Linear Algebra and Matrix Methods
Orthogonality and Projection

A vector subspace is "missing something" from its parent

Let V be a vector space, and W a proper subspace of V ($W \neq V$).

W, then, is somehow "missing" something from V; in particular,

$$dim(W) < dim(V).$$

It takes less vectors to describe elements of W than it does for V.

Multiplying by a matrix transforms a vector...

If we apply an $m \times n$ matrix A to a vector $v \in \mathbb{R}^n$ with decomposition given by the Fundamental Theorem of Linear Algebra as

$$v = \vec{x}_n + c : \ \vec{x}_n \in N(A), \ c \in C(A^t),$$

we get

$$b = Av = A(\vec{x}_n + c) = A\vec{x}_n + Ac = 0 + Ac = Ac,$$

where $b \in C(A)$. We can see that c is the vector of coefficients that determines "how much" of each column vector of A goes into building the vector b.

So what "happens" to the vector $\vec{x}_n$? It contributes nothing to b.

... but might lead to information loss.

If $dim(N(A)) = n - r > 0$, A is not invertible, and there is a kind of "information loss" when applying A: we move from a point in an n-dimensional space,

$$v \in \mathbb{R}^n; \; dim(\mathbb{R}^n) = n,$$

to a point in an r-dimensional space,

$$Av \in C(A); \; dim(C(A)) = r < n.$$

The **image** $C(A)$ does not represent "all" of A, dimension-wise.[1]

The **kernel** $N(A)$ gets its dimension(s) from $\mathbb{R}^n$...

... and sends them to 0.

[1] We are not forgetting that $C(A)$ is a subspace of $\mathbb{R}^m$, not $\mathbb{R}^n$.

Orthogonal Complements

If W is a vector subspace of V, with $dim(W) = r \leq n = dim(V)$, then the **orthogonal complement** of W, denoted $W^{\perp}$ ("W-perp"), is the vector subspace of V such that

$$W \perp W^{\perp} \text{ and } W \oplus W^{\perp} = V.$$

That is, W and $W^{\perp}$ form an orthogonal direct sum that equals V.

Note that

$$dim(W^{\perp}) = n - r \text{ and } (W^{\perp})^{\perp} = W.$$

Counting Basis Vectors: FTLA II: Perp

If W is a vector subspace of V, with $dim(W) = r \leq n = dim(V)$, and if S is a basis for W, then $|S| = r$.

$W^\perp$ has a basis T with $|T| = n - r$.

The union $S \cup T$ is a basis for V.

Fundamental Theorem of Linear Algebra, Part II:

$$N(A) = C(A^t)^\perp \text{ and } N(A^t) = C(A)^\perp.$$

Counting Basis Vectors: Orthogonal Complementarity

If $dim(C(A)) = r$, then any basis of $C(A)$ has r vectors.

Any basis of $N(A^t)$ has $m - r$ vectors, all orthogonal to $C(A)$.

A basis of $N(A^t)$ can be considered the "missing" basis vectors from $C(A)$ to span all of $\mathbb{R}^m$.

Counting Basis Vectors: Orthogonal Complementarity

Likewise for $C(A^t)$ and $N(A)$:

a basis of $C(A^t)$ has r vectors, and a basis of $N(A)$ has $n - r$ vectors, all orthogonal to $C(A^t)$.

The union of these two bases is a basis of $\mathbb{R}^n$.

$dim(C(A)) = dim(C(A^t)) = r = rank(A)$ connects the two views.

Counting Basis Vectors: Rank-Nullity Theorem

This fact is captured generally in the **Rank-Nullity Theorem**.

For any linear transformation $A : \mathbb{R}^n \to \mathbb{R}^m$,

$$\begin{aligned} rank(A) + nullity(A) &= dim(im(A)) + dim(ker(A)) \\ &= dim(C(A)) + dim(N(A)) \\ &= r + (n - r) = n. \end{aligned}$$

Likewise for $A^t : \mathbb{R}^m \to \mathbb{R}^n$,

$$\begin{aligned} rank(A^t) + nullity(A^t) &= dim(im(A^t)) + dim(ker(A^t)) \\ &= dim(C(A^t)) + dim(N(A^t)) \\ &= r + (m - r) = m. \end{aligned}$$

Validating orthogonality: four fundamental subspaces of A

We will check that, for an $m \times n$ matrix $A \in \mathbb{R}^{m \times n}$, we have that

$$N(A) \perp C(A^t) \text{ and } N(A^t) \perp C(A).$$

Recall that, if $A\vec{x} = b$ and $A^t\vec{y} = c$, then

$$\vec{x} \cdot c = b \cdot \vec{y}.$$

Validating orthogonality: four fundamental subspaces of A

First, let $c \in C(A^t)$ and $\vec{x} \in N(A)$ (as columns). Then

$$A\vec{x} = 0 \text{ and } \exists \vec{y} \in \mathbb{R}^m : A^t\vec{y} = c.$$

Then their dot product shows that $\vec{x} \perp c$:

$$\vec{x} \cdot c = \vec{x}^t c = \vec{x}^t(A^t\vec{y}) = (\vec{x}^t A^t)\vec{y} = (A\vec{x})^t\vec{y} = 0^t\vec{y} = 0.$$

The argument for $b \perp \vec{y}$ is similar.

Projections: shadows onto a subspace

A **projection matrix** is a symmetric matrix P such that $P^2 = P$.

(The property $P^2 = P$ is called **idempotency**.)

What does this mean for a vector that is projected by P?

Projections: shadows onto a subspace

Upon repeated projection by the same matrix, no more information is "lost" after the first time. The projection is fixed from then on.

Let $\vec{x} \in \mathbb{R}^n$, and let P be an $n \times n$ projection matrix.

Then

$$P\vec{x} = p$$

for some $p \in \mathbb{R}^n$. This means $p \in C(P)$.

Projections: shadows onto a subspace

But if we apply P again,

$$P^2\vec{x} = P\vec{x} = p$$

as well. Applying the associative property,

$$P^2\vec{x} = P(P\vec{x}) = Pp = p,$$

which means that p maps to itself under P. That is, $Pp = Ip$.

Projections in the context of the FTLA

Let $A \in \mathbb{R}^{m \times n}$ be an $m \times n$ matrix.

Recall that, according to the FTLA, any $b \in \mathbb{R}^m$ can be written as a unique sum

$$b = p + e,$$

of a vector in $p \in C(A)$ and a vector in $e \in N(A^t)$, with $p \perp e$.

We'll use the notation

- p for "projection" (onto $C(A)$), and
- e for "error" (the "lost information", relative to A).

There exists a projection matrix P and $\vec{x} \in \mathbb{R}^n$ such that

$$Pp = A\vec{x} = p, \;\; Pe = A^t e = 0.$$

“Simplest” projection: reduce the number of coordinates

For example, consider the projection matrix

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

which projects a vector $b \in \mathbb{R}^3$ onto the vector in $\mathbb{R}^3$ with only its first and third coordinates.

That is, if $b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$, then $Pb = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ 0 \\ b_3 \end{pmatrix}$.

"Simplest" projection: reduce the number of coordinates

We can write

$$b = p + e = \begin{pmatrix} b_1 \\ 0 \\ b_3 \end{pmatrix} + \begin{pmatrix} 0 \\ b_2 \\ 0 \end{pmatrix}$$

where

$$e = b - p = \begin{pmatrix} 0 \\ b_2 \\ 0 \end{pmatrix}$$

is a dimension's worth of "error" that P "loses" in the projection.

Understanding the projection matrix P of the matrix A

Fix a vector $b \in \mathbb{R}^m$ and a matrix $A \in \mathbb{R}^{m \times n}$.

Then there exists a projection matrix $P \in \mathbb{R}^{m \times m}$
that sends $b \in \mathbb{R}^m$ into $C(A)$: $\exists \vec{x} \in \mathbb{R}^n$ such that

$$Pb = A\vec{x} = p.$$

We also have that $b = p + e$ for some $p \in C(A)$ and $e \in N(A^t)$.

$$\text{Thus, } Pb = P(p+e) = Pp + Pe = Pp + 0 = p; \ \ p \perp e.$$

Understanding the projection matrix P of the matrix A

If

$$A\vec{x} = b = p + e$$

has a solution $\vec{x}$ (unique or not), then $b \in C(A)$,
and projection by P onto $C(A)$ "loses no information";
there is no "error" in solving.

$$\exists \vec{x} : A\vec{x} = b \iff p = b,\ e = 0.$$

Understanding the projection matrix P of the matrix A

If

$$A\vec{x} = b = p + e$$

has *no* solution, then $b \notin C(A)$,
and there is some error in attempting a solution:
projection by P "loses information".

The "closest we can get" is p.

$$\nexists \vec{x} : A\vec{x} = b \iff p \neq b, \; e = b - p \neq 0.$$

Either way,

$$Pb = p; \quad Pe = P(b - p) = Pb - Pp = p - p = 0.$$

Understanding the projection matrix P: projects b to p

We can factor this error equation to learn about how projection works. Since $P^2 = P$, then the matrix

$$P - P^2 = (I - P)P = P(I - P) = 0.$$

If $b = p + e$ such that $Pb = p$ and $Pe = 0$, then

$$\begin{aligned}(P - P^2)b = 0 &\implies (I - P)Pb = 0 \\ &\implies (I - P)p = 0 \therefore p \in N(I - P).\end{aligned}$$

A projection vector p of P is a null (error) vector of $I - P$.

Understanding the matrix $I - P$: also a projection

If P is a projection matrix, then $I - P$ is also a projection matrix: using the facts that I and P are projections, and multiplication by I is commutative:

$$I^2 = I,\ P^2 = P,\ IP = PI = P,$$

we have

$$\begin{aligned}(I - P)^2 = (I - P)(I - P) &= I^2 - PI - IP + P^2 \\ &= I - 2P + P = I - P.\end{aligned}$$

Thus, $I - P$ satisfies the projection matrix property.

Understanding the projection matrix $I - P$: projects b to e

What happens to the P-error vector e under $I - P$?

$$(I - P)e = e - Pe = e - 0 = e.$$

Thus, e is projected onto itself under $I - P$.

To summarize: if P is a projection matrix, then so is $I - P$.

Understanding the projection matrix $I - P$: projects b to e

If $b \in \mathbb{R}^m$ has decomposition $b = p + e$, where

- p is the projection of b by P and
- e is the error under P,

then

- e is the projection of b by $I - P$ and
- p is the error under $I - P$.

Calculating the projection matrix P of the matrix A

Reconsidering P via the identity: if $b \in \mathbb{R}^m$, then the decomposition $b = p + e$ can be written in terms of P by

$$\begin{aligned} I &= P + (I - P) \\ \implies b = Ib &= (P + (I - P))b \\ &= Pb + (I - P)b \\ &= p + e. \end{aligned}$$

What is P, in terms of A?

Calculating the projection matrix P of the matrix A

We will compute P from what we know about the error vector e. If

$$p = Pb = A\hat{x}$$

is the "best fit" solution to the attempted

$$A\vec{x} = b,$$

with $b = p + e$, and P the projection matrix onto $C(A)$, we have

$$\begin{aligned} e &= b - p \\ &= b - Pb = b - A\hat{x} \end{aligned}$$

$$\begin{aligned} \implies A^t e &= A^t(b - A\hat{x}) \\ &= A^t b - A^t A\hat{x} = 0 \text{ (since } e \in N(A^t)) \end{aligned}$$

$$\implies A^t b = A^t A\hat{x}.$$

Calculating the projection matrix P of the matrix A

We will now mention some important aspects of A^tA:

- A^tA is a symmetric matrix with independent columns, and so A^tA is invertible.

With this knowledge, we continue our derivation with $(A^tA)^{-1}$:

$$\begin{aligned}
A^tb &= A^tA\hat{x} \\
\implies (A^tA)^{-1}A^tb &= (A^tA)^{-1}A^tA\hat{x} \\
\implies (A^tA)^{-1}A^tb &= (A^tA)^{-1}(A^tA)\hat{x} \\
\implies (A^tA)^{-1}A^tb &= \hat{x} \\
\implies A(A^tA)^{-1}A^tb &= A\hat{x} = p.
\end{aligned}$$

Our conclusion: $P = A(A^tA)^{-1}A^t$.

The projection matrix P of the matrix A solves $A\hat{x} = Pb$

By this construction of the projection P onto $C(A)$, the matrix

$$P = A(A^tA)^{-1}A^t,$$

we can see that, whether or not the equation

$$A\vec{x} = b$$

can be solved for $\vec{x}$, there is always a solution $\hat{x}$ to the equation

$$A\hat{x} = Pb.$$

That projection solution $\hat{x}$ is, by applying most of P to both sides, and noticing that $A^tP = A^t$,

$$\hat{x} = (A^tA)^{-1}A^tb.$$

Example: Projection onto a line

Suppose A is a column vector $(m \times 1)$. As a vector, call it a.

How do you project the vector $b \in \mathbb{R}^m$ onto the line

$$C(A) = \{ca \,|\, c \in \mathbb{R}\}?$$

If $\exists x \in \mathbb{R}$ such that $ax = b$, then $b \in C(A)$ and you are done.

If there is no such x, then we need to solve the projection equation instead:

$$a\hat{x} = Pb = p \implies b - a\hat{x} = b - p = e.$$

Example: Projection onto a line

From here, we have

$$
\begin{aligned}
& & b - a\hat{x} &= e \\
\Longrightarrow & & a \cdot (b - a\hat{x}) &= a \cdot e = 0 \text{ (since } a \perp e) \\
\Longrightarrow & & a \cdot b &= (a \cdot a)\hat{x} \text{ (since } \hat{x} \text{ is a scalar)} \\
\Longrightarrow & & (a \cdot a)^{-1}(a \cdot b) &= \frac{a \cdot b}{a \cdot a} = \hat{x}.
\end{aligned}
$$

This should look very similar to the general case, where $\hat{x} \in \mathbb{R}^n$:

$$\hat{x} = (A^t A)^{-1} A^t b.$$

Projection: Pythagorean Theorem (what else is new)

The error vector $e = b - p$ of a vector $b \in \mathbb{R}^m$ is the *minimum distance* possible between b and its projection p under A.

Whenever the word "distance" is uttered...

... the Pythagorean Theorem is lurking nearby.

Projection: Pythagorean Theorem (what else is new)

If the error e is the minimum distance between p and b,

and $p \perp e$, then e and p are the legs of a triangle,

and b is the hypotenuse: examining vector lengths, that gives us

$$||b||^2 = ||p||^2 + ||e||^2.$$

Projection: Pythagorean Theorem (error is minimized)

We will verify this fact, and cast the error e as the vector with *minimum* distance, with the *least square* error from the intended "solution" to $A\vec{x} = b$.

Thus, we will call p the **least squares**, or **best fit**, **approximation** to b under A, and e the **least square error**.

Projection = Least squares approximation under A

Let $\vec{x} \in \mathbb{R}^n$ be *any* vector (not necessarily a minimizing one).

Given the decomposition $b = p + e$ for $b \in \mathbb{R}^m$, we can write e in terms of b, p, and *any* $\vec{x} \in \mathbb{R}^n$:

$$\begin{aligned} & b = p + e \\ \implies \; & e = b - p = (A\vec{x} - p) - (A\vec{x} - b), \end{aligned}$$

where, since $p, A\vec{x} \in C(A)$, we have $e \perp A\vec{x}$, and so $e \perp A\vec{x} - p$.

Projection = Least squares approximation under A

Thus, the Pythagorean Theorem also holds under the lengths

$$||A\vec{x} - b||^2 = ||A\vec{x} - p||^2 + ||e||^2.$$

If $p = Pb$ minimizes the error in computing (or failing to compute) $A\vec{x} = b$, then the error between $A\hat{x}$ and p is 0:

$$||A\hat{x} - p|| = 0.$$

This verifies that the least squares solution $\hat{x}$ minimizes the error of any $\vec{x} \in \mathbb{R}^n$:

$$||A\hat{x} - b||^2 = ||e||^2 \leq \inf_{x \in \mathbb{R}^n} ||A\vec{x} - b||^2.$$

Least squares approximation: best fit curve to data

One common application of linear projection is in constructing the **best fit curve** to a set of data points.

Say we have a set of m points in $\mathbb{R}^2$:

$$\{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}.$$

If the data fits the function $y = f(x)$ perfectly, we would be able to write this data set as

$$\{(x_1, y_1 = f(x_1)), (x_2, y_2 = f(x_2)), ..., (x_m, y_m = f(x_m))\}.$$

However, this is not typically the case with real-world data.

Least squares approximation: best fit curve to data

If we declare that f uses $n+1$ coefficient parameters $c_0, c_1, c_2, ..., c_n$ in its definition, what is the vector of parameters

$$c = (c_0, c_1, c_2, ..., c_n)$$

that minimize the error in considering these m data points under f, i.e. minimizes the mean squared error $||Ac - b||^2$?

In this problem, we are given f, and solve for best fit of c.

Least squares example: best fit line

Example

Find the best fit line to the points $\{(0,6),(1,0),(2,0)\}$.

The best fit line is of form $f(x) = c_0 + c_1 x$, so we will solve for the parameter vector $c = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix}$.

This means the system of equations generated by the data is

$$\begin{aligned} c_0 + 0c_1 &= 6 \\ c_0 + 1c_1 &= 0 \\ c_0 + 2c_1 &= 0, \end{aligned}$$

which clearly does not have a solution. We want the best fit.

Least squares example: best fit line

Our system is the matrix equation $Ac = b$, where

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix}, \ c = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix}, \ b = \begin{pmatrix} 6 \\ 0 \\ 0 \end{pmatrix}.$$

The best fit parameter solution $\hat{c}$ is given by

$$\hat{c} = (A^t A)^{-1} A^t b = \begin{pmatrix} 5 \\ -3 \end{pmatrix},$$

which gives the best fit line

$$f(x) = c_0 x + c_1 = 5 - 3x.$$

Least squares example: best fit line

How close is the best fit?

$$p = A\hat{c} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 5 \\ -3 \end{pmatrix} = \begin{pmatrix} 5 \\ 2 \\ -1 \end{pmatrix}$$

$$e = b - p = \begin{pmatrix} 6 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 5 \\ 2 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}$$

$$\implies ||e||^2 = e \cdot e = 6.$$

Least squares example: best fit line with calculus

Let's do the same problem, but with calculus this time.

Compute the error $E(c) = ||e||^2$ for a general pair of parameters for the line $f(x) = c_0 + c_1 x$; this yields the square error

$$\begin{aligned} E(c) &= ||Ac - b||^2 \\ &= \left|\left| \begin{pmatrix} c_0 - 6 \\ c_0 + c_1 \\ c_0 + 2c_1 \end{pmatrix} \right|\right|^2 = (c_0 - 6)^2 + (c_0 + c_1)^2 + (c_0 + 2c_1)^2. \end{aligned}$$

We'll take this square error and minimize it via the second derivative test on c_0 and c_1.

Least squares example: best fit line with calculus

$E(c)$ has a critical point at c when its first partial derivatives are 0:

$$E(c) = (c_0 - 6)^2 + (c_0 + c_1)^2 + (c_0 + 2c_1)^2$$

$$\frac{\partial E}{\partial c_1} = 0 + 2(c_0 + c_1) + 2(c_0 + 2c_1)(2) = 6c_0 + 10c_1$$

$$\frac{\partial E}{\partial c_0} = 2(c_0 - 6) + 2(c_0 + c_1) + 2(c_0 + 2c_1) = 6c_0 + 6c_1 - 12$$

$$\frac{\partial^2 E}{\partial c_1^2} = 10 > 0,\ \frac{\partial^2 E}{\partial c_0^2} = 6 > 0 \text{ (concave up; critical point is a min)}$$

$$\implies 6c_0 + 10c_1 = 0,\ 6c_0 + 6c_1 = 12 \implies c = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = \begin{pmatrix} 5 \\ -3 \end{pmatrix}.$$

Least squares approximation: best fit line, general

In general, the best fit line $f(x) = c_0 + c_1 x$, which takes a parameter $c \in \mathbb{R}^2$, minimizes its error on a set of m data points

$$\{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$$

by solving the projection equation $A\hat{c} = Py$ for the vector $y \in \mathbb{R}^m$ and the matrix $A \in \mathbb{R}^{m \times 2}$ defined by

$$A = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{pmatrix}, \;\; \hat{c} = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix}, \;\; y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}.$$

We can simplify this to the 2×2 system $A^t A\hat{c} = A^t y$, using

$$A^t A = \begin{pmatrix} m & \sum_{i=1}^m x_i \\ \sum_{i=1}^m x_i & \sum_{i=1}^m x_i^2 \end{pmatrix}, \;\; A^t y = \begin{pmatrix} \sum_{i=1}^m y_i \\ \sum_{i=1}^m x_i y_i \end{pmatrix}.$$

Least squares approximation: best fit polynomial, general

In general, the best fit nth degree polynomial

$$f(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_n x^n = \sum_{i=0}^{n} c_i x^i,$$

which takes a parameter $c \in \mathbb{R}^{n+1}$, minimizes its error on a set of m data points

$$\{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$$

by solving the projection equation $A\hat{c} = Py$ for the vector $y \in \mathbb{R}^m$ and matrix $A \in \mathbb{R}^{m \times (n+1)}$ defined by

$$A = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ & & \ddots & & \\ 1 & x_m & x_m^2 & \cdots & x_m^n \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}.$$

A Nice Basis?

In projection and best fitting, we need a matrix A of column vectors that are **linearly independent**. This means the columns of A are a **basis** of $C(A)$.

But to do these computations, we need $A^t A$, which can itself be cumbersome to compute.

If we have a "nice" basis to take columns from, the calculation of $A^t A$ would be easy.

We'll say the "nicest" type of basis is an **orthonormal basis**.

Orthogonal, Orthonormal Set

A set of vectors $\{q_1, q_2, ..., q_n\}$ is called **orthogonal** if they are all pairwise orthogonal. We call the set **orthonormal** if the set is orthogonal and all unit vectors; that is,

$$q_i \cdot q_j = \delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j. \end{cases}$$

δ_{ij} is a function called the **Kronecker delta function**.[2]

[2] Not to be confused with a **Dirac delta function**, which is not a function, but what is called a **generalized function**, or **distribution**, which gives an integral positive weight only at a "point mass". This type of function is used, for example, to write (discrete) probability *mass* functions as probability *densities* with point masses, so you can always write an integral for a CDF.

Orthogonal Matrix, Orthonormal Basis

If a matrix $Q = \begin{pmatrix} q_1 & q_2 & \cdots & q_n \end{pmatrix}$ has an orthonormal set for its columns, then

$$Q^t Q = I,$$

and we call Q an **orthogonal matrix**[3].

If, in addition, Q is square, then $QQ^t = I$, Q is invertible with

$$Q^{-1} = Q^t,$$

and the column set of Q is an **orthonormal basis** for $\mathbb{R}^n$.

[3]Some texts reserve the term **orthogonal matrix** for square matrices Q only.

Orthogonal Matrix Examples: Rotation, Permutation

The simplest nontrivial example of an orthogonal matrix is a **rotation matrix**: for any $0 \leq \theta < 2\pi$,

$$Q = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

will rotate the point $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2$ counterclockwise by θ radians.

Any permutation matrix[4] P is orthogonal:

$$P = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \implies P^t = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \implies P^t P = I.$$

[4] Just because permutation and projection matrices both use P as their representative symbols, they are not the same type of matrix. Context matters.

Orthogonal Matrix Examples: Reflection

If $u \in \mathbb{R}^n$ is a unit column vector, then the **outer product** uu^t is an $n \times n$ matrix (of rank one), and the matrix

$$Q = I - 2uu^t$$

is a **reflection matrix**, under which $Qv \in \mathbb{R}^n$ is the reflection of $v \in \mathbb{R}^n$ across the line spanned by u.

Note: $Q^tQ = I$, and $Q^t = I - 2uu^t = Q$, so reflection matrices are **involutions**; they are their own inverses.

Reflection of a reflection is the original position: $Q^2v = v$.

Orthogonal Matrices are Isometric

An orthogonal matrix preserves the length of a vector it multiplies:

$$||Qv|| = ||v||,$$

meaning Q is a type of operation called an **isometry**.

This is a special case of preserving dot products, meaning Q also preserves angles:

$$(Qv) \cdot (Qw) = (Qv)^t (Qw) = v^t (Q^t Q) w = v^t I w = v \cdot w$$

$$\implies \cos\theta = \frac{(Qv) \cdot (Qw)}{||Qv|| \cdot ||Qw||} = \frac{v \cdot w}{||v|| \cdot ||w||}.$$

In particular, preserving angle means preserving orthogonality.

Orthogonal matrices make easy-to-compute projections

How about projections? We started commenting on orthogonal matrices because their transpose multiplication was easy.

The projection matrix onto the orthogonal matrix Q's column space $C(Q)$ is

$$P = Q(Q^t Q)^{-1} Q^t = QQ^t.$$

Orthogonal matrices make easy-to-compute projections

This is where the distinction between square and non-square Q is crucial.

If Q is square, then Q is invertible, so since every equation $Q\vec{x} = b$ is solvable, $P = I$.

Once again, $Q^t = Q^{-1}$ and $Q\vec{x} = b$ is solved by

$$\vec{x} = Q^{-1}b = Q^t b.$$

$C(Q) = C(Q^t) = \mathbb{R}^n$ and $N(Q^t) = N(Q) = \{0\}$.

Gram-Schmidt orthogonalization: orthonormalize a basis

Say $S = \{a_1, a_2, ..., a_n\}$ is a set of n independent vectors in $\mathbb{R}^n$. Then S is a basis of $\mathbb{R}^n$, but it may be difficult to compute with.

The **Gram-Schmidt** orthogonalization process is a procedure to convert a basis of $\mathbb{R}^n$ into an orthonormal basis.[5]

The order of the basis vectors matters in the process: the first vector determines the first direction, and successive vectors are twisted to be orthogonal to all the previous ones and scaled.

[5] This process can be used on a set of less than n independent vectors, and end up with an orthonormal set. You only end with a basis if you start with one.

Gram-Schmidt orthogonalization: twist, then scale; repeat.

Start with the basis $\{a_1, a_2, ..., a_n\}$.

1. Set $b_1 = a_1$. Then $q_1 = \frac{b_1}{||b_1||}$.
2. Set $b_2 = a_2 - \left(\frac{b_1 \cdot a_2}{b_1 \cdot b_1}\right) b_1$, the orthogonal projection of a_2 onto the line spanned by b_1, subtracted from a_2.
   Then $b_2 \perp b_1$. Scale it: $q_2 = \frac{b_2}{||b_2||}$.
3. Set $b_3 = a_3 - \left(\frac{b_1 \cdot a_3}{b_1 \cdot b_1}\right) b_1 - \left(\frac{b_2 \cdot a_3}{b_2 \cdot b_2}\right) b_2$.
   Then $b_3 \perp b_1$ and $b_3 \perp b_2$. Scale it: $q_3 = \frac{b_3}{||b_3||}$.
4. Successively, continue:

$$b_k = a_k - \sum_{i=1}^{k-1} \left(\frac{b_i \cdot a_k}{b_i \cdot b_i}\right) b_i; \quad q_k = \frac{b_k}{||b_k||}, \quad k = 2, ..., n.$$

End with the orthonormal basis $\{q_1, q_2, ..., q_n\}$.

How the orthogonalization works; matrix form

First, it is clear that $||q_k|| = 1$ for every k. To account for orthogonality:

- q_1 is on the same line as a_1.
- q_2 is in the plane spanned by a_1 and a_2, but $q_2 \perp q_1$.
- q_3 is in the space spanned by a_1, a_2, and a_3, but $q_3 \perp q_1$, q_2.
- $q_k \in span(\{a_1, a_2, ..., a_k\})$ and $q_k \perp q_1, ..., q_{k-1}$.

$A = QR$ properties, least squares solutions

The matrix factorization is $A = QR$, where Q is orthogonal and R is square upper-triangular.

Since $Q^tQ = I$, we also have $R = Q^tA$, where $r_{ij} = q_i \cdot a_j$.
If $i > j$, $r_{ij} = 0$. This is true whether or not A and Q are square.

In fact, if A is not square, but its columns are independent, then we can still use the QR-decomposition to get orthonormal columns in Q, and R will still be square and upper-triangular.

Thus, R is invertible. We can use this fact to compute projection solutions for A.

$A = QR$ properties, least squares solutions

Let $A = QR$. Then

$$A^t A = (QR)^t (QR) = R^t Q^t QR = R^t R.$$

Since R is invertible, so is R^t. Thus, R^{-1} and $(R^t)^{-1} = (R^{-1})^t$ both exist.

The least squares approximation to $A\vec{x} = b$ is

$$\begin{aligned} A^t A\hat{x} = A^t b \implies R^t R\hat{x} = R^t Q^t b \\ \implies R\hat{x} = Q^t b \implies \hat{x} = R^{-1} Q^t b. \end{aligned}$$

As usual, if $A\vec{x} = b$ has a solution, $\hat{x}$ is the projection term.