)^2 + \dots + (C + Dt_m - b_m)^2.$$

4.5 Orthonormal Bases

The vectors $\mathbf{q}_1, \dots, \mathbf{q}_n$ are *orthogonal* when their dot products $\mathbf{q}_1 \cdots \mathbf{q}_j$ are zero— $\mathbf{q}_i^T \mathbf{q}_j = 0$ whenever $i \neq j$. By *dividing each vector by its length*, the vectors become orthogonal *unit vectors*—their lengths are all 1—in this case the basis is called *orthonormal*:

DEFINITION The n vectors $\mathbf{q}_1, \dots, \mathbf{q}_n$ are *orthonormal* if

$$\mathbf{q}_i^T \mathbf{q}_j = \begin{cases} 0 & \text{when } i \neq j \quad (\text{orthogonal vectors}) \\ 1 & \text{when } i = j \quad (\text{unit vectors: } \|\mathbf{q}_i\| = 1) \end{cases}$$

A matrix Q with orthonormal columns has $Q^T Q = I$. Typically $m > n$.

The matrix Q is easy to work with because $Q^T Q = I$ (since each column is orthonormal):

$$Q^T Q = \begin{bmatrix} -\mathbf{q}_1^T - \\ -\mathbf{q}_2^T - \\ -\mathbf{q}_n^T - \end{bmatrix} \begin{bmatrix} | & | & | \\ \mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_n \\ | & | & | \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} = I$$

On the diagonal the unit vectors give $\mathbf{q}_i^T \mathbf{q}_i = \|\mathbf{q}_i\|^2 = 1$. Note an important implication:

When Q is square, $Q^T Q = I$ means that $Q^T = Q^{-1}$: transpose = inverse.

Note that generally if $m \neq n$ then $Q Q^T \neq I$; although $Q^T Q = I$ even when Q is rectangular, Q^T here is only a 1-sided inverse from the left. Only for square matrices do we also have $Q Q^T = I$ (from the definition of the inverse of square matrices as covered earlier).

(The matrix must have full column rank since if its columns were to be orthonormal; so we have $m \geq n$. A mapping by Q in the case where $m > n$ is invertible since it has full column rank, but the transpose (size $n \times m$) doesn't have full column rank and therefore a specific mapping $Q^T \mathbf{x} = \mathbf{b}$ would have infinite solutions—a \mathbf{x} cannot be exactly recovered given a Q^T and \mathbf{b} . As such only applying Q first works.)

If we have a *square* Q with orthonormal columns, its transpose is its inverse from both sides; and as such also has orthonormal columns—both the rows and columns of Q are orthonormal. A square matrix with orthonormal columns is called an *orthogonal matrix*.

Also see that if the columns are only orthogonal and not orthonormal, their dot products still give a diagonal matrix.

(next page)

4.5.1 Multiplication by any orthogonal matrix preserves length and angle

Length is preserved during multiplication with an orthogonal matrix:

Proof $\|Q\mathbf{x}\|^2$ equals $\|\mathbf{x}\|^2$ because $(Q\mathbf{x})^T(Q\mathbf{x}) = \mathbf{x}^T Q^T Q \mathbf{x} = \mathbf{x}^T I \mathbf{x} = \mathbf{x}^T \mathbf{x}$.

If Q has orthonormal columns ($Q^T Q = I$), it leaves lengths unchanged:

$$\text{Same length for } Q\mathbf{x} \quad \|Q\mathbf{x}\| = \|\mathbf{x}\| \text{ for every vector } \mathbf{x}. \quad (3)$$

The dot product, and therefore the angle, is also preserved:

$$Q \text{ also preserves dot products: } (Q\mathbf{x})^T(Q\mathbf{y}) = \mathbf{x}^T Q^T Q \mathbf{y} = \mathbf{x}^T \mathbf{y}.$$

4.5.2 Projections using orthonormal bases

Using orthonormal bases for projection greatly simplifies computation. For projections onto subspaces, all formulas involve $A^T A$; suppose the basis vectors are orthonormal, then $A^T A$ simplifies to $Q^T Q = I$. Our projection becomes

$$\hat{\mathbf{x}} = Q^T \mathbf{b} \quad \text{and} \quad \mathbf{p} = Q\hat{\mathbf{x}} \quad \text{and} \quad P = QQ^T. \quad (4)$$

The least squares solution of $Q\mathbf{x} = \mathbf{b}$ is $\hat{\mathbf{x}} = Q^T \mathbf{b}$. The projection matrix is QQ^T .

See that no inversion is required—this is the point of an orthonormal basis—the best $\hat{\mathbf{x}} = Q^T \mathbf{b}$ is just dot products of $\mathbf{q}_1, \dots, \mathbf{q}_n$ with \mathbf{b} :

$$\text{Projection onto } \mathbf{q}\text{'s} \quad \mathbf{p} = \begin{bmatrix} | & & | \\ \mathbf{q}_1 & \cdots & \mathbf{q}_n \\ | & & | \end{bmatrix} \begin{bmatrix} \mathbf{q}_1^T \mathbf{b} \\ \vdots \\ \mathbf{q}_n^T \mathbf{b} \end{bmatrix} = \mathbf{q}_1(\mathbf{q}_1^T \mathbf{b}) + \cdots + \mathbf{q}_n(\mathbf{q}_n^T \mathbf{b}).$$

Also see that when Q is square (so an orthonormal basis spanning the entire space), $\hat{\mathbf{x}} = Q^T \mathbf{b}$ leads to $\mathbf{p} = Q\hat{\mathbf{x}} = QQ^T \mathbf{b} = I\mathbf{b} = \mathbf{b}$. The projection of \mathbf{b} onto the basis that spans the entire space is \mathbf{b} itself. (See that $QQ^T = I$ only works for square matrices)

4.6 Gram-Schmidt Process

Given three *independent vectors* a, b, c , we intend to construct three *orthogonal* vectors A, B, C . Then, by dividing A, B, C by their lengths, we can produce three orthonormal vectors $q_1 = A/\|A\|$, $q_2 = B/\|B\|$, $q_3 = C/\|C\|$.

Begin by choosing $A = a$ (we make all other vectors orthogonal to this). The next vector B must be perpendicular to A ; we *start with b and subtract its projection along A* . This leaves the perpendicular part, which is the orthogonal vector B :

First Gram-Schmidt step
 B is perpendicular to A

$$B = b - \frac{A^T b}{A^T A} A.$$

We use the one dimensional projection of b onto a to produce a vector orthogonal to a ; then we use it as our next basis vector.

The third direction starts with c . We can say that c is not a combination of A and B because c is not a combination of a and b (see that B is just a combination of b and $A = a$ since \hat{x} in the one dimensional case is just a constant). It is also unlikely that c is already perpendicular to A and B ; the idea here is that we subtract off its components in those two directions to get a perpendicular direction C :

Next Gram-Schmidt step
 C is perpendicular to A and B

$$C = c - \frac{A^T c}{A^T A} A - \frac{B^T c}{B^T B} B.$$

This is the entire idea behind the Gram-Schmidt process; *Subtract from every vector its projections in the directions already set*, repeating for each new basis vector. At the end, or immediately when each one is found, dividing the orthogonal vectors A, B, C, D by their lengths leaves us with *orthonormal* results q_i

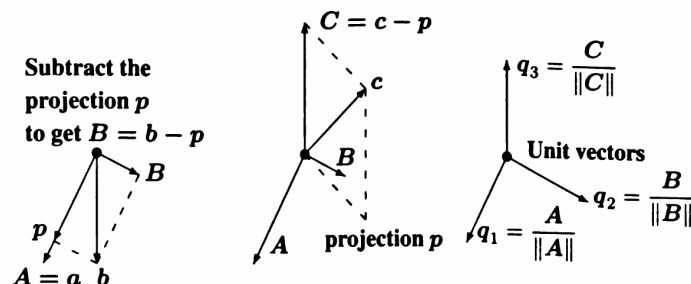


Figure 4.10: First project b onto the line through a and find the orthogonal $B = b - p$. Then project c to p on the AB plane and find $C = c - p$. Divide by $\|A\|$, $\|B\|$, $\|C\|$.

4.7 $A = QR$ factorisation

Using the Gram-Schmidt process we can derive an orthonormal matrix Q with columns q_1, q_2 , and q_3 from a matrix A with columns a, b , and c . The vectors q_1, q_2, q_3 come from linear combinations of a, b, c and vice versa. As such there must be a third matrix that maps Q to A .

Matrix A has independent columns, and so any mapping using its column space has a one to one mapping; a mapping to Q in the same dimension suggests a *square invertible matrix* applied to the right of A , lets call it R^{-1} ; so $AR^{-1} = Q$ (the linear combinations of the columns of A to produce Q must themselves be linearly independent). This square invertible matrix has an inverse R , which when applied to the right of this equation, gives us

$$AR^{-1}R = A = QR$$

so we can say that there is some R connecting the orthonormal Q back to A .

To find this R , we simply multiply the entire expression by Q^T :

$$Q^T A = Q^T QR = R$$

So we have $R = Q^T A$:

$$R = \begin{bmatrix} q_1^T \\ q_2^T \\ q_3^T \end{bmatrix} \begin{bmatrix} a & b & c \end{bmatrix} = \begin{bmatrix} q_1^T a & q_1^T b & q_1^T c \\ q_2^T a & q_2^T b & q_2^T c \\ q_3^T a & q_3^T b & q_3^T c \end{bmatrix}$$

This can be simplified. We know that

- The vectors a and A and q_1 are all along a single line
- The vectors a, b and A, B and q_1, q_2 are all in the same plane
- The vectors a, b, c and A, B, C and q_1, q_2, q_3 are all in the same 3 dimensional subspace

The point here is that q_2 is orthogonal to that entire first line, and q_3 is orthogonal to that entire second plane. Looking back at R , see that $q_2^T a = 0$ by orthogonality, as well as $q_3^T a$ and $q_3^T b$. Our simplified matrix looks like

$$\begin{bmatrix} a & b & c \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix} \begin{bmatrix} q_1^T a & q_1^T b & q_1^T c \\ & q_2^T b & q_2^T c \\ & & q_3^T c \end{bmatrix} \quad \text{or} \quad A = QR.$$

(next page)

Cont.

We had

$$\begin{bmatrix} \mathbf{a} & \mathbf{b} & \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3 \end{bmatrix} \begin{bmatrix} \mathbf{q}_1^T \mathbf{a} & \mathbf{q}_1^T \mathbf{b} & \mathbf{q}_1^T \mathbf{c} \\ & \mathbf{q}_2^T \mathbf{b} & \mathbf{q}_2^T \mathbf{c} \\ & & \mathbf{q}_3^T \mathbf{c} \end{bmatrix} \quad \text{or} \quad \mathbf{A} = \mathbf{Q}\mathbf{R}.$$

Intuitively, at every step $\mathbf{a}_1, \dots, \mathbf{a}_k$ are combinations of $\mathbf{q}_1, \dots, \mathbf{q}_k$, and so column i of \mathbf{R} , which specifies \mathbf{a}_i as a combination of the columns of \mathbf{Q} , will only have nonzero entries in the rows corresponding to the columns of \mathbf{Q} that contribute to that \mathbf{a}_i .

See that this leads to \mathbf{R} being *triangular*—any \mathbf{a}_j only depends on the \mathbf{q}_i where $i < j$ —so all the entries below the j th row in \mathbf{R} will be zero.

(Gram-Schmidt) From independent vectors $\mathbf{a}_1, \dots, \mathbf{a}_n$, Gram-Schmidt constructs orthonormal vectors $\mathbf{q}_1, \dots, \mathbf{q}_n$. The matrices with these columns satisfy $\mathbf{A} = \mathbf{Q}\mathbf{R}$. Then $\mathbf{R} = \mathbf{Q}^T \mathbf{A}$ is *upper triangular* because later \mathbf{q} 's are orthogonal to earlier \mathbf{a} 's.

In the context of Least Squares

See that using our factorisation:

$$\mathbf{A}^T \mathbf{A} = (\mathbf{Q}\mathbf{R})^T \mathbf{Q}\mathbf{R} = \mathbf{R}^T \mathbf{Q}^T \mathbf{Q}\mathbf{R} = \mathbf{R}^T \mathbf{R}$$

the least squares formula

$$\mathbf{A}^T \mathbf{A} \hat{\mathbf{x}} = \mathbf{A}^T \mathbf{b}$$

simplifies to

$$\mathbf{R}^T \mathbf{R} \hat{\mathbf{x}} = (\mathbf{Q}\mathbf{R})^T \mathbf{b}$$

$$\mathbf{R}^T \mathbf{R} \hat{\mathbf{x}} = \mathbf{R}^T \mathbf{Q}^T \mathbf{b}$$

$$\mathbf{R} \hat{\mathbf{x}} = \mathbf{Q}^T \mathbf{b} \quad \text{or} \quad \hat{\mathbf{x}} = \mathbf{R}^{-1} \mathbf{Q}^T \mathbf{b}$$

(\mathbf{R} is invertible since it has full column rank. It has full row rank too so \mathbf{R}^T also has full column rank and is invertible (justifying the last step))

This allows us to solve for the least squares solution $\hat{\mathbf{x}}$ using back substitution (which is very fast). The real cost is the computation for Gram-Schmidt, which is needed to construct the orthogonal \mathbf{Q} and the triangular $\mathbf{R} = \mathbf{Q}^T \mathbf{A}$.

(next page)

An algorithm

Here is an algorithm for the Gram-Schmidt process:

```

for j = 1:n                % modified Gram-Schmidt in n loops
    v = A(:,j);            % v begins as column j of the original A
    for i = 1:j-1          % columns q1 to qj-1 are already settled in Q
        R(i,j) = Q(:,i)' * v; % compute Rij = qiTaj which is qiTv
        v = v - R(i,j)*Q(:,i); % subtract the projection (qiTv)qi
    end                    % v is now perpendicular to all of q1, ..., qj-1
    R(j,j) = norm(v);      % the diagonal entries Rjj in R are lengths
    Q(:,j) = v/R(j,j);     % divide v by its length to get the next qj
end                        % for j = 1 : n produces all of the qj

```

Which essentially executes

$$r_{kj} = \sum_{i=1}^m q_{ik} v_{ij} \text{ and } v_{ij} = v_{ij} - q_{ik} r_{kj} \text{ and } r_{jj} = \left(\sum_{i=1}^m v_{ij}^2 \right)^{1/2} \text{ and } q_{ij} = \frac{v_{ij}}{r_{jj}}.$$

Where r denotes entries of \mathbf{R} and \mathbf{v} a column in the original \mathbf{A} .

The diagonal entries of \mathbf{R} are just the norms of the unnormalised projections in the Gram-schmidt process. For instance

$$\mathbf{q}_2^T \mathbf{b} = \mathbf{q}_2 \cdot \mathbf{b} = \cos(\theta) \|\mathbf{q}_2\| \|\mathbf{b}\| = \cos(\theta) \|\mathbf{b}\| = \|\mathbf{B}\|$$

θ , the angle between \mathbf{b} and its projection \mathbf{q}_2 , is the same angle between \mathbf{b} and its unnormalised projection \mathbf{B} .

4.8 Psuedoinverses

Left Inverse

Consider a ‘tall and thin’ matrix \mathbf{A} , where $m > n$ with rank $r = n$. A matrix like this only has a *left sided* inverse (consider that $\mathbf{x}^T \mathbf{A}$ would produce a matrix whose rows are combinations of linearly dependent rows—this cannot be ‘undone’ to recover \mathbf{x}^T given a result. On the other hand $\mathbf{A}\mathbf{x}$ can be ‘undone’ since this way the mapping \mathbf{A} is unique to each \mathbf{x} .)

In this case \mathbf{A} has independent columns, so $\mathbf{A}^T \mathbf{A}$ ($n \times n$) is invertible, and

$$(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{A} = \mathbf{I}$$

We say that

$$\mathbf{A}_{\text{left}}^{-1} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$$

is the *left inverse* of \mathbf{A} (there may be other left inverses), where $\mathbf{A}_{\text{left}}^{-1} \mathbf{A} = \mathbf{I}$.

Right inverse

In another case consider if \mathbf{A} were ‘short and fat’, where $m < n$ with rank $r = m$. A matrix like this only has a right sided inverse. \mathbf{A} has independent rows, so now $\mathbf{A}\mathbf{A}^T$ is invertible and

$$\mathbf{A}\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1} = \mathbf{I}$$

A nice *right inverse* is therefore:

$$\mathbf{A}_{\text{right}}^{-1} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}$$

where $\mathbf{A}\mathbf{A}_{\text{right}}^{-1} = \mathbf{I}$.

Chapter 5

Determinants

5.1 Properties of determinants

Appendix A

Misc. topics

A.1 Taylor series, Difference approximations

Taylor series

Recall the idea of the Taylor series, where $f(x)$ near some point $x = x_0$ can be approximated as

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \frac{f'''(x_0)}{3!}(x - x_0)^3 + \dots$$

See that this comes from approximating $f(x)$ as a polynomial

$$y(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + a_3(x - x_0)^3 + \dots$$

where we have $a_0 = y(x_0)$; now consider differentiating:

$$\begin{aligned} y'(x) &= a_1 + 2a_2(x - x_0) + 3a_3(x - x_0)^2 + \dots \\ y''(x) &= 2a_2 + (3)(2)a_3(x - x_0) + (4)(3)a_4(x - x_0)^2 + \dots \\ y^{(n)}(x) &= n!a_n + ((n+1) \cdot (n-1) \cdot \dots \cdot 3 \cdot 2)a_{n+1}(x - x_0) + \dots \end{aligned}$$

so we have

$$a_1 = y'(x_0), \quad a_2 = \frac{y''(x_0)}{2!}, \quad \dots, \quad a_n = \frac{y^{(n)}(x_0)}{n!}$$

where substituting into our initial approximation gives us

$$y(x) \approx y(x_0) + y'(x_0)(x - x_0) + \frac{y''(x_0)}{2!}(x - x_0)^2 + \dots + \frac{y^{(n)}(x_0)}{n!}(x - x_0)^n + \dots$$

(next page)

Cont.

We had

$$y(x) \approx y(x_0) + y'(x_0)(x - x_0) + \frac{y''(x_0)}{2!}(x - x_0)^2 + \cdots + \frac{y^{(n)}(x_0)}{n!}(x - x_0)^n + \cdots$$

substituting $x - x_0 = h$ gives us

$$y(x_0 + h) \approx y(x_0) + y'(x_0)(h) + \frac{y''(x_0)}{2!}(h)^2 + \cdots + \frac{y^{(n)}(x_0)}{n!}(h)^n + \cdots$$

Which can be rewritten as

$$y(x + h) \approx y(x) + hy'(x) + \frac{1}{2}h^2y''(x) + \cdots + \frac{1}{n!}h^ny^{(n)}(x) + \cdots$$

Difference formulas

Considering the first few terms of the Taylor series approximation, we have

$$y(x + h) - y(x) \approx hy'(x) + \frac{1}{2}h^2y''(x)$$

$$y(x - h) - y(x) \approx -hy'(x) + \frac{1}{2}h^2y''(x)$$

This allows us to have an approximation to dy/dx :

Centered difference	$\frac{y(x + h) - y(x - h)}{2h} \approx \frac{dy}{dx}$
----------------------------	--

and also to the second derivative

Second difference	$\frac{y(x + h) - 2y(x) + y(x - h)}{h^2} \approx \frac{d^2y}{dx^2}$
--------------------------	---

The individual formulas also yield approximations:

Forward difference	$\frac{dy}{dx} = \frac{y(x + h) - y(x)}{h} + O(h) \text{ error}$
Backward difference	$\frac{dy}{dx} = \frac{y(x) - y(x - h)}{h} + O(h) \text{ error}$